Boomerang Attacks on Hash Function using Auxiliary Differentials

Gaëtan Leurent and Arnab Roy

Université du Luxembourg and SnT

Abstract. In this paper we study boomerang attacks in the chosen-key setting. This is particularly relevant to hash function analysis, since many boomerang attacks have been described against ARX-based designs.
We present a new way to combine message modifications, or auxiliary differentials, with the boomerang attack. We show that under some conditions, we can combine three independent paths instead of two for the classical boomerang attack. Our main result is obtained by applying this technique to round-reduced Skein-256, for which we show a distinguisher on the keyed permutation with complexity only 2^{57}, and a distinguisher on the compression function with complexity 2^{114}. We also discuss application of the technique to Skein-512 and show some problems with the paths used in previous boomerang analysis of Skein-512.

Keywords: hash function, SHA-3 competition, chosen-key, Skein, Threefish, boomerang attack, higher order differential, zero-sum.

1 Introduction

The boomerang attack was proposed by Wagner in 1999 [16] as a cryptanalysis technique against block ciphers. This clever idea allows to combine short differential paths for the top half and the bottom half of a cipher, instead of using a long differential path for the full cipher.

Recently, this idea has been applied to hash function building blocks, as part of the new hash function results inspired by the SHA-3 competition. In [4] Biryukov *et al.* proposed boomerang distinguishers on compression functions and applied it to round-reduced BLAKE. Mendel and Lamberger [11] also independently proposed a boomerang attack on the compression function of SHA-2. More recently at SAC 2011, Yu Sasaki [14] gave a boomerang distinguisher on the full compression function of HAVAL. Boomerang distinguishers have also been applied to Skein/Threefish [1,5].

Another related work by Joux and Peyrin [7] studies boomerangs in the context of hash functions. However this result does not try to build a boomerang property for a hash function, but only uses *auxiliary differential paths*, which are related to the boomerang idea, as a tool for message modifications.

The SHA-3 competition [13] is now at the final phase with 5 remaining hash function candidates; and Skein is one of them. It is one of the two ARX (Addition-Rotation-Xor) designs amongst those candidates.

The most successful attack proposed against Skein is the rotational rebound attack [10], by Khovratovich *et al.*, reaching 57 out of 72 rounds. This work improves upon the rotational cryptanalysis technique [9] which reached 42 rounds for Skein-512 and 39 rounds for Skein-256. Those results are based on rotation-invariant constants in the key schedule. However for the final round of the SHA-3 competition, Skein has been tweaked [6] to avoid those rotation invariant constants, and this technique is no longer applicable.

Various other techniques have also been applied to Skein, which do not depend on rotation-invariant constants. Su *et al.* [15] gave near collisons on 24 rounds for a cost of 2^{60} and 2^{230} compression function calls on Skein-256 and Skein-512 respectively. Aumasson *et al.* [1] also used a boomerang attack to launch a key recovery attack on Threefish-512 for 32, 34 and 35 rounds. In [5] Chen and Jia proposed a bommerang distinguisher with complexity 2^{189} on 32-rounds of Threefish-512, and used it to mount a key-recovery attacks on 33 and 34 rounds, with complexity $2^{324.6}$ and $2^{474.4}$, respectively. However, we show in Section 5.2 that the paths used in this attack are in fact incompatible. Another recent result by Yu *et al.* gives semi-free start near collision for up to 32 rounds of Skein-256 [18] with complexity 2^{105}.

Our Contributions. We study boomerang distinguishers on round-reduced Skein-256. The analysis is based on high probability related-key differential trails in Threefish (probability 1, 2^{-6}, and 2^{-39} for 8, 12, and 16 rounds respectively), like previous analysis [1,5,15].

Our main contribution is a technique using auxiliary differentials, which allows to skip some rounds in the middle of the boomerang in a chosen-key setting. This is similar to previous works using message modifications (*e.g.* on Skein-512 [18,10]) but we use it in a boomerang setting. When applied to the 32-round attack on Skein, we can avoid 8 rounds in the middle, and this results in a significant complexity improvement. Moreover, since the complexity is relatively low, we can experimentally measure the amplification effect, by implementing the attack on 28 rounds. This results in a boomerang distinguisher with complexity 2^{57} for the keyed-permutation (*i.e.* for Threefish-256). When applied to the compression function with the feed-forward, we show that our attack has complexity at most 2^{114}, but we cannot measure experimentally the full effect of the amplification, so we expect the actual complexity to be significantly lower. A summary of our results is given in Table 1.

Additionally, we also discuss the hypothesis of independence between the boomerang paths for ARX primitives, and give an example of previous work where the hypothesis is not valid, and the paths are in fact incompatible.

2 The boomerang attack

The boomerang attack was introduced by David Wagner in 1999 [16] against block ciphers, and the initial idea has been developed through many later results,

Table 1. Summary of the attacks on the compression function (CF) and the keyed permutation (KP) of Skein. We only mention results which are independent of the constants used Skein (*i.e.* which apply to the round-3 version).

Attack	CF/KP	Rounds	CF/KP calls	Reference
Near collisions (Skein-256)	CF	24	2^{60}	[15]
Boomerang dist. (Threefish-512)	KP	32	2^{189}	[5]
Key Recovery (Threefish-512)	KP	33	$2^{324.6}$	[5]
Key Recovery (Threefish-512)	KP	34	$2^{474.4}$	[5]
Near collisions (Skein-256)	CF	32	2^{105}	[18]
Key Recovery (Threefish-512)	KP	32	2^{312}	[1]
Key Recovery (Threefish-512)	KP	35	2^{478}	[1]
Boomerang dist. (Skein-256)	CF and KP	24	2^{18}	Sec. 4.2
Boomerang dist. (Threefish-256)	KP	28	2^{21}	Sec. 4.2
Boomerang dist. (Skein-256)	CF	28	2^{24}	Sec. 4.2
Boomerang dist. (Threefish-256)	KP	32	2^{57}	Sec. 4.2
Boomerang dist. (Skein-256)	CF	32	2^{114}	Sec. 4.2

including [16,8,2,3,4,11]. In this section, we go through the main results of those papers, in order to explain the techniques needed used in our attack on Skein.

The main idea of the boomerang attack is to consider a block cipher as a composition of two sub-ciphers, and to use an encryption oracle as well as a decryption oracle in order to build differential pair for each sub-cipher independently. Given a permutation f that can be decomposed into two sub-permutations f_a and f_b with $f = f_b \circ f_a$ (e.g. a block cipher), one first identifies some high probability differentials[1] $\alpha \to \alpha'$ with probability p_a for f_a and $\gamma \to \gamma'$ with probability p_b for f_b, relative to a group operation $+$ (in practice, the group operation is either the exclusive or $\oplus$, or the modular addition $\boxplus$).

The attacker selects two plain-texts $P^{(0)}$ and $P^{(1)}$ with $P^{(1)} = P^{(0)} + \alpha$, and requests the corresponding cipher-texts $C^{(0)}$ and $C^{(1)}$. Then he builds the cipher-texts $C^{(2)} = C^{(0)} + \gamma'$ and $C^{(3)} = C^{(1)} + \gamma'$, and requests the corresponding plain-texts $P^{(2)}$ and $P^{(3)}$. This is illustrated by Figure 1. With this construction, we expect that $P^{(3)} = P^{(2)} + \alpha$ with probability $p_a^2 p_b^2$. This comes from the following observations:

(i) With probability p_a, $(P^{(0)}, P^{(1)})$ is a good pair for the differential $\alpha \to \alpha'$ in f_a, and we have $X^{(1)} = X^{(0)} + \alpha'$, where $X^{(i)} = f_a(P^{(i)})$.

(ii) With probability p_b^2, $(C^{(0)}, C^{(2)})$ and $(C^{(1)}, C^{(3)})$ are good pairs for the differential $\gamma' \to \gamma'$ in f_b^{-1}, and we have $X^{(2)} - X^{(0)} = X^{(3)} - X^{(1)} = \gamma$, where $X^{(i)} = f_b^{-1}(C^{(i)})$.

(iii) If *(i)* and *(ii)* are satisfied, then we have

$$X^{(3)} - X^{(2)} = (X^{(3)} - X^{(1)}) - (X^{(2)} - X^{(0)}) + (X^{(1)} - X^{(0)}) = \alpha'.$$

[1] We use this to denote $\Pr_{x,k}[f_a(x + \alpha) = f_a(x) + \alpha'] = p_a$.

With probability p_a, $(X^{(2)}, X^{(3)})$ is a good pair for the differential $\alpha' \to \alpha$ in f_a^{-1}, and we have $P^{(3)} = P^{(2)} + \alpha$.

This basic attack gives a distinguisher for f, and it can be extended to a key-recovery attack using partial encryption/decryption.

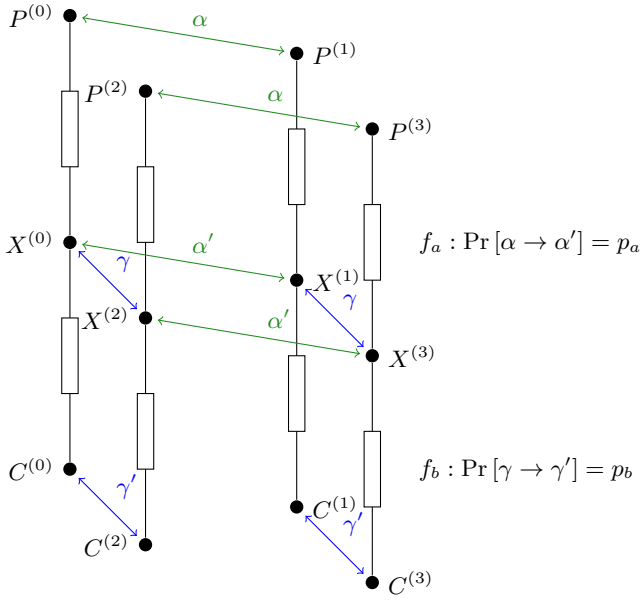

Fig. 1. The boomerang attack

Boomerang attack are particularly efficient on ARX-like designs because the probability of differential paths drops quickly when the number of rounds grows. It is usually possible to find good differential for a few rounds, but extending them leads to more diffusion and very bad probabilities. More generally, if we denote the probability of the best differential for function f by $\mathrm{bp}(f)$, then the boomerang attack is better than classical differential attack if:

$$\mathrm{bp}(f_a)^2 \, \mathrm{bp}(f_b)^2 > \mathrm{bp}(f_b \circ f_a)$$

2.1 Amplified probabilities

We can compute a better estimate of the complexity of a boomerang attack if we remark that we don't actually need to specify the differences α' and γ. As long as the two pairs $(C^{(0)}, C^{(2)})$ and $(C^{(1)}, C^{(3)})$ reach the same difference $X^{(2)} - X^{(0)} = X^{(3)} - X^{(1)}$, the boomerang attack will work. We can compute the complexity by summing over all possible α' and γ, which is equivalent to

replacing the probabilities p_a and p_b by the following values:

$$\hat{p}_a = \sqrt{\sum_{\alpha'} \Pr\left[\alpha \to \alpha'\right]} \qquad \hat{p}_b = \sqrt{\sum_{\gamma} \Pr\left[\gamma \to \gamma'\right]}$$

These are sometimes called *amplified* probabilities, but this is unrelated to the amplified bommerang attack of [8].

We can further improve the complexity by considering two independent differentials $\alpha_0 \to \alpha'_0$ and $\alpha_1 \to \alpha'_1$ in f_a, and $\gamma_0 \to \gamma'_0$ and $\gamma_1 \to \gamma'_1$ in f_b. The paths can be used for a boomerang attack as long as $\alpha'_1 - \alpha'_0 = \gamma_1 - \gamma_0$, as shown in [16].

In practice, the amplified probabilities are often estimated experimentally with random values.

2.2 Related-key boomerang

The boomerang attack can also be used with related-key differentials, instead of fixed-key differentials, as shown in [3]. In this case, we use a differential[2] $\alpha, \alpha_k \to \alpha'$ with probability p_a for f_a, and $\beta, \beta_k \to \beta'$ with probability p_b for f_b.

Starting from a random plain-text $P^{(0)}$, we compute

$$\begin{aligned}
P^{(1)} &= P^{(0)} + \alpha \\
C^{(0)} &= f(P^{(0)}, k) & C^{(1)} &= f(P^{(1)}, k + \alpha_k) \\
C^{(2)} &= C^{(0)} + \beta' & C^{(3)} &= C^{(1)} + \beta' \\
P^{(2)} &= f^{-1}(C^{(2)}, k + \beta_k) & P^{(3)} &= f^{-1}(C^{(3)}, k + \alpha_k + \beta_k)
\end{aligned}$$

and we obtain $P^{(3)} = P^{(2)} + \alpha$ with probability $p_a^2 p_b^2$.

2.3 Application to the known-key setting

In a known-key or chosen-key setting, a boomerang property can be used to distinguish a given permutation from a random one, as first used in [4] and [11]. The boomerang attack can generate quartets $(C^{(i)}, P^{(i)})_{i=0}^{3}$ with:

$$\begin{aligned}
C^{(i)} &= f(P^{(i)}) \\
P^{(1)} - P^{(0)} = P^{(3)} - P^{(2)} &= \alpha & C^{(2)} - C^{(0)} = C^{(3)} - C^{(1)} &= \gamma'
\end{aligned} \qquad (1)$$

Alternatively, we can consider the boomerang as a zero-sum property [4], or higher-order differential collision [11] since a quartet satisfies:

$$\begin{aligned}
\Delta P^{(i)} &= (P^{(3)} - P^{(2)}) - (P^{(1)} - P^{(0)}) = 0 \\
\Delta C^{(i)} &= (C^{(3)} - C^{(2)}) - (C^{(1)} - C^{(0)}) = (C^{(3)} - C^{(1)}) - (C^{(2)} - C^{(0)}) = 0
\end{aligned}$$

[2] We use this to denote $\Pr_{x,k}\left[f_a(x + \alpha, k + \alpha_k) - f_a(x, k) = \alpha'\right] = p_a$.

Moreover, in a known-key or chosen-key setting, it is possible to start from the middle. First one selects some values $X^{(0)}, X^{(1)}, X^{(2)}, X^{(3)}$ with $X^{(2)} - X^{(0)} = X^{(3)} - X^{(1)} = \gamma$ and $X^{(1)} - X^{(0)} = X^{(3)} - X^{(2)} = \alpha'$; then he compute $P^{(i)} = f_a^{-1}(X^{(i)})$ and $C^{(i)} = f_b(X^{(i)})$. This allows to select specific $X^{(i)}$'s that satisfy the paths with better probability that a random quartet.

For an n-bit random permutation, the generic complexity for obtaining a quartet satisfying (1) with fixed α, γ' is 2^n. However, if only the difference $(P^{(3)} - P^{(2)}) = (P^{(1)} - P^{(0)}) = \alpha$ is fixed, the complexity is only $2^{n/2}$. If we only want $\Delta P^{(i)} = 0$ and $\Delta C^{(i)} = 0$, the complexity is lower bounded by $2^{n/3}$, but the best known attack still takes time $2^{n/2}$.

2.4 Application to hash function

Boomerang attacks have been applied to hash function, in order to attack some of the components of the design. A bommerang attack can readily be applied to the block-cipher or permutation inside most of the designs. It can also be extended to a distinguisher against the compression function of most block-cipher based designs, as shown in [11] and [4]. For instance let us consider a compression function following the MMO construction: $\mathrm{CF}(h, m) = E_h(m) + m$, and a quartet $P^{(i)}, C^{(i)}$ for the block cipher E under the related keys $K^{(i)}$. The quartet satisfies:

$$\begin{aligned} C^{(i)} &= E_{K^{(i)}}(P^{(i)}) \\ \Delta K^{(i)} &= (K^{(3)} - K^{(2)}) - (K^{(1)} - K^{(0)}) = 0 \\ \Delta P^{(i)} &= (P^{(3)} - P^{(2)}) - (P^{(1)} - P^{(0)}) = 0 \\ \Delta C^{(i)} &= (C^{(3)} - C^{(1)}) - (C^{(2)} - C^{(0)}) = 0. \end{aligned}$$

Moreover, we have

$$\begin{aligned} &\Delta\,\mathrm{CF}(K^{(i)}, P^{(i)}) \\ &\quad = \Big[\mathrm{CF}(K^{(3)}, P^{(3)}) - \mathrm{CF}(K^{(2)}, P^{(2)})\Big] - \Big[\mathrm{CF}(K^{(1)}, P^{(1)}) - \mathrm{CF}(K^{(0)}, P^{(0)})\Big] \\ &\quad = \Big[(C^{(3)} + P^{(3)}) - (C^{(2)} + P^{(2)})\Big] - \Big[(C^{(1)} + P^{(1)}) - (C^{(0)} + P^{(0)})\Big] \\ &\quad = (C^{(3)} - C^{(1)}) - (C^{(2)} - C^{(0)}) + (P^{(3)} - P^{(2)}) - (P^{(1)} - P^{(0)}) = 0 \end{aligned}$$

This is a zero-sum property for the compression function. For an n-bit random compression function the best known attack to build a quartet with a zero-sum output takes time $2^{n/3}$ using Wagner's generalized birthday attack [17]. If we also want the inputs to be a zero-sum, the best known attack takes time $2^{n/2}$.

3 Boomerang Attack using Auxiliary Differentials

The main idea of our attack is to use auxiliary paths as message modifications. This idea has already be applied to hash function cryptanalysis by Joux and

Peyrin in [7], in the context of a classical differential attack. Here, we apply it to the boomerang setting, with related-key paths. The main idea is to build boomerang quartet in an inside-out fashion, and to use auxiliary paths to efficiently generate values in the middle, so that they conform to several rounds.

We consider a function f that can be decomposed into three sub-functions $f = f_c \circ f_b \circ f_a$, and we consider a differential in each of those sub-functions:

- for f_a, we use a differential $\alpha \to \alpha'$ with probability p_a
- for f_b, we use a set $\mathcal{B}$ of b differentials $\beta_j \to \beta'_j$ with probability p_b
- for f_c, we use a differential $\gamma \to \gamma'$ with probability p_c

We describe the idea with fixed-key differential for simplicity, but it works in the same way with related key differentials. We start with a boomerang quartet $(U^{(0)}, U^{(1)}, U^{(2)}, U^{(3)}) \to (V^{(0)}, V^{(1)}, V^{(2)}, V^{(3)})$ for f_b, with

$$U^{(1)} = U^{(0)} + \alpha' \quad U^{(3)} = U^{(2)} + \alpha' \quad V^{(2)} = V^{(0)} + \gamma \quad V^{(2)} = V^{(1)} + \gamma$$

Using an auxiliary path $\beta_j \to \beta'_j$, we construct $U_*^{(i)} = U^{(i)} + \beta_j$. With probability p_b^4, we obtain a new quartet $(U_*^{(0)}, U_*^{(1)}, U_*^{(2)}, U_*^{(3)}) \to (V_*^{(0)}, V_*^{(1)}, V_*^{(2)}, V_*^{(3)})$, where $V_*^{(i)} = V^{(i)} + \beta'_j$. Then, we have

$$U_*^{(1)} = U_*^{(0)} + \alpha' \quad U_*^{(3)} = U_*^{(2)} + \alpha' \quad V_*^{(2)} = V_*^{(0)} + \gamma \quad V_*^{(2)} = V_*^{(1)} + \gamma$$

We compute the plain-texts and cipher-texts corresponding to these values, and with probability $p_a^2 \cdot p_c^2$, this will result in a boomerang quartet for f, as shown in Figure 2.

If the initial cost to build a quartet for f_b is C, then we can build a quartet for the full f with complexity:

$$\frac{1}{p_a^2 p_c^2} \left(\frac{C}{b \cdot p_b^4} + 1 \right)$$

For the application to Skein, we use properties of the key-schedule to build a set of related-key differentials β_k with probability 1. This results in $C \ll b \cdot p_c^4$, and we essentially skip rounds in the middle of the permutation for free.

4 Application to Skein

Brief description of Skein The compression function of Skein is based on the block cipher Threefish. Let $U_{r,i}$ be the ith word of the encrypted state after r rounds and n_w be the number of words in a state. Then for each round we have

$$V_{r,i} = \begin{cases} U_{r,i} + K_{r/4,i} & \text{if } r \bmod 4 = 0 \\ U_{r,i} & \text{otherwise} \end{cases}$$

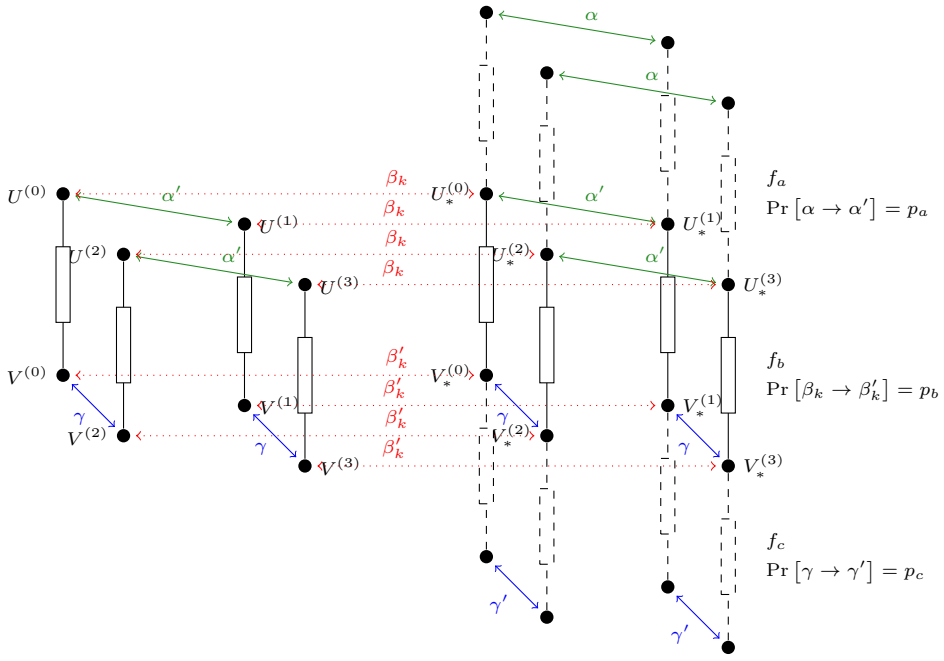

Fig. 2. Using auxiliary paths in a boomerang distinguisher

where $K_{r/4,i}$ is the ith word of the round key at round $r/4$. The state $U_{r+1,i}$ (for $i = 0, 1, .., n_w$) after round $r + 1$ is obtained from $V_{r,i}$ by applying a MIX transformation and a permutaion of n_w words as following:

$$\begin{array}{lll} (X_{r,2k}, X_{r,2k+1}) & := \texttt{MIX}_{r,k}(V_{r,2k}, V_{r,2k+1}) & \text{for } k = 0, 1, .., n_w/2 \\ U_{r+1,i} & := X_{r,\sigma(i)} & \text{for } i = 0, 1, .., n_w \end{array}$$

where σ is a permutaion specified in [6] and $(c, d) = \texttt{MIX}_{r,k}(a, b)$ is described as

$$\begin{aligned} c &= (a + b) \bmod 2^{64} \\ d &= (b \lll R_{r \bmod 8,k}) \oplus c \end{aligned}$$

The rotations $R_{r \bmod 8,k}$ are speicifed in [6]. The key scheduling algorithm of Threefish produces 18 round keys from a tweak (T_0, T_1) and a key as following

$$\begin{array}{ll} K_{l,i} = K_{(l+i) \bmod (n_w+1)} & \text{for } i = 0, 1, .., n_w - 4 \\ K_{l,i} = K_{(l+i) \bmod (n_w+1)} + T_{l \bmod 3} & \text{for } i = n_w - 3 \\ K_{l,i} = K_{(l+i) \bmod (n_w+1)} + T_{l \bmod 3} & \text{for } i = n_w - 2 \\ K_{l,i} = K_{(l+i) \bmod (n_w+1)} + l & \text{for } i = n_w - 1 \end{array} .$$

where $K_{n_w} = C_{240} \oplus \bigoplus_{i=0}^{n_w-1} K_i$ with C_{240} a constant specified in [6], and $T_2 = T_0 \oplus T_1$. The compression function F for Skein is given as $F = E_{CV,T}(M) \oplus M$. For Skein-256 $n_w = 4$ and word size is 8 byte. We use the notation $k^{(l)}$ to denote the expanded key used at round $4l$, *i.e.* $k^{(l)} = K_{l,0}, K_{l,1}, \ldots K_{l,n_w-1}$.

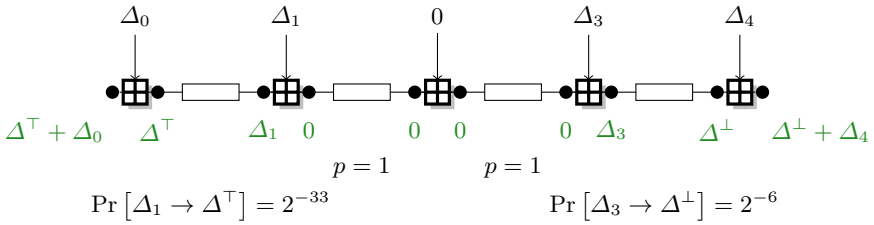

Fig. 3. Linearized differential path for Skein

Table 2. Subkey differential trails

Round	Subkey Trail$_1$: K_3, K_4, T_0, T_2				Round	Subkey Trail$_2$: K_2, K_3, T_0, T_1			
0	K_0	$K_1 + \underline{T_0}$	$K_2 + T_1$	$\underline{K_3} + 0$	16	K_4	$K_0 + \underline{T_1}$	$K_1 + T_2$	$\underline{K_2} + 4$
4	K_1	$K_2 + T_1$	$\underline{K_3} + \underline{T_2}$	$\underline{K_4} + 1$	20	K_0	$K_1 + T_2$	$\underline{K_2} + \underline{T_0}$	$\underline{K_3} + 5$
8	K_2	$\underline{K_3} + \underline{T_2}$	$\underline{K_4} + \underline{T_0}$	$K_0 + 2$	24	K_1	$\underline{K_2} + \underline{T_0}$	$\underline{K_3} + \underline{T_1}$	$K_4 + 6$
12	$\underline{K_3}$	$\underline{K_4} + \underline{T_0}$	$K_0 + T_1$	$K_1 + 3$	28	$\underline{K_2}$	$\underline{K_3} + \underline{T_1}$	$K_4 + T_2$	$K_0 + 7$
16	$\underline{K_4}$	$K_0 + T_1$	$K_1 + \underline{T_2}$	$K_2 + 4$	32	$\underline{K_3}$	$K_4 + T_2$	$K_0 + \underline{T_0}$	$K_1 + 8$

4.1 Round-reduced Differential Trails in Skein-256

Due to the key schedule of Skein, it is possible to build differential trails over 8 rounds with probability one, using a difference in the tweak T. To do this, we just use a key difference and tweak difference that cancel each other at a given round r, and we compute the corresponding key differences for rounds $r+4$ and $r-4$. By injecting this difference in the state, we obtain an 8-round path.

In order to achieve the best probability, we use a difference Δ_{msb} on the most significant bit of both tweaks used at round r, and on the corresponding keys. This results in a 12-round path with probability 2^{-6} and a 16-round path with probability 2^{-43} (we don't consider the key addition for those probabilities). The path is shown in Figure 3. For a boomerang attack on 32-round Skein, we use this with $r = 8$ and $r = 24$, and the corresponding key-differential as shown in Table 2. Previous analysis of Skein[1,5,15,18] are based on the same trails.

For our attack, we also need a set of auxiliary paths. We build this set using the same 8-round paths with probability one, but we do not restrict ourselves to a difference on the most significant bit. We can set the tweak T to an arbitrary value, and recompute the key in order the have the same expanded key $k^{(4)}$ at rounds 16. This gives a set of 2^{128} paths with probability one.

4.2 Description of the attack on Skein-256

Our attack on skein is similar to a boomerang attack on 32 rounds using two 16-round trails, but we build a valid quartet starting from the middle, and we use auxiliary trails to avoid paying the probabilities of 8 rounds in the middle. We proceed with three consecutive steps, as shown by Figure 5, page 16.

First step. The first part of the attack considers rounds 16 to 20. We try to build a quartet $u^{(i)} = R(t^{(i)})$ with

$$t^{(0)} \oplus t^{(1)} = t^{(2)} \oplus t^{(3)} = \Delta^{\perp} \oplus \Delta_4 \tag{2}$$

$$u^{(0)} \oplus u^{(2)} = u^{(1)} \oplus u^{(3)} = \Delta_1 \tag{3}$$

One possible way to build such a quartet is to start with a set of $t^{(i)}$ that satisfies (2) and $t^{(0)} \oplus t^{(2)} = t^{(1)} \oplus t^{(3)} = \Delta^{\perp} \oplus \Delta_4$, and to compute the corresponding $u^{(i)} = R(t^{(i)})$. The quartet will satisfy (3) with probability 2^{-66}, but we can fix some bits of the state in order to improve this complexity.

Actually, it is more efficient to follow the steps of a boomerang attack:

- start from a pair $t^{(0)}, t^{(1)}$ with $t^{(0)} \oplus t^{(1)} = \Delta^{\perp} \oplus \Delta_4$;
- compute $u^{(0)} = R(t^{(0)})$, $u^{(1)} = R(t^{(1)})$, $u^{(2)} = u^{(0)} \oplus \Delta_1$, $u^{(3)} = u^{(1)} \oplus \Delta_1$;
- compute $t^{(2)} = R^{-1}(u^{(2)})$ and $t^{(3)} = R^{-1}(u^{(3)})$; check whether (2) holds.

Using this procedure allows us to benefit from amplified probabilities, since we do not specify the path from $(u^{(0)}, u^{(2)})$ to $(t^{(0)}, t^{(2)})$ and from $(u^{(1)}, u^{(3)})$ to $(t^{(1)}, t^{(3)})$, respectively, we only check that the differences are the same. Experimentations show that this step costs around 2^{18}.

Second step. The second part of the attack concerns rounds 12 to 16. We start with a quartet $u^{(i)} = R(t^{(i)})$ satisfying (2) and (3), and we want to extend it with $s^{(i)} = R^{-1}(t^{(i)} - k_4^{(i)})$ so that

$$s^{(0)} \oplus s^{(1)} = s^{(2)} \oplus s^{(3)} = \Delta_3. \tag{4}$$

The main idea is to use the key injection at round 16 in order to randomize the state, until we find pairs that follows the differential $\Delta^{\perp} \to \Delta_3$. First we select the keys that result in:

$$(t^{(0)} - k_4^{(0)}) \oplus (t^{(1)} - k_4^{(1)}) = (t^{(2)} - k_4^{(2)}) \oplus (t^{(3)} - k_4^{(3)}) = \Delta^{\perp} \qquad \text{with}$$
$$k_4^{(0)} \oplus k_4^{(1)} = k_4^{(2)} \oplus k_4^{(3)} = \Delta_4 \qquad \text{and} \qquad k_4^{(0)} \oplus k_4^{(2)} = k_4^{(1)} \oplus k_4^{(3)} = \Delta_0.$$

We can find the suitable solution by solving a simple system of additions and xors. Then, we compute the corresponding $s^{(i)}$, and we check whether(4) is satisfied. On average, we expect this step to cost 2^{12}. According to our experimentations, however, there seem to be some dependency between the paths, and the average cost is about 2^{18}.

This step can be seen as an application of the technique of Section 3. We use a trivial related-key differential where the key difference just cancels the state difference in order to extend a 4-round quartet into 8-round quartets.

Third step. This is the main step of the attack, following the ideas of Section 3. We start with a quartet $u^{(i)} = R(R(s^{(i)}) + k_4^{(i)})$, and we use probability-1

differentials to build many more quartets, until the top and bottom paths are satisfied.

The best result are achieved using the modular difference, because the key-additions are modular additions. Note that we include the initial and final key-addition in our 32-round reduced Threefish/Skein. More precisely, for each quartet generated for rounds 12–16 verifying (3) and (4), we compute the corresponding plain-text and cipher-text and we check whether

$$\Delta^{\boxplus} P^{(i)} = (P^{(3)} \boxminus P^{(2)}) \boxminus (P^{(1)} \boxminus P^{(0)}) = 0 \tag{5}$$

$$\Delta^{\boxplus} C^{(i)} = (C^{(3)} \boxminus C^{(2)}) \boxminus (C^{(1)} \boxminus C^{(0)}) = 0 \tag{6}$$

Experimentally, a quartet satisfies (5) with probability 2^{-36} and (6) with probability 2^{-21} (see Appendix A). This gives a distinguisher for the keyed permutation with complexity around 2^{57}. Note that if we do an analysis similar to the one in [5], we would expect this attack to have a complexity of around 2^{95}; the amplification effect detected in practice is much stronger than predicted by [5].

If we want to build a distinguisher for the compression function, we will instead use the xor-difference, because the feed-forward is an xor operation. Therefore, we will check whether:

$$\Delta^{\oplus} P^{(i)} = P^{(0)} \oplus P^{(1)} \oplus P^{(2)} \oplus P^{(3)} = 0 \tag{7}$$

$$\Delta^{\oplus} C^{(i)} = C^{(0)} \oplus C^{(1)} \oplus C^{(2)} \oplus C^{(3)} = 0 \tag{8}$$

Experimentally, we find that (8) is verified with probability 2^{-24}. The probability for (7) is too low to check experimentally, but we can estimate it from the probability of (5): a quartet satisfying (5) is composed of two pairs of plain-text with $(P^{(1)} \boxminus P^{(0)}) = (P^{(3)} \boxminus P^{(2)}) = \Delta$, where Δ has weight around 34. For each active position, there is a probability 1/3 that the carry extension in $(P^{(0)}, P^{(1)})$ is the same as in $(P^{(2)}, P^{(3)})$, which leads to:

$$\Pr\left[\Delta^{\oplus} P^{(i)} = 0\right] \geq \Pr\left[\Delta^{\boxplus} P^{(i)} = 0\right] \times (1/3)^{34} \geq 2^{-90}.$$

This gives a distinguisher on the compression function with complexity 2^{114}. In practice we expect the complexity to be significantly lower, because a quartet satisfying (7) does not necessarily satisfy (5).

Attack on 24 and 28 rounds can be build with the same approach.

5 Extensions and Limitations

The technique described in the previous sections can be applied to improve almost any chosen-key boomerang distinguisher. The main limitation is that we need to be able to generate an initial quartet for the middle rounds, similarly to what we do in steps one and two of the attack on Skein-256. We note that any successful boomerang attack does provide such quartets; therefore, as long as a

standard boomerang attack works, our improved attack with auxiliary differentials will also work.

However, an often overlooked problem of boomerang attacks is that we need the top and bottom paths to be somehow independent. More precisely, in a standard boomerang attack as depicted by Figure 1, we expect that if a pair $(P_0, P_1) \to (X_0, X_1)$ with is a good pair for f_a (*i.e.* $P_1 = P_0 + \alpha$ and $X_1 = X_0 + \alpha'$), then the pair $(X_0 + \gamma, X_1 + \gamma)$ behaves like a random pair regarding f_a, and will satisfy the differential with probability p_a. However, in practice this may not be the case.

In the following section, we discuss cases where boomerang attacks on ARX design can fail because of this property. This problem was already discussed by Murphy in [12], where he gives examples of non-compatible paths for the DES and the AES. It has also been discussed by Sasaki in [14] for boomerang attacks on Haval.

5.1 Extension to more rounds

We tried to extend the attack by adding middle rounds, at the bottom of the top path, or at the top of the bottom path. For instance, using a 16-round path for the bottom part should only increase the complexity by a factor of roughly 2^{12}. However, this usually results in incompatible paths, for which no valid quartet exist. In particular linearized paths are incompatible, and we have not been able to build compatible paths.

By studying those incompatible paths, we found that very simple patterns can lead to incompatibilities. Figure 4 gives an example of a pattern that results in incompatible paths. A quartet following those paths would have to satisfy:

$$x^{(0)} \oplus x^{(2)} = x^{(1)} \oplus x^{(3)} = \mathtt{01} \quad y^{(0)} \oplus y^{(2)} = y^{(1)} \oplus y^{(3)} = \mathtt{00} \quad \textit{(Top)} \tag{9}$$

$$x^{(0)} \oplus x^{(1)} = x^{(2)} \oplus x^{(3)} = \mathtt{01} \quad y^{(0)} \oplus y^{(1)} = y^{(2)} \oplus y^{(3)} = \mathtt{01} \quad \textit{(Bottom)} \tag{10}$$

$$x^{(0)} \boxplus y^{(0)} = x^{(1)} \boxplus y^{(1)} \qquad x^{(2)} \boxplus y^{(2)} = x^{(3)} \boxplus y^{(3)} \tag{11}$$

Without loss of generality, we can assume that $\mathrm{lsb}(x^{(0)}) = 0$. This implies $\mathrm{lsb}(x^{(1)}) = 1$ from (10), and $\mathrm{lsb}(x^{(2)}) = 1$ and $\mathrm{lsb}(x^{(3)}) = 0$ from (9). We can deduce $y^{(0)} = y^{(1)} \boxplus 1$ and $y^{(3)} = y^{(2)} \boxplus 1$ from (11). Combined with (10) this yields $\mathrm{lsb}(y^{(0)}) = 1$, $\mathrm{lsb}(y^{(1)}) = 0$, $\mathrm{lsb}(y^{(2)}) = 0$, and $\mathrm{lsb}(y^{(3)}) = 1$. This is incompatible with (9).

This pattern seems to appear very frequently when using linearized paths in ARX designs, and shows that some very natural paths cannot be combined in a boomerang attack.

5.2 Application to Skein-512

By applying our technique to Skein-512, we would expect distinguishers with a similar complexity for the same number of rounds. However, in order to apply the technique, we need to be able to generate quartets for the middle rounds, and we

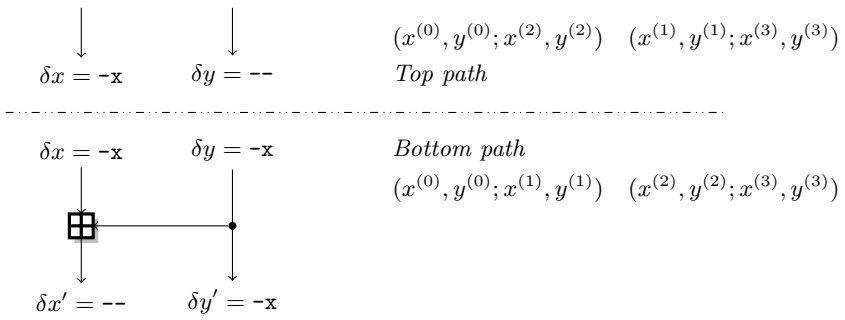

Fig. 4. Example of incompatible paths

failed to build any such quartet for a 32-round attack. Since a boomerang attack on Skein-512 was presented in [5], we studied the paths used in this attacks, and we found that they are in fact not compatible.

Following the notations of [5], the path for rounds 0–16 has a difference $e_{15,5}[10, 39, 49, *64]$. If this path is applied to states $(e^{(0)}, e^{(1)})$ and $(e^{(2)}, e^{(3)})$, this implies that on bit 49, we have:

$$v_{15,5}^{(0)} = 0 \qquad v_{15,5}^{(1)} = 1 \qquad v_{15,5}^{(2)} = 0 \qquad v_{15,5}^{(3)} = 1 \tag{12}$$

Assuming there are no carries, the path for round 16–32 has $e_{16,5}[-34, -50]$ and $e_{16,2}[5, 11, 16, 41, 44, 47]$. Since $e_{16,2}, e_{16,5} = \texttt{MIX}(e_{15,4}, e_{15,5})$, and the rotation used at that step is 56, we have $e_{15,5} = (e_{16,2} \oplus e_{16,5})^{\ggg 56}$. This results in $e_{15,5}$ being active on bits 13, 19, 24, 42, 49, 52, 55, and 58. If this path is applied to states $(e^{(0)}, e^{(2)})$ and $(e^{(1)}, e^{(3)})$, this implies that on bit 49, we have $v_{15,5}^{(0)} \neq v_{15,5}^{(2)}$ and $v_{15,5}^{(0)} \neq v_{15,5}^{(2)}$. This is contradictory with (12).

We tried to fix the paths using a carry extension from bit 34 to 41 in $e_{16,5}$, by using different signs in the paths $(e^{(0)}, e^{(1)})$ and $(e^{(2)}, e^{(3)})$, or by using a carry extension in $e_{15,5}[49]$, but this always ends up with a similar contradiction. However, we note that if the attack of [5] can be fixed, then our technique is expected to yield a distinguisher on Skein-512 with complexity similar to the distinguisher on Skein-256.

6 Conclusions

In this paper we have presented a technique to improve the boomerang attack in the chosen-key setting and applied it to obtain an efficient distinguisher for 32 rounds of the compression function of Skein-256. We also discuss extension of the attack, and application of the technique to Skein-512. This technique can essentially be used to improve any chosen-key boomerang distinguisher. However, we explain that boomerang attack on ARX-design can fail because of incompatible paths. This is not a limitation of the auxiliary paths, but of the underlying boomerang technique.

Acknowledgment. Gaëtan Leurent is supported by the AFR grant PDR-10-022 of the FNR Luxembourg.

References

1. Aumasson, J.P., Calik, C., Meier, W., Ozen, O., Phan, R.C.W., Varici, K.: Improved cryptanalysis of Skein. In: ASIACRYPT. Volume 5912 of LNCS., Springer (2009) 542–559
2. Biham, E., Dunkelman, O., Keller, N.: The Rectangle Attack - Rectangling the Serpent. In Pfitzmann, B., ed.: EUROCRYPT. Volume 2045 of Lecture Notes in Computer Science., Springer (2001) 340–357
3. Biham, E., Dunkelman, O., Keller, N.: Related-Key Boomerang and Rectangle Attacks. In Cramer, R., ed.: EUROCRYPT. Volume 3494 of Lecture Notes in Computer Science., Springer (2005) 507–525
4. Biryukov, A., Nikolic, I., Roy, A.: Boomerang attacks on BLAKE-32. In Joux, A., ed.: FSE. Volume 6733 of Lecture Notes in Computer Science., Springer (2011) 218–237
5. Chen, J., Jia, K.: Improved related-key boomerang attacks on round-reduced Threefish-512. In Kwak, J., Deng, R.H., Won, Y., Wang, G., eds.: ISPEC. Volume 6047 of Lecture Notes in Computer Science., Springer (2010) 1–18
6. Ferguson, N., Lucks, S., Schneier, B., Whiting, D., Bellare, M., Kohno, T., Callas, J., Walker, J.: The Skein hash function family. Submission to NIST (2008/2010)
7. Joux, A., Peyrin, T.: Hash functions and the (amplified) boomerang attack. In Menezes, A., ed.: CRYPTO. Volume 4622 of Lecture Notes in Computer Science., Springer (2007) 244–263
8. Kelsey, J., Kohno, T., Schneier, B.: Amplified boomerang attacks against reduced-round MARS and Serpent. In Schneier, B., ed.: FSE. Volume 1978 of Lecture Notes in Computer Science., Springer (2000) 75–93
9. Khovratovich, D., Nikolic, I.: Rotational cryptanalysis of ARX. In Hong, S., Iwata, T., eds.: FSE. Volume 6147 of Lecture Notes in Computer Science., Springer (2010) 333–346
10. Khovratovich, D., Nikolic, I., Rechberger, C.: Rotational rebound attacks on reduced Skein. In Abe, M., ed.: ASIACRYPT. Volume 6477 of Lecture Notes in Computer Science., Springer (2010) 1–19
11. Lamberger, M., Mendel, F.: Higher-order differential attack on reduced SHA-256. Cryptology ePrint Archive, Report 2011/037 (2011) `http://eprint.iacr.org/`.
12. Murphy, S.: The return of the cryptographic boomerang. IEEE Transactions on Information Theory **57**(4) (2011) 2517–2521
13. National Institute of Standards and Technology: Cryptographic hash algorithm competition. `http://csrc.nist.gov/groups/ST/hash/sha-3/index.html`
14. Sasaki, Y.: Boomerang distinguishers on MD4-based hash functions: First practical results on full 5-pass HAVAL. In: SAC. (2011)
15. 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. Volume 6467 of Lecture Notes in Computer Science., Springer (2010) 124–139
16. Wagner, D.: The boomerang attack. In Knudsen, L.R., ed.: FSE. Volume 1636 of Lecture Notes in Computer Science., Springer (1999) 156–170

17. Wagner, D.: A Generalized Birthday Problem. In Yung, M., ed.: CRYPTO. Volume 2442 of Lecture Notes in Computer Science., Springer (2002) 288–303
18. Yu, H., Chen, J., Ketingjia, Wang, X.: Near-collision attack on the step-reduced compression function of Skein-256. Cryptology ePrint Archive, Report 2011/148 (2011) http://eprint.iacr.org/.

A Boomerang quartets for 28 round Threefish

In this section, we give examples of quartets, to show that the techniques used for the 32-round attack are valid. Table 3 gives is a zero-sum for rounds 4–32 of Threefish, with $\Delta^{\boxplus} P'^{(i)} = 0$ and $\Delta^{\boxplus} C^{(i)} = 0$. Generating such a quartet costs around 2^{21}. Table 4 gives is a zero-sum for rounds 0–28 of Threefish, with $\Delta^{\boxplus} P^{(i)} = 0$ and $\Delta^{\boxplus} C'^{(i)} = 0$. Generating such a quartet costs around 2^{36}.

Table 3. A quartet that satisfies the paths for rounds 4–32

Plain-text before round 4				
$P'^{(0)}$	fe5ab24b9481e005	dcf5504b75b919e5	076e43e18e3a50ce	d31433344b540c75
$P'^{(1)}$	fe5ab24b9481e005	dcf5504b75b919e5	076e43e18e3a50ce	531433344b540c75
$P'^{(2)}$	64309deb8a55633f	71f578e8ddfd6e89	4e299c34006568b0	95847e8860164845
$P'^{(3)}$	64309deb8a55633f	71f578e8ddfd6e89	4e299c34006568b0	15847e8860164845
Key				
$K^{(0)}$	674dfabf537e5a73	92a94934d0ca3e21	90ce87c17d8540d1	ff65c869e8cdadd4
$K^{(1)}$	674dfabf537e5a73	92a94934d0ca3e21	90ce87c17d8540d1	7f65c869e8cdadd4
$K^{(2)}$	674dfabf537e5a73	92a94934d0ca3e21	10ce87c17d8540d1	7f65c869e8cdadd4
$K^{(3)}$	674dfabf537e5a73	92a94934d0ca3e21	10ce87c17d8540d1	ff65c869e8cdadd4
Tweak				
$T^{(0,1)}$	182d916b255ae5e8	5cba243a3b82278e	982d916b255ae5e8	5cba243a3b82278e
$T^{(2,2)}$	982d916b255ae5e8	dcba243a3b82278e	182d916b255ae5e8	dcba243a3b82278e

Table 4. A quartet that satisfies the paths for rounds 0–28

Plain-text before round 0				
$P^{(0)}$	d9d7934ee20a9c9a	d7c7d25a8f42f324	25ac377afcb411bb	424daed3f2425bc1
$P^{(1)}$	d4d82358b1e9945a	58c7c2587f63fb24	25ec3b7b6eb84fb7	c20daed37e421dc5
$P^{(2)}$	0404cce3c56e92df	1887e00caa229acc	5bdad7995f5f036a	a7b69f1a1274d559
$P^{(3)}$	ff055ced954d8a9f	9987d00a9a43a2cc	5c1adb99d1634166	27769f199e74975d
Key				
$K^{(0)}$	8cb950f444a069e3	48380fb03c6b84c6	2034665dbf7fbfb9	59a45c529130786a
$K^{(1)}$	8cb950f444a069e3	48380fb03c6b84c6	2034665dbf7fbfb9	d9a45c529130786a
$K^{(2)}$	8cb950f444a069e3	48380fb03c6b84c6	a034665dbf7fbfb9	d9a45c529130786a
$K^{(3)}$	8cb950f444a069e3	48380fb03c6b84c6	a034665dbf7fbfb9	59a45c529130786a
Tweak				
$T^{(0,1)}$	684e3541ef841667	b3a8cd11bb94bb5d	e84e3541ef841667	b3a8cd11bb94bb5d
$T^{(2,3)}$	e84e3541ef841667	33a8cd11bb94bb5d	684e3541ef841667	33a8cd11bb94bb5d

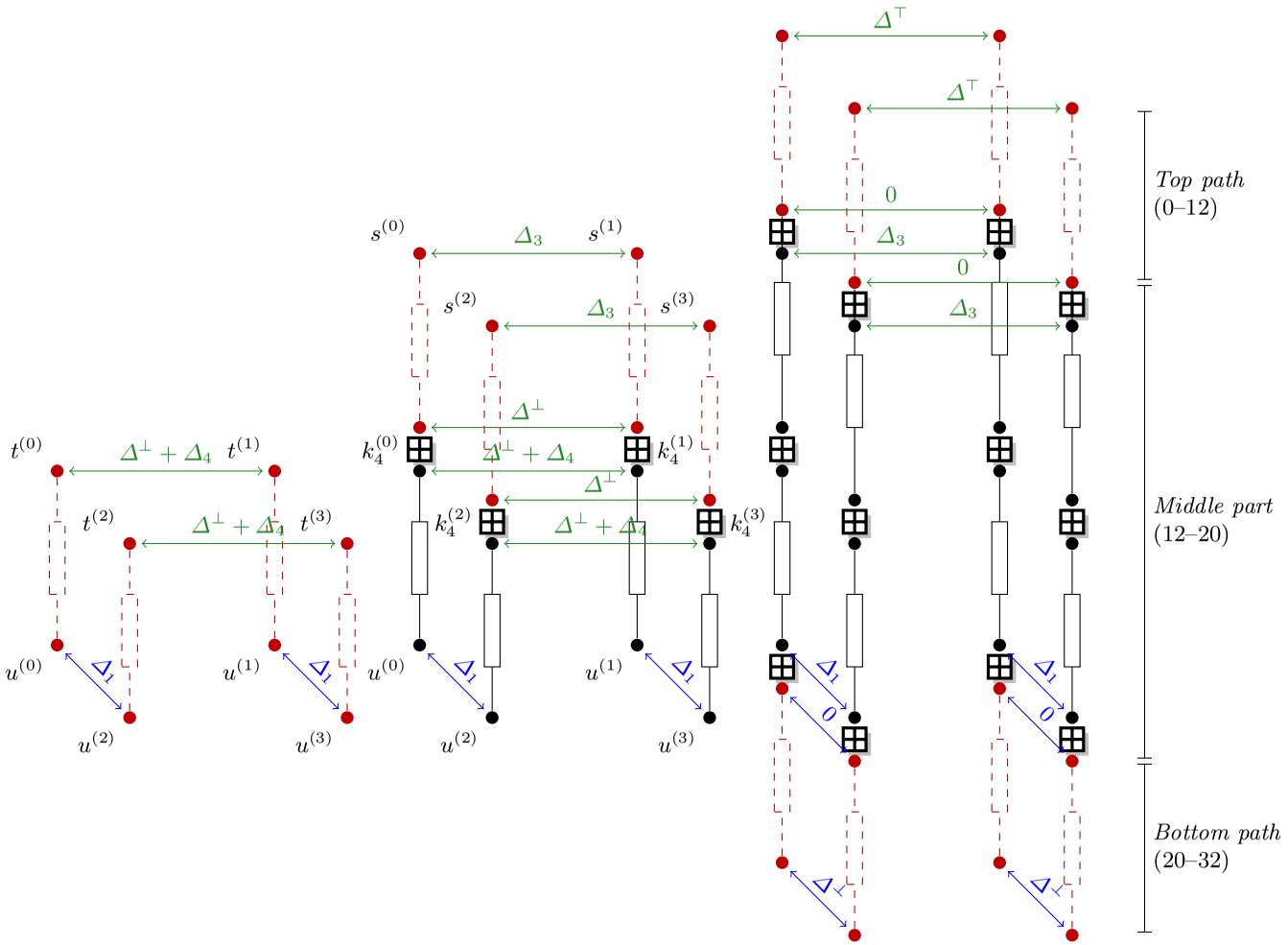

Fig. 5. Overview of the attack, showing the three consecutive steps.