Bicliques for Permutations: Collision and Preimage Attacks in Stronger Settings

Dmitry Khovratovich

Microsoft Research, USA, and Infotecs, Russia

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 the primitives with these parameters fixed, i.e. on permutations. We introduce the concept of sliced bicliques, which is a translation of regular bicliques to the framework with permutations.

The new framework allows to convert preimage attacks into collision attacks and derive the first collision attacks on the reduced SHA-3 finalist Skein in the hash function setting up to 11 rounds. We also demonstrate new preimage attacks on the reduced Skein and the output transformation of the reduced 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, 23]. 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 this reduces 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, 24]. 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, 24],

X. Wang and K. Sako (Eds.): ASIACRYPT 2012, LNCS 7658, pp. 544–561, 2012.

which culminated in the concept of bicliques [16]. 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 on how these states map to each other under different sub-transformations.

Bicliques. The new biclique technique [8, 16] led to a few surprising attacks on AES, though many of them had only a constant factor improvement over exhaustive search. The attack has influenced those reducing the security level of the full Square [18], Kasumi [13], IDEA [15]. All these attacks need a small but noticeable number of operations to test a single key, 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 is a crucial element in the biclique attacks. In Section 2 we show how to enumerate N message candidates with only $2\sqrt{N}$ states.

However, there are several settings where an attacker can not manipulate a scheduled parameter, or there is no schedule at all. For example, preimage attacks on blockcipher-based hash functions first consider a compression function and produce pseudo-preimages, and then run a computationally expensive meet-in-the-middle attack to produce real preimages. If an attacker wants to reduce the cost by avoiding the second step, then he has to assign the chaining value (CV) with the original initial value (IV). If the compression function is based on the Matyas-Meyer-Oseas mode with E_K as a block cipher,

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

where M stands for a message block, then the attacker analyze the permutation $E_{IV}(\cdot)$.

Another example is the SHA-3 finalist Grøstl with output transformation $x \leftarrow \text{Truncate}(x \oplus P(x))$, where P is a fixed permutation. Therefore, the translation of the biclique technique to permutations is quite promising.

Permutations have been subject to a few recent attacks [22, 30], which use a predecessor of biclique — initial structure. A natural question is whether 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. Collision attacks typically fix the chaining value, so in the DM mode

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

an attacker is able to manipulate the round injections through the modification of M, while in the MMO mode he is able to choose only the input. From our point of view, famous collision attacks on the MD4/SHA family [5, 29] 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. Related-key attacks on the full AES [6] hint that the former setting is more suitable.

The hash function Skein follows the MMO mode and is an object of our analysis. The existing near-collision attacks on the compression function of Skein [4,26] 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 [17] on Skein is a great step forward, as we discuss in the further text.

Our Contributions

In Section 2 we introduce a new notion of *sliced biclique* as a translation of a regular biclique to permutations. The new concept helps to carry out the meet-in-the-middle attacks and the biclique technique to permutations without modifiable parameters. We call *parameters* both keys and messages.

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 (Section 4). We obtain the first collision attacks on the reduced round Skein hash function (Section 5). The new attacks are also translated to new preimage attacks on Skein (see the extended version of this paper [14]).

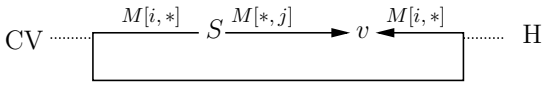
Then we consider the output transformation of the SHA-3 finalist Grøstl-256 and derive the first shortcut 6-round attack (Section 6). Finally, we analyze a procedure from earlier meet-in-the-middle attacks called message compensation (Section 7). Previously ad-hoc, it gets a clear interpretation as a sliced biclique (see the details in the extended version) [14]).

2 Splice-and-Cut Attacks and Bicliques

Splice-and-cut attacks [2, 23] 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 block 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 partitions the message space into sets, which are represented as two-dimensional array of messages $\{M[i, j]\}$, and process each set independently. Given $\{M[i, j]\}$, he selects 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 assigns S with an arbitrary value 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 a single set $\{M[i,j]\}$, and the complexity 2^{n-d} for the pseudo-preimage search. To find a full preimage the adversary generates $2^{d/2}$ pseudo-preimages, computes $2^{n-d/2}$ CVs out of the initial value, and checks for a matching pair. The total complexity is $2^{n-d/2+1}$ without optimizations (which are not always possible), 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.

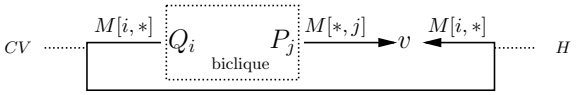
A biclique is an extension for the first step of the attack, which is based upon an earlier informal concept of *initial structure* [3, 24]. Instead of a single state S, a *biclique* is defined over a *sub-cipher* — a part of the primitive, typically several rounds long — and for a particular group of 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:



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

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.

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

3 Bicliques for Permutations

The simplest way to turn a permutation into a preimage-resistant function is to xor the input to the output:

$$F(x) = E(x) \oplus x. \tag{2}$$

Our goal is to construct a preimage search algorithm, which recovers x from given $H = F(x)$. We proceed as follows.

Using a specific algorithm, we select a sub-permutation g within E and an internal state V in E but not in g. Denote the input state of g by Q, and the output state of g by P. We partition the space of all states into sets $\{Q_{i,j}\}$, which we represent as a two-dimensional array of states. Here i, j are d-bit values for some d. We test independently each set if it contains a state that correspond to a valid preimage x. Let us denote the g-image of $Q_{i,j}$ by $P_{i,j}$:

$$Q_{i,j} \xrightarrow{g} P_{i,j}.$$

We will explain how to choose g and partition of Q in a subsection "Construction algorithms", and it will also become clear why we use two indices to enumerate states Q.

The rest of this section is devoted to finding an improved way to test a single set of states. A straightforward way to check if one of $\{Q_{i,j}\}$ is a solution to (2) is to compute for each i, j the state V two times. First, as a function of P in the forward direction, let us denote this computation by $\overrightarrow{F}$. Second, compute V as a function of Q in the backward direction: computing x, then $E(x) = H \oplus x$, and then V; let us denote this computation by $\overleftarrow{F}$. Hence we check if

$$\exists i, j: \quad \overrightarrow{F}(P_{i,j}) = \overleftarrow{F}(Q_{i,j}). \tag{3}$$

This algorithm is equivalent to the exhaustive search and requires 2^{2d} computations of E.

The complexity can be reduced as follows. Let $v \subseteq V$ be an internal variable, and $\overrightarrow{f_v}$ and $\overleftarrow{f_v}$ be the projections of $\overrightarrow{F}$ and $\overleftarrow{F}$, resp., to v. We say that the states $Q_{i,j}$ and $P_{i,j}$ form a *sliced biclique*, if the following conditions hold:

$$\begin{aligned} \forall i, j \quad & \overleftarrow{f_v}(Q_{i,j}) = \overleftarrow{f_v}(Q_{i,0}); \\ \forall i, j \quad & \overrightarrow{f_v}(P_{i,j}) = \overrightarrow{f_v}(P_{0,j}). \end{aligned}$$

Therefore, the necessary condition in Equation (3) can be reformulated as follows:

$$\begin{aligned} \exists i, j: \quad \overrightarrow{F}(P_{i,j}) = \overleftarrow{F}(Q_{i,j}) \implies \exists i, j: \quad & \overrightarrow{f_v}(P_{i,j}) = \overleftarrow{f_v}(Q_{i,j}) \iff \\ \iff \exists i, j: \quad & \overrightarrow{f_v}(P_{0,j}) = \overleftarrow{f_v}(Q_{i,0}). \end{aligned} \tag{4}$$

Let us denote $\overrightarrow{f_v}(P_{0,j})$ by $\overrightarrow{v_j}$ and $\overleftarrow{f_v}(Q_{i,0})$ by $\overleftarrow{v_i}$. Hence one of $\{Q_{i,j}\}$ is a solution if

$$\exists i, j: \quad \overrightarrow{v_j} = \overleftarrow{v_i}. \tag{5}$$

To check it we need to call $\overrightarrow{f_v}$ and $\overleftarrow{f_v}$ 2^d times each, which is less than 2^d calls of E. The computations are depicted in Figure 1. The matching candidates yields a pair (i, j) and the state $Q_{i,j}$, which we retest as a preimage candidate. For the full attack we need to partition the full input domain into the groups of size 2^{2d} and construct bicliques for them.

If the complexity of constructing a biclique and retesting the false alarms is small compared to 2^d, then 2^{2d} states are tested with complexity 2^d, and the set of all n-bit states is tested with complexity 2^{n-d}. In the most of our attacks we test only a subset of states of cardinality 2^r with complexity 2^{r-d} The parameter d is called a *dimension* of sliced biclique.

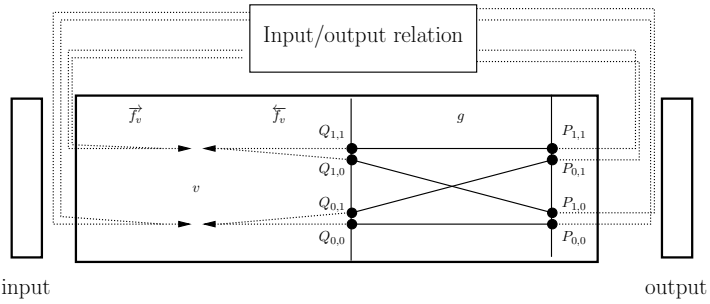


Fig. 1. Sliced biclique for a permutation

Construction Algorithms. Let us describe a construction algorithm for sliced bicliques, and then discuss its modifications. First we choose (below we will explain how) state $Q_{0,0}$ and two sets of differences $\{\Delta_i\}$ and $\{\nabla_j\}$, $i, j > 0$. We construct a biclique where

$$Q_{i,j} = Q_{i,0} \oplus \Delta_j; \tag{6}$$

$$P_{i,j} = P_{0,j} \oplus \nabla_i. \tag{7}$$

We proceed as follows

1. Compute $P_{0,0} \leftarrow g(Q_{0,0})$.
2. Set $Q_{0,j} \leftarrow Q_{0,0} \oplus \Delta_j$, compute $P_{0,j}$ for all $j > 0$.
3. Set $P_{i,j} \leftarrow P_{0,j} \oplus \nabla_i$, compute $Q_{i,j}$ for all i, j.

Hence Equation (7) is fulfilled by definition, and we need to prove Equation (6). We claim that it is fulfilled if the states $Q_{0,0}, Q_{i,0}, Q_{i,j}, Q_{0,j}$ form a boomerang quartet [28] over f with differences Δ_i and ∇_j, as demonstrated in Figure 2, a).

Indeed, $Q_{0,j} = Q_{0,0} \oplus \Delta_j$ by definition. We also have

$$P_{i,j} = g(Q_{0,j}) \oplus \nabla_i; \quad P_{i,0} = g(Q_{0,0}) \oplus \nabla_i.$$

Therefore, $g^{-1}(P_{i,j}) \oplus g^{-1}(P_{i,0}) = Q_{i,j} \oplus Q_{i,0} = \Delta_i$ if

$$(Q_{0,0}, Q_{i,0}, Q_{i,j}, Q_{0,j}) \text{ — boomerang quartet.} \tag{8}$$

In order to figure out sufficient conditions for the latter statement to hold, we consider two groups of differential trails. The trails in the first group are called Δ-trails and describe the evolution of differences Δ_j:

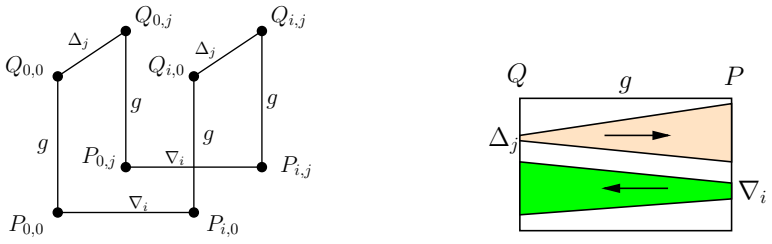
$$\begin{array}{l} Q_{i,0} \xrightarrow{g} P_{i,0}; \\ Q_{i,j} \xrightarrow{g} P_{i,j}. \end{array} \quad \Longrightarrow \quad \Delta_j \rightarrow P_{i,0} \oplus P_{i,j}.$$

The trails in the second group are called ∇-trails and describe the evolution of differences ∇_i:

$$\begin{array}{l} P_{0,j} \xrightarrow{g^{-1}} Q_{0,j}; \\ P_{i,j} \xrightarrow{g^{-1}} Q_{i,j}. \end{array} \quad \Longrightarrow \quad \nabla_i \rightarrow Q_{0,j} \oplus Q_{i,j}.$$

As proved in [16], Condition (8) holds with probability 1 if the Δ- and ∇-trails share no active nonlinear elements (Figure 2, b)). Such bicliques are called *based on non-interleaving trails*. A straightforward way to achieve this property is to select Δ_j and ∇_i so that their diffusion is minimum. A more sophisticated approach is to choose the state $Q_{0,0}$ so that the diffusion is minimum.

If Δ- and ∇-trail share nonlinear elements, we say that such bicliques are *based on interleaving trails*.



Biclique states as a boomerang quartet. Non-interleaving differential trails in g.

Fig. 2. Differential properties of sliced biclique

4 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 [16] based on regular bicliques. The parameter $M[i, j]$ in the biclique equation (1) is the chaining value. As a result, in the preimage attack on the compression function the attacker has to work with multiple CV's to get a pseudo-preimage. A full preimage requires another meet-in-the-middle procedure (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.

Equipped with the concept of sliced bicliques, we can fix the chaining value and attack the permutation $E_{IV}()$. Hence 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 [22,30], 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 [17]. The method produces pseudo-collisions out of biclique preimage attacks as follows. Assume we have a biclique of dimension d and are able to match deterministically on some l hash value bits. Then the adversary generates partial pseudo-preimages to a hash value with these 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. The reason is the diffusion of differences Δ and ∇ to the whole state while computing v, whereas in the regular biclique attack the effect of those differences is postponed. Nevertheless, our approach is more interesting, since the real collision attacks are considered a much stronger setting as compared to pseudo-collisions (cf. collisions for MD5).

Memory. The straightforward 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 [27], 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.

5 Collision Attacks on Skein

Here we present the first collision attacks on the reduced Skein hash function. The MMO mode is considered to be difficult for collision search, since most methods require a fixed chaining value when attacking the compression function. Since the round injections in the MMO mode come from the chaining value, the cryptanalyst has no freedom there, and hence is unable to construct local collisions, apply message modification techniques, etc.. As a result, previous attacks on Skein [4, 26] 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. We consider three members of the Skein hash function family: Skein-512, Skein-256, and Skein-512-256. Skein-512 [10] operates on the internal state of eight 64-bit words, while Skein-256 works with a state of four words. We denote the state words by $S^0, S^1, \ldots, S^7$. All the versions have 72 rounds. Skein-512-256 merely 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:

$$\begin{aligned} y_0 &= x_0 + x_1 \pmod{2^{64}};\\ y_1 &= (x_1 \lll_{R_{(d \bmod 8)+1,j}}) \oplus y_0. \end{aligned}$$

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, also by an addition of a linear function of the chaining value and the tweak (constants in our scenario).

The only published attack on the Skein hash function is a preimage attack [16] on 22 rounds of Skein-512.

5.1 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 bicliques based on non-interleaving trails are likely to cover two rounds only. We present bicliques of different kind that are capable to cover up to 4 rounds, and give some hints on how to construct longer bicliques.

2-Round Biclique. Our first examples deal with short bicliques of high dimension. As a result, the attacks have a significant advantage over brute-force. We use an additional enumeration of rounds in a biclique, starting with 0.

We use an algorithm from Section 3 with non-interleaving trails. We choose an arbitrary $Q_{0,0}$ and construct bicliques of dimension 64 out of the following differences Δ and ∇:

$\Delta_j =$ [0|0|0|0|0|j|0|0] after MIX in round 0 of the biclique. $j = 1 \dots 2^{64} - 1$

$\nabla_i =$ [0|0|0|0|0|0|i|0] after MIX in round 2 of the biclique. $i = 1 \dots 2^{64} - 1$

It is easy to check that the Δ- and ∇-trails do not share active non-linear components and hence produce a biclique (Figure 3).

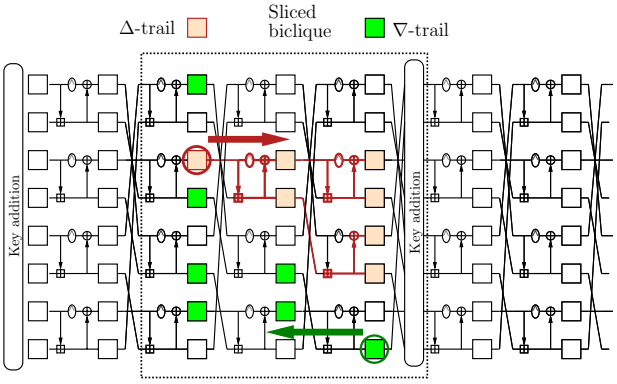


Fig. 3. Non-interleaving differential trails in a sliced biclique of dimension 64 in Skein-512

Only three rounds are required to diffuse a 64-bit word onto the full state. Hence we expect the matching part be two rounds long in both directions. A straightforward attack on 6 rounds uses a biclique in rounds 2-4 of Skein and word S^1 of the output of the 6-round transformation as the matching variable v (Figure 4).

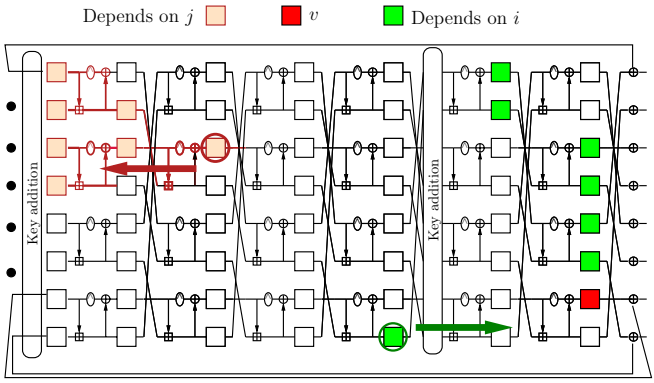


Fig. 4. Matching in 6-round Skein-512

However, we extend it by one round with the idea of the indirect partial matching [1]. Consider the state word S^0 after round 6 as a function of $P_{i,j}$. It is easy to check that

$$P_{i,j} = P_{0,j} \oplus i \implies S^0(P_{i,j}) = S^0(P_{0,j}) + i.$$

Therefore, if we set $v = S^0$, we get $\overrightarrow{v_{i,j}} = \overrightarrow{v_{0,j}} + i$. As a result,

$$\exists i,j : \overrightarrow{v_{i,j}} = \overleftarrow{v_{i,j}} \Leftrightarrow \exists i,j : \overrightarrow{v_j} = \overleftarrow{v_i} + i,$$

which can be checked with complexity 2^{64}. Hence we generate 2^{64} 64-bit partial preimages with cost about 2^{64}.

To produce new bicliques, we generate a new $Q_{0,0}$ and repeat the procedure. Full 7-round collisions are found within $2^{(512-64)/2} = 2^{224}$ partial preimages with the cost 2^{224}. Since the total number of states Q needed for the attack is less than 2^{256}, it is unlikely that two identical states are produced.

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

3-Round Biclique. If we decrease dimension to 20 and lower, the diffusion takes 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.

First, we carefully choose the position of the biclique in the compression function and bits where the difference is applied in Q and P. Since the rotation constants in each MIX function are distinct, the diffusion properties may change significantly when we shift the biclique over rounds and the active bits over the 64-bit word. The best configuration we have found places the biclique in rounds 5–7 (or $8k$ rounds further, because the rotation constants repeat every

8 rounds), where the states Q are defined before the MIX operation in round 5, and the states P are defined after the MIX operation in round 7. The Δ- and ∇-differences are defined for as follows:

$$\Delta_j = \boxed{0}\boxed{0}\boxed{0}\boxed{0}\boxed{0}\boxed{j \ll 45}\boxed{0} \; j = 1 \ldots 2^{19} - 1$$
$$\nabla_i = \boxed{0}\boxed{i}\boxed{0}\boxed{0}\boxed{0}\boxed{0}\boxed{0}\boxed{0} \; i = 1 \ldots 2^{19} - 1$$

For $d < 19$ we simply set the most significant bits of Δ and ∇ to zero. We additionally require that the least 45 significant bits of the word S^6 in Q be equal to 0 in order to trails from interleaving. There is no other restriction on Q, so we can generate the states $Q_{0,0}$ in message sets simply by changing the words $S^0, \ldots, S^3$. Since we need less than 2^{256} states, it is unlikely that there would be a collision. This configuration produces a 3-round sliced biclique. Note that reducing dimension does not make the trails to interleave.

The length of the matching part decreases as the dimension grows. We have checked the diffusion on a PC and figured out that the matching part covers 7 rounds for $d = 17$. In this configuration we match on bits 30–33 of word S^2 and bits 20–32 of word S^3 of input to the compression function. The matching is not deterministic, as for some bits the difference is equal to zero with probability $p_i < 1$. We have calculated the type-I error probability as $\prod_i p_i = 0.6$ and conclude that probability 0.4 we miss a solution. Therefore, the total complexity is about two times larger compared to the deterministic case and is equal to about 2^{248}

For $d = 10$ the matching part takes 8 rounds. The matching variable consists of bits 17–21 of word S^0 and bits 24–31 of word S^2. The type-I error probability does not exceed 0.2, and the total complexity is 2^{251}. The other values are given in Table 1.

4-Round Biclique. A regular biclique in the preimage attack on Skein [16] covers 4 rounds with two key additions. If we consider these rounds without the key addition, we get exactly a sliced biclique of the same dimension. The diffusion in the matching part will be slightly different because of the rotation constants, but we still can bypass 10 rounds. Though the cost of the biclique construction is quite expensive — 2^{209} — there are 815 bit degrees of freedom left, of which 303 are in the internal state. We propose to use this freedom to amortize the biclique construction cost and generate new $Q_{0,0}$, so that a 14-round partial preimage is found 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. Another problem is that the construction cost of a single biclique is very high, and we are unaware of how to exploit the degrees of freedom over so many rounds given no freedom in the injections.

Table 1. Collision attacks on reduced Skein with large memory requirements (close to the computational complexity). Memoryless attacks add a small constant to the exponent.

Skein-256		Skein-512	
Rounds	Complexity	Rounds	Complexity
2	2^{85}	5	2^{192}
4	2^{96}	7	2^{224}
8	2^{120}	10	2^{248}
9	2^{124}	11	2^{251}
12	$2^{126.5}$	14	$2^{254.5}$

5.2 Skein-256

Diffusion in Skein-256 is generally faster, because the internal state consists of four words only. Typically it takes one round less to affect the whole state. As a result, non-interleaving biclique trails and the matching part are shorter. We figured out that 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 found to be less sensitive to the smaller state size. The low-dimension attacks for Skein-512 lose two rounds when being translated to Skein-256 (Table 1).

The biclique construction, including trail details and partition of the state space, is very similar to that in Skein-512, so we do not give much details. The 2-round biclique yields 2- and 4-round attacks, which correspond to 5- and 7-round attacks on Skein-512. The 3-round biclique with dimension 17 yields an 8-round attack.

6 Certificational Preimage Attack on the Reduced Grøstl Output Transformation

In this section we present a certificational attack, i.e. it has only a small advantage over the exhaustive search, on Grøstl [11] — 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 [19, 21, 25], though only few of them violated collision or preimage resistance of the hash function [20, 30]. The paper [30] 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 [19] 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.

Attack. We denote the internal states within 6 rounds from #1 to #13, as depicted in Figure 5. We construct a sliced biclique of dimension 1 in states #4–#9, which contains the Super S-box layer in states #5–#8. The matching variable is a linear function of the variables in states #12 and #13 not affected by Δ- and ∇-differences. The Δ-difference has a single active byte, marked as lightblue. Its influence on the internal states within the matching part is also depicted as lightblue in Figure 5. The ∇-difference and its influence are depicted as green.

The matching condition is a linear function of the bytes not affected by the differences. Let us elaborate on this statement. Consider the rightmost columns of states #12 and #13 and denote them by A and B, respectively. Let us note that B is a linear function of A. In turn, 7 bytes of A do not depend on i, and 7 bytes of B do not depend on j. If the state $Q_{i,j}$ corresponds to a preimage, then a system of $8 - (8 - 7) - (8 - 7) = 6$ linear equations should hold. All the equations have form

$$A_j \oplus B_i = 0,$$

which is easily transformed to Equation (5).

We construct a sliced biclique based on interleaving trails. A biclique of dimension 1 is equivalent to a single boomerang quartet (Figure 2, left). In contrast to attacks on Skein, all the four relevant differences are distinct.

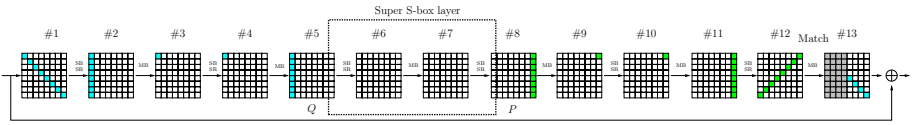


Fig. 5. Preimage attack on the reduced Grøstl-256 output transformation

Bicliques are constructed as follows. First, we arbitrarily choose $\Delta_{0,1} \neq \Delta_{1,1}$ and $\nabla_{1,0} \neq \nabla_{1,1}$, which are all active in one byte only, as specified earlier. We construct the states $\{Q_{i,j}\}_{i,j=0,1}$, which satisfy the following equations:

$$Q_{0,0} \oplus Q_{0,1} = \Delta_{0,1}; \quad Q_{1,0} \oplus Q_{1,1} = \Delta_{1,1}; \tag{9}$$

$$P_{0,0} \oplus P_{1,0} = \nabla_{1,0}; \quad P_{0,1} \oplus P_{1,1} = \nabla_{1,1}. \tag{10}$$

First, we derive the differences in #5 and in #8. Then we reformulate Equations (9) for each Super S-box, and solutions are found independently by exhaustive search with a total complexity around 2^{70}. The solutions are then concatenated. For the details, we refer to the long biclique attack on AES [8], which gives a description of an equivalent algorithm.

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}.

7 Message Compensation

The message compensation procedure [1, 16] 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 majority of this section is left for an extended version of this paper [14].

We propose the following algorithm a generic message schedule. Suppose you construct a biclique in rounds N_1–N_2, and want to describe a message set $\{M[i,j]\}$ such that

1. Injections $W^{N_0}, W^{N_0+1}, \ldots, W^{N_1-1}$ do not depend on j;
2. Injections $W^{N_2+1}, W^{N_2+2}, \ldots, W^{N_3}$ do not depend on i.

We propose to construct a sliced biclique without a matching point in rounds N_1–N_3. The difference Δ_j is defined before round N_1. To satisfy the first condition,

we assign the words of Δ_j that correspond to $W^{N_0}, W^{N_0+1}, \ldots, W^{N_1-1}$ to zero. If some words are left undefined, then we get a freedom in these values and can use it to manipulate the difference propagation in the Δ-trails.

We define ∇_i after round N_3. To satisfy the second condition, we assign the words of ∇_i that correspond to $W^{N_2+1}, W^{N_2+2}, \ldots, W^{N_3}$ to zero. Again, if some words are undefined, we keep this freedom.

Finally, we construct a sliced biclique based on non-interleaving trails. We use undefined parts of Δ_j and ∇_i to control the diffusion on the word level, and select $M[0,0]$ to control the diffusion, if necessary, on the bit level. We may also choose other round indices for a biclique, if this makes the difference selection more clear. We may also have to deal with other constraints like padding, which further reduce the freedom in Δ and ∇. Finally, we may have to construct a biclique based on interleaving trails, if non-interleaving ones are impossible because of the diffusion.

8 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.

Acknowledgements. The author thanks Maria Naya-Plasencia, Andrey Labunets, and Alexander Shalimov for fruitful discussions on the paper topic and possible improvements. The author also thanks reviewers of the paper, whose comments helped to improve the readability. Any further review or comment is warmly welcome.

References

1. Aoki, K., Guo, J., Matusiewicz, K., Sasaki, Y., Wang, L.: Preimages for Step-Reduced SHA-2. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 578–597. Springer, Heidelberg (2009)
2. Aoki, K., Sasaki, Y.: Preimage Attacks on One-Block MD4, 63-Step MD5 and More. In: Avanzi, R.M., Keliher, L., Sica, F. (eds.) SAC 2008. LNCS, vol. 5381, pp. 103–119. Springer, Heidelberg (2009)
3. Aoki, K., Sasaki, Y.: Meet-in-the-Middle Preimage Attacks Against Reduced SHA-0 and SHA-1. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 70–89. Springer, Heidelberg (2009)
4. Aumasson, J.-P., Çalık, Ç., Meier, W., Özen, O., Phan, R.C.-W., Varıcı, K.: Improved Cryptanalysis of Skein. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 542–559. Springer, Heidelberg (2009)
5. Biham, E., Chen, R.: Near-Collisions of SHA-0. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 290–305. Springer, Heidelberg (2004)
6. Biryukov, A., Khovratovich, D.: Related-Key Cryptanalysis of the Full AES-192 and AES-256. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 1–18. Springer, Heidelberg (2009)
7. Black, J., Rogaway, P., Shrimpton, T.: Black-Box Analysis of the Block-Cipher-Based Hash-Function Constructions from PGV. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 320–335. Springer, Heidelberg (2002)
8. Bogdanov, A., Khovratovich, D., Rechberger, C.: Biclique Cryptanalysis of the Full AES. In: Lee, D.H. (ed.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 344–371. Springer, Heidelberg (2011)
9. Diffie, W., Hellman, M.: Special feature exhaustive cryptanalysis of the NBS Data Encryption Standard. Computer 10, 74–84 (1977)
10. Ferguson, N., Lucks, S., Schneier, B., Whiting, D., Bellare, M., Kohno, T., Callas, J., Walker, J.: The Skein hash function family. Submission to NIST, Round 3 (2010), `http://www.skein-hash.info/sites/default/files/skein1.3.pdf`
11. Gauravaram, P., Knudsen, L.R., Matusiewicz, K., Mendel, F., Rechberger, C., Schläffer, M., Thomsen, S.S.: Grøstl – a SHA-3 candidate. Submission to NIST (2008), `http://www.groestl.info/Groestl.pdf`
12. Gilbert, H., Peyrin, T.: Super-Sbox Cryptanalysis: Improved Attacks for AES-Like Permutations. In: Hong, S., Iwata, T. (eds.) FSE 2010. LNCS, vol. 6147, pp. 365–383. Springer, Heidelberg (2010)
13. Jia, K., Yu, H., Wang, X.: A meet-in-the-middle attack on the full KASUMI. Cryptology ePrint Archive, Report 2011/466 (2011)
14. Khovratovich, D.: Bicliques for permutations: collision and preimage attacks in stronger settings. Cryptology ePrint Archive, Report 2012/141 (2012), `http://eprint.iacr.org/2012/141`
15. Khovratovich, D., Leurent, G., Rechberger, C.: Narrow-Bicliques: Cryptanalysis of Full IDEA. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 392–410. Springer, Heidelberg (2012)

16. Khovratovich, D., Rechberger, C., Savelieva, A.: Bicliques for Preimages: Attacks on Skein-512 and the SHA-2 Family. In: Canteaut, A. (ed.) FSE 2012. LNCS, vol. 7549, pp. 244–263. Springer, Heidelberg (2012), http://eprint.iacr.org/2011/286.pdf
17. Li, J., Isobe, T., Shibutani, K.: Converting Meet-In-The-Middle Preimage Attack into Pseudo Collision Attack: Application to SHA-2. In: Canteaut, A. (ed.) FSE 2012. LNCS, vol. 7549, pp. 264–286. Springer, Heidelberg (2012)
18. Mala, H.: Biclique cryptanalysis of the block cipher Square. Cryptology ePrint Archive, Report 2011/500 (2011), http://eprint.iacr.org/
19. Mendel, F., Rechberger, C., Schläffer, M., Thomsen, S.S.: The Rebound Attack: Cryptanalysis of Reduced Whirlpool and Grøstl. In: Dunkelman, O. (ed.) FSE 2009. LNCS, vol. 5665, pp. 260–276. Springer, Heidelberg (2009)
20. Mendel, F., Rechberger, C., Schläffer, M., Thomsen, S.S.: Rebound Attacks on the Reduced Grøstl Hash Function. In: Pieprzyk, J. (ed.) CT-RSA 2010. LNCS, vol. 5985, pp. 350–365. Springer, Heidelberg (2010)
21. Peyrin, T.: Improved Differential Attacks for ECHO and Grøstl. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 370–392. Springer, Heidelberg (2010)
22. Sasaki, Y.: Meet-in-the-Middle Preimage Attacks on AES Hashing Modes and an Application to Whirlpool. In: Joux, A. (ed.) FSE 2011. LNCS, vol. 6733, pp. 378–396. Springer, Heidelberg (2011)
23. Sasaki, Y., Aoki, K.: Preimage Attacks on Step-Reduced MD5. In: Mu, Y., Susilo, W., Seberry, J. (eds.) ACISP 2008. LNCS, vol. 5107, pp. 282–296. Springer, Heidelberg (2008)
24. Sasaki, Y., Aoki, K.: Finding Preimages in Full MD5 Faster Than Exhaustive Search. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 134–152. Springer, Heidelberg (2009)
25. Sasaki, Y., Li, Y., Wang, L., Sakiyama, K., Ohta, K.: Non-full-active Super-Sbox Analysis: Applications to ECHO and Grøstl. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 38–55. Springer, Heidelberg (2010)
26. Su, B., Wu, W., Wu, S., Dong, L.: Near-Collisions on the Reduced-Round Compression Functions of Skein and BLAKE. In: Heng, S.-H., Wright, R.N., Goi, B.-M. (eds.) CANS 2010. LNCS, vol. 6467, pp. 124–139. Springer, Heidelberg (2010)
27. van Oorschot, P.C., Wiener, M.J.: Parallel collision search with cryptanalytic applications. J. Cryptology 12(1), 1–28 (1999)
28. Wagner, D.: The Boomerang Attack. In: Knudsen, L.R. (ed.) FSE 1999. LNCS, vol. 1636, pp. 156–170. Springer, Heidelberg (1999)
29. Wang, X., Yin, Y.L., Yu, H.: Finding Collisions in the Full SHA-1. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 17–36. Springer, Heidelberg (2005)
30. Wu, S., Feng, D., Wu, W., Guo, J., Dong, L., Zou, J.: (Pseudo) Preimage Attack on Reduced-Round Grøstl Hash Function and Others. In: Canteaut, A. (ed.) FSE 2012. LNCS, vol. 7549, pp. 127–145. Springer, Heidelberg (2012)