Journal of the Brazilian Computer Society, 2026, 32:1, doi: 10.5753/jbcs.2026.5390

Improved Biclique Cryptanalysis of the Lightweight Cipher FUTURE

Gabriel de Carvalho [**Universidade do Estado do Rio de Janeiro** | ***gabrielc@eng.uerj.br***]
Luis Kowada [**Universidade Federal Fluminense** | ***luis@ic.uff.br***]

Department of Computer and Systems Engineering (DESC), Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro, R. São Francisco Xavier, 524 - Maracanã, Rio de Janeiro - RJ, 20550-013, Brazil.

Received: 31 December 2024 • **Accepted:** 08 August 2025 • **Published:** 25 March 2026

Abstract. In the past decade, lightweight cryptography has been of much interest in the academia, especially regarding the cryptanalysis of such ciphers. The National Institute of Standards and Technology (NIST) is one of the entities responsible for this interest, given that they promoted in 2019 a public process to choose the American standard for lightweight cryptography. In 2022, the FUTURE cipher was published and has since been the target of much cryptanalysis, including integral, meet-in-the-middle and differential cryptanalysis in a very short period of time. The objective of this paper is to present four biclique attacks that are better than the one previously published, in terms of time, memory and data complexities, obtained through semi-automatic search. Our fastest attack requires $2^{124.38}$ full computations of the cipher to run, while requiring only 2^{24} data pairs and negligible memory. We also present the fastest unbalanced biclique attack and star attack to our knowledge. Only one integral attack on FUTURE has been published that is faster than our attacks, $2^{123.70}$ without using the full codebook of data, i.e. less than 2^{64} pairs of plaintexts/ciphertexts, requiring 2^{63} pairs. Still, when compared to it, our attacks use much less data while being only slightly slower, which presents a good trade-off.

Keywords: Cryptanalysis, Symmetric Cryptography, Block Ciphers, Biclique Cryptanalysis.

1 Introduction

A Lightweight Cipher is a cipher designed to be used in very constrained environments, such as embedded systems, radio devices, and sensor networks McKay *et al*. [2016]. FUTURE Gupta *et al*. [2022] is one of the most recent developments in this field and one of the most academically relevant to the study of block cipher cryptanalysis. It is a Substitution-Permutation Network (SPN), using structures very similar to those of AES Daemen and Rijmen [2013] with 10 rounds, 64-bit block size and secret key of size 128 bits.

FUTURE has been the target of several attacks in a very short period of time Xu *et al*. [2024]; Schrottenloher and Stevens [2023]; Roy *et al*. [2024]; Mondal *et al*. [2024]; İlter and Selçuk [2022] since its publication in 2022. Some of these attacks target round-reduced versions of FUTURE, while others are applied to the full-round cipher. Since our attack targets the full version, those that also consider the full-round cipher are more relevant to this work.

Schrottenloher and Stevens published a simple meet-in-the-middle attack on several ciphers, with FUTURE being one of them. Their attack uses the full codebook and recovers the secret key in 2^{126} full computations of the FUTURE cipher. The codebook is defined as the set of all plaintext–ciphertext pairs for the cipher with the secret key Schrottenloher and Stevens [2023].

Roy, Dey, Mondal and Adhikari published at the end of 2023 the first biclique cryptanalysis on FUTURE. They presented two attacks: one using a simple balanced biclique and the other using a new variation proposed by the authors, which uses two bicliques simultaneously in the attack. The simplest one uses a 32-dimensional balanced biclique to achieve a time complexity of $2^{125.8875}$, while the second one uses $2^{125.5365}$ full computations of the cipher to perform the attack. Both use 2^{48} for data and approximately to 4GB of memory Roy *et al*. [2024].

In 2024, Xu, Cui, Hu and Wang published the fastest attacks on FUTURE currently available, including one that uses the full codebook and another that uses half of it. One of the attacks requires $2^{123.7}$ full computations of the cipher using 2^{63} of data and another, even faster, requires 2^{112}, but also depends on the full codebook (2^{64} pairs) Xu *et al*. [2024]. Throughout this paper, the term "faster" is used to specifically refer to a lower time complexity in the comparison of attacks.

Biclique cryptanalysis is most famous for its application to the AES cipher Bogdanov *et al*. [2011], constituting the first attack to its full-round version. In the following years, many others built upon their work, including Tao and Wu [2015]; de Carvalho *et al*. [2023a]; Bogdanov *et al*. [2015]. It was also applied to several others in the past decade, such as ARIA Chen and Xu [2014], Serpent de Carvalho and Kowada [2020], HIGHT Hong *et al*. [2011], and, most recently, FUTURE Roy *et al*. [2024].

The use of software tools for automatic or semi-automatic searches to assist cryptanalysts has become common over time. Most modern cryptanalysis employs tools in some capacity. Modeling attacks as a MILP or SAT problem is one of the most widely used approaches to automate part of the work. Recent works Schrottenloher and Stevens [2023]; Shi *et al*. [2023]; de Carvalho *et al*. [2023b] are examples of the use of MILP modeling for the problem of searching for MITM attacks on many ciphers, while Bellini *et al*. [2024]

introduced in 2024 a Python library called **CLAASP**, which enables the automation of various forms of cryptanalysis for different types of block ciphers.

SBSeg2024 Contributions. All previously published attacks on FUTURE have the same limitation: the amount of required data is too large for it to be relevant. The original paper accepted in SBSeg2024 presented two attacks: a balanced biclique attack that is the fastest of its kind, requiring significantly less data than any other attack on the cipher, and a star attack that requires the minimum possible amount of data, while being only slightly slower than other attacks. Both attacks are based on distinct generator sets (GS-Biclique), using bicliques found through an automatic search for related-key differentials — a variation introduced by de Carvalho *et al*. [2022] — with the *BicliqueFinder* tool[1]. The balanced attack arguably presents the best time-data trade-off among all known attacks that do not rely on the full codebook.

Paper Contributions. This paper presents two new attacks, one balanced biclique and one unbalanced one. Both outperform the attack presented at SBSeg2024, which was already the fastest biclique attack at the time, although the new attacks require slightly more data. This performance improvement is achieved through the concept of partial word dependency, where certain words in the matching phase are affected by certain differentials while remaining unaffected by others. This is exploited to the fullest inside within the model typically used to estimate biclique complexity (i.e., the number of active S-boxes). The *BicliqueFinder* software was updated to support the discovery and accurate complexity estimation of such attacks. In addition, the complexity of the star attack presented at SBSeg2024, as well as most of the figures, have been revised. Table 1 presents a comparison between our attacks and other known attacks on the full-round FUTURE cipher.

Paper structure. Section 2 describes the biclique cryptanalysis the concept of generator sets for key bits. The FUTURE cipher and its notation are described in Section 3. The following sections are dedicated to each attack. The balanced biclique from SBSeg2024 is described in Section 4. The new attacks are both explained and detailed in Sections 5 and 6, the first for the new balanced attack and the second for the unbalanced one. The star attack, presented in SBSeg2024 and update here is presented in Section 7. Finally, Section 8 concludes the paper.

2 Biclique Cryptanalysis

In this section we present the basic steps associated with the balanced biclique attack with the biclique being built in the last rounds of the cipher, as used in this paper to attack the FUTURE cipher.

- **Preparation phase.** An adversary partitions the key space into groups with 2^{2d} keys for some d. Each key group is associated with a $2^d \times 2^d$ matrix K, where each element $K[i,j]$ represents a key in the group. Let k be the number of bits of the secret key. In this case we have 2^{k-2d} groups. Also, the cipher $\Gamma = f \circ g \circ h$ being attacked is a composition of three subciphers f, g and h. This phase is the most important of the attack, since most of the complexities derive from the choices made here. For this reason, Section 2.1 is dedicated to this phase. The three steps below are then done for each key group.

1. **Building the biclique.** A biclique structure is built over the subcipher f, resulting in a structure that satisfies the condition
$$\forall i,j : S_j \xrightarrow[f]{K[i,j]} C_i,$$
where $0 \leq i,j < 2^d$, S_j are internal states of the cipher and C_i are ciphertexts. Section 2.2 presents the fundamentals for building such structures.
2. **Obtain plaintexts.** This is a chosen ciphertext attack. Consequently, we have at our disposal a decryption oracle, which is used to obtain the plaintext P_i for each ciphertext C_i.
$$\forall i : C_i \xrightarrow[E^{-1}]{decryption\ oracle} P_i.$$
3. **Meet-in-the-Middle.** For each key $K[i,j]$ in the group it is tested if
$$\exists i,j : P_i \xrightarrow[g]{K[i,j]} S_j.$$
If one of the $K[i,j]$ is the secret key, then the above condition is satisfied. Therefore, every key that satisfies it is a candidate to the secret key. This can be done through any meet-in-the-middle method. Here, we use the Matching with precomputations technique, explained in Section 2.3.

2.1 Preparation Phase

Here we informally define the core concepts of the preparation phase from the biclique cryptanalysis.

Let $E = f \circ g \circ h$ be a cipher, with s subkeys of k' bits each, generated from a key schedule over the secret key with k bits. We call *key space* the interval $[0, 2^k - 1]$. The total *key bits* is then $m = s \cdot k'$. A *generator set of key bits* is any set of key bits that is enough to generate all m key bits of cipher E through some algorithm. Each family of differentials in the attack can be defined over a different generator set, if necessary. This concept was created by de Carvalho *et al*. [2022].

The biclique attack can be seen as an optimization of an exhaustive search. This is due to every single key being tested, except that it is done in such a way that is faster than simply testing each possible key through the cipher. For that to happen, it is necessary to partition the whole key space into disjoint sets in a way that each set is tested separately through the biclique. Most of the improvement in speed comes from this choice.

Each group has a representative, called a *base key*. The base key of the group has some bits fixed to 0 while all other vary from group to group. The bits that are fixed to 0 are the ones that go through every possible value when the key

[1] It can be found at `https://github.com/Clique33/BicliqueFinder`

Table 1. Time, data and memory complexity of the attacks on full-round FUTURE. The $\approx \mathbf{0}$ sign symbolizes a negligible amount.

Attack	Time	Data	Memory	Reference
Meet-in-the-Middle	2^{126}	2^{64}	2^{36}	Schrottenloher and Stevens [2023]
Balanced Biclique	$2^{125.8875}$	2^{48}	2^{32}	Roy *et al*. [2024]
Multiple Biclique	$2^{125.5365}$	2^{48}	2^{32}	Roy *et al*. [2024]
Integral	$2^{123.7}$	2^{63}	$\approx \mathbf{0}$	Xu *et al*. [2024]
Integral	$\mathbf{2^{112}}$	2^{64}	$\approx \mathbf{0}$	Xu *et al*. [2024]
$(4)-$Balanced Biclique	$2^{125.18}$	$\mathbf{2^{20}}$	$\approx \mathbf{0}$	Section 4
$(2 \cdot 4)-$Balanced Biclique	$\mathbf{2^{124.38}}$	2^{24}	$\approx \mathbf{0}$	Section 5
$(4, 2 \cdot 4)-$Unbalanced Biclique	$2^{124.80}$	2^{24}	$\approx \mathbf{0}$	Section 6
Star Attack	$2^{126.14}$	$\mathbf{1}$	$\approx \mathbf{0}$	Section 7

difference of each related-key differential in the biclique is applied to them.

For example, suppose a base key in which bytes 0 and 1 are fixed to 0. This means that there has to be a total of 2^{16} related-key differentials in the cipher, and every one of them influence one or both of those bytes in a way that no key is repeated nor is not tested. The base key must also be a generator set (usually a subkey or consecutive subkeys), so that all the key bits can be generated to carry out the attack.

2.2 Building the Biclique

Here, we take a look at the biclique structure built on the subcipher f of the target cipher $E = f \circ g \circ h$. Due to its simplicity, we show the structure of the balanced biclique. However, the star biclique and bicliques with multiple differentials are explained in their respective sections.

Let f be the subcipher that maps an state S to the ciphertext C using the key K thorugh the subcipher f (*i.e.* $f_K(S) = C$). A *dimension d biclique over f* or *$d-$dimensional biclique over f* is the 3-tuple $(\{S_j\}, \{C_i\}, \{K[i,j]\})$, where $0 \leq i, j < 2^d$ such that

$$\forall i, j : f_{K[i,j]}(S_j) = C_i.$$

One way to achieve this condition is using related-key differentials. It is important to highlight the fact that this is a single key attack. The related-key model is used only within the key groups. Let $K[0,0]$ be the *base key*, *i.e.* the key that maps the internal state S_0 to the ciphertext C_0. This is called the *base computation*

$$S_0 \xrightarrow[f]{K[0,0]} C_0.$$

The next step is defining the Δ_i*-differentials* and ∇_j*-differentials* using related-key differentials. Δ_i*-differentials* map the input difference 0 to the output difference Δ_i, using the key difference Δ_i^K, where $\Delta_0 = \Delta_0^K = 0$ and $0 \leq i < 2^d$

$$0 \xrightarrow[f]{\Delta_i^K} \Delta_i,\ \Delta_0 = \Delta_0^K = 0.$$

In contrast, ∇_j*-differentials* map the input difference 0 to the output difference ∇_j, using the key difference ∇_j^K on the opposite direction, from f^{-1}, where $\nabla_0 = \nabla_0^K = 0$ and $0 \leq j < 2^d$

$$\nabla_j \xleftarrow[f^{-1}]{\nabla_j^K} 0,\ \nabla_0 = \nabla_0^K = 0.$$

Two families of differentials are independent if there are no bits of states or subkeys of f in which ∇_j*-differentials* and Δ_i*-differentials* affect simultaneously. If both sets of differentials are independent, then it is possible to combine them into (Δ_i, ∇_j)*-differentials*

$$\nabla_j \xrightarrow[f]{\nabla_j^K \oplus \Delta_i^K} \Delta_i.$$

By definition, the base computation conforms to both sets of differentials and hence, it is possible to substitute it to the combined differentials

$$S_0 \oplus \nabla_j \xrightarrow[f]{K[0,0] \oplus \nabla_j^K \oplus \Delta_i^K} \Delta_i \oplus C_0.$$

By letting

$$S_j = S_0 \oplus \nabla_j,$$
$$C_i = \Delta_i \oplus C_0 \text{ and}$$
$$K[i,j] = K[0,0] \oplus \nabla_j^K \oplus \Delta_i^K$$

we have the definition of a dimension d biclique over f.

Building a biclique this way costs only 2^{d+1} computations of f, since it is possible to choose the key differences and base computation, and then, independently, compute the Δ_i*-differentials* and ∇_j*-differentials*.

2.3 Matching with Precomputations

This technique uses the knowledge that only parts of the cipher are affected by the differentials of the biclique to do the meet-in-the-middle step faster. It can be further exploited if instead of meeting an entire internal state, we meet in only a part of the state, namely v. This way we only look at the parts *affected by* the differentials *and* that *affect* v.

Let $E = f \circ g \circ h$, where the biclique was built over f. An adversary then computes and stores 2.2^d *full computations* of the cipher up to the variable v: 2^d computations of h and 2^d computations of g^{-1}

$$\forall i : P_i \xrightarrow[h]{K[i,0]} v_i^1 \text{ and } \forall j : v_j^2 \xleftarrow[g^{-1}]{K[0,j]} S_j.$$

This means that every internal state and subkeys of s and t up to v have to be stored. This is the *precomputation phase*.

Then comes the *recomputation phase*, where the parts that differ from the stored values must be recomputed. The cost of this method is rather variable depending on the diffusion properties of the cipher of interest, both the diffusion related to the key schedule and the states. The number of rounds also influences the cost.

2.4 Complexities

This attack can be seen as an improved exhaustive search, since every key will be tested, but not the whole cipher will be computed in each step. Three types of complexities are of interest: memory, data and time.

The memory complexity is dominated by the Precomputation Phase due to requiring the storage of whole states and keys of subciphers g and h. So if the biclique has dimension d, the memory complexity will be 2^{d+1} computations of $g \circ h$.

The data complexity depends only on how many bits of C_i are affected by the Δ_i-differentials, since all possible relevant C_i must be turned into P_i for the meet-in-the-middle step, which in turn depends essentially on the amount of rounds covered by the biclique, as well as on its dimension, and on the diffusion properties of the cipher.

Finally, the time complexity is where most of the analysis is necessary. It is basically the number of key groups times the time complexity of each iteration. each iteration builds the biclique and then does the matching with precomputations, which is divided into precomputation phase and recomputation phase. In the end, we have

$$C_{time} = 2^{k-2d}(C_{biclique}+C_{precomp}+C_{recomp}+C_{falspos}).$$

The false positives are the keys that pass on the test in the recomputation phase, meaning that they are secret key candidates. Thus it is necessary to check if they are the secret key.

3 The FUTURE Cipher

FUTURE Gupta *et al*. [2022] is a 10 round Substitution-Permutation Network cipher (SPN). The key size is 128 bits while its block size is 64 bits. Each subkey has 64 bits, seen as a bitstring of 16 nibbles (chunks of 4 bits each).

$$FUTURE = AK_{10} \circ SR \circ SC \circ AK_9 \circ R_8 \circ R_7 \circ ... \circ R_1 \circ R_0$$

Each round $R_i = SR \circ MC \circ SC \circ AK_i$, for $0 \leq i < 9$. Rounds vary from 0 to 9 and there are 11 subkeys, derived from the secret key, indexed from K^0 (or \$0) to K^{10} (or \$10), and 41 states, where state jth is denoted as denoted as $\#j$, where $P = \#0$ and $C = \#40$. Each state $\#j$ can be graphically represented by a 4×4 matrix of nibbles (chunks of 4 consecutive bits). The nibbles are enumerated from the leftmost to the rightmost and then down to the next word. The words are enumerated from top to bottom. The h-th nibble of the state S é denoted as s_h. Next we have the graphic representation of an internal state of the cipher.

s_0	s_1	s_2	s_3
s_4	s_5	s_6	s_7
s_8	s_9	s_{10}	s_{11}
s_{12}	s_{13}	s_{14}	s_{15}

The *SubCell* operation SC looks up the S-box shown in Table 2 and substitutes each nibble of the state according to the it.

The *ShiftRows* SR is, similar to the AES, the rotation of nibbles of each row from the state. There are 4 rows, from 0 to 3. Row i is rotated i times to the right.

s_0	s_1	s_2	s_3
s_4	s_5	s_6	s_7
s_8	s_9	s_{10}	s_{11}
s_{12}	s_{13}	s_{14}	s_{15}

$\longrightarrow$

s_0	s_1	s_2	s_3
s_7	s_4	s_5	s_6
s_{10}	s_{11}	s_8	s_9
s_{13}	s_{14}	s_{15}	s_{12}

The *MixColumns* MC is, again very similar to the AES. Each of the four columns of the state are multiplied by a Maximum Distance Separable (MDS) matrix. This multiplication is done in $GF(2^4)$, with the primitive polynomial $x^4 + x + 1$. The matrix follows.

$$\begin{pmatrix} 8 & 9 & 1 & 8 \\ 2 & 2 & 9 & 9 \\ 2 & 3 & 8 & 9 \\ 9 & 9 & 8 & 1 \end{pmatrix}$$

The key scheduling for this cipher is remarkably simple: The 128 bit key is partitioned into two, the 64 leftmost bits become X and the other 64 become Y. Then, each K^i is equal to $X <<< (5 \cdot (\frac{i}{2}))$, if i is even and $Y <<< (5 \cdot (\frac{i}{2}))$ if i is odd. It does not use any Sboxes in its scheduling.

4 $(4)-$Dimensional Biclique

In this section we present the fastest biclique attack on the FUTURE cipher up to the SBSeg2024 conference. The attack also required much less data than any other attacks on the full-round FUTURE. This biclique was found through the BicliqueFinder tool. It can be found at `https://github.com/Clique33/BicliqueFinder`.

4.1 Preparation Phase

We partition the key into $2^{128-2\cdot4} = 2^{120}$ groups, since FUTURE has a 128 bit secret key and our biclique is $4-$dimensional. We define FUTURE as the composition $FUTURE = f \circ g \circ h$, where h enciphers the plaintext into state $\#17$, g enciphers state $\#17$ into state $\#25$ and f enciphers state $\#25$ into the ciphertext.

Due to the key scheduling of the cipher, any two subkeys are a generator set for the key if the index of one is even and the index of the other is odd. Hence, we define both related-key differentials through subkeys \$0 and \$1. The nibble 3 of \$0 is the only active nibble of Δ^K and nibble 8 of \$1 is the only active nibble for ∇^K. These key differences create Δ_i-differentials and ∇_j-differentials that are independent from each other, as shown in Figure 1. It is a $4-$dimensional biclique because the differentials share no Sboxes and there are 2^4 possible C_i and S_j. This biclique covers the last 4

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	3	0	2	7	E	4	D	9	A	C	6	F	5	8	B

Table 2. S-box of the FUTURE cipher.

rounds of the cipher, which goes from state #25 through to state #40.

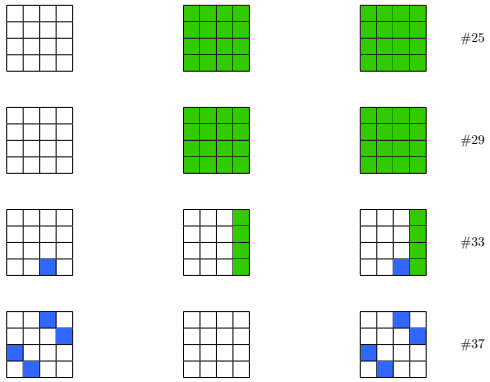

Figure 1. The nibbles of relevant states affected by Δ_i^K (left) and the ones affected by ∇_j^K (middle) in the building of biclique phase of the $4-$dimensional biclique. The rightmost shows they are independent.

4.2 Matching with Precomputations over 6 rounds

In this part of the attack, we check if the secret key belongs to the group $\{K[i,j]\}$, *i.e.* if $K_{secret} \in \{K[i,j]\}$. First we precompute 2^5 values of v, which we define as being the nibble 3 of state #17, and save them, together with all internal states and subkeys involved in these precomputations. Then, we have

$$P_i \xrightarrow[h]{K[i,0]} v_{i,0}^1 \text{ and } v_{0,j}^2 \xleftarrow[g^{-1}]{K[0,j]} S_j$$

for each i and j, recomputing only those parts that differ from the ones saved in memory. If $v_{i,j}^1 = v_{i,j}^2$, then $K[i,j]$ is a key candidate.

Next, we observe the recomputation that has to be done in the matching with precomputations step. Figure 2 shows graphically the nibbles that need to be recomputed.

In the forward direction we are interested in the difference between the computation of $P_i \xrightarrow[h]{K[i,j]} v$ and the precomputed values of $P_i \xrightarrow[h]{K[i,0]} v_i^1$, given by the influence of ∇_j^K on the subkeys from \$0 to \$4. Since FUTURE does not use Sboxes in the key schedule, they are irrelevant for the recomputation step. Only 1 nibble of #5 is influenced by ∇_j^K. Next, states #9 and #13 require the recomputation of 4 nibbles each. Therefore, we only have to recompute 9 Sboxes for the states in the forward direction.

Similarly to the forward recomputation, we look at the difference between $v_j^2 \xleftarrow[g^{-1}]{K[i,j]} S_j$ and the precomputed $v_j^2 \xleftarrow[g^{-1}]{K[0,j]} S_j$, given by the influence of Δ_i^K in the subkeys

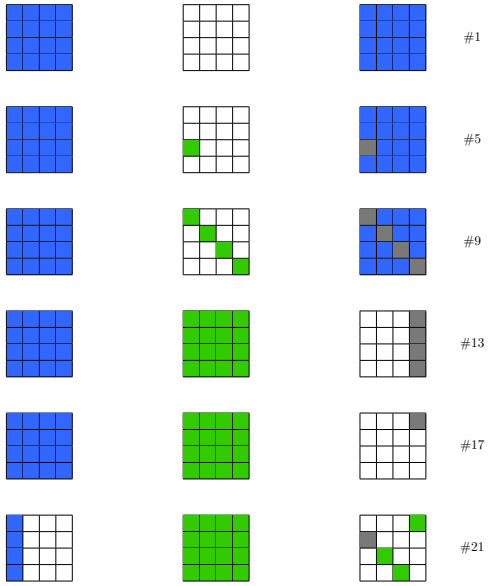

Figure 2. The nibbles of relevant states affected by Δ_i^K (left) and ∇_j^K (middle), in the matching phase of the $4-$dimensional biclique. In the rightmost, grey represents the ones affected by both.

\$5 to \$6. It is possible to see that only one nibble of both states #21 and #17, since these are the only ones that affect variable v, and thus, only 2 Sboxes must be recomputed.

Therefore, only $9 + 2 = 11$ Sboxes out of 160 total in the cipher must be recomputed.

4.3 Complexities

Firstly we have

$$C_{total} = 2^{k-2d}(C_{biclique} + C_{precomp} + C_{recomp} + C_{falpos}).$$

The default approach to time complexity in biclique cryptanalysis is the percentage of Sboxes required to perform the attack, compared to the total number of Sboxes in the cipher. This is due to it usually being the bottleneck for time complexity in S-box based cipher. For the FUTURE cipher the total number of Sboxes is 160.

The complexity of building the biclique is given by the computation of the base, then 2^4 computations of Δ as well as 2^4 computations of ∇, which means counting the amount of Sboxes needed in each. For Δ it is only 5 Sboxes, while ∇ requires 36 Sboxes. For the base, it is enough to calculate the percentage of the cipher needed $C_{biclique} = (16/41) + (2^4 - 1) \cdot (5/160) + (2^4 - 1) \cdot (36/160) = 2^{2.0820}$.

For the precomputation it suffices to calculate how many times the subcipher $g \circ h$ is calculated, $C_{precomp} = 2^4 \cdot (25/41) = 2^{3.2863}$. On average, there will be 2^4 false positives per iteration, while the cost to test it is given by the total recomputation from P_i to #17 and from S_j to #17, instead of only meeting in v. This is about 37 Sboxes in the forward direction (1 S-box in #5, 4 Sboxes in #9, 16

Sboxes in #13 and 16 Sboxes in #17) and 20 Sboxes in the backward direction (4 Sboxes in #21 and 16 Sboxes in #17), Totalling 57 Sboxes. Hence, $C_{falpos} = 2^{2d}/2^{|v|} \cdot \text{cost} = 2^8/2^4 \cdot (57/160) = 2^{2.5110}$).

It remains to find C_{recomp}. As discussed in the last section, the total number of Sboxes to be recomputed is only 11. Then $C_{recomp} = (2^8 - 2^4) \cdot (11/160) = 2^{4.0444}$.

In the end, we obtain approximately

$$C_{total} = 2^{120}(2^{2.0820}+2^{3.2863}+2^{4.0444}+2^{2.5110}) = 2^{125.18}$$

FUTURE full computations.

The data complexity is given by how many active nibbles are in the ciphertext state. It is possible to notice in Figure 3 that only 5 nibbles of C_i are affected and thus, only 2^{20} pairs of plaintexts/ciphertexts are necessary for the attack, *i.e.* there are only 2^{20} possibilities for the ciphertexts.

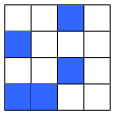

Figure 3. Nibbles affected by Δ_i in the last state.

In terms of memory, the attack is upper limited by 2^4 computations of $g \circ h$ The full computation of $g \circ h$ consists of 25 states and 6 subkeys, with 16 nibbles each. Therefore the memory complexity is $2^4 \cdot (25 + 16) \cdot 16 = 10496$ nibbles, which is 5248 bytes. That is a negligible amount.

5 $(2 \cdot 4)-$Dimensional Biclique

In this section we present the fastest balanced biclique attack on the FUTURE cipher to our knowledge. The attack requires 16 times more data then the $(4)-$dimensional biclique from Section 4, although it is still much less data than other attacks on the full-round FUTURE.

5.1 Building Bicliques with Multiple Key Differences

In Section 2.2, we defined the $d-$dimensional biclique, but that only captures the balanced bicliques with a single set of differences from S to C (Δ_i^K) and one on the opposite direction (∇_j^K). Here we expand this definition for bicliques with multiple key differences.

Let the cipher $E = f \circ g \circ h$ and f be the subcipher that maps an state S to the ciphertext C using the key K through the subcipher f (*i.e.* $f_K(S) = C$). A $(n \cdot d)-$*dimensional biclique over* f is the 3-tuple $(\{S_{j_0,\ldots,j_{k-1}}\}, \{C_{i_0,\ldots,i_{k-1}}\}, \{K[i_0,\ldots,i_{k-1},j_0,\ldots,j_{k-1}]\})$, where $0 \leq i_0,\ldots,i_{k-1},j_0,\ldots,j_{k-1} < 2^d$ and $0 \leq k < n$ such that

$$f_{K[i_0,\ldots,i_{k-1},j_0,\ldots,j_{k-1}]}(S_{j_0,\ldots,j_{k-1}}) = C_{i_0,\ldots,i_{k-1}}.$$

In this section, we are interested in a $(2 \cdot 4)-$dimensional biclique, therefore our biclique is simply $(\{S_{j_0,j_1}\}, \{C_{i_0,i_1}\}, \{K[i_0,i_1,j_0,j_1]\})$. The procedure is then quite similar, to before, but with one set of differences for each and, consequently, one family of differentials for each:

$$\begin{aligned}
S_{j_0} &= S_0 \oplus \nabla_{j_0},\\
S_{j_1} &= S_0 \oplus \nabla_{j_1},\\
C_{i_0} &= \Delta_{i_0} \oplus C_0,\\
C_{i_1} &= \Delta_{i_1} \oplus C_0,\\
K[i_0,i_1,j_0,j_1] &= K[0,0] \oplus \Delta_{i_0}^K \oplus \Delta_{i_1}^K \oplus \nabla_{j_0}^K \oplus \nabla_{j_1}^K.
\end{aligned}$$

The cost to build it is as low as the more independent they are. If all four are independent in the biclique, then the cost to build it is $2^4 + 2^4 + 2^4 + 2^4 = 2^6$, instead of the brute force approach that would cost 2^{16}. Our work only searched for those bicliques in which the deltas are independent from nablas, but not necessarily between each other, which in turn makes the cost of building $2^8 + 2^8 = 2^9$ computations. This was done because of current limitations in our BicliqueFinder software.

The downsides are that the more differentials, the harder it is to find independent bicliques and it can increase the amount of data required to carry the attack. Conversely, it speeds up all independent processes in greater magnitude since the reduction of complexity from 2^{2d} to 2^{d+1} is exponential.

5.2 Preparation Phase

We partition the key into $2^{128-4\cdot4} = 2^{112}$ groups, since FUTURE has a 128 bit secret key and our biclique is $(2 \cdot 4)-$dimensional. We define FUTURE identically to the $4-$dimensional attack, as the composition $FUTURE = f \circ g \circ h$, where h enciphers the plaintext into state #17, g enciphers state #17 into state #25 and f enciphers state #25 into the ciphertext.

Due to the key scheduling of the cipher, any two subkeys are a generator set for the key if the index of one is even and the index of the other is odd. Hence, we define all related-key differentials through subkeys \$0 and \$1. The nibble 11 of \$0 is the only active nibble of $\Delta_{i_0}^K$ and nibble 3 of \$0 is the only active nibble of $\Delta_{i_1}^K$. Nibble 0 of \$1 is the only active nibble for $\nabla_{j_0}^K$ and nibble 1 of \$1 is the only active nibble for $\nabla_{j_1}^K$. These key differences create Δ_{i_0}-differentials, Δ_{i_1}-differentials, ∇_{j_0}-differentials and ∇_{j_1}-differentials such that the deltas are independent from the nablas, as shown in Figure 5. It is a $(2 \cdot 4)-$dimensional biclique because the Δ_{i_0} and Δ_{i_1} differentials share no Sboxes with ∇_{j_0} and ∇_{j_1}. This biclique covers the last 4 rounds of the cipher, which goes from state #25 through to state #40.

5.3 Matching with Precomputations

In this part of the attack, we check if the secret key belongs to the group $\{K[i_0,i_1,j_0,j_1]\}$, *i.e.* if $K_{secret} \in \{K[i_0,i_1,j_0,j_1]\}$. First we precompute 2^6 values of v, 2^4 for each family of differentials, which we define as being the nibble 3 of state #17, and save them, together with all internal states and subkeys involved in these precomputations.

Up to here it is the same as other attacks, but we are able to further exploit the idea of precomputing to compute the nibbles affected by $(\Delta_{i_0}, \Delta_{i_1}, \nabla_{j_0})$, $(\Delta_{i_0}, \Delta_{i_1}, \nabla_{j_1})$, $(\Delta_{i_0}, \nabla_{j_0}, \nabla_{j_1})$ and $(\Delta_{i_1}, \nabla_{j_0}, \nabla_{j_1})$ by precomputing 2^8 for each of the pairs of Δs and ∇s, but only storing the parts that are necessary to compute the relevant Sboxes. One last time, we may choose to compute and store for the 2^{12} computations of any of the triples to compute the remaining Sboxes affected by all of them.

In the first step, we precompute and store all states and subkeys from h and g that affect v for each differential:

$$P_{i_0,0} \xrightarrow[h]{K[i_0,0,0,0]} v^1_{i_0,0,0,0} \text{ and } v^2_{i_0,0,0,0} \xleftarrow[g^{-1}]{K[i_0,0,0,0]} S_{0,0},$$

$$\vdots$$

$$P_{0,0} \xrightarrow[h]{K[0,0,0,j_1]} v^1_{0,0,0,j_1} \text{ and } v^2_{0,0,0,j_1} \xleftarrow[g^{-1}]{K[0,0,0,j_1]} S_{0,j_1}.$$

Afterwards, we compute

$$P_{i_0,i_1} \xrightarrow[h]{K[i_0,i_1,0,0]} v^1_{i_0,i_1,0,0} \text{ and } v^2_{i_0,i_1,0,0} \xleftarrow[g^{-1}]{K[i_0,i_1,0,0]} S_{0,0},$$

and

$$P_{0,0} \xrightarrow[h]{K[0,0,j_0,j_1]} v^1_{0,0,j_0,j_1} \text{ and } v^2_{0,0,j_0,j_1} \xleftarrow[g^{-1}]{K[0,0,j_0,j_1]} S_{j_0,j_1},$$

by recomputing and storing only those parts that differ from the ones saved in memory and affect v. In Figure 4, the nibbles in dark blue are affected only by the deltas and the dark green only by nablas. The same process is used to compute the ones that are affected by three of the differentials, but only the triples $(\Delta^K_{i_0},\Delta^K_{i_1},\nabla^K_{j_k})$ and $(\Delta^K_{i_{k'}},\nabla^K_{j_0},\nabla^K_{j_1})$, for any $k, k' \in \{0, 1\}$ need to be stored. Those are the nibbles in salmon. Finally, they are used to recompute the final

$$P_{i_0,i_1} \xrightarrow[h]{K[i_0,i_1,j_0,j_1]} v^1_{i_0,i_1,j_0,j_1}$$

and

$$v^2_{i_0,i_1,j_0,j_1} \xleftarrow[g^{-1}]{K[i_0,i_1,j_0,j_1]} S_{j_0,j_1}.$$

If $v^1_{i_0,i_1,j_0,j_1} = v^2_{i_0,i_1,j_0,j_1}$, then $K[i_0, i_1, j_0, j_1]$ is a key candidate.

We now analyze the relevant states, i.e. the input states of the SubCell operations. In state #1 there is no influence from neither $\nabla^K_{j_0}$ nor $\nabla^K_{j_1}$. However, all 16 nibbles are affected by both $\Delta^K_{i_0}$ and $\Delta^K_{i_1}$. The next relevant state #5 all 14 nibbles from 2 to 15 are affected only by $\Delta^K_{i_0}$ and $\Delta^K_{i_1}$. Nibble 0 is affected by both deltas and by $\nabla^K_{j_0}$, while nibble 1 is affected by both deltas and by $\nabla^K_{j_1}$. All nibbles in state #9 are affected by both $\Delta^K_{i_0}$ and $\Delta^K_{i_1}$, but 8 are so only by them. There are 4 nibbles also affected by $\nabla^K_{j_0}$ and 4 other by $\nabla^K_{j_1}$. State #13, interestingly, has 12 nibbles that are ignored. That is because they do not affect the computation of variable v, i.e. nibble 3 os state #17. All other though, are affected by all differentials. Nibble 3 from state #17, by its nature also depends on all differentials. The last relevant state is #21, in which only 4 nibbles are relevant to the computation of v, all of them affected by $\nabla^K_{j_0}$ and $\nabla^K_{j_1}$, but nibble 4 also depends on $\Delta^K_{i_1}$ and nibble 14 depends also on $\Delta^K_{i_0}$. Figure 4 summarizes the relation of the affected nibbles from the relevant states. Therefore:

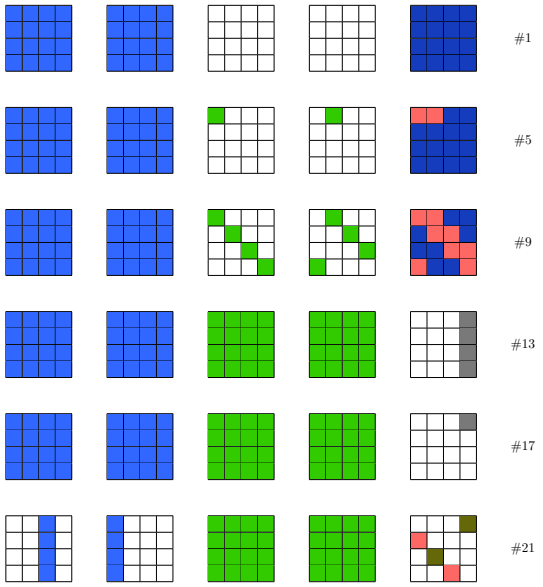

Figure 4. The nibbles of relevant states affected by $\Delta^K_{i_0}$, $\Delta^K_{i_1}$, $\nabla^K_{j_0}$ and $\nabla^K_{j_1}$, from left to right, respectively, in the matching phase of the $(2 \cdot 4)-$biclique. In the rightmost, salmon represents words that are affected by three families of differentials, dark blue the ones affected only by both deltas, dark green the ones affected only by both nablas and grey the ones affected by all.

- 38 Sboxes are affected only by $(\Delta^K_{i_0}, \Delta^K_{i_1})$,
- 2 are affected by $(\nabla^K_{j_0}, \nabla^K_{j_1})$;
- 5 are affected by $(\Delta^K_{i_0}, \Delta^K_{i_1}, \nabla^K_{j_0})$;
- 5 are affected by $(\Delta^K_{i_0}, \Delta^K_{i_1}, \nabla^K_{j_1})$;
- 1 is affected by $(\Delta^K_{i_0}, \nabla^K_{j_0}, \nabla^K_{j_1})$;
- 1 is affected by $(\Delta^K_{i_1}, \nabla^K_{j_0}, \nabla^K_{j_1})$;
- 5 are affected by $(\Delta^K_{i_0}, \Delta^K_{i_1}, \nabla^K_{j_0}, \nabla^K_{j_1})$.

5.4 Complexities

Before we had

$$C_{total} = 2^{k-2d}(C_{biclique} + C_{precomp} + C_{recomp} + C_{falpos}).$$

However, since the approach here is a little unlike the original one, in the since that there are many steps precomputations and recomputations, we use the simpler

$$C_{total} = 2^{k-2d}(C_{biclique} + C_{matching} + C_{falpos}).$$

We continue to approach time complexity in biclique cryptanalysis as the percentage of Sboxes required to perform the attack, compared to the total number of Sboxes in the cipher. That is, the time required to test a key through exhaustive search is given by the amount of Sboxes computed. A full computation of the FUTURE cipher requires the computation of 160 Sboxes.

The complexity of building the biclique is given by the computation of the base, then $2^4 \cdot 2^4$ computations of the Δs, since they are not independent, as well as $2^4 \cdots 2^4$ computations of the ∇s, which means counting the amount of Sboxes

needed in each. In fact, they are completely dependent, i.e. all Sboxes affected by one is also by the other, as shown in Figure 5. For the Δs it is only 5 Sboxes, while ∇ requires 36 Sboxes. For the base, it is enough to calculate the percentage of the cipher needed $C_{biclique} = (2^9) \cdot (41/160) = 2^{7.0356}$.

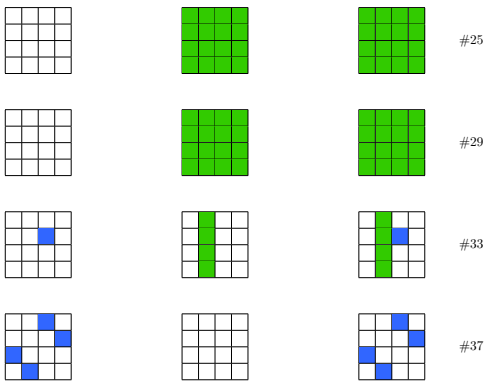

Figure 5. The nibbles of relevant states affected by both $\Delta^K_{i_0}$ and $\Delta^K_{i_1}$ (left) and the ones affected by both $\nabla^K_{j_0}$ and $\nabla^K_{j_1}$ (middle) in the building of biclique phase of the $(2 \cdot 4)-$biclique. The rightmost shows they are independent.

On average, there will be 2^{12} false positives per iteration, while the cost to test it is given by the total recomputation from P_{i_0,i_1} to #17 and from S_{j_0,j_1} to #17, instead of only meeting in v. This is about 37 Sboxes in the forward direction (1 S-box in #5, 4 Sboxes in #9, 16 Sboxes in #13 and 16 Sboxes in #17) for each ∇^K and 20 Sboxes in the backward direction (4 Sboxes in #21 and 16 Sboxes in #17) for each Δ^K, Totalling 114 Sboxes. Hence, $C_{falpos} = 2^{2 \cdot n \cdot d}/2^{|v|} \cdot \text{cost} = 2^{16}/2^4 \cdot (114/160) = 2^{11.5110}$).

It remains to find $C_{matching}$. As discussed in the last subsection, the time complexity will be given by each step: $C_{matching} = C_{singles} + C_{pairs} + C_{triples} + C_{quadruple}$. $C_{singles}$ is the complexity of each differential individually, similar to the precomputation step on the $4-$dimensional biclique. That is, every Sbox from relevant states (96), minus the ones that do not affect the computation of v, 12 from state #13, 15 from #17 and 12 #21, plus v is computed twice. So $C_{singles} = 4 \cdot 2^4 \cdot (58/160) = 2^{4.5361}$, Next, C_{pairs} is given by the time to compute the pairs of nablas and deltas. Since part has already been computed in $C_{singles}$ it is subtracted here. So there is a total of $2^8 - 2^5$ computations of the Sboxes that depend only on them both, a total of 40. Therefore, $C_{pairs} = (2^8 - 2^5) \cdot 40/160 = 2^{5.8074}$ Similarly, $C_{triples} = (2^{12} - 2^8 - 2^4) \cdot 12/160 = 2^{8.1639}$. Finally, $C_{quadruple} = (2^{16} - 2^{12} - 2^4) \cdot 5/160 = 2^{10.9065}$. Hence, the time complexity of the full matching with precomputations step is

$$C_{matching} = 2^{4.5361} + 2^{5.8074} + 2^{8.1639} + 2^{10.9065} = 2^{11.1583}$$

In the end, we obtain approximately

$$C_{total} = 2^{112}(2^{7.0356} + 2^{11.1583} + 2^{11.5110}) = 2^{124.38}$$

FUTURE full computations.

The data complexity is given by how many active nibbles are in the ciphertext state. It is possible to notice in Figure 6 that only 6 nibbles of C are affected and thus, only 2^{24} pairs of plaintexts/ciphertexts are necessary for the attack, *i.e.* there are only 2^{24} possibilities for the ciphertexts.

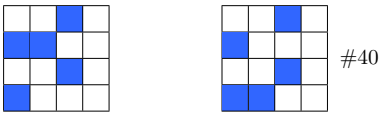

Figure 6. Nibbles affected by delta differentials in the last state, Δ_{i_0} to the left and Δ_{i_1} to the right.

In terms of memory, the attack is upper limited by the storage of all states and subkey for 2^6 computations of $g \circ h$, 2^4 for each family of differentials, plus the storage of 2^9 computations of $g \circ h$, 2^8 for each pair of deltas and nablas, plus 2^{12} for one of the triples. A full computation of $g \circ h$ consists of 25 states and 6 subkeys, with 16 nibbles each. Therefore the memory complexity is $(2^6 + 2^9 + 2^{12}) \cdot (25 + 16) \cdot 16 = 3064832$ nibbles, which is 1532416 bytes, which is less than $2MB$: a negligible amount.

6 $(4, 2 \cdot 4)-$Dimensional Biclique

In this section we present the fastest unbalanced biclique attack on the FUTURE cipher to our knowledge. The attack requires just as much data the $(2 \cdot 4)-$dimensional biclique from Section 5 although it loses some speed in comparison.

6.1 Building Bicliques with Multiple Key Differences

Once more we expand the definition of bicliques given in Section 5.1 for unbalanced bicliques. There are many similarities

Let the cipher $E = f \circ g \circ h$ and f be the subcipher that maps an state S to the ciphertext C using the key K through the subcipher f (*i.e.* $f_K(S) = C$). A $(n_1 \cdot d_1, n_2 \cdot d_2)-$*dimensional biclique over* f is the 3-tuple $(\{S_{j_0,\ldots,j_{k'-1}}\}, \{C_{i_0,\ldots,i_{k-1}}\}, \{K[i_0,\ldots,i_{k-1},j_0,\ldots,j_{k'-1}]\})$, where $0 \leq i_0,\ldots,i_{k-1} < 2^{d_0}$, $0 \leq j_0,\ldots,j_{k'-1} < 2^{d_1}$, $0 \leq k < n_0$ and $0 \leq k' < n_1$ such that

$$f_{K[i_0,\ldots,i_{k-1},j_0,\ldots,j_{k'-1}]}(S_{j_0,\ldots,j_{k'-1}}) = C_{i_0,\ldots,i_{k-1}}.$$

In this section, we are interested in a $(4, 2 \cdot 4)-$dimensional biclique, therefore our biclique is simply $(\{S_{j_0,j_1}\}, \{C_i\}, \{K[i, j_0, j_1]\})$. The procedure is then quite similar, to before, but with one set of differences for each and, consequently, one family of differentials for each:

$$\begin{aligned}
S_{j_0} &= S_0 \oplus \nabla_{j_0}, \\
S_{j_1} &= S_0 \oplus \nabla_{j_1}, \\
C_i &= \Delta_i \oplus C_0, \\
K[i, j_0, j_1] &= K[0, 0] \oplus \Delta^K_i \oplus \nabla^K_{j_0} \oplus \nabla^K_{j_1}.
\end{aligned}$$

Again, the cost to build it is as low as the more independent they are. If all four are independent in the biclique, then the cost to build it is $2^4 + 2^4 + 2^4 \approx 2^{5.6}$, instead of the brute force approach that would cost 2^{12}. Our work only searched for those bicliques in which the deltas are independent from nablas, but not necessarily between each other, which in turn makes the cost of building $2^4 + 2^8 \approx 2^{8.1}$ computations. This was done because of current limitations in our BicliqueFinder software.

6.2 Preparation Phase

We partition the key into $2^{128-3\cdot4} = 2^{116}$ groups, since FUTURE has a 128 bit secret key and our biclique is $(4, 2\cdot4)-$dimensional. We define FUTURE identically to all other attacks, as the composition $FUTURE = f \circ g \circ h$, where h enciphers the plaintext into state $\#17$, g enciphers state $\#17$ into state $\#25$ and f enciphers state $\#25$ into the ciphertext.

We define all related-key differentials through subkeys \$0 and \$1. The nibble 11 of \$0 is the only active nibble of $\Delta_{i_0}^K$ and nibble 3 of \$0 is the only active nibble of $\Delta_{i_1}^K$. Nibble 0 of \$1 is the only active nibble for $\nabla_{j_0}^K$ and nibble 1 of \$1 is the only active nibble for $\nabla_{j_1}^K$. These key differences create Δ_i-differentials, ∇_{j_0}-differentials and ∇_{j_1}-differentials such that the delta is independent from the nablas, as shown in Figure 8. It is a $(4, 2\cdot4)-$dimensional biclique because the Δ_i differentials share no Sboxes with ∇_{j_0} and ∇_{j_1}. This biclique covers the last 4 rounds of the cipher, which goes from state $\#25$ through to state $\#40$.

6.3 Matching with Precomputations

In this part of the attack, we check if the secret key belongs to the group $\{K[i, j_0, j_1]\}$, *i.e.* if $K_{secret} \in \{K[i, j_0, j_1]\}$. First we precompute $2^5 + 2^4$ values of v, 2^4 for each family of differentials, which we define as being the nibble 3 of state $\#17$, and save them, together with all internal states and subkeys involved in these precomputations.

Similarly to the $(2\cdot4)-$dimensional biclique, we are able to further exploit the idea of precomputing to compute the nibbles affected by $(\Delta_i, \nabla_{j_0}, \nabla_{j_1})$ by precomputing 2^8 the pair of ∇s, but only storing the parts that are necessary to compute the relevant Sboxes.

In the first step, we precompute and store all states and subkeys from h and g that affect v for each differential:

$$P_i \xrightarrow[h]{K[i,0,0]} v^1_{i,0,0} \text{ and } v^2_{i,0,0} \xleftarrow[g^{-1}]{K[i,0,0]} S_{0,0},$$

$$P_0 \xrightarrow[h]{K[0,j_0,0]} v^1_{0,j_0,0} \text{ and } v^2_{0,j_0,0} \xleftarrow[g^{-1}]{K[0,j_0,0]} S_{j_0,0}$$

and

$$P_0 \xrightarrow[h]{K[0,0,j_1]} v^1_{0,0,j_1} \text{ and } v^2_{0,0,j_1} \xleftarrow[g^{-1}]{K[0,0,j_1]} S_{0,j_1},$$

Afterwards, we compute

$$P_0 \xrightarrow[h]{K[0,j_0,j_1]} v^1_{0,j_0,j_1} \text{ and } v^2_{0,j_0,j_1} \xleftarrow[g^{-1}]{K[0,j_0,j_1]} S_{j_0,j_1}.$$

by recomputing and storing only those parts that differ from the ones saved in memory and affect v. In Figure 7, the nibbles in dark green are affected only by the ∇^Ks. The nibbles in salmon represent the ones affected by Δ_i^K and one of the ∇^Ks. Finally, they are used to recompute the final

$$P_i \xrightarrow[h]{K[i,j_0,j_1]} v^1_{i,j_0,j_1}$$

and

$$v^2_{i,j_0,j_1} \xleftarrow[g^{-1}]{K[i,j_0,j_1]} S_{j_0,j_1}.$$

If $v^1_{i,j_0,j_1} = v^2_{i,j_0,j_1}$, then $K[i, j_0, j_1]$ is a key candidate.

We now analyze the relevant states, i.e. the input states of the SubCell operations. In state $\#1$ there is no influence from neither $\nabla_{j_0}^K$ nor $\nabla_{j_1}^K$. However, all 16 nibbles are affected by Δ_i^K. The next relevant state $\#5$, 14 nibbles are affected only by Δ_i^K. Nibble 6 is affected by Δ_i^K and by $\nabla_{j_0}^K$, while nibble 15 is affected by both Δ_i^K and by $\nabla_{j_1}^K$. All nibbles in state $\#9$ are affected by Δ_i^K, but 8 are so only by them. There are 4 nibbles also affected by $\nabla_{j_0}^K$ and 4 other by $\nabla_{j_1}^K$. State $\#13$, has 12 nibbles that are ignored due to they not affecting the computation of variable v. All others are affected by all differentials. Nibble 3 from state $\#17$, is v and, therefore, depends on all differentials. The last relevant state is $\#21$, in which only 4 nibbles are relevant to the computation of v, all of them affected by $\nabla_{j_0}^K$ and $\nabla_{j_1}^K$, but nibble 3 also depends on Δ_i^K. Figure 7 summarizes the relation of the affected nibbles from the relevant states. Therefore:

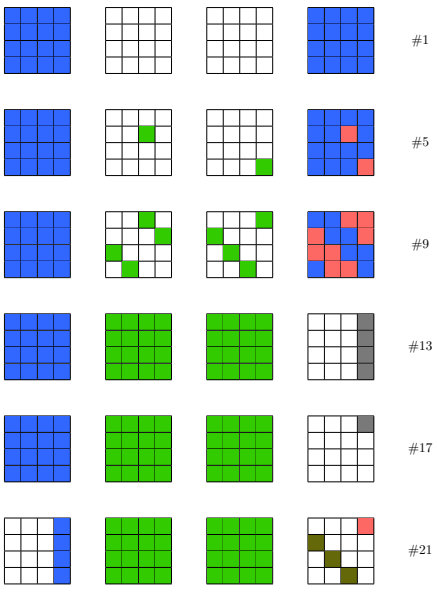

Figure 7. The nibbles of relevant states affected by Δ_i^K, $\nabla_{j_0}^K$ and $\nabla_{j_1}^K$, from left to right, respectively, in the matching phase of the $(4, 2\cdot4)-$biclique. In the rightmost, salmon represents words that are affected by two families of differentials, Δ_i^K and one of the ∇^Ks, dark green the ones affected only by both ∇^Ks and grey the ones affected by all.

- 38 Sboxes are affected only by Δ_i^K,
- 3 are affected by $(\nabla_{j_0}^K, \nabla_{j_1}^K)$;
- 11 are affected by $(\Delta_i^K, \nabla_{j_0}^K, \nabla_{j_1}^K)$;
- 5 are affected by $(\Delta_i^K, \nabla_{j_0}^K, \nabla_{j_1}^K)$.

6.4 Complexities

The time complexity analysis is closely related to the $(2 \cdot 4)-$dimensional biclique and thus, we use the same approach:

$$C_{total} = 2^{k-2d}(C_{biclique} + C_{matching} + C_{falpos}).$$

The complexity of building the biclique is given by the computation of the base, then 2^4 computations of Δ_i, as well as $2^4 \cdots 2^4$ computations of the ∇s, which means counting the amount of Sboxes needed in each. As shown in Figure 8. For Δ_i it is only 5 Sboxes, while the ∇s require 36 Sboxes each. Hence, it is enough to calculate the percentage of the cipher needed:

$$C_{biclique} = 2^8 \cdot (72/160) + 2^4 \cdot (5/160) = 2^{6.8542}.$$

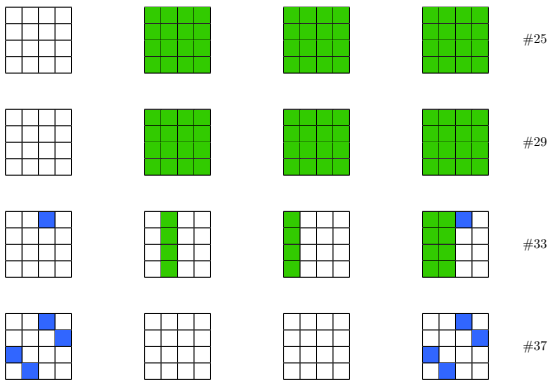

Figure 8. The nibbles of relevant states affected by Δ_i^K (left) and the ones affected by $\nabla_{j_0}^K$ (middle left) and $\nabla_{j_1}^K$ (middle right) in the building of biclique phase of the $(4, 2 \cdot 4)-$biclique. The rightmost shows they are independent.

On average, there will be 2^8 false positives per iteration, while the cost to test it is given by the total recomputation from P_i to #17 and from S_{j_0,j_1} to #17, instead of only meeting in v. This is about 37 Sboxes in the forward direction (1 S-box in #5, 4 Sboxes in #9, 16 Sboxes in #13 and 16 Sboxes in #17) for each ∇^K and 20 Sboxes in the backward direction (4 Sboxes in #21 and 16 Sboxes in #17), Totalling 94 Sboxes. Hence, $C_{falpos} = 2^{n_1 \cdot d_1 \cdot n_2 \cdot d_2}/2^{|v|} \cdot \text{cost} = 2^{12}/2^4 \cdot (94/160) = 2^{7.2327}$).

It remains to find $C_{matching}$. The time complexity will be given by each step: $C_{matching} = C_{singles} + C_{pairs} + C_{triple}$. $C_{singles}$ is the complexity of each differential individually, similar to the precomputation step on the $4-$dimensional biclique. That is, every Sbox from relevant states (96), minus the ones that do not affect the computation of v, 12 from state #13, 15 from #17 and 12 #21, plus v is computed twice. So $C_{singles} = 3 \cdot 2^4 \cdot (58/160) = 2^{4.1210}$, Next, C_{pairs} is given by the time to compute the pairs, either of ∇s or Δ and a ∇. Since part has already been computed in $C_{singles}$ it is subtracted here. So there is a total of $2^8 - 2^5$ computations of the Sboxes that depend only on them both, a total of 13. Therefore, $C_{pairs} = (2^8 - 2^5) \cdot 13/160 = 2^{4.1859}$. Finally, $C_{triple} = (2^{12} - 2^8 - 2^4) \cdot 6/160 = 2^{7.1639}$. Hence, the time complexity of the full matching with precomputations step is

$$C_{matching} = 2^{4.1210} + 2^{4.1859} + 2^{7.1639} = 2^{7.4838}$$

In the end, we obtain approximately

$$C_{total} = 2^{116}(2^{6.8542} + 2^{7.4838} + 2^{7.2327}) = 2^{124.80}$$

FUTURE full computations.

The data complexity is given by how many active nibbles are in the ciphertext state. It is possible to notice in Figure 9 that only 6 nibbles of C are affected and thus, only 2^{24} pairs of plaintexts/ciphertexts are necessary for the attack, *i.e.* there are only 2^{24} possibilities for the ciphertexts.

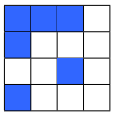

Figure 9. Nibbles affected by Δ_i in the last state.

In terms of memory, the attack is upper limited by the storage of all states and subkey for $2^5 + 2^4$ computations of $g \circ h$, 2^4 for each family of differentials, plus the storage of 2^8 computations of $g \circ h$, the pair of ∇s. A full computation of $g \circ h$ consists of 25 states and 6 subkeys, with 16 nibbles each. Therefore the memory complexity is $(2^5 + 2^4 + 2^8) \cdot (25 + 16) \cdot 16 = 199424$ nibbles, which is 99712 bytes, Which is less than $128KB$: a negligible amount.

7 Star attack

This attack uses a variation of the biclique cryptanalysis known as the *star attack*. In this variant, there are two or more differentials of the same kind (either all Δ or all ∇) and neither of the other. They are typically constructed at the beginning of the cipher and, as a result, only $\Delta-$differentials are involved. The most notable characteristic of the star attack is that it requires only a single plaintext-ciphertext pair. This is the case because, when constructed this way, no words are active in the plaintext or the ciphertext, meaning that only one pair of data is necessary to execute the attack.

Similarly to the attack presented in Section 4, we use a $4-$dimensional balanced biclique with two families of differentials created from $\Delta_{i_0}^K$ and $\Delta_{i_1}^K$. The generator sets for both key differences are the subkey \$0 and \$1, where nibble 0 of \$1 is active in $\Delta_{i_0}^K$, and nibble 2 of \$1 is active in $\Delta_{i_1}^K$.

Our star is built in the beginning of the cipher, from the plaintext up to #10. All other steps are executed in a similar way, where one of the deltas is treated as nabla for the precomputation step, and only the nibbles affected by both need to be recomputed. The variable v is chosen as nibble 3 from state #21. The biclique is independent as show in Figure 10.

This attack is slower when compared to the previous one to compensate its lack of data requirements. The forward recomputation requires the recomputation of all 16 Sboxes in state #13, 4 Sboxes in state #17 and only one in state #21. In the backward direction, all 16 Sboxes in state #29 and 4 Sboxes in state #25 have to be recomputed. This results in 41 Sboxes. This can be seen in Figure 11.

The time complexity for the recomputation is $C_{recomp} = (2^8 - 2^5) \cdot (41/160) = 2^{5.8430}$. The precomputation complexity is the total number of Sboxes outside the biclique

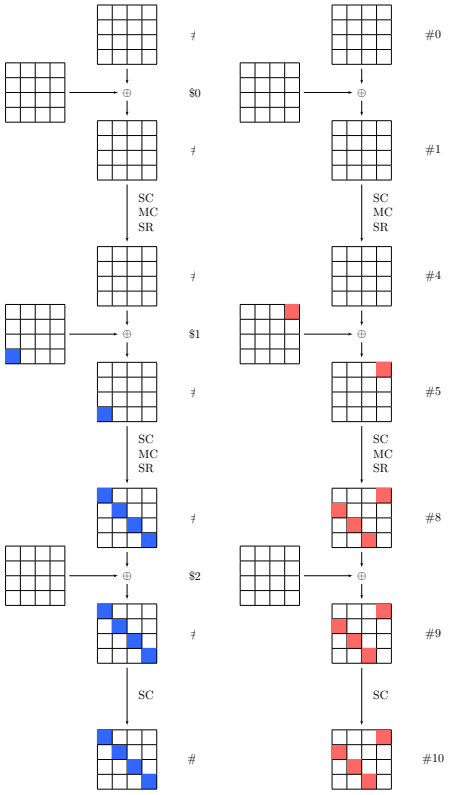

Figure 10. Δ_{i_0}-differentials and Δ_{i_1}-differentials in the star.

minus the ones that do not affect the computation of v, $C_{precomp} = 2^4 \cdot ((112 - 38)/160) = 2^{2.8875}$, since the matching step goes from state $\#10$ up to the ciphertext. The number of false positives is 2^4, but the cost of each test is higher when compared to the $4-$dimensional attack, since the star covers less states than our other attacks. That is a total of 80 Sboxes minus the ones already computed for v. Hence, $C_{falpos} = 2^{2d}/2^{|v|} \cdot \text{cost} = 2^8/2^4 \cdot (39/160) = 2^{1.9635}$). Lastly, $C_{biclique} = (2^5) \cdot (10/160) = 2^1$, since the biclique computes only 10 Sboxes total in which each Δ involves only 5 Sboxes.

In the end, we obtain approximately

$$C_{total} = 2^{120}(2^1 + 2^{2.8875} + 2^{5.8430} + 2^{1.9635}) = 2^{126.14}$$

FUTURE full computations.

8 Concluding Remarks

We presented here four biclique attacks on the full-round FUTURE cipher, three of them being faster than all other biclique attacks in literature to our knowledge and one with the lowest possible data complexity.

Both the $4-$dimensional balanced biclique attack, which requires only 2^{20} plaintext-ciphertext pairs and has a time complexity of $2^{125.18}$, and the star attack, with a time complexity of $2^{126.14}$, were presented at SBSeg2024. However, several figures and the time complexity of the star attack have since been revised. These improvements were made possible by our tool, which automates the search for bicliques and semi-automates the creation of propagation diagrams. Complete diagrams for the attacks can be found in the Appendix.

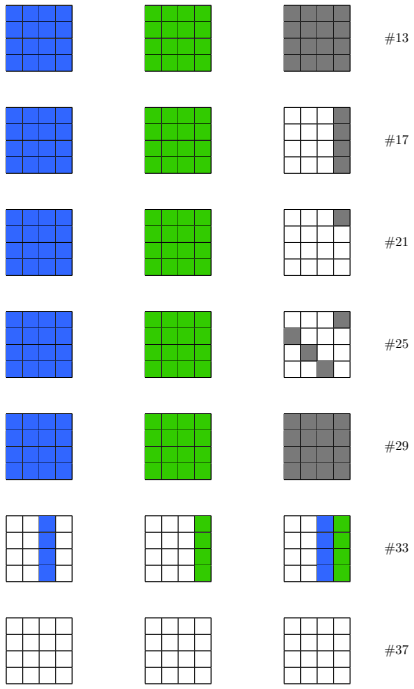

Figure 11. The nibbles of relevant states affected by $\Delta^K_{i_0}$ (left) and $\Delta^K_{i_1}$ (middle), in the matching phase of the star attack. In the rightmost, grey represents the ones affected by both.

This paper also presents two bicliques that are faster than the previous ones, although with a slight increase in data complexity. The first is a $(2 \cdot 4)-$dimensional biclique with a data complexity of 2^{24} and a time complexity of $2^{124.38}$. The second is an unbalanced $(4, 2 \cdot 4)-$dimensional biclique — the first of its kind on the FUTURE cipher. Unfortunately, it is not as fast as the former, requiring the same amount of data and a time complexity of $2^{124.80}$.

Future work includes the search for better bicliques. In this paper, we limited our search to bicliques where the Δs are independent from ∇s, but not necessarily from each other. This constraint was imposed due to current limitations in our *BicliqueFinder* software. Additionally, exploring star bicliques of various dimensions may also yield promising results. The same methodology can also be applied to other lightweight ciphers, such as the GIFT family.

Declarations

Authors' Contributions

Luis Kowada and Gabriel de Carvalho contributed to the conception of this study and analysis. Gabriel de Carvalho is the main contributor and writer of this manuscript. Luis Kowada and Gabriel de Carvalho participated in the validation of the study, review and final editing. Luis Kowada contributed to this work's methodology, and supervision. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no conflict of interests.

Availability of data and materials

The datasets generated, diagrams that are mentioned and the software developed during the current study are available in `https://github.com/Clique33/BicliqueFinder`.

References

Bellini, E., Gerault, D., Grados, J., Huang, Y. J., Makarim, R., Rachidi, M., and Tiwari, S. (2024). Claasp: A cryptographic library for the automated analysis of symmetric primitives. In Carlet, C., Mandal, K., and Rijmen, V., editors, *Selected Areas in Cryptography – SAC 2023*, pages 387–408, Cham. Springer Nature Switzerland. DOI: 10.1007/978-3-031-53368-6$_1$9.

Bogdanov, A., Chang, D., Ghosh, M., and Sanadhya, S. K. (2015). Bicliques with minimal data and time complexity for aes. In *Information Security and Cryptology-ICISC 2014: 17th International Conference, Seoul, South Korea, December 3-5, 2014, Revised Selected Papers 17*, pages 160–174. Springer. DOI: 10.1007/978-3-319-15943-0$_1$0.

Bogdanov, A., Khovratovich, D., and Rechberger, C. (2011). Biclique cryptanalysis of the full aes. In *Advances in Cryptology–ASIACRYPT 2011: 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings 17*, pages 344–371. Springer. DOI: 10.1007/978-3-642-25385-0$_1$9.

Chen, S.-z. and Xu, T.-m. (2014). Biclique key recovery for ARIA-256. *IET Information Security*, 8(5):259–264. DOI: 10.1049/iet-ifs.2012.0353.

Daemen, J. and Rijmen, V. (2013). *The design of Rijndael: AES-the advanced encryption standard*. Springer Science & Business Media. DOI: 10.1007/978-3-662-04722-4.

de Carvalho, G. *et al*. (2022). Generator sets for the selection of key differences in the biclique attack. In *Anais do XXII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*, pages 1–14. SBC. DOI: 10.5753/sbseg.2022.224083.

de Carvalho, G. *et al*. (2023a). Revisiting the biclique attack on the aes. In *Anais do XXIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*, pages 153–166. SBC. DOI: 10.5753/sbseg.2023.232855.

de Carvalho, G. C. and Kowada, L. A. (2020). The first biclique cryptanalysis of serpent-256. In *Anais do XX Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*, pages 29–42. SBC. DOI: 10.5753/sbseg.2020.19225.

de Carvalho, G. C., Neto, T. S., and do Rêgo Sousa, T. (2023b). Automated security proof of square, led and clefia using the milp technique. In *Anais do XXIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*, pages 445–455. SBC. DOI: 10.5753/sbseg.2023.232871.

Gupta, K. C., Pandey, S. K., and Samanta, S. (2022). Future: a lightweight block cipher using an optimal diffusion matrix. In *International Conference on Cryptology in Africa*, pages 28–52. Springer. DOI: 10.1007/978-3-031-17433-9$_2$.

Hong, D., Koo, B., and Kwon, D. (2011). Biclique attack on the full HIGHT. In *International Conference on Information Security and Cryptology*, pages 365–374. Springer. DOI: 10.1007/978-3-642-31912-9$_2$4.

İlter, M. B. and Selçuk, A. A. (2022). Milp-aided cryptanalysis of the future block cipher. In *International Conference on Information Technology and Communications Security*, pages 153–167. Springer. Available at:`https://eprint.iacr.org/2022/1398`.

McKay, K., Bassham, L., Sönmez Turan, M., and Mouha, N. (2016). Report on lightweight cryptography. DOI: 10.6028/nist.ir.8114.

Mondal, S. K., Rahman, M., Sarkar, S., and Adhikari, A. (2024). Yoyo cryptanalysis on future. *International Journal of Applied Cryptography*, 4(3-4):238–249. DOI: 10.1504/ijact.2024.138453.

Roy, H. S., Dey, P., Mondal, S. K., and Adhikari, A. (2024). Cryptanalysis of full round future with multiple biclique structures. *Peer-to-Peer Networking and Applications*, 17(1):397–409. DOI: 10.1007/s12083-023-01600-y.

Schrottenloher, A. and Stevens, M. (2023). Simplified modeling of mitm attacks for block ciphers: New (quantum) attacks. *IACR Transactions on Symmetric Cryptology*, 2023:146–183. DOI: 10.46586/tosc.v2023.i3.146-183.

Shi, D., Sun, S., Song, L., Hu, L., and Yang, Q. (2023). Exploiting non-full key additions: Full-fledged automatic demirci-selcuk meet-in-the-middle cryptanalysis of skinny. Available at:`https://eprint.iacr.org/2023/255`.

Tao, B. and Wu, H. (2015). Improving the biclique cryptanalysis of aes. In *Information Security and Privacy: 20th Australasian Conference, ACISP 2015, Brisbane, QLD, Australia, June 29–July 1, 2015, Proceedings 20*, pages 39–56. Springer. DOI: 10.1007/978-3-319-19962-7$_3$.

Xu, Z., Cui, J., Hu, K., and Wang, M. (2024). Integral attack on the full future block cipher. *Tsinghua Science and Technology*. DOI: 10.26599/tst.2024.9010007.

Appendix

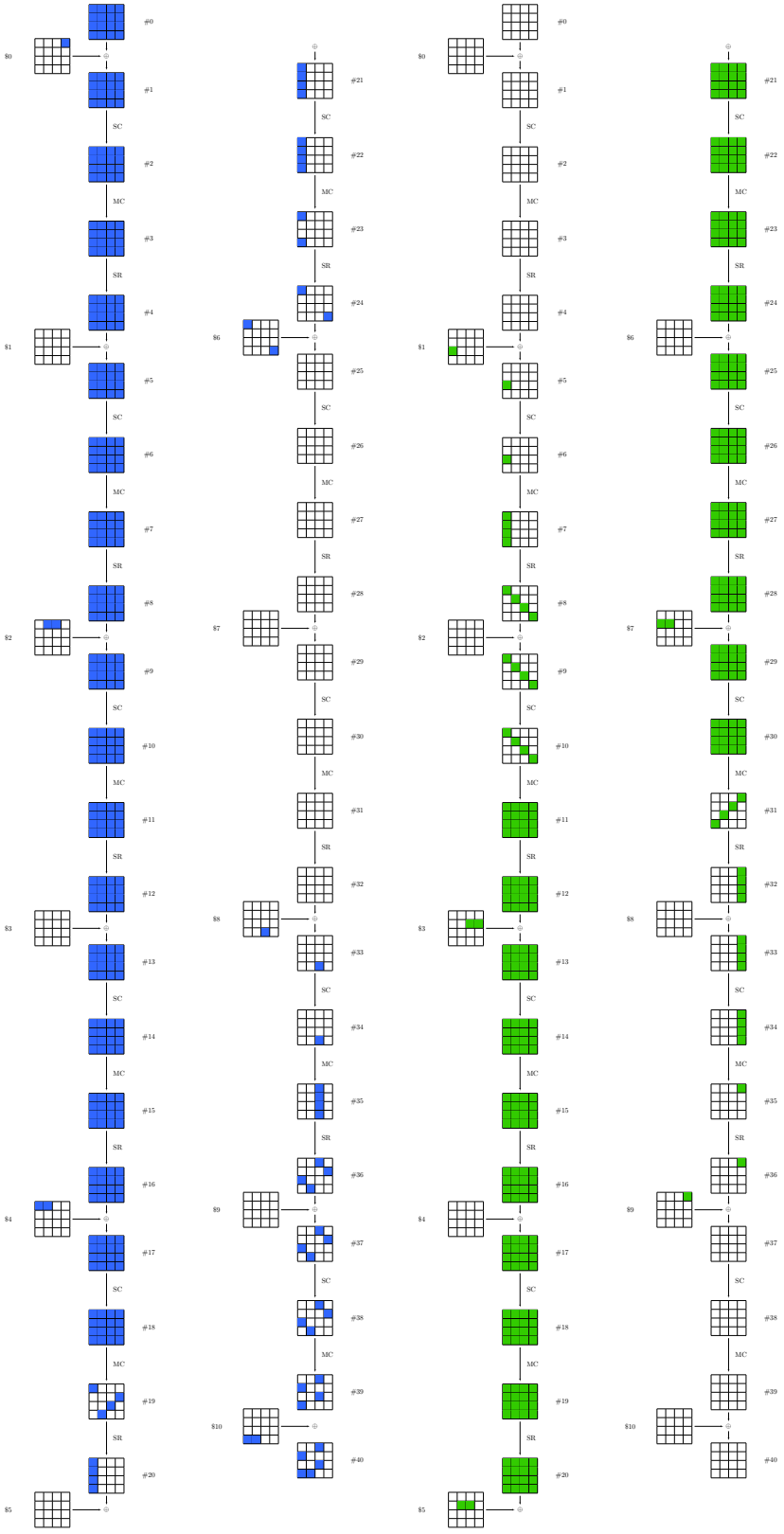

Figure 12. Complete propagation's of the $4-$dimensional biclique attack.

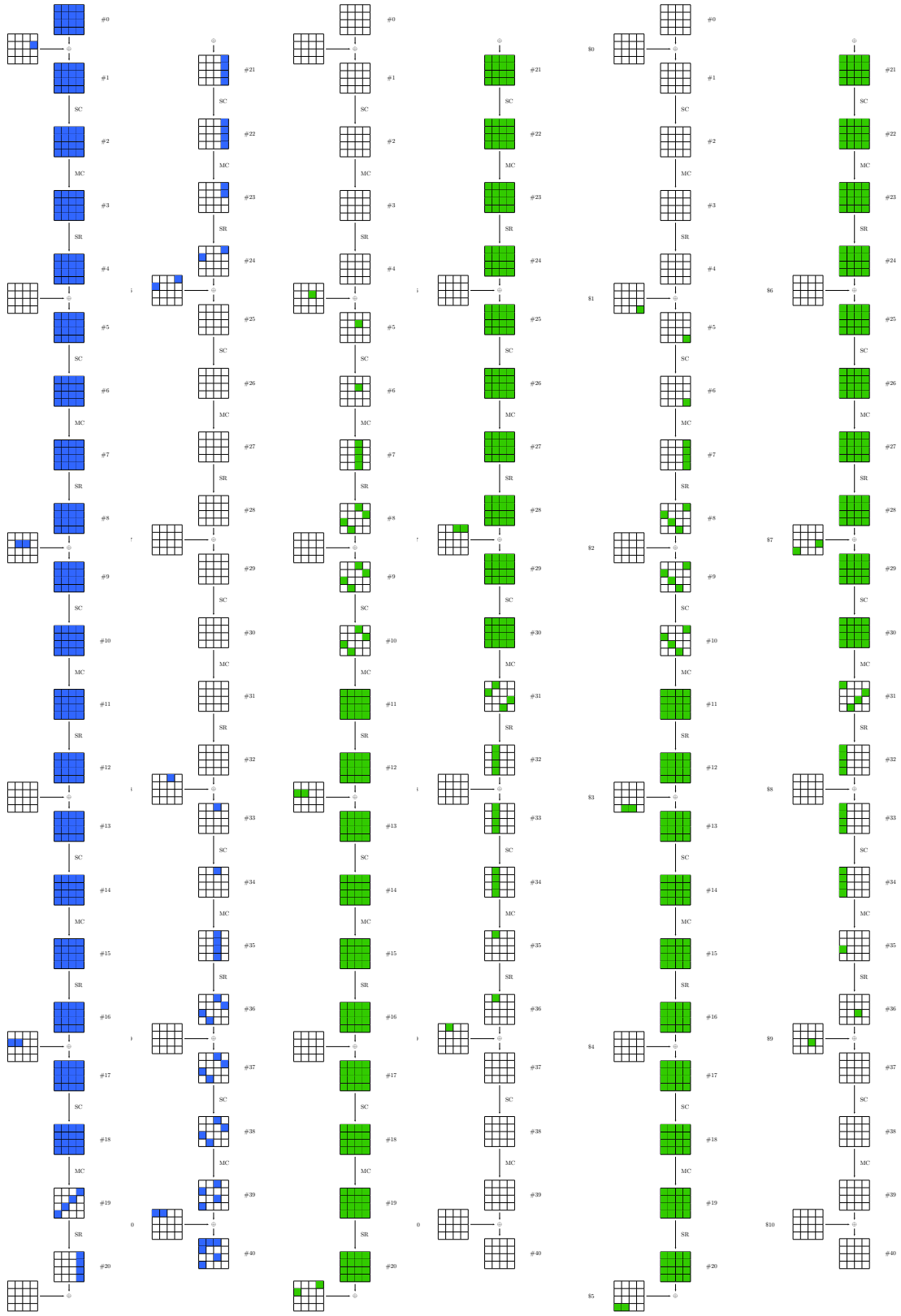

Figure 13. Complete propagation's of the $(4, 2 \cdot 4)-$dimensional biclique attack.

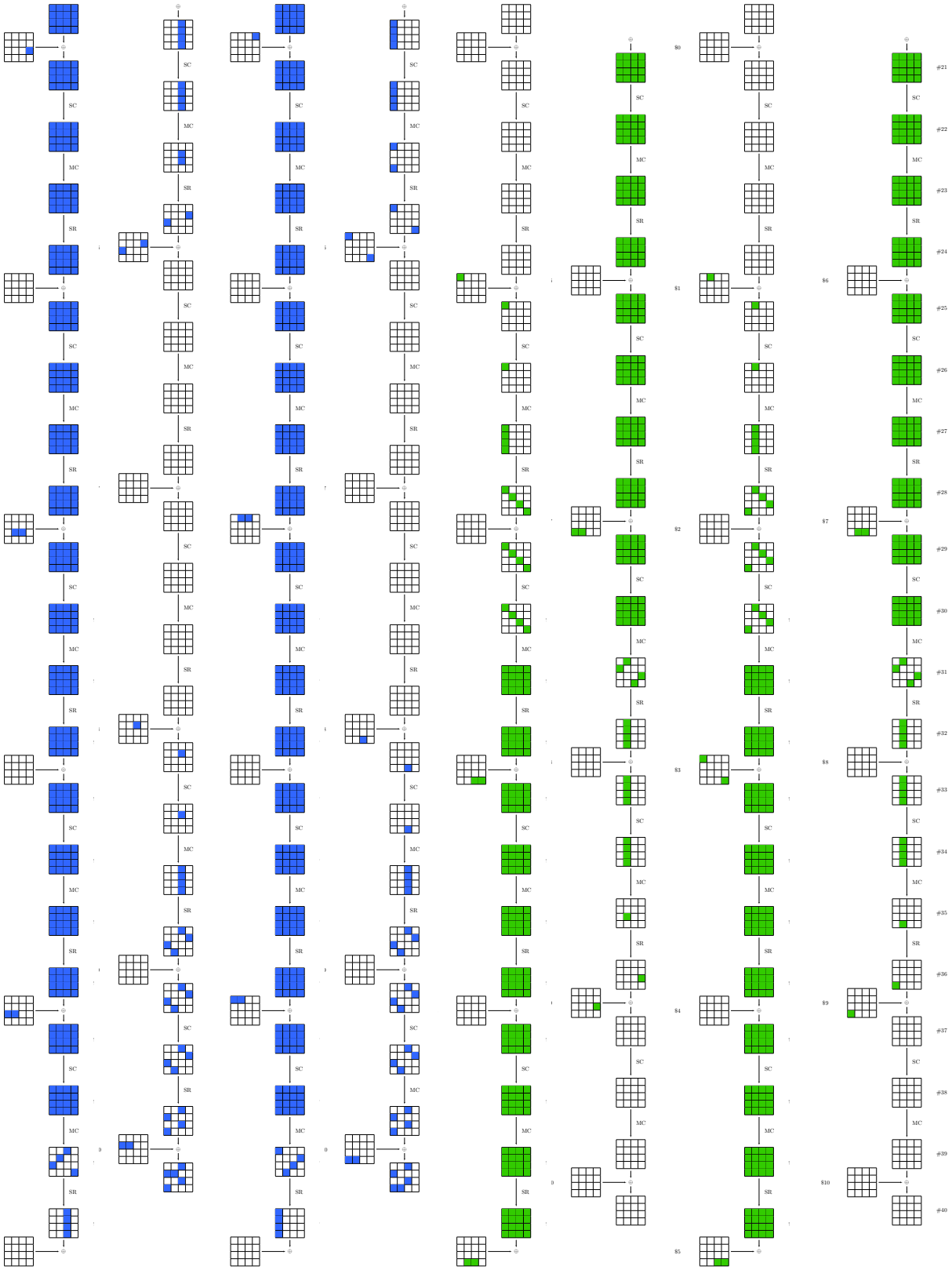

Figure 14. Complete propagation's of the $(2 \cdot 4)-$dimensional biclique attack.

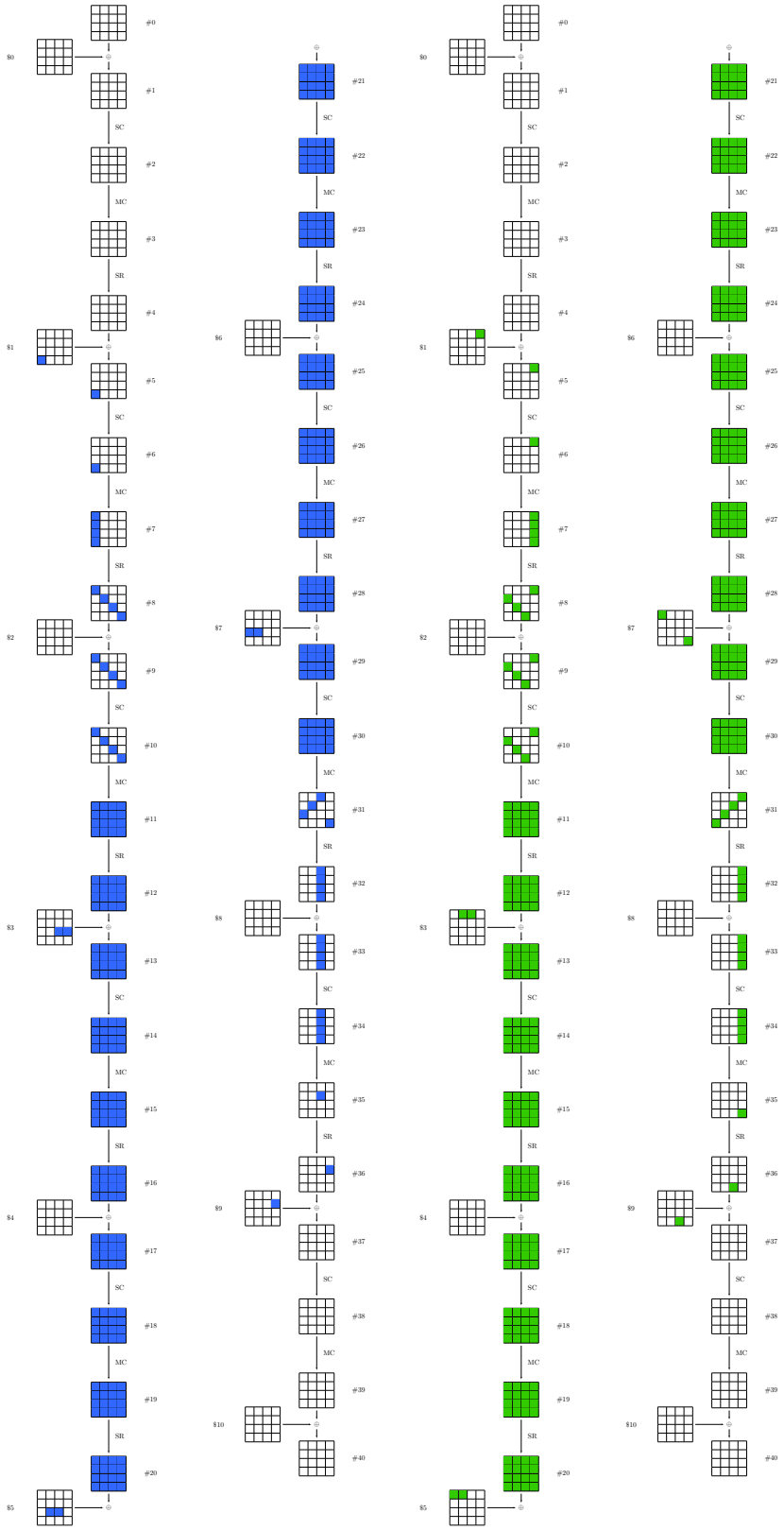

Figure 15. Complete propagation's of the star attack.