Bicliques for permutations: collision and preimage attacks in stronger settings

Dmitry Khovratovich
Microsoft Research

March 15, 2012

Abstract

We extend and improve biclique attacks, which were recently introduced for the cryptanalysis of block ciphers and hash functions. While previous attacks required a primitive to have a key or a message schedule, we show how to mount attacks on permutations with fixed or no parameters. To link the new methods with older ones, we introduce the concept of phantom schedule.

The new framework allows to convert preimage attacks into collision attacks and derive the first collision attacks on the reduced SHA-3 finalist Skein. We also demonstrate new preimage attacks on the reduced Skein and the output transformation of Grøstl. Finally, the sophisticated technique of message compensation gets a simple explanation with bicliques.

Keywords: Skein, SHA-3, hash function, collision attack, preimage attack, biclique, permutation, Grøstl.

1 Introduction

Meet-in-the-middle attacks have been known in cryptanalysis at least since the analysis of Double-DES [9], but got less attention in 90s and early 2000s because of more difficult key schedules in contemporary block ciphers. They regained prominence with the introduction of the splice-and-cut framework by Aoki and Sasaki for hash functions [2, 22]. Aoki and Sasaki considered various designs and demonstrated how to construct pseudo-preimages for compression functions based on block ciphers. Pseudo-preimages can be converted to regular preimages, though the procedure halves the advantage previously gained over brute force.

While the first splice-and-cut attacks were quite simple, they quickly became more sophisticated as cryptanalysts tried to increase the number of rounds broken [1, 23]. That number for the first attacks was determined by the length of *chunks* — two sections of a primitive each independent of its own set of key/message bits called *neutral bits*. For example, two DES calls in Double-DES are chunks each independent of half of the key. Later research showed how to start the attack with a sophisticated construction (so called *initial structure*) over several rounds to increase the total number of rounds in the attack [3, 23], which culminated in the concept of bicliques [15]. While initial structures relied on slow diffusion, bicliques do not need that condition. In turn, they translated the condition on internal states being suitable for meet-in-the-middle attacks to the requirements how these states map to each other under different parameter values.

Bicliques. Breaking the diffusion barrier, the biclique technique led to the first (and so far the only) attack on 8-round AES-128 [8], which does not involve any elements of the exhaustive key search. The attack, later called a long biclique, has influenced those reducing the security level of the full AES [8], Square [17], Kasumi [13], IDEA [14]. All these attacks need a small but noticeable number of operations to test a single, and in our opinion they have smaller potential. Indeed, even a single operation for each key implies a lower bound on the complexity which is not far from exhaustive search.

Also from the technical point of view, the use of bicliques in those settings is not much different from earlier use of initial structures.

From parametrized transformations to permutations. The key/message schedule plays a very active role in the biclique attacks. As we will show in Section 2, they help to construct colliding computations and enumerate N message candidates with only $2\sqrt{N}$ states.

However, there are special settings where an attacker does not have freedom in the schedule or it is just absent. For example, preimage attacks on blockcipher-based compression functions produce pseudo-preimages and require significantly more computations to produce real preimages. If an attacker avoids the second step and fixes the chaining value, he either faces a block cipher with a known plaintext/ciphertext pair (for the Davies-Meyer mode) or a single non-modifiable permutation (for the Matyas-Meyer-Oseas mode). Another example is the SHA-3 finalist Grøstl, whose output transformation converts x to Truncate$(x \oplus P(x))$, where P is a fixed permutation. Hence the translation of the biclique technique to permutations is quite promising.

A few attacks on permutations have appeared recently [21, 29] but make use of a simple version of initial structure only. This raises natural questions as if the more general concept of bicliques can be carried out to this setting and even if so whether the advantages of long bicliques can be used similarly to AES.

Collisions for the MMO-based primitives. While the Matyas-Meyer-Oseas (MMO) and Davies-Meyer (DM) modes are equally resistant to generic attacks [7], they are way more different when dedicated methods are considered. In the DM mode an attacker is able to manipulate the round injections, while in the MMO mode he is able to choose the input. From our point of view, famous collision attacks on the MD4/SHA family [5,28] demonstrate that the first setting is much more friendly to the attacker. Indeed, the most powerful collision search method — differential cryptanalysis — works with related-key characteristics in the DM mode, and with regular characteristics in the MMO mode. Devastating related-key attacks on AES [6] hint that the former setting is more suitable.

One should not be confused, however, by near-collision attacks on the compression function of Skein [4, 25], as they are essentially free-start collisions, i.e. they inject the difference in the chaining value or the tweak. Therefore, we conclude that mounting a regular collision attack on the hash function based on MMO is quite difficult. The very recent pseudo-collision attack [16] on Skein is a great step forward, as we discuss in the further text.

Our contributions

We introduce a new notion of *sliced biclique* as a modification of a regular biclique. The new concept helps to carry out the meet-in-the-middle attacks and the biclique technique to permutations without modifiable parameters. We also show a conversion to the regular biclique attack via *phantom schedule*, which replaces the absent modifiable parameters.

The applications are manyfold. First, we improve a very recent technique of finding pseudo-collisions with pseudo-preimages and show how to get regular collision attacks on the MMO-based primitives. We obtain the first collision attacks on the reduced round Skein hash function. The new attacks are also translated to new preimage attacks on Skein.

Then we consider the output transformation of the SHA-3 finalist Grøstl-256, which consists of a single permutation. In this setting we target the maximum number of rounds rather than lower complexity, and derive the first shortcut 6-round attack.

Finally, we simplify the neutral bit search procedure from earlier meet-in-the-middle attacks called message compensation. It appeared to be very ad-hoc and complicated tool and was difficult to check. It gets a clear interpretation as a sliced biclique, which will hopefully lead for better attacks.

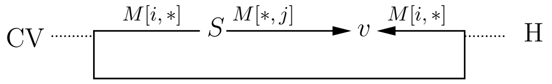
There is little difference in the application of the theory to key and message schedules, so let us call *parameters* both keys and messages.

2 Splice-and-cut attacks and bicliques

Splice-and-cut attacks [2, 22] were designed as a preimage search method. A simple splice-and-cut attack is applied to the Davies-Meyer-based compression function F:

$$F(CV, M) = E_M(CV) \oplus CV,$$

where CV is a chaining value, M is a message block, $E_K(\cdot)$ is a cipher. An attacker is given an n-bit hash value H and has to find a preimage M. The preimage search is organized as follows. The attacker selects a message set $\{M[i, j]\}$, an internal state S and an internal variable v such that v as a function of S in one direction does not depend on i, and in the other direction does not depend on j:



Then he takes an arbitrary value of S and computes v in the forward direction for all possible j (denoted by $\overrightarrow{v}_j$) and in the other direction for all possible i (denoted by $\overleftarrow{v}_i$), computing CV and using H on the way. The overlap of the resulting two sets yields preimage candidates which are tested on the full state width. The indices i and j typically belong to $[0;\ 2^d - 1]$ for some d, which yields the matching probability 2^{2d-n} for one message set, and the complexity 2^{n-d} for the pseudo-preimage search. To find a full preimage the adversary generates $2^{d/2}$ pseudo-preimages and matches one of them computing $2^{n-d/2}$ CVs out of the initial value. The total complexity is $2^{n-d/2+1}$, so only $d \geq 3$ provides an advantage over brute force.

The basic attack was carried out to other modes and even block ciphers. For the latter, the encryption oracle plays the role of the feedforward to link the input and the output. Though block ciphers are out of our scope, we note that the following concepts can be easily adapted for them.

A biclique is an extension for the first step of the attack, which is based upon an earlier informal concept of *initial structure* [3,23].. Instead of a single state S, a *biclique* is defined over a *sub-cipher* — a part of the primitive, typically several rounds long — and for a particular group of keys or messages that are subject to test. A biclique over f for parameters $\{M[i, j]\}$ is pair of state sets

$$\{Q_i\}, \{P_j\}$$

such that

$$Q_i \xrightarrow[f]{M[i,j]} P_j. \tag{1}$$

A biclique tests parameters $\{M[i, j]\}$ in the same way as in the basic attack. The matching variable v is computed in both directions:

$$P_j \xrightarrow{M[*,j]} v \xleftarrow{M[i,*]} Q_i. \tag{2}$$

The condition (1) guarantees that if $M[i, j]$ is a right value then the computations from P_j and Q_i meet in a biclique exactly as at the matching point. Those computations are depicted in Figure 1.

Because of non-ideal diffusion several rounds around v may be computed only partially. Those rounds that have to be computed in full are called *chunks*. The chunk length is usually limited by the number of rounds with independent parameter injections (16 in SHA-0/1/2, 1 in AES-128, 2 in AES-256).

The crucial property of a biclique is that it enumerates 2^{2d} parameters with only 2^{d+1} internal states. The value d is called *dimension* of a biclique, and the number of rounds in f — *length* of a biclique. A biclique of dimension 2, i.e. the one mapping four states to another four states with 16 parameters is depicted in Figure 1.

The computational advantage of a biclique attack is the same as in the basic attack, and hence is proportional to the dimension.

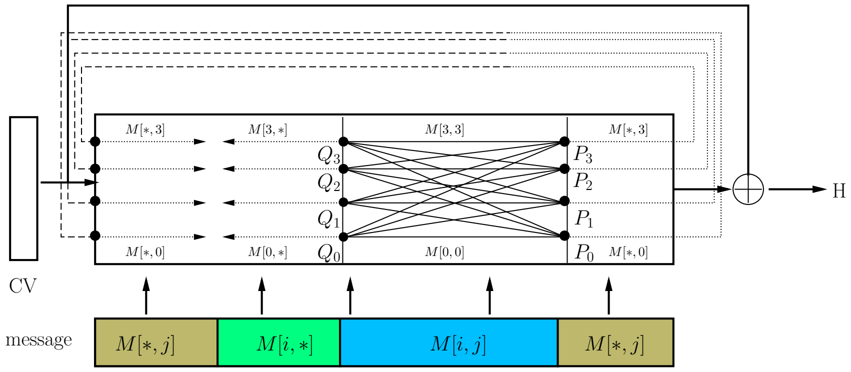


Figure 1: Biclique of dimension 2 for the Davies-Meyer-based compression function.

3 Bicliques for permutations

We have explained how to use bicliques to find a right value of a parameter and hence mount an attack. However, this gives no clue on what to do if there is no parameter. For instance, consider the Grøstl output transformations:

$$H = \text{Truncate}(x \oplus P(x)),$$

where P is a permutation. It is necessary to invert it in a preimage attack on Grøstl, but there is no parameter for the biclique (Equation (1)). Simple initial structures have been constructed [21, 29], but they strongly rely on slow diffusion.

First, we realize that a solution is a (input) state, not a parameter. Our next idea is to describe as many possible solutions, i.e. states, with a biclique. We adapt the biclique concept as follows:

- Do not require that the computations in a biclique converge. Instead, minimize the difference between them.
- Select the difference in vertices so that the difference in the matching variable is predictable (preferably zero).

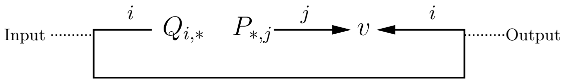
Let us introduce a more formal definition. Let E be a permutation with sub-cipher f, and assume E is wrapped with a kind of feedforward or oracle such that some variable v can be computed from internal states of f directly or via the feedforward/oracle. In contrast to the regular biclique, we include the matching variable into a definition. A *sliced biclique* over f for v is a pair of sets 2^{2d} states each:

$$Q_{i,j} \underset{f}{\to} P_{i,j}.$$

such that $\forall i, i_1, i_2, j, j_1, j_2$ pairs

$$(P_{i_1,j}, P_{i_2,j}) \text{ and } (Q_{i,j_1}, Q_{i,j_2}) \text{ are indistinguishable when computing } v. \qquad (3)$$

The property (3) essentially means that for every i all $Q_{i,j}$ belong to some class of equivalence, which we denote by $Q_{i,*}$. The same holds for states P, which are grouped into classes $P_{*,j}$. Therefore, we compute only



for each i and each j separately. If one of biclique states is a solution to the main problem (the right key or valid preimage), it will be detected in the matching phase (Figure 2). Adapting the notation from the regular biclique attack, we check if

$$\exists i,j : \overrightarrow{v_{*,j}} \overset{?}{=} \overleftarrow{v_{i,*}}, \qquad (4)$$

efficiently, instead of checking if exist i, j such that

$$\overrightarrow{v_{i,j}} \overset{?}{=} \overleftarrow{v_{i,j}}. \tag{5}$$

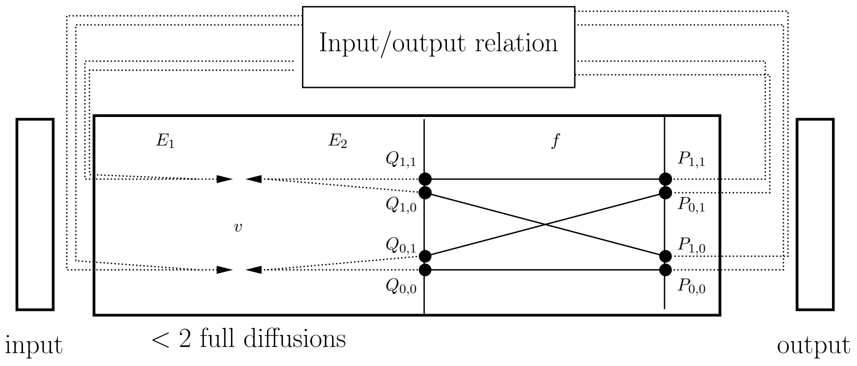


Figure 2: Sliced biclique for a permutation.

Differential view. Property (3) implies that

- The differences $P_{i_1,j} \oplus P_{i_2,j}$ and $Q_{i,j_1} \oplus Q_{i,j_2}$ are small;
- Diffusion properties of the primitive limit the difference expansion so that there is no difference in the matching variable.

Therefore, a sliced biclique attack may be illustrated with two groups of differential trails. The first group starts in states Q and contains for each i all possible differences $Q_{i,j_1} \oplus Q_{i,j_2}$. Then we follow the difference expansion and control that the difference in the matching variable v is zero. Actually, the latter condition can be relaxed to any fixed value. Similarly, we construct differential trails from P to v.

Construction algorithms. Algorithms for sliced bicliques are inherited from algorithms for regular bicliques [15]. Apparently, the absence of parameters does not modify general principles of the construction. All the algorithms employ a differential view on bicliques.

Bicliques from non-interleaving trails. These algorithms are the simplest. A sliced biclique of dimension 1 requires two differential trails for f that do not share active non-linear components: one that starts with a difference Δ in the forward direction, and the other one that starts with ∇ in the backward direction (Figure 3). Then a sliced biclique is constructed as follows:

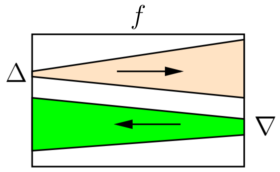


Figure 3: Non-interleaving differential trails for f. The forward trail starts with Δ, the backward trail ends with ∇.

1. Take an arbitrary state $Q_{0,0}$.
2. Derive $Q_{0,1} = Q_{0,0} \oplus \Delta$.

3. Compute $P_{0,j} = f(Q_{0,j})$ for $j = 0, 1$.
4. Derive $P_{1,j} = P_{0,j} \oplus \nabla$ for $j = 0, 1$.
5. Compute $Q_{1,j} = f^{-1}(P_{1,j})$ for $j = 0, 1$.

We automatically get $Q_{1,1} \oplus Q_{1,0} = \Delta$ because the trails do not interleave. This algorithm easily generalizes to multiple Δ and ∇: we simply use multiple Δ's in step 2, ∇'s in step 4, and change i and j accordingly. For the proof of its correctness we refer to [15].

Bicliques from interleaving trails are more complicated. We adapt two different strategies. First, we consider regular bicliques constructed from interleaving trails following [8, 15]. Then we select a part of the biclique sub-cipher that does not invoke any parameter, and hence get a sliced biclique. We apply this strategy to Skein. Alternatively, we split the sub-cipher into independent blocks, e.g. Super S-boxes, and consider bicliques of dimension 1 for each block separately, assuming that they can be generated with little amortized cost. We apply this strategy to Grøstl.

Degrees of freedom. Since there is no key/message schedule, the amount of freedom is limited to the internal state size. A cryptanalyst may erroneously impose too much restrictions on the bicliques so that there will be few of them, and no solution would be eventually produced. An attacker should ensure that at least one solution appears in a biclique, e.g., by a counting argument.

4 Conversion to the regular biclique attack with a phantom parameter

Here we show a conversion from a sliced biclique to a regular biclique with an alternative description of the permutation. The idea is to replace the difference in states with an injection of a phantom parameter. First, we assume that the differences in Q are located at the same l bit positions. Let us introduce state differences which are non-zero at these positions:

$$q_j = (0, 0, \ldots, j, 0, \ldots, 0); \qquad p_i = (0, 0, \ldots, i, 0, \ldots, 0).$$

Then we assume the following:

$$\forall i \; Q_{i,j_1} \oplus Q_{i,j_2} = q_{j_1} \oplus q_{j_2}; \qquad \forall j \; P_{i_1,j} \oplus P_{i_2,j} = p_{i_1} \oplus p_{i_2}.$$

Suppose that i and j have bitsize l, and consider $L = (l_1 || l_2)$ as a $2l$-bit parameter. Then prepend f with two consecutive XORs of l_1, and append f with two consecutive XORs of l_2:

$$\text{Before:} \qquad S \xrightarrow[f]{} S';$$

$$\text{After:} \quad S \xrightarrow[\oplus l_1]{} \xrightarrow[\oplus l_1]{} \xrightarrow[f]{} \xrightarrow[\oplus l_2]{} \xrightarrow[\oplus l_2]{} S'.$$

Evidently, the resulting transformation is equivalent to the original. However, now the middle part admits the construction of a regular biclique:

$$S \xrightarrow[\oplus l_1]{} \underbrace{\xrightarrow[\oplus l_1]{} \xrightarrow[f]{} \xrightarrow[\oplus l_2]{}}_{f'} \xrightarrow[\oplus l_2]{} S';$$

$$\text{Biclique:} \qquad Q_i \xrightarrow[f']{l_1 = j,\, l_2 = i} P_j,$$

where

$$\forall i, j \; Q_i = Q_{i,j} \oplus q_j, \quad P_j = P_{i,j} \oplus p_i.$$

The new construction uses a fictive, or *phantom* parameter, and provides an alternative view on the construction of a sliced biclique. We stress, however, that the role of this parameter and the one of a key/message in regular attacks are different. While the first is used only for simplification, the second one forms the search space.

5 Framework of new preimage and collision attacks on Skein

The SHA-3 finalist Skein [10] employs the Matyas-Meyer-Oseas mode to construct a compression function. It takes the block cipher Threefish (denoted by $E_K(\cdot)$) and computes:

$$F(CV, T, M) = E_{CV,T}(M) \oplus M,$$

where CV is the chaining value, and T is the tweak value. Due to difficulties in mounting collision attacks on the MMO mode, the only published attack on the Skein hash function is the preimage attack [15] based on regular bicliques. The parameter $M[i, j]$ in the biclique equation (1) comes from the chaining value. As a result, the attacker generate different CV's for each pseudo-preimage in the first step of the attack, and has to use another standard meet-in-the-middle procedure to get full preimages (Section 2). The first step must have complexity 2^{n-3} or smaller to yield an advantage over brute-force, which implies that only bicliques of dimension 3 or larger should be used.

We are now able to fix the chaining value and attack the resulting permutation with the concept of sliced bicliques. As a result, we can generate full preimages without the pseudo-preimage step. The complexity drops to 2^{n-d} instead of $2^{n+1-d/2}$, and restrictions on the biclique dimension do not hold anymore. Meet-in-the-middle attacks on the first call of the MMO and similar modes exist [21, 29], but do not use the long biclique approach yet, and were not applied to Skein.

Collision attacks. A more interesting property of the MMO mode comes out if we consider a very recent pseudo-collision attack which uses regular bicliques [16]. The method produces pseudo-collisions out of the splice-and-cut preimage attacks to as follows. Assume we have a biclique of dimension d and are able to match deterministically on l hash value bits. Then the adversary generates partial pseudo-preimages to a hash value with those l bits equal to an arbitrarily chosen constant h. Hence 2^{2d-l} l-bit partial pseudo-preimages to h can be generated with cost 2^d. Note that they collide on l output bits. The adversary generates $2^{n/2-l/2}$ such preimages and expect a pair of them to collide on the remaining $(n - l)$ bits by the birthday paradox. Since chaining values and schedule inputs are not fixed in the attack, this yields a pseudo-collision with the expected complexity $2^{(n/2-l/2)+(d)-(2d-l)} = 2^{n/2+l/2-d}$. The approach both for DM and MMO modes.

The optimal d satisfies the equation $d = 2d - l$, which implies $d = l$. The attack is optimal if all preimages are generated out of a single biclique, which implies

$$l = n/2 - l/2 \Leftrightarrow l = n/3.$$

Hence the minimum complexity of collision search is $2^{n/3}$.

Again,the chaining value can be fixed in the MMO mode if we apply the sliced biclique concept. Then we can generate real collisions instead of pseudo-collisions. However, we can break fewer rounds compared to the pseudo-collision attacks because sliced bicliques can not benefit from long chunks.

Memory. The default version of the attack requires to store all the pseudo-preimages generated, which makes the memory complexity be of the same order as the time complexity. However, as the preimage step is non-deterministic, we can employ memoryless collision search methods [26], which multiply the time complexity by a small constant. Therefore, all the attacks described in the further text, except for the marginal ones, have memoryless equivalents.

6 Collision attacks on Skein

Here we present the first collision attacks on the reduced Skein hash function. The MMO mode is difficult for collision attacks as the round injections come from the chaining value, and the adversary is unable to construct local collisions, apply message modification techniques, etc.. As a result, previous attacks on Skein [4, 25] dealt with the compression function only. The attacks are grouped according to the number of rounds covered by a biclique. Though we aim for the maximal dimension and the number of rounds attacked, for clarity we do not push the concept to the extreme and try to avoid complicated bicliques. Hence our attacks can be improved in the future.

Short description of Skein. Skein-512 [10] has an internal state of eight 64-bit words, while Skein-256 has a state of four words. We denote the state words by $S^0, S^1, \ldots$. Both versions have 72 rounds, and Skein-512-256 just truncates the output of Skein-512 to 256 bits. Each round of Skein-512 consists of four (two in Skein-256) simple transformations called MIX:

$$\begin{aligned} y_0 &= x_0 + x_1 \\ y_1 &= (x_1 \lll_{R_{(d \bmod 8)+1,j}}) \oplus y_0 \end{aligned}$$

where R is a constant depending on the round number d. The invocations of MIX are followed by a word permutation and, every four rounds, an injection of a linear function of the chaining value and the tweak.

The only published attack on the Skein hash function is a preimage attack [15] on 22 rounds of Skein-512.

6.1 Skein-512

As few as three rounds of Skein-512 are required to diffuse the contents of a single word to the full state. As a result, the initial structure technique would be bounded by two rounds at most. We present bicliques that are capable to cover up to 8 rounds.

2-round biclique. Our first examples deal with smaller number of rounds and bicliques of high dimension. As a result, the attacks have a non-marginal advantage over brute-force.

We use a simple algorithm with non-interleaving trails (Section 3) because of high dimension. For dimension 64 we consider 2^{64} possible differences in a 64-bit word. The position of this word is different for P and Q and is depicted at Figure 4. We additionally specify the difference in P:

$$P_{i,j} = P_{0,j} + (i, 0, 0, \ldots, 0) \tag{6}$$

Only three rounds are required to diffuse a 64-bit word onto the full state. Hence we would expect that the matching part would two rounds only in both directions. However, we can extend it by one round with the indirect partial matching [1]. Let the matching variable v be the state word S^0 three rounds after a biclique. As can be seen from Figure 4, Equation (6) implies that:

$$S^0_{i,j} = S^0_{0,j} + i.$$

We can not obtain Equation (4) directly, but we can derive a similarly efficient variant. Indeed, Equation (5) resolves into

$$\overrightarrow{v_{0,j}} \stackrel{?}{=} \overleftarrow{v_{i,*}} - i,$$

which is a regular matching condition with 64-bit filtering. Hence we generate 2^{64} 64-bit partial preimages with cost 2^{64}. Full 7-round collisions are found within $2^{(512-64)/2} = 2^{224}$ such partial preimages with the cost 2^{224}.

Collisions on the smaller number of rounds can be found with bicliques of dimension 128. Though they can be two rounds long as well, the matching part diffusion takes one round less in each direction, which gives only a 5-round collision. The complexity is 2^{192}.

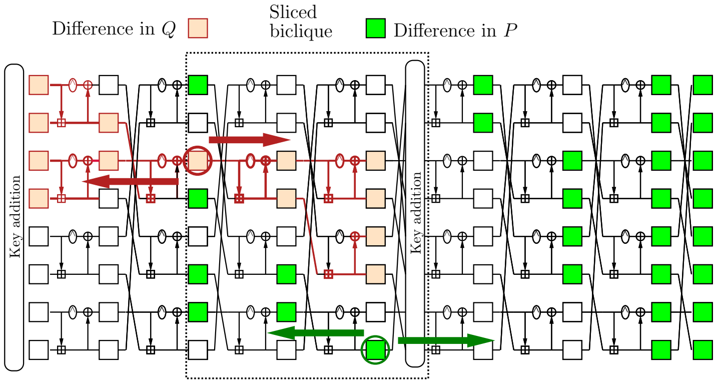


Figure 4: Sliced biclique of dimension 64 in 7-round Skein-512.

3-round biclique. If we decrease dimension to less than a half the word size, the diffusion will take more than three rounds. As a result, we can construct 3-round bicliques of dimension close to 20. We use an algorithm with non-interleaving trails with some modifications.

The exact values depend on the difference position and rotation constants. The maximum dimension is obtained if we consider rounds 5–7 (or $8k$ rounds further), as the rotation constants in these rounds allow for the maximum number of bits to not interact. We consider a 19-bit difference in Q in bits 63-45 of word 0, and a 19-bit difference in P located in bits 18-0. We additionally require that the word S^0 be equal to 0 in states P to prevent carries from interleaving with trails starting from Q. The length of the matching part decreases as the dimension grows. We have checked the diffusion on a PC and figured out that it covers 8 rounds for $d = 19$ and 10 rounds for $d = 8$. Therefore, we can generate 2^{19} 19-bit partial preimages for 11 rounds with the total collision attack complexity of 2^{245}. The other values are given in Table 1.

4-round biclique. We take a biclique the preimage attack on Skein [15], where it covers 4 rounds with two key additions. Removing these additions, we get a sliced biclique of dimension 3. We are able to use the same estimates of the diffusion, so the number of rounds decreases by the chunk length, i.e. by 8 rounds. Hence we get a 14-round partial preimages with complexity 2^3. Full collisions are found with complexity $2^{254.5}$.

Longer bicliques. Bicliques of dimension 1 can be constructed up to 8 rounds, but the advantage over brute-force attacks is really marginal. The number of rounds covered is close to 20, and the complexity grows to 2^{255}.

6.2 Skein-256

Diffusion in Skein-256 generally needs one round less to cover the full state. As a result, collision attacks on Skein-256 with bicliques of the same dimension lag 2-3 rounds behind the attacks on Skein-512. For instance, bicliques of dimension 64 and 128 cover one round only, and the matching part is two rounds shorter. This results in 2-round collisions with complexity 2^{85} and 4-round collisions with complexity 2^{96}.

Bicliques of smaller dimension are less sensitive to the smaller state size. Hence the low-dimension attacks for Skein-512 lose two rounds when being translated to Skein-256 (Table 1).

Skein-256		Skein-512	
Rounds	Complexity	Rounds	Complexity
2	2^{85}	5	2^{192}
4	2^{96}	7	2^{224}
8	2^{120}	11	2^{245}
9	2^{124}	13	2^{252}
12	$2^{126.5}$	14	$2^{254.5}$
18	2^{127}	20	2^{255}

Table 1: Collision attacks on reduced Skein with large memory requirements. Memoryless attacks add a small constant to the exponent.

7 Preimage attacks on Skein

One would expect that new preimage attacks on Skein follow directly from partial-preimage attacks described in the previous section. However, preimage attacks require significantly more pseudo-preimages ($O(2^n)$ compared to $O(2^{n/2})$) than collision attacks do. As a result, only the attacks that do not utilize too much freedom may be converted to preimage attacks. This reduces the applicability of long bicliques. Indeed, since the message and the chaining value have the same length, only one preimage on average is expected for the first call of the compression function. Hence only bicliques that cover all possible states are relevant in this attack, and they are generally short.

Still, the wide-pipe design Skein-512-256 becomes vulnerable as the number of preimages for the last call is about 2^{256}. Hence longer bicliques can be used.

2-round biclique. Biclique of dimension 64 based on non-interleaving trails. We get a 7-round attack with complexity 2^{192}.

3-round biclique. Biclique of dimension 19 based on non-interleaving trails with additional constraints on the state values. We get a 10-round attack with complexity 2^{237}.

4-round biclique and longer. Biclique of dimension 3 based on interleaving trails. We get a 14-round attack with complexity $2^{251.4}$. The 8-round biclique has dimension 1 and is based on strongly interleaving trails. We get a marginal attack on 21 rounds with complexity $2^{254.3}$.

Rounds	Complexity
7	2^{192}
10	2^{235}
14	$2^{251.4}$
21	$2^{254.3}$

Table 2: Preimage attacks on Skein-512-256.

8 Certificational preimage attack on the Grøstl output transformation

Grøstl [11] is a SHA-3 finalist with a compression function not based on a block cipher. It invokes two permutations P and Q, both AES-based, and updates the chaining value CV as follows:

$$CV \leftarrow CV \oplus Q(M) \oplus P(M \oplus CV),$$

where M is a message block. The final call of the compression function is followed by the output transformation

$$F(x) = \text{Truncate}(x \oplus P(x)),$$

where the truncation operation takes half of the state to get 256- and 512-bit outputs. Hence Grøstl-256 operates on a 512-bit state and permutations P and Q, and Grøstl-512 operates on a 1024-bit state.

Permutations P and Q follow the AES design with very similar operations: SubBytes, ShiftBytes, MixBytes (8-byte analogue of MixColumns), and AddRoundConstant. The ShiftBytes operation in Grøstl-256 rotates i-th row by i positions to the left; details of the other operations are irrelevant for our attack. The sequence SubBytes–ShiftBytes–MixBytes–AddRoundConstant–SubBytes is equivalent to 8 (for Grøstl-256) parallel 64-bit Super S-boxes [12]. Due to the design simplicity, Grøstl has been the target of numerous cryptanalytic attacks [18, 20, 24], though only few of them violated collision or preimage resistance of the hash function [19, 29]. The paper [29] addresses virtually the same problem as we do, and obtains preimage attacks on the 5-round version of the compression function, including the preimage attack on the 5-round output transformation.

To run a preimage attack, and the first preimage attack in particular, it is desirable to invert the output transformation of Grøstl. As it is also claimed to be one-way, it serves as a natural target for sliced biclique attacks.

We adapt a differential view as it provides a simple explanation of the attack in differential trails, making it similar to both rebound attacks [18] and recent biclique attacks on AES. The main distinction is that there is no round without a difference because there is no schedule. However, the difference expansion in the outbound phase must be deterministic unless we have additional degrees of freedom in the inbound phase.

Here we present a 6-round attack on Grøstl-256 only. The same approach holds for Grøstl-512, but due to the space constraints we leave it for the future work.

6 rounds. The trail for the 6-round attack on Grøstl-256 is depicted in Figure 5. We denote the internal states from #1 to #13. We construct a sliced biclique for the Super S-box layer covering states #5-#8, with the matching point in the last MixBytes transformation of round 6. We choose the difference in P and Q states of a biclique as an expansion of a 1-byte difference in states #9 and #4, respectively. These 1-byte differences expand to 8-byte differences in states #12 and #13. The matching condition is a linear function of the bytes not affected by the differences.

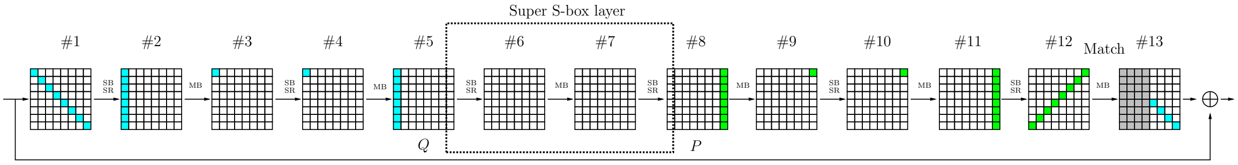


Figure 5: 6-round trail for the preimage attack on the Grøstl-256 output transformation.

Bicliques are constructed as follows. We specify two differences in #4, which must not be equal to each other, and derive the differences Δ_1 and Δ_2 in #5. Similarly we derive differences ∇_1 and ∇_2 in #8. Our goal is to find the states Q, P for a sliced biclique, which satisfy the following equations:

$$Q_{0,0} \oplus Q_{0,1} = \Delta_1; \quad Q_{1,0} \oplus Q_{1,1} = \Delta_2; \tag{7}$$

$$P_{0,0} \oplus P_{1,0} = \nabla_1; \quad P_{0,1} \oplus P_{1,1} = \nabla_2. \tag{8}$$

These equations are reformulated for each Super S-box separately, and solutions are found independently by exhaustive search and then concatenated with a total complexity around 2^{70}. We refer to the long biclique attack on AES [8], which gives a detailed description of the algorithm. We also note that the equations (7) describe a *boomerang quartet* [27], and the probability estimates also follow from the theory of boomerang attacks.

The complexity is amortized as follows. Each Super S-box has 7 inactive input s-boxes. There exist $2^{56-8} = 2^{48}$ alternative values for them which do not affect the active output S-box. Hence we can generate $2^{48\cdot 8} = 2^{392}$ sliced bicliques out of a single one. As the hash value contains 256 bits only, we have enough freedom for the attack. For each biclique, i.e. 2^2 states, we recompute only a portion of the S-boxes in each round, with $2 \cdot (8 + 16 + 2 + 7 + 56 + 8) = 194$ S-boxes or 2^{-3} calls of the permutation. Hence the amortized cost of a single state test is 2^{-5}, and the total attack complexity is 2^{251}.

9 Message compensation

The message compensation procedure [1,15] instructs how to select message groups in the splice-and-cut attack in case of a strong, nonlinear message schedule. Existing applications are very ad-hoc and complicated. It is possible, however, to give a unified view on the message compensation problem and existing solutions with bicliques for permutations. The biclique concept also allows for longer chunks in the MITM attacks because a more sophisticated interaction of message differences is allowed.

The message schedule in the SHA family is the generalized Feistel structure over 16 words. Each round updates one word and shifts the other words to the left hence giving space for the new word. Denote the round messages by $W_1, \ldots, W_r$. The adversary has to select the message group $M[i, j]$ so that the backward chunk messages are independent of j and the forward chunk messages are independent of i.

The adversary chooses the rounds for chunks in advance, supposedly the backward chunk in rounds b_1–b_2 and the forward chunk in rounds f_1–f_2. Denote the message group to be tested by $M[i, j]$. Then he needs that:

1. For every i backward message words $W_{b_1}, \ldots, W_{b_2}$ are identical for all $M[i, j]$;

2. For every j forward message words $W_{f_1}, \ldots, W_{f_2}$ are identical for all $M[i, j]$.

These restrictions well fit into the sliced biclique concept. The states $Q_{i,j}$ and $P_{i,j}$ of the biclique correspond to internal state of the message schedule in rounds $b_2 + 1$ and $f_1 - 1$ for messages $M[i, j]$. The requirements (1), (2) transform into the restriction on the diffusion of the difference between input as well as output states.

Hence the proper construction of a message group may look as follows. We construct a (sliced) biclique in rounds $(b_2 + 1)$–$(f_1 - 1)$ so that

- the difference between $P_{i_1,j}$ and $P_{i_2,j}$ does not affect $W_{f_1}, \ldots, W_{f_2}$;

- the difference between $Q_{i_1,j}$ and Q_{i,j_2} does not affect $W_{b_1}, \ldots, W_{b_2}$.

One may also apply the concept of the phantom schedule to convert a sliced biclique into a regular one for possible clarity.

In practice, the message compensation, as well as its interpretation in bicliques, does not fix the full message state within a biclique. It is possible if some message words are not updated in the biclique rounds. Then a cryptanalyst may leave those words undefined and use it subsequently to amortize the construction cost of the schedule biclique or the state biclique.

Constants in the message compensation [1, 15] are interpreted as internal variables that define a biclique. Message words for chunks are defined as functions of those constants, which can be shown to be independent of particular neutral bits. This view is no longer necessary, since in the biclique concept the independence follows from the form of the differential trails outside of a biclique.

Message compensation in the biclique attack on SHA-256

Let us consider the recent attack on SHA-256 [15], where a biclique is constructed for rounds 17–22, and chunks are located in rounds 2–16 and 23–36. An interpretation of the message compensation within the biclique framework will be as follows.

First, we fix the partition of SHA-256 into biclique and chunk rounds. The message schedule of SHA-256 follows a generalized Feistel structure with 16 words. We have to select the message group $M[i, j]$ so that $W_2, \ldots, W_{16}$ do not depend on j, and $W_{23}, \ldots, W_{36}$ do not depend on i. In terms of biclique, this is equivalent to the following conditions:

- Difference in the states Q (green) does not diffuse to words $W_2, \ldots, W_{16}$. As can be seen in Figure 6 (left), this condition is satisfied if the difference is concentrated in the rightmost word after round 17.

- Difference in the states P (lightblue) does not diffuse to words $W_{23}, \ldots, W_{36}$. As can be seen in Figure 6 (left), this condition is satisfied if the difference is concentrated in word W_{22} after round 29.

To describe the rest with a regular biclique, we introduce two phantom injections: P^F, which targets the active bits in Q (green) before round 18, and P^B, which targets the difference in P after round 29. Let us construct truncated differentials based on those injections. The differential trails interleave only in modular additions and hence can be considered independent modulo 2^{32}. This yields a biclique in the message schedule for rounds 18–29 (Figure 6, right).

To summarize, a message group to be tested is constructed in the biclique framework as follows. Fix an arbitrary message state between the injections of P^B and compute for all P^B the states between the injections of P^F. Take one of those states and compute for all P^F all possible states between the injections of P^B. Going beyond the phantom injections, we derive the full message group $M[i, j]$.

The biclique we have constructed has a large amount of bits that can amortize the construction cost. For instance, a modification in word W_{18} results in a pre-determined addition modulo 2^{32} to several message words. Constants in the original message compensation are also neutral words.

10 Conclusions

We have introduced sliced bicliques as a new tool for the analysis of permutations in the context of preimage and collision attacks. We have demonstrated that the advantage in the number of rounds from the long biclique idea can be obtained also for permutations. The application of our concept to different design has interesting consequences.

First, our collision attacks on Skein demonstrate that the MMO mode may not be as resistant to collision attacks and the differential cryptanalysis in particular as it was considered. The fundament of our attacks is the new pseudo-collision search technique that has been recently introduced. Though we employ some elements of differential cryptanalysis, the details are completely different from the famous collision attacks on the SHA family. Hence we suppose that the potential of differential cryptanalysis for high-profile hash functions has not been exhausted.

Secondly, our preimage attacks on the Grøstl output transformation show that the concept of the Super S-box contributes not only to the biclique attacks on the designs with the key schedule (AES), but also on the ones without the schedule. We expect this type of attack to progress alongside with the future techniques for the Super S-box.

Finally, we explained the message compensation in the biclique terms. We expect that the designers of future meet-in-the-middle attacks on SHA-2 will be able to provide a compact two-step description of their results. First, a biclique in the schedule is constructed, and secondly, it is used to construct a biclique in the state. We are looking forward to new techniques that would combine these bicliques in an optimal way.

We leave a significant amount of targets for the future work. 7-round Grøstl-256, 9- and 10-round Grøstl-512, Whirlpool, BLAKE are natural targets. Construction of bicliques of high dimension out of interleaving trails remains an open problem.

References

[1] Kazumaro Aoki, Jian Guo, Krystian Matusiewicz, Yu Sasaki, and Lei Wang. Preimages for step-reduced SHA-2. In *ASIACRYPT'09*, volume 5912 of *Lecture Notes in Computer Science*, pages 578–597. Springer, 2009.

[2] Kazumaro Aoki and Yu Sasaki. Preimage attacks on one-block MD4, 63-step MD5 and more. In *Selected Areas in Cryptography'08*, volume 5381 of *Lecture Notes in Computer Science*, pages 103–119. Springer, 2008.

[3] Kazumaro Aoki and Yu Sasaki. Meet-in-the-middle preimage attacks against reduced SHA-0 and SHA-1. In *CRYPTO'09*, volume 5677 of *Lecture Notes in Computer Science*, pages 70–89. Springer, 2009.

[4] Jean-Philippe Aumasson, Çagdas Çalik, Willi Meier, Onur Özen, Raphael C.-W. Phan, and Kerem Varici. Improved cryptanalysis of Skein. In *ASIACRYPT'09*, volume 5912 of *Lecture Notes in Computer Science*, pages 542–559. Springer, 2009.

[5] Eli Biham and Rafi Chen. Near-collisions of SHA-0. In *CRYPTO'04*, volume 3152 of *Lecture Notes in Computer Science*, pages 290–305. Springer, 2004.

[6] Alex Biryukov and Dmitry Khovratovich. Related-key cryptanalysis of the full AES-192 and AES-256. In *ASIACRYPT'09*, volume 5912 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2009.

[7] John Black, Phillip Rogaway, and Thomas Shrimpton. Black-box analysis of the block-cipher-based hash-function constructions from PGV. In *CRYPTO'02*, volume 2442 of *Lecture Notes in Computer Science*, pages 320–335. Springer, 2002.

[8] Andrey Bogdanov, Dmitry Khovratovich, and Christian Rechberger. Biclique cryptanalysis of the full AES. In *ASIACRYPT'11*, volume 7073 of *Lecture Notes in Computer Science*, pages 344–371. Springer, 2011.

[9] Whitfield Diffie and Martin Hellman. Special feature exhaustive cryptanalysis of the NBS Data Encryption Standard. *Computer*, 10:74–84, 1977.

[10] Niels Ferguson, Stefan Lucks, Bruce Schneier, Doug Whiting, Mihir Bellare, Tadayoshi Kohno, Jon Callas, and Jesse Walker. The skein hash function family,`http://www.skein-hash.info/sites/default/files/skein1.3.pdf`. Submission to NIST (Round 3), 2010.

[11] Praveen Gauravaram, Lars R. Knudsen, Krystian Matusiewicz, Florian Mendel, Christian Rechberger, Martin Schläffer, and Søren S. Thomsen. Grøstl – a SHA-3 candidate, available at `http://www.groestl.info/Groestl.pdf`. Submission to NIST, 2008.

[12] Henri Gilbert and Thomas Peyrin. Super-sbox cryptanalysis: Improved attacks for AES-like permutations. In *FSE'10*, volume 6147 of *Lecture Notes in Computer Science*, pages 365–383. Springer, 2010.

[13] Keting Jia, Honbo Yu, and Xiaoyun Wang. A meet-in-the-middle attack on the full KASUMI. Cryptology ePrint Archive, Report 2011/466, 2011.

[14] Dmitry Khovratovich, Gaëtan Leurent, and Christian Rechberger. Narrow-bicliques: Cryptanalysis of the full IDEA. In *EUROCRYPT'12, to appear*, 2012.

[15] Dmitry Khovratovich, Christian Rechberger, and Alexandra Savelieva. Bicliques for preimages: Attacks on Skein-512 and the SHA-2 family. Available online at `http://eprint.iacr.org/2011/286.pdf`, to appear at FSE 2012, 2012.

[16] Ji Li, Takanori Isobe, and Kyoji Shibutani (private communication). Converting meet-in-the-middle preimage attack into pseudo collision attack: Application to SHA-2. In *FSE'12, to appear*, 2012.

[17] Hamid Mala. Biclique cryptanalysis of the block cipher Square. Cryptology ePrint Archive, Report 2011/500, 2011. `http://eprint.iacr.org/`.

[18] Florian Mendel, Christian Rechberger, Martin Schläffer, and Søren S. Thomsen. The rebound attack: Cryptanalysis of reduced Whirlpool and Grøstl. In *FSE'09*, volume 5665 of *Lecture Notes in Computer Science*, pages 260–276. Springer, 2009.

[19] Florian Mendel, Christian Rechberger, Martin Schläffer, and Søren S. Thomsen. Rebound attacks on the reduced Grøstl hash function. In *CT-RSA'10*, volume 5985 of *Lecture Notes in Computer Science*, pages 350–365. Springer, 2010.

[20] Thomas Peyrin. Improved differential attacks for ECHO and Grøstl. In *CRYPTO'10*, volume 6223 of *Lecture Notes in Computer Science*, pages 370–392. Springer, 2010.

[21] Yu Sasaki. Meet-in-the-middle preimage attacks on AES hashing modes and an application to Whirlpool. In *FSE'11*, volume 6733 of *Lecture Notes in Computer Science*, pages 378–396. Springer, 2011.

[22] Yu Sasaki and Kazumaro Aoki. Preimage attacks on step-reduced MD5. In *ACISP'08*, volume 5107 of *Lecture Notes in Computer Science*, pages 282–296. Springer, 2008.

[23] Yu Sasaki and Kazumaro Aoki. Finding preimages in full MD5 faster than exhaustive search. In *EUROCRYPT'09*, volume 5479 of *Lecture Notes in Computer Science*, pages 134–152. Springer, 2009.

[24] Yu Sasaki, Yang Li, Lei Wang, Kazuo Sakiyama, and Kazuo Ohta. New non-ideal properties of AES-based permutations: Applications to ECHO and Grøstl. In *ASIACRYPT'10*, volume 6477 of *Lecture Notes in Computer Science*, pages 38–55. Springer, 2010.

[25] Bozhan Su, Wenling Wu, Shuang Wu, and Le Dong. Near-collisions on the reduced-round compression functions of Skein and BLAKE. In *CANS'10*, volume 6467 of *Lecture Notes in Computer Science*, pages 124–139. Springer, 2010.

[26] Paul C. van Oorschot and Michael J. Wiener. Parallel collision search with cryptanalytic applications. *J. Cryptology*, 12(1):1–28, 1999.

[27] David Wagner. The boomerang attack. In *FSE'99*, volume 1636 of *Lecture Notes in Computer Science*, pages 156–170. Springer, 1999.

[28] Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu. Finding collisions in the full SHA-1. In *CRYPTO'05*, volume 3621 of *Lecture Notes in Computer Science*, pages 17–36. Springer, 2005.

[29] Shuang Wu, Dengguo Feng, Wenling Wu, Jian Guo, Le Dong, and Jian Zou. [(pseudo).

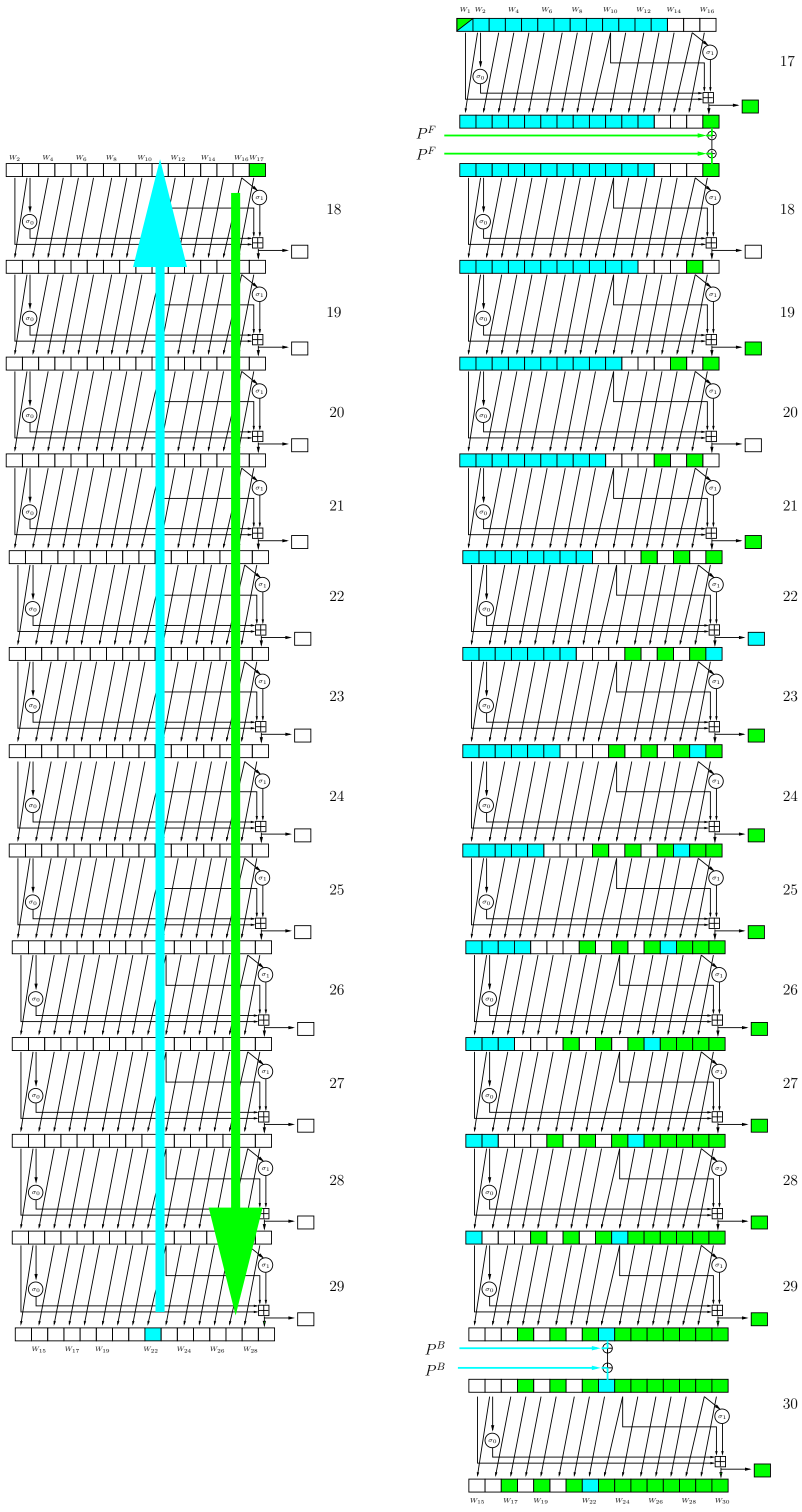

w_2 w_4 w_6 w_8 w_{10} w_{12} w_{14} w_{16} w_{17}
σ_0
σ_1
18
19
20
21
22
23
24
25
26
27
28
29
w_{15} w_{17} w_{19} w_{22} w_{24} w_{26} w_{28}
w_1 w_2 w_4 w_6 w_8 w_{10} w_{12} w_{14} w_{16}
17
P^F
P^F
P^B
P^B
30
w_{15} w_{17} w_{19} w_{22} w_{24} w_{26} w_{28} w_{30}

Figure 6: Message compensation as a biclique in the SHA-256 attack.