Exploiting Non-Full Key Additions: Full-Fledged Automatic Demirci-Selçuk Meet-in-the-Middle Cryptanalysis of SKINNY

Danping Shi[1,2], Siwei Sun[3(✉)], Ling Song[4], Lei Hu[1,2], and Qianqian Yang[1,2]

[1] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{shidanping, hulei, yangqianqian}@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3] School of Cryptology, University of Chinese Academy of Sciences, Beijing, China
sunsiwei@ucas.ac.cn
[4] Jinan University, Guangzhou, China songling.qs@gmail.com

Abstract. The Demirci-Selçuk meet-in-the-middle (DS-MITM) attack is a sophisticated variant of differential attacks. Due to its sophistication, it is hard to efficiently find the best DS-MITM attacks on most ciphers *except* for AES. Moreover, the current automatic tools only capture the most basic version of DS-MITM attacks, and the critical techniques developed for enhancing the attacks (e.g., differential enumeration and key-dependent-sieve) still rely on manual work. In this paper, we develop a full-fledged automatic framework integrating all known techniques (differential enumeration, key-dependent-sieve, and key bridging, etc) for the DS-MITM attack that can produce key-recovery attacks directly rather than only search for distinguishers. Moreover, we develop a new technique that is able to exploit partial key additions to generate more linear relations beneficial to the attacks. We apply the framework to the SKINNY family of block ciphers and significantly improved results are obtained. In particular, all known DS-MITM attacks on the respective versions of SKINNY are improved by at least 2 rounds, and the data, memory, or time complexities of some attacks are reduced even compared to previous best attacks penetrating less rounds.

Keywords: Demirci-Selçuk MITM Attacks, Differential Enumeration, Key-dependent Sieve, SKINNY

1 Introduction

DS-MITM attack was introduced by Demirci and Selçuk [6] to attack AES in FSE 2008. Let $\{P^0, P^1, \ldots, P^{255}\}$ be a set of 2^8 plaintexts for 4-round AES such that the i-th $(0 \leq i < 16)$ byte of these plaintexts traversing $\mathbb{F}_2^8$ and all other bytes of them are fixed to some constant. Basically, Demirci and Selçuk in [6] showed that the value of the sequence $C^0[j]||C^1[j]||\dots||C^{255}[j]$ formed by concatenating the jth byte of the corresponding ciphertexts $\{C^0, C^1, \dots, C^{255}\}$ of $\{P^0, P^1, \dots, P^{255}\}$ can be fully determined by 25 8-bit parameters. Moreover, it is observed in [7] that the value of the sequence $C^0[j] \oplus C^1[j]||C^0 \oplus C^2[j]||\dots, C^0 \oplus C^{255}[j]$ can be fully determined by 24 8-bit parameters. Therefore, $C^0[j] \oplus C^1[j]||C^0 \oplus C^2[j]||\dots, C^0 \oplus C^{255}[j]$ can take at most $(2^8)^{24}$ different values, while for a random 255-byte sequence, it has $(2^8)^{255}$ possibilities. Obviously, this behavior forms a distinguisher. In this work, we say that the degree of freedom of the output sequence is 24 bytes.

Since then, many improvement techniques have been proposed to enhance the attack [13,8,10,16,9], and DS-MITM produces the best cryptanalytic results on AES in the single-key model [10,16,17]. In 2010, Dunkelman et al. introduced the so-called *differential enumeration* technique to reduce the degree of freedom of the output sequence, where the input plaintext set is constructed such that it contains one message conforming to a given truncated differential [13]. The *differential enumeration* technique was further improved in [8]. Also, Dunkelman et al. exploited the algebraic relations (named as *key bridges*) to reduce the space of the candidate keys [13]. Another improvement is to consider a multiset, i.e. an unordered set with multiplicity, other than an ordered sequence, which reduces the possibilities by a factor 4 [13]. The key-dependent-sieve technique was introduced in [16] to further reduce the degree of freedom of the output sequence by considering the relations induced by the key-schedule algorithm on the parameters that fully determine the value of the output sequence.

In order to find DS-MITM attack efficiently, some tools have been proposed in the literature. In [8,9], a dedicated search algorithm for DS-MITM attacks implemented in `C/C++` was presented by Derbez and Fouque. Shi et al. proposed a constraint programming (CP) based approach for automatizing the search of DS-MITM distinguishers, whose most important advantage is the decoupling of the modeling and resolution processes of the cryptanalytic technique [20]. However, the CP-based model presented in [20] only capture the most basic version of the DS-MITM attack, and those critical techniques developed for enhancing the attacks (e.g., differential enumeration and key-dependent-sieve) still rely on manual work.

Our Contributions. We develop a full-fledged automatic framework for DS-MITM attacks on *tweakable* block ciphers that integrates all known techniques, including but not limited to differential enumeration, key-dependent-sieve, and key bridging techniques. This tool makes full use of the ability of choosing tweaks when the target cipher is a tweakable block cipher, and our tool is able to output a configuration of a DS-MITM key-recovery attack directly, and thus avoid trapping into the situation where an optimal distinguisher may lead to a sub-optimal key-recovery attack. Note that the automation of the differential enumeration technique is highly nontrivial, and it is enabled by a thorough analysis on how to synthesize the objective function from the variables involved in the model.

Moreover, we propose a method for describing the dependencies between the variables linked by a linear transformation and a *non-full* key addition based on the rank of a matrix derived from the linear transformation. With this method, the dependencies within the rounds of an iterative block cipher due to non-full key additions can be fully exploited to reduce the degree of freedom of the output sequence. Note that this technique alone can improve the previous best DS-MITM attack on SKINNY-128-384 by 1 round in the single-key and single-tweak setting as shown in Section G.

We apply the framework to the SKINNY family of block ciphers and the results are summarized in Table 1, from which we can see that all known DS-MITM attacks on the respective versions of SKINNY are improved by at least 2 rounds, and the data, memory, or time complexities of some attacks are reduced even compared to previous best attacks penetrating less rounds. We note that most of the key-recovery attacks listed in Table 1 are not extended from the best distinguishers we can find by changing the objective of the model to identify the optimal distinguishers instead of the best key-recovery attacks.

Organization. In Sect. 2, we give a brief description of DS-MITM attack and SKINNY block cipher. Then in Sect. 3, we present the generalized new non-full key-addition technique. In Sect.4, we present a unified full-fledged automatic framework integrating all known techniques (e.g. differential enumeration, key-dependent-sieve, tweak-difference cancellation, non-full key-addition) for the DS-MITM attack and apply it to SKINNY. Sect. 5 presents the results of SKINNY. Finally, we propose some discussions in Sect. 6. Please refer to the full version [21] for more details. Relevant source codes can be found via https://github.com/shidanping/DS-MITM.

2 Primarily

2.1 Notations

The following notations will be used in this paper.

- The input state of rth round is denoted by $\mathbf{S}_r$ and jth cell of n-cell state $\mathbf{S}_r$ is represented by $\mathbf{S}_r[j]$. Let P^k represent kth plaintext and C^k represent associated ciphertext. The parameter of P^k in the internal cell $\mathbf{S}_r[j]$ is denoted by $P^k[\mathbf{S}_r[j]]$. Let $P \oplus P'[\mathbf{S}_r[j]]$ represent $P[\mathbf{S}_r[j]] \oplus P'[\mathbf{S}_r[j]]$.
- Assume $\mathcal{B} = [\mathbf{S}_r[j_0], \mathbf{S}_r[j_1], \dots, \mathbf{S}_r[j_t]]$ is a sequence of positions. Then the concatenation $P[\mathbf{S}_r[j_0]]||P[\mathbf{S}_r[j_1]] \dots ||P[\mathbf{S}_r[j_t]]$ of P ($P \oplus P'$ respectively) in positions specified by $\mathcal{B}$ is denoted by $P[\mathcal{B}]$ ($P \oplus P'[\mathcal{B}]$ respectively). The set of $\{P[\mathbf{S}_r[j_0]], P[\mathbf{S}_r[j_1]], \dots, P[\mathbf{S}_r[j_t]]\}$ is also represented by $\{P[j] : j \in \mathcal{B}\}$.
- Let $\mathcal{E}_1$ and $\mathcal{E}_r$ be 1-round and r-round function of an iterative block cipher respectively. $\mathcal{E}_1$ maps input state $\mathbf{S}_r$ to output state $\mathbf{S}_{r+1} = \mathcal{E}_1(\mathbf{S}_r)$.
- $|*|$ represents the size of a set or table $*$.

A δ-set was first proposed by Daemen and Rijmen [5], which is a structure of 256 plaintexts by traversing one byte while sharing same value in other bytes. Lin et al. extended the definition of δ-set to multiple active bytes [19].

Table 1. Summary results of SKINNY in the single-key setting, where ID, ZC, Int and MITM denote the impossible differential, zero correlation, integral and classic meet-in-the-middle attack respectively

Version	Approach	R_{attack}	Time	Data	Memory	CT	Ref.
SKINNY-128-128	ID	17	$2^{120.8}$	$2^{118.5}$	$2^{97.5}$	✗	[23]
	ID	17	$2^{116.51}$	$2^{116.37}$	2^{80}		[15]
	DS-MITM	17	$2^{122.06}$	2^{96}	$2^{118.91}$		Sect. L, Fig. 35
SKINNY-128-256	ID	19	$2^{119.8}$	2^{62}	2^{110}	✗	[23]
	ID	19	$2^{219.23}$	$2^{117.86}$	2^{208}		[15]
	DS-MITM	19	$2^{238.26}$	2^{96}	$2^{210.99}$		[14]
	DS-MITM	19	$2^{235.05}$	2^{96}	$2^{207.7}$		Sect. I, Fig. 29
	DS-MITM	20	$2^{254.28}$	2^{96}	$2^{250.99}$		Sect. H, Fig. 27
	DS-MITM	21	$2^{234.84}$	2^{96}	$2^{183.52}$	✓	Sect. A, Fig. 13(`8-bit tweak`)
	DS-MITM	21	$2^{234.99}$	2^{64}	$2^{231.86}$		Sect. C, Fig. 17(`8-bit tweak`)
	Int	22	2^{216}	$2^{113.58}$	2^{216}		[15]
SKINNY-128-384	ID	22	$2^{373.48}$	$2^{92.22}$	$2^{147.22}$	✗	[22]
	ID	21	$2^{347.35}$	$2^{122.89}$	2^{336}		[15]
	MITM	23	2^{368}	2^{120}	2^{16}		[2]
	DS-MITM	22	$2^{366.28}$	2^{96}	$2^{370.99}$		[4]
	DS-MITM	23	2^{372}	2^{96}	$2^{352.46}$		Sect. G, Fig. 25
	DS-MITM	25	$2^{363.83}$	2^{96}	$2^{336.39}$	✓	Sect. 5.2, Fig. 11(`8-bit tweak`)
	Int	26	2^{344}	2^{121}	2^{340}		[15]
SKINNY-64-128	ID	18	2^{116}	2^{60}	2^{112}	✗	[12]
	ID	19	$2^{119.8}$	2^{60}	2^{112}		[23]
	ID	19	$2^{110.34}$	$2^{60.86}$	2^{104}		[15]
	DS-MITM	18	$2^{126.32}$	2^{32}	$2^{61.91}$		[14]
	DS-MITM	19	$2^{123.43}$	2^{52}	$2^{126.95}$		Sect. N, Fig. 39
	DS-MITM	21	$2^{119.32}$	2^{60}	$2^{114.81}$	✓	Sect. D, Fig. 19(`8-bit tweak`)
	ZC/Integral	20	$2^{97.5}$	$2^{68.4}$	2^{82}		[1]
	Int	22	2^{110}	$2^{57.58}$	2^{108}		[15]
SKINNY-64-192	ID	22	$2^{183.97}$	$2^{47.84}$	$2^{74.84}$	✗	[22]
	ID	21	$2^{174.42}$	$2^{62.43}$	2^{168}		[15]
	MITM	23	2^{188}	2^{52}	2^{4}		[11]
	MITM	23	2^{188}	2^{28}	2^{4}		[2]
	MITM	23	2^{184}	2^{60}	2^{8}		[2]
	DS-MITM	21	$2^{186.63}$	2^{60}	$2^{133.99}$		[14]
	DS-MITM	21	$2^{180.01}$	2^{44}	$2^{191.55}$		Sect. K, Fig. 33
	DS-MITM	23	$2^{179.9}$	2^{32}	$2^{183.49}$	✓	Sect. F, Fig. 23(`8-bit tweak`)
	DS-MITM	23	$2^{174.9}$	2^{56}	$2^{179.46}$		Sect. E, Fig. 21(`16-bit tweak`)
	ZC/Integral	23	$2^{155.6}$	$2^{73.2}$	2^{138}		[1]
	Int	26	2^{172}	2^{61}	2^{172}		[15]
SKINNY-64-64	ID	17	$2^{61.8}$	$2^{59.5}$	$2^{49.6}$	✗	[23]
	ID	17	2^{59}	$2^{58.79}$	2^{40}		[15]
	DS-MITM	17	$2^{62.06}$	2^{48}	$2^{61.91}$		Sect. O, Fig. 41

[1] ✓represents chosen-tweak model (CT).

Definition 1 **($\delta(\mathcal{A})$-set).** *A set of messages $\{P^0, P^1, \ldots, P^N\}$ that are all different in positions specified by $\mathcal{A}$ ($P^0 \oplus P^k[\mathcal{A}] = k$) and all equal in other positions, where $\mathcal{A} = [\mathbf{S}_r[j_0], \mathbf{S}_r[j_1], \ldots, \mathbf{S}_r[j_s]]$ is a sequence of positions.*

An ordered difference sequence of the associated $\delta(\mathcal{A})$-set expressed in definition 2 will be utilized in DS-MITM attack.

Definition 2 **($\Delta\mathcal{E}_r(\delta(\mathcal{A}))[\mathcal{B}]$-sequence).** *An ordered sequence $P^0[\mathcal{B}] \oplus P^1[\mathcal{B}]||$ $P^0[\mathcal{B}] \oplus P^2[\mathcal{B}]|| \ldots ||P^0[\mathcal{B}] \oplus P^N[\mathcal{B}]$ in positions specified by $\mathcal{B}$ of the associated $\delta(\mathcal{A})$-set by encrypting the $\delta(\mathcal{A})$-set $\{P^0, P^1, \ldots, P^N\}$ by function $\mathcal{E}_r$, where $\mathcal{A} =$ $[\mathbf{S}_{r_0}[j_0], \mathbf{S}_{r_0}[j_1], \ldots, \mathbf{S}_{r_0}[j_s]]$ and $\mathcal{B} = [\mathbf{S}_{r_1}[i_0], \ldots, \mathbf{S}_{r_1}[i_t]]$ represent two sequences of positions.*

2.2 Basic DS-MITM attack

In this section, we present a brief overview of the previous DS-MITM attack. A cipher is usually split into three consecutive parts of r_0, r_1, and r_2 rounds, respectively. The DS-MITM attack consists of a precomputation phase and an online phase.

Precomputation phase. The precomputation phase is to construct a distinguisher on the second part of r_1 rounds. Constructing a distinguisher is to find a pair of $(\mathcal{A}, \mathcal{B})$ to construct a $\delta(\mathcal{A})$-set satisfying that the size of the space of the values that the output sequence $\Delta\mathcal{E}_{r_1}(\delta(\mathcal{A}))[\mathcal{B}]$ may take is less than that for a random sequence. For a reduced block cipher $\mathcal{E}_{r_1}$, $\Delta\mathcal{E}_{r_1}(\delta(\mathcal{A}))[\mathcal{B}]$ sequence is usually uniquely determined by several internal parameters. Then the size of the space of the values that $\Delta\mathcal{E}_{r_1}(\delta(\mathcal{A}))[\mathcal{B}]$ may take is portrayed by the size of all possible values space of these internal parameters. A lookup table will be built to save all possible values that $\Delta\mathcal{E}_{r_1}(\delta(\mathcal{A}))[\mathcal{B}]$ may take for all possible values of these internal parameters, which will be represented by $Tab_{\Delta\mathcal{E}_{r_1}(\delta(\mathcal{A}))[\mathcal{B}]}$ below.

A basic distinguisher on toy cipher is described by proposition 1. We will give concrete examples of the following concepts on a 3-round toy SPN block cipher with a 4-byte block size (Fig. 1). The round function of the toy block cipher consists of a Substitution layer $\mathbb{SB}$ (substitute each cell by a Sbox), a linear layer $\mathbb{L}$ (update state by left-multiplying a binary matrix $[[0,1,1,1],[1,0,1,1],[1,1,0,1],$ $[1,1,1,0]]$) and a key addition layer $\mathbb{AK}$ (update the state by XORing the round keys). To make the description clearer for the SPN block cipher, let $\mathbf{S}_i$ represent the input state of ith round and $\mathbf{S}_i^{\mathbb{SB}}$ be the output state of the Substitution layer below.

Proposition 1. *Let $\mathcal{A} = [\mathbf{S}_0[3]], \mathcal{B} = [\mathbf{S}_3[1]]$. Construct a $\delta(\mathcal{A})$-set $\{P^0, P^1, \ldots,$ $P^{255}\}$ satisfying that $P^0 \oplus P^i[\mathcal{A}] = i, i \in \{1, \ldots, 255\}$, the output difference sequence $\Delta\mathcal{E}_3(\delta(\mathcal{A}))[\mathcal{B}] = P^0 \oplus P^1[\mathbf{S}_3[1]]|| \ldots ||P^0 \oplus P^{255}[\mathbf{S}_3[1]]$ can be uniquely determined by 7 internal parameters:*

$$P^0[\mathbf{S}_0[3]], \{P^0[\mathbf{S}_1[j]] : j \in [0,1,2]\}, \{P^0[\mathbf{S}_2[j]] : j \in [0,2,3]\}.$$

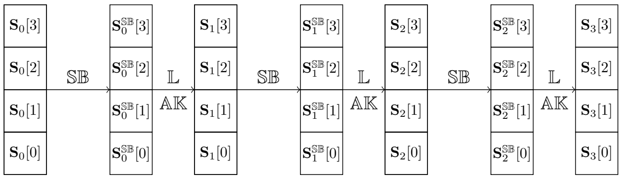


Fig. 1. A 3-round toy SPN block cipher

Proof. For each plaintext $P^i, i \in \{1, \dots, 255\}$, $P^0 \oplus P^i[\mathbf{S}_0[j]] = 0, \forall j \in [0, 1, 2]$ and $P^0 \oplus P^i[\mathbf{S}_0[3]] = i$ from $\delta(\mathcal{A})$-set definition. So only the difference in $\mathbf{S}_0[3]$ is non-zero. Thus with the knowledge of $P^0[\mathbf{S}_0[3]]$, $P^0 \oplus P^i[\mathbf{S}_0^{\mathbb{SB}}[3]]$ can be deduced, while $\forall j \in [0, 1, 2], P^0 \oplus P^i[\mathbf{S}_0^{\mathbb{SB}}[j]] = 0$. $\{P \oplus P^i[\mathbf{S}_1[j]] : \forall j \in [0, \dots, 3]\}$ can be deduced and $P^0 \oplus P^i[\mathbf{S}_1[3]] = 0$. Iterate this process, $\Delta\mathcal{E}_3(\delta(\mathcal{A}))[\mathcal{B}]$ can be uniquely determined by the above 7 internal parameters. Thus $\Delta\mathcal{E}_3(\delta(\mathcal{A}))[\mathcal{B}]$ can take at most $(2^8)^7$ possible values, while it has $(2^8)^{255}$ possibilities for a random 255-byte sequence. A distinguisher is constructed and a lookup table $Tab_{\Delta\mathcal{E}_{r_1}(\delta(\mathcal{A}))[\mathcal{B}]}$ is built to save all possible values of $\Delta\mathcal{E}_3(\delta(\mathcal{A}))[\mathcal{B}]$.

Online phase. The online phase is to guess round-keys involved in r_0 rounds to identify a $\delta(A)$-set for the distinguisher. Then guess round-keys involved in r_2 rounds to compute the value of $\Delta\mathcal{E}_{r_1}(\delta(\mathcal{A}))[\mathcal{B}]$ by partially decrypting the associated $\delta(A)$-set through r_2 rounds. Check whether the sequence in the lookup table $Tab_{\Delta\mathcal{E}_{r_1}(\delta(\mathcal{A}))[\mathcal{B}]}$, obtain the candidate of guessed round-keys involved in r_0, r_2 rounds that pass the test.

2.3 Techniques for Enhancing the DS-MITM Attack

Several improvement techniques are introduced to further reduce the time or memory complexity in the precomputation phase and online phase.

Differential Enumeration Technique. The main bottleneck technique is the differential enumeration technique introduced by Dunkelman et al. in Asiacrypt 2010 [13]. Try many pairs of messages to find one pair of (P, P') conforming to a truncated differential characteristic and construct a $\delta(\mathcal{A})$-set from P ($P \in \delta(\mathcal{A})$), which leads to a reduction of the possible values space of the internal parameters. In [8], Derbez et al. introduced the improved differential enumeration technique by finding that many values of the internal parameters are not reached if the $\delta(\mathcal{A})$-set constructed from a message conforming to a specified truncated differential characteristic.

Property 1 (Differential property of S-box). Given an input and output difference pair of $(\Delta_{in}, \Delta_{out})$ of an Sbox, the equation $\mathrm{Sbox}(x) \oplus \mathrm{Sbox}(x \oplus \Delta_{in}) = \Delta_{out}$ has one solution on average.

For example in proposition 1, assume (P^0, P') conforms to the truncated differential trail shown in Fig. 2 and $P^0 \in \delta(\mathcal{A})$-set. Then 6 parameters of $\{P^0[\mathbf{S}_0[3]]\} \cup$

$\{P^0[\mathbf{S}_1[j]], j \in [0,1,2]\} \cup \{P^0 \oplus P'[\mathbf{S}_0^{\mathbb{SB}}[3]]\} \cup \{P^0 \oplus P'[\mathbf{S}_3[1]]\}$ can determine the output sequence in proposition 1. Because three parameters of $P^0[\mathbf{S}_2[j]](j \in [0,2,3])$ can be deduced from $P^0 \oplus P'[\mathbf{S}_2[j]]$ and $P^0 \oplus P'[\mathbf{S}_2^{\mathbb{SB}}[j]])$ according to property 1, while it is obvious that $P^0 \oplus P'[\mathbf{S}_2[j]]$ and $P^0 \oplus P'[\mathbf{S}_2^{\mathbb{SB}}[j]]$ can be deduced from the above 6 internal parameters. Then $\Delta\mathcal{E}_3(\delta(\mathcal{A}))[\mathcal{B}]$ can take at most $(2^8)^6$ possible values, and the size of the precomputation table $Tab_{\Delta\mathcal{E}_3(\delta(\mathcal{A}))[\mathcal{B}]}$ is reduced by 1 byte.

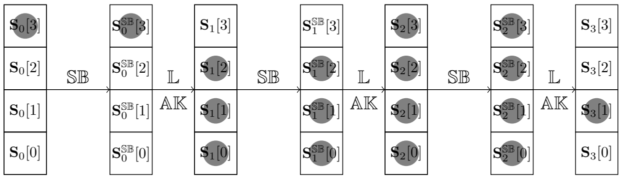


Fig. 2. A truncated differential trail on toy cipher

Key-dependent-sieve Technique. In [16], Li et al. introduced this technique to reduce the possibilities of the values that the internal parameters may reach, which is achieved by utilizing the relations on round keys deduced from these internal parameters.

Tweak-difference Cancellation Technique. In [18], the difference in tweak is utilized to cancel a difference in the state, called tweak-difference cancellation in this paper. Then differences of a $\delta(\mathcal{A})$-set at more cells will be zero, which leads to fewer internal parameters that determine the output sequence.

Key-bridging Technique. The technique utilizes the dependent relations on keys involved in the key-recovery phase to reduce the guessed keys space [13], which is a general method used in most key-recovery attacks.

2.4 Brief Description of SKINNY Block Cipher

SKINNY is a family of tweakable block cipher [3]. SKINNY-64 and SKINNY-128 have 64-bit and 128-bit block size respectively. In both versions, the states are arranged as 4 × 4-array, where the size of each cell is 4-bit in SKINNY-64 case and 8-bit in SKINNY-128 case. The input state of rth round is denoted by

$$\mathbf{S}_r = \begin{pmatrix} \mathbf{S}_r[0] & \mathbf{S}_r[1] & \mathbf{S}_r[2] & \mathbf{S}_r[3] \\ \mathbf{S}_r[4] & \mathbf{S}_r[5] & \mathbf{S}_r[6] & \mathbf{S}_r[7] \\ \mathbf{S}_r[8] & \mathbf{S}_r[9] & \mathbf{S}_r[10] & \mathbf{S}_r[11] \\ \mathbf{S}_r[12] & \mathbf{S}_r[13] & \mathbf{S}_r[14] & \mathbf{S}_r[15] \end{pmatrix}.$$

For each block size n, SKINNY-n can take three tweakey size $t = n, t = 2n$, and, $t = 3n$. SKINNY-n-t denotes the version with block size n and tweakey size t. t-bit tweakey can be arranged as t/n 4×4-array $TKz, z \in \{1, \ldots, t/n\}$. Each tweakey is first updated by a permutation PT ($TKz[j] \leftarrow TKz[PT[j]], z \in \{1, 2, 3\}$) for each round. Then every cell of the first and second rows of TK2 and TK3 is individually updated with an LFSR (The details of LFSR can be found in [3]).

$$PT = (9, 15, 8, 13, 10, 14, 12, 11, 0, 1, 2, 3, 4, 5, 6, 7).$$

The round function is composed of 5 operations: SubCells (SB), AddConstants (AC), AddRoundTweakey (AK), ShiftRows (SR) and MixColumns (MC). In the following, let $\mathbf{S}_r^{\mathbb{SB}}$, $\mathbf{S}_r^{\mathbb{AK}}$ and $\mathbf{S}_r^{\mathbb{SR}}$ denote the output state of SubCells, AddRoundTweakey and ShiftRows respectively (Fig. 10). The sum of updated t/n tweakey arrays is denoted by RK_r, which is the round-key of rth round.

SubCells is to substitute each cell by a 4-bit ($n = 64$) or 8-bit ($n = 128$) Sbox. AddConstants is to update the state by XORing constants, which is omitted because constants have no effect on this attack. AddRoundTweakey is to update the state by XORing the first two rows of state with t/n tweakey arrays, i.e. $\mathbf{S}_r^{\mathbb{AK}}[j] = \mathbf{S}_r^{\mathbb{SB}}[j] \oplus RK_r[j], 0 \leq j \leq 7$. ShiftRows is to rotate i-th row to the right by i cells. MixColumns is to multiply each column by the binary matrix MC:

$$MC = \begin{pmatrix} 1\,0\,1\,1 \\ 1\,0\,0\,0 \\ 0\,1\,1\,0 \\ 1\,0\,1\,0 \end{pmatrix}.$$

3 The Non-full Key-Addition Technique

A new general improvement technique, referred to as *non-full key-addition technique*, is introduced for block ciphers where partial states are updated by the round keys. The previous best DS-MITM attack on SKINNY-128-384 can directly be improved by one round by utilizing the technique alone (in Sect. G). When partial states are updated by round keys, states between two consecutive rounds are not totally independent. Many dependencies within internal parameters are ignored in previous attacks, which are effective for further reducing the space of the values that internal parameters may take.

Assume only the first two bytes are updated by XORing the round-keys RK_r in the key addition layer of the toy cipher. Then $\mathbf{S}_{r+1} = \mathbb{L}(\mathbf{S}_r^{\mathbb{SB}}) \oplus (RK_r[0], RK_r[1], 0, 0)$. Then variables in $\{\mathbf{S}_r[0], \mathbf{S}_r[1], \mathbf{S}_r[2], \mathbf{S}_r[3], \mathbf{S}_{r+1}[2], \mathbf{S}_{r+1}[3]\}$ are not independent and linked by Sbox and linear layer without round-key knowledge. For example, the degree of freedom of $\{\mathbf{S}_1[0], \mathbf{S}_1[1], \mathbf{S}_1[2], \mathbf{S}_2[2], \mathbf{S}_2[3]\}$ is 4. This dependency will lead to that 5 parameters of $\{P^0[\mathbf{S}_1[j]] : j \in [0, 1, 2]\} \cup \{P^0[\mathbf{S}_2[j]] : j \in [2, 3]\}$ in proposition 1 can take at most $(2^8)^4$ possible values. So the space of values that the output sequence may take is further reduced by 1 byte. This example can be seen as taking $g_1 = (1, 1, 1, 0, 1, 1)$ defined in the below property 2.

This article will describe the non-full key-addition technique in property 2 from a more general and comprehensive perspective. For simplicity, we will introduce this technique on a regular round function. Note that the technique for other round functions can be considered in a similar way. The constant addition is omitted in this description as it has no effect on the property.

Property 2. Assume $\mathbf{S}_{r+1} = \mathbb{L}(\mathbf{S}_r^{\mathbb{SB}}) \oplus (RK_r[0], \ldots, RK_r[s-1], 0, \ldots), \mathbf{S}_r^{\mathbb{SB}} = (\mathrm{Sbox}(\mathbf{S}_r[0]), \ldots, \mathrm{Sbox}(\mathbf{S}_r[n-1]))$, where s partial cells are updated by the round-key RK_r and $\mathbb{L}$ is the linear transformation matrix. Introduce a vector $g_r = (g_r[0], g_r[1], \ldots, g_r[2n-s-1]) \in \mathbb{F}_2^{2n-s}$ corresponding to $(\mathbf{S}_r[0], \ldots, \mathbf{S}_r[n-1], \mathbf{S}_{r+1}[s], \ldots,\ \mathbf{S}_{r+1}[n-1])$. For each possible value of g_r, compute the rank β_{g_r} of the matrix consisting of $\{\overrightarrow{\mathbf{e}_j} : g_r[j] = 1, j \in [0, \ldots, n-1]\}$ and $\{\mathbb{L}_j : g_r[n+j-s] = 1, j \in [s, \ldots, n-1]\}$, where $\overrightarrow{\mathbf{e}_j}$ is the n-dimensional unit vector with jth bit 1 and $\mathbb{L}_j$ is the jth row of the linear transformation matrix. For any plaintext P, parameters of $\{P[\mathbf{S}_r[j]] : g_r[j] = 1, j \in [0, \ldots, n-1]\} \cup \{P[\mathbf{S}_{r+1}[j]] : g_r[n+(j-s)] = 1, j \in [s, \ldots, n-1]\}$ can take at most $(2^c)^{\beta_{g_r}}$ possible values, where c is the size of each cell. We also say that the possible values space of these internal parameters is reduced by $\sum\limits_{j=0}^{2n-1-s} g_r[j] - \beta_{g_r}$ cells.

As each $\mathbf{S}_r^{\mathbb{SB}}[j]$ can be expressed by $\mathrm{Sbox}(\mathbf{S}_r[j])$. Thus the relations on $\mathbf{S}_r^{\mathbb{SB}}[j]$ can be converted to that on $\mathbf{S}_r[j]$ directly. Note that all possible values of $(g_r[0], \ldots, g_r[2n-1-s], \sum\limits_{j=0}^{2n-1-s} g_r[j] - \beta_{g_r})$ can be built directly from $\mathbb{L}$. This way of description makes the technique easy to be modeled in the full-fledged search framework in Section 4.5.

4 Full-fledged Framework with New Improvement Techniques

4.1 A High Level Overview

Before stating our new framework of modelling DS-MITM attack with four additional new techniques that have not been included in the basic model in Asiacrypt 2018 [20], we would like to give a high-level description of the unified framework which supports a full package of techniques. In particular, we highlight the variables that will be introduced for realizing these functions.

Basic DS-MITM distinguisher. Impose constraints over three types (typeX, typeY, typeZ) of 0-1 variables to describe the basic distinguisher [20].

Differential enumeration. Note that the automation of the differential enumeration technique is highly nontrivial and the key point is to synthesize internal parameters that will uniquely determine the output sequence from the combination of the basic model and truncated differential trail. It is enabled by introducing an important proposition (proposition 3). To modelling the differential enumeration technique, a new type, i.e., typeT, of 0-1 variables for each cell are first introduced to describe the traditional truncated differential trail. Two new types (typeGT, typeGZ) of 0-1 variables for each cell are introduced to synthesize the internal parameters that determine the output sequence, where typeGT variables describe the internal parameters whose values will be bounded by truncated differential trail and typeGZ variables describe the remaining internal parameters.

Key-dependent sieve. To modelling the key-dependent-sieve technique, the internal parameters that determine the output sequence described by typeGT and typeGZ variables are unified by a new type, i.e., typeV, of 0-1 variables, and a new type typeK of 0-1 variables are introduced to describe the round-keys deduced from these internal parameters.

Non-full key-addition. To modelling the non-full key-addition technique, introduce integer variables for each round to describe the reduced cells with typeV variables introduced for the key-dependent-sieve technique.

Tweak-difference cancellation. Note that the tweak values input to each round are known to the attackers, which can be treated as constants in the computation of the output difference. But we need to consider the injected tweak-difference by tweak addition operation when imposing constraints over typeX variables following the forward differential propagation rule. We will introduce typeX variables for each tweak cell and describe forward differential trail propagation for both tweak addition and tweak schedule.

Key-recovery phase. The methods for modelling the phase of deducing the guessed round-keys to construct $\delta(\mathcal{A})$-set and obtain $\Delta\mathcal{E}_{r_1}(\delta(\mathcal{A}))[\mathcal{B}]$ sequence by partially decrypting the associated $\delta(\mathcal{A})$-set can refer to Shi et al.'s work [20], which are achieved by introducing typeM variables involved in first r_0 rounds and typeW type variables involved in last r_2 rounds and impose constraints over typeM variables to form *a backward differential trail* and constraints over typeW variables to form *a forward determination trail.* To consider the differential enumeration in this paper, we also need to model the phase of obtaining a pair conforming to the truncated differential trail of the distinguisher. We will introduce new type, i.e., typeE, of 0-1 variables for each cell involved in r_0 and r_2 rounds, and impose constraints over typeE to form *a backward differential trail* through the first r_0 rounds and *a forward differential trail* through the last r_2 rounds. And typeE-$\mathbf{S}_{r_0}$ should be equal to typeT-$\mathbf{S}_{r_0}$, while typeE-$\mathbf{S}_{r_0+r_1}$ should be equal to typeT-$\mathbf{S}_{r_0+r_1}$.

4.2 Modelling the Basic DS-MITM Distinguisher

In [20], Shi et al. proposed a modelling method for the basic DS-MITM attack based on constraints programming (CP). In this section, we review and describe Shi et al.'s modelling method for finding DS-MITM distinguisher in a more unified way. We encourage the readers to go through this section since new terminologies are introduced and will be used to enhance the expressiveness of our framework.

As defined in [20], three types (typeX, typeY, typeZ) of 0-1 variables for each cell are introduced. Let typeX-$*$, typeY-$*$ and typeZ-$*$ denote the type variables in a cell or a state $*$ respectively below. Constraints over typeX variables follows the so-called *forward differential propagation rule.* Constraints over typeY variables follows so-called *backward determination propagation rule*. Assume the distinguisher is constructed on the second part of r_1 rounds $(r_0, r_0+1, \dots, r_0+r_1-1)$.

typeX Variables

- Generalized propagation rule for typeX variables is presented in definition 3, and typeX variables form a so-called *forward differential trail.*

  Definition 3 (forward differential trail). *Let* $\mathbf{S}_{i+1} = f(\mathbf{S}_i)$ *for* $0 \le i \le r-1$, *where* f *is the round function of an iterative block cipher and* $\mathbf{S}_i = (\mathbf{S}_i[0], \mathbf{S}_i[1], \dots, \mathbf{S}_i[n-1])$ *is the* n*-cell input state of the* i*th round. Introduce typeX variables for each cell: typeX-*$\mathbf{S}_i = ($*typeX-*$\mathbf{S}_i[0]$, *typeX-*$\mathbf{S}_i[1], \dots,$ *typeX-*$\mathbf{S}_i[n-1]) \in \{0,1\}^n, 0 \le i \le r$. *Define* $\overline{\mathcal{A}_i} = [\mathbf{S}_i[j] : typeX\text{-}\mathbf{S}_i[j] = 0, j \in [0, \dots, n-1]]$. *We call* $(typeX\text{-}\mathbf{S}_0 \xrightarrow{f} typeX\text{-}\mathbf{S}_1 \xrightarrow{f} \dots \xrightarrow{f} typeX\text{-}\mathbf{S}_r)$ *a valid forward differential trail if for each pair of* (P, P') *satisfying* $P \oplus P'[j] = 0, \forall j \in \overline{\mathcal{A}_i}$, *obtain*
  $$P \oplus P'[j] = 0, \forall j \in \overline{\mathcal{A}_{i+1}}.$$

- typeX variables are defined with the following implications.
  $$\text{typeX-}* = \begin{cases} 0(\square) \\ 1(▨) \end{cases}$$

- Informally, constraints over typeX variables follow the differential propagation rule with probability 1. And typeX-$\mathbf{S}_{i+1}[j] = 0$ indicates that $\mathbf{S}_{i+1}[j]$ is always a in-active cell (internal difference at $\mathbf{S}_{i+1}[j]$ is always 0) for any pair of (P, P') satisfying $P \oplus P'[\mathbf{S}_0[j]] = 0, \forall j \in \overline{\mathcal{A}_0}$. A valid forward differential trail on the toy cipher is shown in Fig. 3. In the figure, $\overline{\mathcal{A}_0} = [\mathbf{S}_0[j] : j \in [0,1,2]], \overline{\mathcal{A}_1} = [\mathbf{S}_1[3]]$, and typeX-$\mathbf{S}_0 = (0,0,0,1) \longrightarrow$ typeX-$\mathbf{S}_1 = (1,1,1,0)$ is valid. Because output difference $P \oplus P'[\mathbf{S}_1[3]]$, for any pair of (P, P') inactive at positions specified by $[\mathbf{S}_0[j] : j \in [0,1,2]]$, is always 0. Imposed constraints over typeX variables following this propagation rule through all specific operations (S-box, $MC, \dots$) please refer to [20].

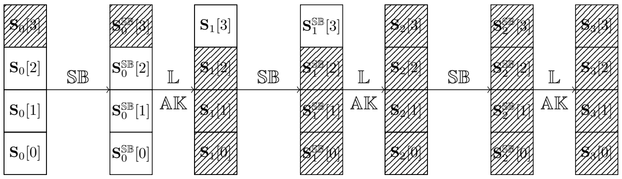


Fig. 3. A valid forward differential trail on toy cipher

An opposite direction *backward differential trail* is also presented in definition 4 and a backward differential trail on toy cipher is shown in Fig. 4.

Definition 4 (backward differential trail). *Let* $\mathbf{S}_{i+1} = f(\mathbf{S}_i)$ *for* $0 \leq i \leq r-1$*, where* f *is the round function of an iterative block cipher and* $\mathbf{S}_i = (\mathbf{S}_i[0], \mathbf{S}_i[1], \ldots, \mathbf{S}_i[n-1])$ *is the* n*-cell input state of* i*th round. Introduce typeX variables for each cell:* $typeX\text{-}\mathbf{S}_i = (typeX\text{-}\mathbf{S}_i[0], typeX\text{-}\mathbf{S}_i[1], \ldots, typeX\text{-}\mathbf{S}_i[n-1]) \in \{0,1\}^n, 0 \leq i \leq r$*. Define* $\overline{\mathcal{A}_i} = \{\mathbf{S}_i[j] : typeX\text{-}\mathbf{S}_i[j] = 0, j \in [0, \ldots, n-1]\}$*. We call* $(typeX\text{-}\mathbf{S}_0 \xleftarrow{f^{-1}} typeX\text{-}\mathbf{S}_1 \xleftarrow{f^{-1}} \ldots \xleftarrow{f^{-1}} typeX\text{-}\mathbf{S}_r)$ *a valid backward differential trail if for any pair of* (P, P') *satisfying* $P \oplus P'[j] = 0, \forall j \in \overline{\mathcal{A}_{i+1}}$*, obtain*

$$P \oplus P'[j] = 0, \forall j \in \overline{\mathcal{A}_i}.$$

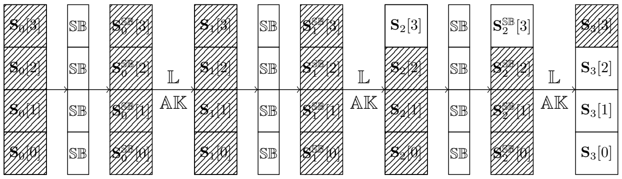


Fig. 4. A valid backward differential trail on toy cipher

typeY Variables

- Generalized propagation rule for typeY variables is proposed in definition 5, and typeY variables form the so-called *backward determination trail.*

  Definition 5 (backward determination trail). *Let* $\mathbf{S}_{i+1} = f(\mathbf{S}_i)$ *for* $0 \leq i \leq r-1$*, where* f *is the round function of an iterative block cipher and* $\mathbf{S}_i = (\mathbf{S}_i[0], \mathbf{S}_i[1], \ldots, \mathbf{S}_i[n-1])$ *is the* n*-cell input state of* i*th round. Introduce typeY variables for each cell:* $typeY\text{-}\mathbf{S}_i = (typeY\text{-}\mathbf{S}_i[0], \ldots, typeY\text{-}\mathbf{S}_i[n-1]) \in \{0,1\}^n, 0 \leq i \leq r$*. Define* $\mathcal{B}_i = \{\mathbf{S}_i[j] : typeY\text{-}\mathbf{S}_i[j] = 1, j \in [0, \ldots, n-1]\}$*. We call* $(typeY\text{-}\mathbf{S}_0 \xrightarrow{f} typeY\text{-}\mathbf{S}_1 \xrightarrow{f} \ldots \xrightarrow{f} typeY\text{-}\mathbf{S}_r)$ *a valid backward determination trail if for any pair of* (P, P')*, each difference among*

  $$\{P \oplus P'[j] : j \in \mathcal{B}_{i+1}\}$$

  can be uniquely determined by

  $$\{P \oplus P'[j], P[j] : j \in \mathcal{B}_i\}.$$

- This variable is defined with the following implications.

  $$\text{typeY-}* = \begin{cases} 0(\square) \\ 1(▨) \end{cases}$$

- Informally, typeY-$\mathbf{S}_i[j] = 0$ indicates that difference in each cell in $\mathcal{B}_{i+1}$ is independent of the knowledge of $\mathbf{S}_i[j]$. A valid backward determination trail on the toy cipher is shown in Fig. 5. In the figure, $\mathcal{B}_2 = [\mathbf{S}_2[j] : j \in [0,2,3]], \mathcal{B}_3 = [\mathbf{S}_3[1]]$, and typeY-$\mathbf{S}_2 = (1,0,1,1) \longrightarrow$ typeY-$\mathbf{S}_3 = (0,1,0,0)$ is valid, because $P \oplus P'[\mathbf{S}_3[1]]$ can be uniquely determined by $\{P \oplus P'[\mathbf{S}_2[j]], P[\mathbf{S}_2[j]] : j \in [0,2,3]\}$. Imposed constraints over typeY variables following the propagation rule through all specific operations please refer to [20].

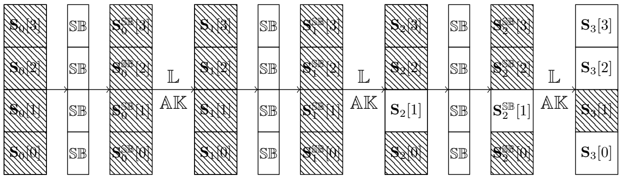

$\mathbf{S}_0[3]$ $\mathbf{S}_0[2]$ $\mathbf{S}_0[1]$ $\mathbf{S}_0[0]$
SB SB SB SB
$\mathbf{S}_0^{\mathbb{SB}}[3]$ $\mathbf{S}_0^{\mathbb{SB}}[2]$ $\mathbf{S}_0^{\mathbb{SB}}[1]$ $\mathbf{S}_0^{\mathbb{SB}}[0]$
$\mathbb{L}$
AK
$\mathbf{S}_1[3]$ $\mathbf{S}_1[2]$ $\mathbf{S}_1[1]$ $\mathbf{S}_1[0]$
SB SB SB SB
$\mathbf{S}_1^{\mathbb{SB}}[3]$ $\mathbf{S}_1^{\mathbb{SB}}[2]$ $\mathbf{S}_1^{\mathbb{SB}}[1]$ $\mathbf{S}_1^{\mathbb{SB}}[0]$
$\mathbb{L}$
AK
$\mathbf{S}_2[3]$ $\mathbf{S}_2[2]$ $\mathbf{S}_2[1]$ $\mathbf{S}_2[0]$
SB SB SB SB
$\mathbf{S}_2^{\mathbb{SB}}[3]$ $\mathbf{S}_2^{\mathbb{SB}}[2]$ $\mathbf{S}_2^{\mathbb{SB}}[1]$ $\mathbf{S}_2^{\mathbb{SB}}[0]$
$\mathbb{L}$
AK
$\mathbf{S}_3[3]$ $\mathbf{S}_3[2]$ $\mathbf{S}_3[1]$ $\mathbf{S}_3[0]$

Fig. 5. A valid backward determination on toy cipher

An opposite direction *forward determination trail* is also defined in [20].

Remark 1. In the backward determination trail definition, the $\{P[\mathbf{S}_i[j]], j \in \mathcal{B}_i\}$ can be omitted in case of f is a linear operation.

typeZ Variables

- This variable is imposed for each cell satisfying the rule that typeZ-$*$ (▩) equals 1 if and only if typeX-* = 1 (▨) and typeY-* = 1 (▧).

Objective Function

Proposition 2 ([20]). *Assume (typeX-$\mathbf{S}_{r_0} \xrightarrow{f}$ typeX-$\mathbf{S}_{r_0+1} \xrightarrow{f} \ldots \xrightarrow{f}$ typeX-$\mathbf{S}_{r_0+r_1}$) is a forward differential trail and (typeY-$\mathbf{S}_{r_0} \xrightarrow{f}$ typeY-$\mathbf{S}_{r_0+1} \xrightarrow{f} \ldots \xrightarrow{f}$ typeY-$\mathbf{S}_{r_0+r_1}$) is a backward determination trail. Impose constraints over (typeX-$\mathbf{S}_i[j]$, typeY-$\mathbf{S}_i[j]$,typeZ-$\mathbf{S}_i[j]$) for each cell following the rule that typeZ-$\mathbf{S}_i[j] = 1$ if and only if typeX-$\mathbf{S}_i[j] = 1$ and typeY-$\mathbf{S}_i[j] = 1$. Let $\mathcal{A} = \mathcal{A}_{r_0} = [\mathbf{S}_{r_0}[j]$: typeX-$\mathbf{S}_{r_0}[j] = 1, j \in [0,\ldots,n]], \mathcal{B} = \mathcal{B}_{r_0+r_1} = [\mathbf{S}_{r_0+r_1}[j]$: typeY-$\mathbf{S}_{r_0+r_1}[j] = 1, j \in [0,\ldots,n]]$. For any constructed $\delta(\mathcal{A})$-set $\{P^0, P^1, \ldots, P^{N-1}\}$, the output difference sequence $\Delta\mathcal{E}_{r_1}(\delta(\mathcal{A}))[\mathcal{B}]$ can be uniquely determined by the following internal parameters:*

$$\{P^0[\mathbf{S}_i[j]] : \textit{typeZ-}\mathbf{S}_i[j] = 1, r_0 \le i \le r_0 + r_1 - 1, j \in [0,\ldots,n]\}.$$

In the basic model, the objective function can be obtained from proposition 2. The example in proposition 1 can be obtained directly from this proposition 2, which is also illustrated in Fig. 6. The lookup table $Tab_{\Delta\mathcal{E}_r(\delta(\mathcal{A}))[\mathcal{B}]}$ will be built to save all values of $\Delta\mathcal{E}_{r_1}(\delta(\mathcal{A}))[\mathcal{B}]$ for all possible values of $\{P^0[\mathbf{S}_i[j]]$: typeZ-$\mathbf{S}_i[j] = 1, r_0 \le i \le r_0 + r_1 - 1, j \in [0,\ldots,n]\}$. And the smaller the size of

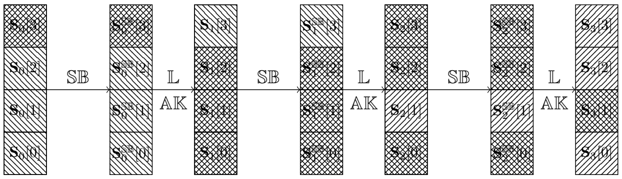


Fig. 6. A valid typeZ trail on toy cipher

the table is, the better the distinguisher is. Thus in the basic model, the objective function of the distinguisher is constrained to Minimize $(\sum_{i=r_0}^{r_0+r_1-1}\sum_{j=0}^{n-1}\text{typeZ-}\mathbf{S}_i[j])$.

Remark 2. Sometimes multiset is considered instead of the output difference sequence, i.e. an unordered set with multiplicity. The model for searching the ordered sequence and unordered multiset are almost the same. For simplicity, the objective function is defined for the ordered sequence below, while the experiments for SKINNY are both done by considering the ordered sequence and unordered multiset.

4.3 Modelling the Differential Enumeration Technique

The basic idea of the differential enumeration technique is to try many pairs of messages to find one pair of (P, P') conforming to a specified truncated differential characteristic and construct a $\delta(\mathcal{A})$-set from P. The space of the values that the output sequence may take is reduced because of fewer internal parameters that determine the sequence.

It is highly nontrivial to synthesize objective function, which is enabled by an important proposition 3. Three types (typeT, typeGT, typeGZ) of 0-1 variables for each cell are introduced. Constraints over typeT variables will follow a valid truncated differential propagation rule. In order to automatically synthesize the internal parameters from the combination of the basic distinguisher and truncated differential trail. Two new types (typeGT, typeGZ) of 0-1 variables for each cell are introduced to describe the internal parameters that determine the output sequence. And the parameters whose values are bounded by the truncated differential trail are represented by typeGT variables while the remaining internal parameters are described by typeGZ variables.

typeT Variables

- Constraints over typeT variables follow the traditional valid truncated differential propagation rule in the encryption direction. typeT variable is defined with the following implications.

$$\text{typeT-}* = \begin{cases} 0(\square) : \text{if the cell is in-active in the truncated differential trail,} \\ 1(\boxdot) : \text{if the cell is active in the truncated differential trail.} \end{cases}$$

- (typeT-$\mathbf{S}_0 \xrightarrow{f}$ typeT-$\mathbf{S}_1 \xrightarrow{f} \ldots \xrightarrow{f}$ typeT-$\mathbf{S}_r$) represents a valid truncated differential trail through the encryption round function f. A valid truncated differential trail on the toy cipher is shown in Fig. 7.

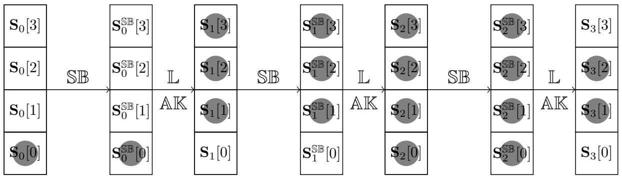


Fig. 7. A truncated differential trail on the toy cipher

typeGT Variables

- Generalized propagation rule for typeGT variables is proposed in definition 6 and typeGT variables form a so-called *typeT-based backward determination trail.*

  Definition 6 (typeT-based backward determination trail). *Let* $\mathbf{S}_{i+1} = f(\mathbf{S}_i)$ *for* $0 \leq i \leq r-1$, *where* f *is the round function of an iterative block cipher,* $\mathbf{S}_i = (\mathbf{S}_i[0], \mathbf{S}_i[1], \ldots, \mathbf{S}_i[n-1])$ *is the* n*-cell input state of* i*th round, and (typeT-*$\mathbf{S}_0 \xrightarrow{f}$ *typeT-*$\mathbf{S}_1 \xrightarrow{f} \ldots \xrightarrow{f}$ *typeT-*$\mathbf{S}_r$*) is a valid truncated differential trail. Introduce typeGT variables for each cell: typeGT-*$\mathbf{S}_i =$ *(typeGT-*$\mathbf{S}_i[0], \ldots,$ *typeGT-*$\mathbf{S}_i[n-1]) \in \{0,1\}^n, 0 \leq i \leq r$. *Define* $\mathcal{G}_i = [\mathbf{S}_i[j] :$ *typeGT-*$\mathbf{S}_i[j] = 1, j \in [0, \ldots, n-1]]$. *We call (typeGT-*$\mathbf{S}_0 \xrightarrow{f}$ *typeGT-*$\mathbf{S}_1 \xrightarrow{f} \ldots \xrightarrow{f}$ *typeGT-*$\mathbf{S}_r$*) a typeT-based backward determination trail if for each pair of* (P, P') *conforming to the truncated differential trail defined by typeT variables, obtain each difference in*

  $$\{P \oplus P'[j] : j \in \mathcal{G}_{i+1}\}$$

  can be uniquely determined by

  $$\{P \oplus P'[j], P[j] : j \in \mathcal{G}_i\}.$$

- typeGT variable for each cell is defined with the following implications.

$$\text{typeGT-}* = \begin{cases} 0(\square) \\ 1(\blacksquare) \end{cases} \tag{1}$$

- Informally, typeGT-$\mathbf{S}_i[j] = 0$ indicates whether the difference in each cell in $\mathcal{G}_{i+1}$ is independent of knowledge of $\mathbf{S}_i[j]$ or $\mathbf{S}_i[j]$ is in-active in the truncated differential trail (typeT-$\mathbf{S}_i[j] = 0$), which is different from *backward determination definition 5*. A valid typeT-based backward determination trail on the toy cipher is shown in Fig. 8. In the figure, $\mathcal{G}_0 = [\mathbf{S}_0[0]], \mathcal{G}_1 = [\mathbf{S}_1[j] : j \in [1,2,3]]$, and typeGT-$\mathbf{S}_0 = (1,0,0,0) \rightarrow$ typeGT-$\mathbf{S}_1 = (0,1,1,1)$ is valid. $\{P \oplus P'[\mathbf{S}_1[j]] : j \in [1,2,3]\}$ can be uniquely determined by $\{P \oplus P'[\mathbf{S}_0[0]], P[\mathbf{S}_0[0]]\}$ because $\forall j \in [1,2,3], P \oplus P'[\mathbf{S}_0[j]] = 0$ if (P, P') conforms to the truncated differential trail defined by ▣. From the comparison between Fig. 5 and Fig. 8, *typeT-based backward determination trail* is different from the previous *backward determination trail*. This will lead to a reduction of the internal parameters that determine the output sequence by combining the truncated differential with the basic DS-MITM distinguisher.

- Constraints over typeGT variables following the propagation rule through all operations can be imposed in two steps. Firstly, introduce dummy 0-1 variables for each cell: Dm-$\mathbf{S}_i = ($Dm-$\mathbf{S}_i[0], \ldots,$ Dm-$\mathbf{S}_i[n-1]) \in \{0,1\}^n$, and impose constraints over (Dm-$\mathbf{S}_i$, typeGT-$\mathbf{S}_{i+1}$) following backward determination propagation rule (definition 5). Secondly, impose constraints over (Dm-$\mathbf{S}_i[j]$, typeT-$\mathbf{S}_i[j]$, typeGT-$\mathbf{S}_i[j]$) following the rule that typeGT-$\mathbf{S}_i[j] = 1$ if and only if Dm-$\mathbf{S}_i[j] =$ typeT-$\mathbf{S}_i[j] = 1$, which can be easily generated by using convex hull computation method.

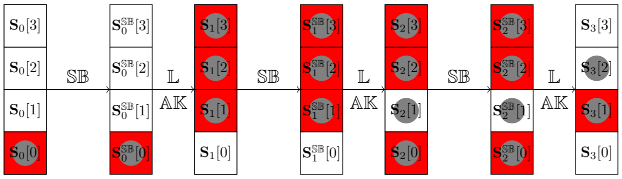


Fig. 8. A typeT-based backward determination trail on toy cipher

A *typeT-based forward determination trail* is also presented in definition 7.

Definition 7 (typeT-based forward determination trail). *Let* $\mathbf{S}_{i+1} = f(\mathbf{S}_i)$ *for* $0 \leq i \leq r-1$*, where* f *is the round function of an iterative block cipher,* $\mathbf{S}_i = (\mathbf{S}_i[0], \mathbf{S}_i[1], \ldots, \mathbf{S}_i[n-1])$ *is the* n*-cell input state of* i*th round, and* (*typeT*-$\mathbf{S}_0 \rightarrow$ *typeT*-$\mathbf{S}_1 \rightarrow \cdots \rightarrow$ *typeT*-$\mathbf{S}_r$) *is a valid truncated differential trail. Introduce typeGT variables for each cell: typeGT*-$\mathbf{S}_i = ($*typeGT*-$\mathbf{S}_i[0], \ldots,$ *typeGT*-$\mathbf{S}_i[n-1]) \in \{0,1\}^n, 0 \leq i \leq r$*. Define* $\mathcal{G}_i = \{\mathbf{S}_i[j] :$ *typeGT*-$\mathbf{S}_i[j] = 1, j \in [0, \ldots, n-1]\}$*. We call* (*typeGT*-$\mathbf{S}_0 \xleftarrow{f^{-1}}$ *typeGT*-$\mathbf{S}_1 \xleftarrow{f^{-1}} \ldots \xleftarrow{f^{-1}}$ *typeY*-$\mathbf{S}_r$) *a typeT-based forward determination differential trail if for each pair of* (P, P') *conforming to the truncated differential propagation trail defined by typeT variables, obtain each difference in*

$$\{P \oplus P'[j] : j \in \mathcal{G}_i\}$$

can be uniquely determined by

$$\{P \oplus P'[j] : j \in \mathcal{G}_{i+1}\}, \{P[j] : j \in \mathcal{G}_i\}.$$

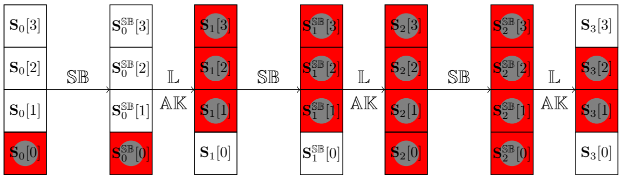


Fig. 9. A typeT-based forward determination trail on toy cipher

typeGZ Variables

- This variable is imposed following the rule of Eq. (2).

$$\text{typeGZ-}* = \begin{cases} 1(\text{🟩}) : \text{if typeZ-}* = 1 \text{ and typeGT-}* = 0, \\ 0(\square) : \text{otherwise.} \end{cases} \tag{2}$$

- These variables are utilized to consider the remaining internal parameters that determine the output difference except those covered by typeGT variables (typeGT-* $= 1$). Constraints over these variables following the rule can be easily generated by the convex hull computation method.

Objective Function Based on Differential Enumeration Technique. An important proposition 3 is proposed based on the six types variables (typeX, typeY, typeZ, typeT, typeGT, typeGZ). Then this proposition is applied to automatically synthesize the objective function for the distinguisher based on the differential enumeration technique.

The following description is based on the assumption that Sbox has differential property 1, which is usually true. And the differential property 1 will be utilized where *typeT-backward determination trail* and *typeT-based forward determination trail* meet. Denote R_M be the round where two trails meet. Let $\mathbf{S}_{R_M}$ and $\mathbf{S}_{R_M}^{\mathbb{SB}}$ represent the input and output state of the $\mathbb{SB}$-layer of round R_M, respectively. To combine the basic DS-MITM distinguisher and the differential enumeration technique, typeGT variables in R_M should be initialized by typeZ and typeT variables as shown in Eq. (3) and Eq. (4) for two reasons. Firstly, the differential property 1 can be utilized in the active Sbox of the truncated differential trail. Secondly, we only care about internal parameters that determine the output difference (typeZ-* $= 1$). For a complete search of r_1 rounds distinguisher, all possible R_M should be tried. The following description of the proposition is for an individual model with a fixed (r_1, R_M).

$$\text{typeGT-}\mathbf{S}_{R_M}[j] = \begin{cases} 1, \text{ if typeT-}\mathbf{S}_{R_M}[j] = 1 \text{ and typeZ-}\mathbf{S}_{R_M}[j] = 1, \\ 0, \text{ otherwise.} \end{cases} \tag{3}$$

$$\text{typeGT-}\mathbf{S}_{R_M}^{\mathbb{SB}}[j] = \begin{cases} 1, \text{ if typeT-}\mathbf{S}_{R_M}^{\mathbb{SB}}[j] = 1 \text{ and typeZ-}\mathbf{S}_{R_M}^{\mathbb{SB}}[j] = 1, \\ 0, \text{ otherwise.} \end{cases} \tag{4}$$

The constraints over (typeGT-$\mathbf{S}_{R_M}[j]$, typeT-$\mathbf{S}_{R_M}[j]$, typeZ-$\mathbf{S}_{R_M}[j]$) following the rules can be easily generated by convex hull computation method.

Proposition 3 (New Objective Function). *Impose constraints over three types (typeX,typeY,typeZ) of 0-1 variables on r_1 rounds $(r_0, r_0+1, \ldots, r_0+r_1-1)$. typeX variables and typeY variables form a forward differential trail and a backward determination trail respectively. Impose constraints over (typeX-$\mathbf{S}_i[j]$, typeY-$\mathbf{S}_i[j]$,typeZ-$\mathbf{S}_i[j]$) for each cell following the rule that typeZ-$\mathbf{S}_i[j] = 1$ if and only if typeX-$\mathbf{S}_i[j]$ = 1 and typeY-$\mathbf{S}_i[j]$ = 1. (typeGT-$\mathbf{S}_{r_0} \xrightarrow{\mathcal{E}_1}$ typeGT-$\mathbf{S}_{r_0+1} \xrightarrow{\mathcal{E}_1} \ldots \xrightarrow{\mathcal{E}_1}$ typeGT-$\mathbf{S}_{R_M}$) and (typeGT-$\mathbf{S}^{\mathbb{SB}}_{R_M} \xleftarrow{\mathcal{E}_1-1}$ typeGT-$\mathbf{S}_{R_M+1} \xleftarrow{\mathcal{E}_1^{-1}} \ldots \xleftarrow{\mathcal{E}_1^{-1}}$ typeGT-$\mathbf{S}_{r_0+r_1}$) form a typeT-based backward determination trail (definition 6) and a typeT-based forward determination trail (definition.7) respectively. typeGT-$\mathbf{S}_{R_M}$ and typeGT- $\mathbf{S}^{\mathbb{SB}}_{R_M}$ are initialized by Eq. (3) and Eq. (4). Define $\mathcal{A} = [\mathbf{S}_{r_0}[j] :$ typeX-$\mathbf{S}_{r_0}[j] = 1, j \in [0, \ldots, n-1]], \mathcal{B} = [\mathbf{S}_{r_0+r_1}[j] :$ typeY-$\mathbf{S}_{r_0+r_1}[j] = 1, j \in [0, \ldots, n-1]]$. Assume (P^0, P') conforms to the truncated differential trail defined by typeT variables. For any $\delta(\mathcal{A})$-set $\{P^0, P^1, \ldots, P^{N-1}\}$ constructed from the message P^0 ($P^0 \in \delta(\mathcal{A})$), the output difference sequence $\Delta\mathcal{E}_{r_1}(\delta(\mathcal{A}))[\mathcal{B}]$ can be uniquely determined by the following internal parameters:*

$$\begin{aligned}
&\{P^0 \oplus P'[\mathbf{S}_{r_0}[j]] : \textit{typeGT-}\mathbf{S}_{r_0}[j] = 1, j \in [0, \ldots, n-1]\},\\
&\{P^0 \oplus P'[\mathbf{S}_{r_0+r_1}[j]] : \textit{typeGT-}\mathbf{S}_{r_0+r_1}[j] = 1, j \in [0, \ldots, n-1]\},\\
&\{P^0[\mathbf{S}_i[j]] : \textit{typeGT-}\mathbf{S}_i[j] = 1, r_0 \le i \le r_0+r_1-1, i \ne R_M, j \in [0, \ldots, n-1]\},\\
&\{P^0[\mathbf{S}_i[j]] : \textit{typeGZ-}\mathbf{S}_i[j] = 1, r_0 \le i \le r_0+r_1-1, j \in [0, \ldots, n-1]\}.
\end{aligned} \tag{5}$$

Proof. According to proposition 2, the output sequence can be uniquely determined by $\{P^0[\mathbf{S}_r[j]] :$ typeZ-$\mathbf{S}_r[j] = 1, r_0 \le r \le r_0+r_1-1, j \in [0, \ldots, n-1]\}$, in which all except $\{P^0[\mathbf{S}_{R_M}[j]] :$ typeZ-$\mathbf{S}_{R_M}[j] = 1, j \in [0, \ldots, n-1]\}$ have been included by $\{P^0[\mathbf{S}_r[j]] :$ typeGT-$\mathbf{S}_r[j] = 1$ or typeGZ-$\mathbf{S}_r[j] = 1, r \ne R_M, j \in [0, \ldots, n-1]\}$ from the definition of typeGZ variables shown in Eq. (2). If we can prove that $\{P^0[\mathbf{S}_{R_M}[j]] :$ typeZ-$\mathbf{S}_{R_M}[j] = 1, j \in [0, \ldots, n-1]\}$ can be uniquely determined by above internal parameters, the proof is complete.

Firstly, if (P^0, P') conforms to the truncated differential described by typeT variables and (typeGT-$\mathbf{S}_{r-1} \xrightarrow{\mathcal{E}_1}$ typeGT-$\mathbf{S}_r$) forms a typeT-based backward determination trail, then $\{P^0 \oplus P'[\mathbf{S}_r[j]] :$ typeGT-$\mathbf{S}_r[j] = 1, j \in [0, \ldots, n-1]\}$ can be uniquely determined by $\{P^0 \oplus P'[\mathbf{S}_{r-1}[j]] :$ typeGT-$\mathbf{S}_{r-1}[j] = 1, j \in [0, \ldots, n-1]\}$ and $\{P^0[\mathbf{S}_{r-1}[j]] :$ typeGT-$\mathbf{S}_{r-1}[j] = 1, j \in [0, \ldots, n-1]\}$ (definition 6). Thus iterate the process on r to obtain that $\{P^0 \oplus P'[\mathbf{S}_{R_M}[j]] :$ typeGT-$\mathbf{S}_{R_M}[j] = 1, j \in [0, \ldots, n-1]\}$ can be uniquely determined by $\{P^0[\mathbf{S}_r[j]] :$ typeGT-$\mathbf{S}_r[j] = 1, r_0 \le r \le R_M-1, j \in [0, \ldots, n-1]\} \cup \{P^0 \oplus P'[\mathbf{S}_{r_0}[j]] :$ typeGT-$\mathbf{S}_{r_0}[j] = 1, j \in [0, \ldots, n-1]\}$.

Secondly, if (P^0, P') conforms to the truncated differential trail described by typeT variables and (typeGT-$\mathbf{S}_r \xleftarrow{\mathcal{E}_1^{-1}}$ typeGT-$\mathbf{S}_{r+1}$) forms a typeT-based forward determination trail (definition 7). Then $\{P^0 \oplus P'[\mathbf{S}_r] :$ typeGT-$\mathbf{S}_r[j] = 1, j \in [0, \ldots, n-1]\}$ can be uniquely determined by $\{P^0 \oplus P'[\mathbf{S}_{r+1}[j]] :$ typeGT-$\mathbf{S}_{r+1}[j] =$

$1, j \in [0, \ldots, n-1]\}$ and $\{P^0[\mathbf{S}_r[j]] : \text{typeGT-}\ \mathbf{S}_r[j] = 1, j \in [0, \ldots, n-1]\}$. Iterate this process on r to obtain that $\{P^0 \oplus P'[\mathbf{S}^{\mathbb{SB}}_{R_M}[j]] : \text{typeGT-}\mathbf{S}^{\mathbb{SB}}_{R_M}[j] = 1, j \in [0, \ldots, n-1]\}$ can be uniquely determined by $\{P^0[\mathbf{S}_r[j]] : \text{typeGT-}\mathbf{S}_r[j] = 1, R_M + 1 \leq r \leq r_0 + r_1 - 1, j \in [0, \ldots, n-1]\} \cup \{P^0 \oplus P'[\mathbf{S}_{r_0+r_1}\ [j]] : \text{typeGT-}\mathbf{S}_{r_0+r_1}[j] = 1, j \in [0, \ldots, n-1]\}$.

Apply differential property 1 on $(P^0 \oplus P'[\mathbf{S}_{R_M}[j]], P^0 \oplus P'[\mathbf{S}^{\mathbb{SB}}_{R_M}[j]])$ to deduce $\{P^0[\mathbf{S}_{R_M[j]}] : \text{typeGT-}\mathbf{S}_{R_M}[j] = 1, j \in [0, \ldots, n-1]\}$. Thus $\{P^0[\mathbf{S}_{R_M}[j]] : \text{typeZ-}\mathbf{S}_{R_M}[j] = 1, j \in [0, \ldots, n-1]\}$ are uniquely determined with the remaining parameters of $\{P^0[\mathbf{S}_{R_M}[j]] : \text{typeGZ-}\mathbf{S}_{R_M}[j] = 1, j \in [0, \ldots, n-1]\}$.

The objective function is constrained to Minimize OBJ, where

$$\begin{aligned} OBJ = &\sum_{i=r_0}^{i=r_0+r_1-1} \sum_{j=0}^{j=n-1} \text{typeGZ-}S_i[j] + \sum_{i=r_0, i \neq R_M}^{i=r_0+r_1-1} \sum_{j=0}^{j=n-1} \text{typeGT-}S_i[j] \\ &+ \sum_{i \in \{r_0, r_0+r_1\}} \sum_{j=0}^{j=n-1} \text{typeGT-}S_i[j]. \end{aligned} \tag{6}$$

Remark 3. The above objective function is a unified expression for the generalized model with the differential enumeration technique. The actual objective function OBJ_{Dis} of the distinguisher is OBJ minus the reduced space by the key-dependent-sieve or non-full key-addition techniques et al.. Details of the attack phase on SKINNY will be given in proposition 4 as an example of the proof, which can be automatically deduced by proposition 3.

4.4 Modelling Key-Dependent-Sieve Technique

Some round keys can be deduced from the internal parameters that determine the output difference sequence. The key-dependent-sieve technique is to utilize the dependent relations on these round keys to reduce the possible values space of the internal parameters, which is an important technique and has not been included in the previous automatic search model in [20].

One type typeV of 0-1 variable for each state cell and one type typeK of 0-1 variable for each round-key cell will be introduced to describe whether the round-key cell can be deduced from the internal parameters listed in Eq. (5) that determine the output difference sequence. We will give a description of modelling the technique on a regular round function. Assume $\mathbf{S}_{r+1} = \mathbb{L}(\mathbf{S}^{\mathbb{SB}}_r) \oplus RK_r$, where $\mathbb{L}$ is a linear transformation matrix and RK_r is the round-key.

typeV Variables

– This variable is imposed following the rule listed in Eq. (7).

$$\text{typeV-}* = \begin{cases} 1 : \text{if typeGZ-}* = 1 \text{ or typeGT-}* = 1, \\ 0 : \text{otherwise.} \end{cases} \tag{7}$$

According to proposition 3, the internal parameter satisfying typeGZ-$*=1$ or typeGT-$*=1$ is needed to determine the output sequence, which is unified by typeV-$*=1$.

typeK Variables

- Describe $RK_r[j]$ as $\mathbf{S}_{r+1}[j] \oplus \mathbb{L}(\mathbf{S}_r^{\mathbb{SB}}[j_0], \mathbf{S}_r^{\mathbb{SB}}[j_1], \ldots, \mathbf{S}_r^{\mathbb{SB}}[j_s]])$. A new type typeK of 0-1 variables are introduced for each round-key cell following the rule listed in Eq. (8).

$$\text{typeK-}RK_r[j] = \begin{cases} 1(\blacksquare) : \text{if typeV-}\mathbf{S}_{r+1}[j] = 1, \text{typeV-}\mathbf{S}_r^{\mathbb{SB}}[j_i] = 1, \forall i, \\ 0(\square) : \text{otherwise.} \end{cases} \tag{8}$$

The possible values space of the internal parameters can be reduced by the number of relations on these deduced round-key cells satisfying typeK-$RK_r[j] = 1$. And the various relations for specified cipher can be included in the model dynamically.

Model of Key-Dependent-Sieve for SKINNY. For SKINNY described in Section 2.4, each round-key cell $RK_r[j]$ is related to only one position of each master tweakey array TKz. $RK_r[j]$ can be uniquely determined by $\{TKz[PT^r[j]] : z \in \{1, \ldots, t/n\}\}$. PT^r (PT^{-r} respectively) represents the composite permutation of $PT \circ \cdots \circ PT$ ($PT^{-1} \circ \cdots \circ PT^{-1}$ respectively). For each $j \in \{0, 1, \ldots, 15\}$, $\{RK_r[PT^{-r}[j]] : r \in \{r_0, \ldots, r_0+r_1-1\}\}$ are uniquely determined by $\{TKz[j] : z \in \{1, \ldots, t/n\}\}$. In attack figures, j will be listed in each round-cell $RK_r[PT^{-r}[j]]$. Assume N_j cells in $\{RK_r[PT^{-r}[j]] : r \in \{r_0, \ldots, r_0 + r_1 - 1\}\}$ will be deduced from the internal parameters that determine the output sequence. Then $N_j = \sum_{r=r_0}^{r_0+r_1-1} \text{typeK-}RK_r[PT^{-r}[j]]$. Each relation on these N_j round-key cells can be converted to a relation on those internal parameters. As $\{RK_r[PT^{-r}[j]] : r \in \{r_0, \ldots, r_0+r_1-1\}\}$ are uniquely determined by $\{TKz[j] : z \in \{1, \ldots, t/n\}\}$, the possible values space of internal parameters can be reduced by $N_j - t/n$ cells from $N_j - t/n$ independent relations on these N_j round-keys if $N_j > t/n$.

Introduce an integer variable $Cut_{keysieve_j}$ for each position j to represent the reduced cells. Constraints over $\{Cut_{keysieve_j}, \sum_{r=r_0}^{r_0+r_1-1} \text{typeK-}RK_r[PT^{-r}[j]]\}$ are imposed satisfying $Cut_{keysieve_j} = \text{Max}\ (0, \sum_{r=r_0}^{r_0+r_1-1} \text{typeK-}RK_r[PT^{-r}[j]] - t/n)$. Then the overall reduced cells by key-dependent-sieve technique are $\sum_{j=0}^{15} Cut_{keysieve_j}$ and denoted by $Cut_{keysieve}$ listed in attack figures.

4.5 Modelling the Non-full Key-Addition Technique

The non-full key-addition exploits the relations on the parameters that determine the output difference sequence and proposition 3 shows that these internal parameters satisfy typeGT-* = 1 or typeGZ-* =1. And typeGT-* = 1 or typeGZ-* =1 has been unified by typeV-$*$ = 1 in Section 4.4. Property 2 shows how to exploit all possible dependencies within parameters. Introduce an integer variable Cut_r for each round to represent the reduced cells, restrict (typeV-$\mathbf{S}_r[0], \ldots,$ typeV-$\mathbf{S}_r[n-1],$ typeV-$\mathbf{S}_{r+1}[s] \ldots,$ typeV-$\mathbf{S}_{r+1}[n-1], Cut_r$) to take values in the subset of all possible values of $(g_r[0], \ldots, g_r[2n-1-s], \sum_{j=0}^{2n-1-s} g_r[j] - \beta_{g_r})$ shown in property 2. Constraints over these variables can be imposed by a system of linear inequalities by using the convex hull computation method.

Model of Non-full Key-Addition for SKINNY. The round function of SKINNY is a little different from that defined in property 2. The technique can be also considered in a similar way. For SKINNY, the first two rows of state before the ShiftRows will be updated by XORing the round-keys. We will model the technique for each column of SKINNY, and the property for all columns are the same. Thus each column of SKINNY can be simply described by $(y_0, y_1, y_2, y_3) = \mathbb{L}(x_0 \oplus rk_0, x_1 \oplus rk_1, x_2, x_3)$, where $\mathbb{L} = MC \circ SR$ is the composite linear transformation matrix. For example, $y_i = \mathbf{S}_{r+1}[4 \cdot i]$ and $x_i = \mathbf{S}_r^{\mathbb{SB}}[4 \cdot i + (-i)\%4]$ for 0th column. In SKINNY case, introduce vector $g = (g[0], \ldots, g[5]) \in \mathbb{F}_2^6$ corresponding to $(x_2, x_3, y_0, y_1, y_2, y_3)$. For each possible value of g, obtain the rank β_g of matrix consisting of $\{\mathbb{L}_j^{-1} : g[j-2] = 1, j \in [2,3]\}$ and $\{\overrightarrow{\mathbf{e}_j} : g[j+2] = 1, j \in [0, \ldots, 3]\}$. Introduce an integer Cut_1 for each column to describe the reduced cells. According to property 2, restrict (typeV-x_2, typeV-x_3, typeV-$y_0, \ldots,$ typeV-y_3, Cut_1) to take values in the subset of all possible values of $(g[0], \ldots, g[5], \sum_{j=0}^{5} g[j] - \beta_g)$. The reduced cells in each column of SKINNY by this technique are listed below of MC in attack figures of SKINNY. The overall reduction number by utilizing the technique is denoted by $Cut_{nonfull}$ in attack figures.

4.6 Modelling the Tweak-Difference Cancellation Technique

For tweakable block cipher, the attack considers the output sequence of the associated $\delta(\mathcal{A})$-set by encrypting a plaintext-tweak combination $\{(P^0, TW^0), \ldots, (P^N, TW^N)\}$, where TW^N represents the selected tweak for P^i. The tweak differences can be controlled to cancel the state difference in one round, then differences of the $\delta(\mathcal{A})$-set at more internal cells will be zero, which leads to fewer internal parameters that determine the output sequence. The tweak-difference is proved to be an effective technique for attacks of SKINNY and has not been included in the previous automatic search model in [20].

Assume the tweak difference will be injected to the state by tweak addition. The tweak addition operation and tweak schedule should be considered when imposing constraints over typeX variables following *the forward differential propagation rule* (informally differential propagation with probability 1). We need to introduce typeX variables for each tweak cell, and impose the constraints over typeX variables through each tweak addition following forward differential propagation rule in definition 3 except for the round with tweak-difference cancellation. For the round with tweak-difference cancellation, the tweak materials are controlled to cancel state difference. Assume the tweak addition operation is expressed by $y = x \oplus rT$, where rT represents the tweak material input to the internal state cell.

The propagation rules for the round with tweak-difference cancellation is

$$\text{typeX-}y = \begin{cases} 0 : \text{typeX-}x = \text{typeX-}rT = 0, \\ 1 : \text{typeX-}x \oplus \text{typeX-}rT = 1, \\ 0 \text{ or } 1 : \text{typeX-}x = \text{typeX-}rT = 1, \end{cases} \tag{9}$$

while the propagation rule for other rounds is the forward differential propagation rule presented in definition 3:

$$\text{typeX-}y = \begin{cases} 0 : \text{typeX-}x = \text{typeX-}rT = 0, \\ 1 : \text{others}. \end{cases} \tag{10}$$

The constraints over $(\text{typeX-}x, \text{typeX-}rT, \text{typeX-}y)$ following the rules of Eq. (9) or Eq. (10) can be imposed by using the convex hull computation method. Note that tweak differences are known to attackers and can be treated as constants except in the description of *forward differential trail*. In order to inject fewer differences from the tweak, the tweak differences are usually controlled to cancel the state difference in the first round of the distinguisher.

Model of Tweak-Difference Cancellation for SKINNY. We will introduce typeX variables for each round-key cell $RK_r[j]$. If one of $\{TKz[j] : z \in \{1, \ldots, t/n\}\}$ is loaded with tweak material, the tweak difference will propagate to round-key $RK_r[PT^{-r}[j]]$. If $\text{typeX-}RK_r[j] = 1$ (▨), at least one of $\{TKz[PT^r[j]] : z \in \{1, \ldots, t/n\}\}$ is loaded with tweak material. Tweak difference introduced in one position is more controllable and sufficient to cancel state difference in the first round of distinguisher. For simplicity, we will give the description of attacks by loading tweak material on the positions of $TK1$. For example in Fig. 10 and Fig. 11, tweak materials will be loaded in $TK1[1]$ and are controlled to cancel state difference in 4th round.

4.7 Modelling the Key-Recovery Phase

Firstly, the key-recovery phase is to find a pair of plaintext (P, P') conforming to the truncated differential trail. Secondly, guess round-keys involved in r_0 rounds to construct a $\delta(\mathcal{A})$-set from P for the distinguisher. Finally, guess round-keys involved in last r_2 rounds $(r_0 + r_1, r_0 + r_1 + 1, \ldots, r_0 + r_1 + r_2 - 1)$ to obtain the value of $\Delta Enc_{r_1}(\delta(\mathcal{A}))[\mathcal{B}]$ sequence by partially decrypting the associated $\delta(\mathcal{A})$-set. The methods for modelling the last two phases please refer to Shi et al.'s work [20], which are achieved by introducing two new type variables following a backward differential propagation rule through the first r_0 rounds and a forward determination propagation rule through the last r_2 rounds. And a key-bridging technique is performed for SKINNY [4].

Here we also need to model the phase of constructing a plaintext structure to find a pair of (P, P') conforming to the truncated differential trail of the distinguisher. In order to construct plaintext structure and observe ciphertext difference for each pair of plaintexts in the structure, we should propagate the input difference of the distinguisher with probability 1 from round r_0 to plaintext and propagate the output difference of the distinguisher with probability 1 from round $r_0 + r_1$ to ciphertext. Introduce typeE type variables for each state involved in first r_0 rounds and last r_2 rounds. Impose constrains over typeE variables satisfying that (typeE-$\mathbf{S}_0 \xleftarrow{\mathcal{E}_1^{-1}} \dots \xleftarrow{\mathcal{E}_1^{-1}}$ typeE-$\mathbf{S}_{r_0}$) form a *backward differential trail* and ($typeE{-}\mathbf{S}_{r_0+r_1} \xrightarrow{\mathcal{E}_1} \dots \xrightarrow{\mathcal{E}_1}$ typeE-$\mathbf{S}_{r_{r_0+r_1+r_2}}$) form a *forward differential trail*. Besides, typeE-$\mathbf{S}_{r_0}[j]$ = typeT-$\mathbf{S}_{r_0}[j]$ and typeE-$\mathbf{S}_{r_0+r_1}[j]$ = typeT-$\mathbf{S}_{r_0+r_1}[j], \forall j \in \{0, \dots, n-1\}$ will be imposed. According to the definition of forward and backward differential trails, we have the following observation. The plaintext structure to find a pair conforming to the truncated differential can be constructed by a $\delta(\mathcal{A}^T)$-set of $\{P^0, P^i, \dots, P^N\}$ satisfying that $P^0 \oplus P^i[j] = 0, \forall j \notin \mathcal{A}^T$, where $\mathcal{A}^T = [\mathbf{S}_0[j] :$ typeE-$\mathbf{S}_0[j] = 1, j \in [0, \dots, n-1]]$. It is fairly straightforward to see the online phase of attacks on SKINNY in Section 2.4 and Fig. 11.

5 Results of SKINNY Block Cipher

All of the known improvement techniques (differential enumeration, key-dependent-sieve, non-full key-addition, tweak-difference cancellation, key-bridging) are integrated into the automatic search for the best DS-MITM attack on SKINNY. This full-fledged automatic model for SKINNY makes full use of the ability to choose tweaks and output the DS-MITM key-recovery attack directly.

The results are summarized in Table 1. All known DS-MITM attacks on the respective versions of SKINNY are improved, and the data, memory, or time complexities of some attacks are reduced even compared to previous best attacks penetrating less rounds. The previous best 10.5-round distinguisher for SKINNY-128-384 is also improved by 2.5 rounds by changing the objective of the model to identify the best distinguishers, which is presented in Sections P.

5.1 Brief Illustration of Figures and Complexity Computation

The attack phase can be easily verified from all figures, and so does the attack complexities. We would give a brief illustration of attack figures on the SKINNY family and the unified attack complexities computation methods. We will only give one detailed attack phase on SKINNY-128-384 (Fig. 10 and Fig. 11) to help readers understand and check the model.

Figure illustration in distinguisher figures.

- ▨ and ▧ form a forward differential trail (Definition 3) and a backward determination trail (Definition 5) respectively.
- 🟩 cells represent the internal parameters that determine the output difference according to proposition 2.
- The reduced cells by applying the non-full key-addition technique on 🟩 cells of each column are listed below of the operation MC. The total reduced number is represented by $Cut_{NonFull}$.
- 🟧 cells denote the round-keys deduced from these 🟩 cells. The $Cut_{Keysieve}$ represents the reduced number by utilizing the key-dependent-sieve technique on these deduced 🟧 round-keys.
- The number j listed in the round-key cell represents that this round-key cell can be uniquely determined by $\{TKz[j] : z \in \{1 \ldots, t/n\}\}$. If ▨ is drawn in this round-key cell, then $TK1[j]$ is loaded by tweak material.

Complexity in precomputation phase. The time complexity for constructing a lookup table to save all possibilities is $N \cdot 2^{c \cdot OBJ_{Dis}} \cdot \rho$, where N is the size of the $\delta(\mathcal{A})$-set ($N = |\delta(\mathcal{A})|$), c is the length of each cell, ρ is typically computed by the number of active S-box (▩) divided by total number of S-box in attacked rounds of SKINNY, and OBJ_{Dis} is the objective function of the distinguisher defined in remark 3. And the memory complexity is $(N-1) \cdot (|\mathcal{B}| \cdot c) \cdot 2^{OBJ_{Dis} \cdot c}$, where $|\mathcal{B} \cdot c|$ is the length of each output sequence $\varDelta\mathcal{E}_r(\delta(\mathcal{A}))[\mathcal{B}]$.

Figure illustration in the online key-recovery phase

- ▨ cells form a backward differential trail (Definition 4).
- 🟧 cells denote the round-keys involved to construct a plaintext structure to identify a $\delta(\mathcal{A})$-set and obtain the output sequence by partially decrypting the associated $\delta(\mathcal{A})$-set. The key-bridging technique can be utilized in these round-keys, which is also presented in the following attack on SKINNY-128-384.

Complexity in the online phase. The time complexity in the online phase is $N \cdot 2^{OBJ_{KC} \cdot c} \cdot \rho_1$, where N is the size of the $\delta(\mathcal{A})$-set, c is the length of the cell, ρ_1 is typically computed by number of active Sbox (▨) divided by total number of S-box, and OBJ_{KC} represents objective function of the key-recovery attack defined by the number of guessed round-keys. The data complexity is $2^{N_{data} \cdot c}$, where N_{data} is the number of ▨ at Round 0.

5.2 25 Rounds Attack on SKINNY-128-384 (376-bit key, 8-bit tweak)

Load the 8-bit tweak material in $TK1[1]$, which will propagate to round-keys $RK_r[PT^{-r}[j]]$: $\{RK_0[1], RK_1[9], RK_2[0] \ldots\}$.

Precomputation phase.

Proposition 4 (11-round distinguisher on SKINNY-128-384 (Fig. 10)). *Define $\mathcal{A} = [\mathbf{S}_4[2]], \mathcal{B} = [\mathbf{S}_{15}[10]]$. Construct a $\delta(\mathcal{A})$-set of $\{P^0, P^1, \ldots, P^{N-1}\}$ and a tweak material set of $\{TW^0, TW^1, \ldots, TW^N\}$ satisfying that $P^i[\mathbf{S}_4[2]] \oplus P^0[\mathbf{S}_4[2]] = i$ and $TW^i[RK_4[2]] = P^i[\mathbf{S}_4^{\mathbb{SB}}[2]] \oplus P^0[\mathbf{S}_4^{\mathbb{SB}}[2]], \forall i \in \{0, 1, \ldots, N-1\}$. Then $\Delta\mathcal{E}_{11}(\delta(\mathcal{A}))[\mathcal{B}]$ sequence can only take at most $(2^8)^{41}$ values.*

Proof. After the tweak-difference cancellation, $P^i \oplus P^0[\mathbf{S}_6[j]] = 0, j \in [0, \ldots, 15]$, $\forall i \in \{1, \ldots, N-1\}$. Then $P^i \oplus P^0[\mathbf{S}_7[9]] = TW^i \oplus TW^0[RK_6[4]]$, so $\{P^0, P^1, \ldots, P^N\}$ also identify a $\delta(\mathcal{A}')$-set for $\mathcal{A}' = [\mathbf{S}_7[9]]$. It is trivial from proposition 2 that the output difference sequence $\Delta\mathcal{E}_{11}(\delta(\mathcal{A}))[\mathcal{B}]$ can be uniquely determined by following 46-cell internal parameters (🟩):

$$P^0[\mathbf{S}_7[9]], \{P^0[\mathbf{S}_8[j]] : j \in [3, 11, 15]\}, \{P^0[\mathbf{S}_9[j]] : j \in [1, 2, 3, 7, 9, 11, 13, 15]\}$$
$$\{P^0[\mathbf{S}_{10}[j]] : j \notin [4, 10, 12, 13]\}, \{P^0[\mathbf{S}_{11}[j]] : j \notin [0, 4, 5, 13, 15]\}$$
$$\{P^0[\mathbf{S}_{12}[j]] : j \in [1, 3, 5, 8, 11, 14]\}, \{P^0[\mathbf{S}_{13}[j]] : j \in [1, 7, 10]\}, \{P^0[\mathbf{S}_{14}] : j \in [5, 8]\}.$$

Non-full key-addition technique. According to property 2, the possible values space of the internal parameters can be reduced by 5 bytes from following relations on internal parameters in the 9th round, 10th round, 11th round:

- 1 in $\{P^0[\mathbf{S}_9[15]], P^0[\mathbf{S}_{10}[2]], P^0[\mathbf{S}_{10}[14]]\}$ as $P^0[\mathbf{S}_9[15]] = P^0[\mathbf{S}_{10}[2]] \oplus P^0[\mathbf{S}_{10}[14]]$,
- 1 in $\{P^0[\mathbf{S}_9[9]], P^0[\mathbf{S}_{10}[7]], P^0[\mathbf{S}_{10}[15]]\}$ as $P^0[\mathbf{S}_9[9]] = P^0[\mathbf{S}_{10}[7]] \oplus P^0[\mathbf{S}_{10}[15]]$,
- 2 in $\{P^0[\mathbf{S}_{10}[8]], P^0[\mathbf{S}_{10}[15]], P^0[\mathbf{S}_{11}[2]], P^0[\mathbf{S}_{11}[6]], P^0[\mathbf{S}_{11}[14]]\}$ as $P^0[\mathbf{S}_{10}[8]] = P^0[\mathbf{S}_{11}[6]] \oplus P^0[\mathbf{S}_{11}[14]]$, $P^0[\mathbf{S}_{10}[15]] = P^0[\mathbf{S}_{11}[2]] \oplus P^0[\mathbf{S}_{11}[14]]$,
- 1 in $\{P^0[\mathbf{S}_{11}[11]], P^0[\mathbf{S}_{11}[14]], P^0[\mathbf{S}_{12}[1]], P^0[\mathbf{S}_{12}[5]]\}$ as $P^0[\mathbf{S}_{11}[11]] \oplus P^0[\mathbf{S}_{11}[14]] = P^0[\mathbf{S}_{12}[1]] \oplus P^0[\mathbf{S}_{12}[5]]$.

Thus $\Delta\mathcal{E}_{11}(\delta(\mathcal{A}))[\mathcal{B}]$ sequence can be uniquely determined by 41-cell internal bytes. $N = |\delta(\mathcal{A})| = 43$ is enough to construct the distinguish for SKINNY, because there are $2^{8 \cdot 42}$ possibilities for a random 42-byte sequence. Build a lookup table $Tab_{\Delta\mathcal{E}_{11}(\delta(\mathcal{A}))[\mathcal{B}]}$ to save all of the $2^{8 \cdot 41}$ possibilities.

Complexity. The time complexity to construct a hash table in the precomputation phase to save all possibilities is $43 \cdot 2^{41 \cdot 8} \cdot \frac{47}{16 \cdot 25} \approx 2^{330.31}$. And the memory complexity is $42 \cdot 8 \cdot 2^{41 \cdot 8} \approx 2^{336.39}$.

Online Phase.

The 25-round attack on SKINNY-128-384 can be extended by adding 4 rounds at the start and 10 rounds at the end (Fig. 11). 🟧 cells represent involved guessed round-keys in the online phase.

- Query an arbitrate plaintext-tweak combination of P^0 and TW^0 such that $TW^0[RK_4[2]] = 0$ to obtain the corresponding ciphertext C^0.
- For each possible value of these active round-keys (🟧):

step 1 Deduce internal parameters of P^0 in active cells (▨): $P^0[\mathbf{S}_4[2]]$, $P^0[\mathbf{S}_3[15]]$, $\{P^0[\mathbf{S}_2[j]] : j \in [0, 6, 9, 12]\}$, $\{P^0[\mathbf{S}_1[j]] : j \in [2, 4, 5, 7, 8, 10, 13]\}$.

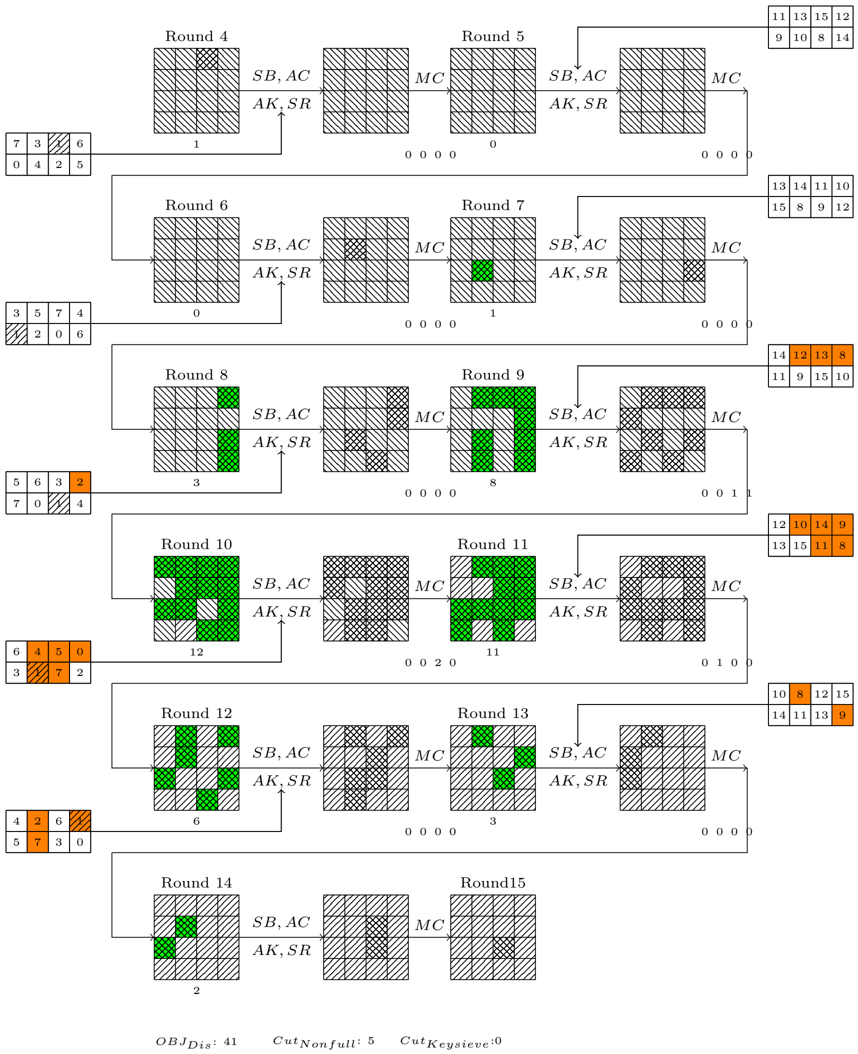


Fig. 10. 11-round Distinguisher of SKINNY-128-384

$\mathcal{A} = [\mathbf{S}_4[2]], \mathcal{B} = [\mathbf{S}_{15}[10]]$. Load tweak material in $TK1[1]$. The output difference can be uniquely determined by 46-cells internal parameters (🟩). 5 bytes are reduced by the non-full key-addition technique, which are listed below of each MC. $|\delta(\mathcal{A})| = 43$ is enough to construct distinguish. As the output sequence of SKINNY-128-384 can take at most $2^{8 \cdot 41}$ possible values, while there are $2^{8 \cdot 42}$ possibilities for a random 42-byte sequence. Each number j listed in round-key cell represents that these round-key cells can be uniquely determined by $\{TKz[j] : z \in \{1, \ldots, t/n\}\}$. ▨ on the round-key cell represents that the tweak material loaded in $TK1[j]$ will propagate to this cell, where tweak-difference may be injected.

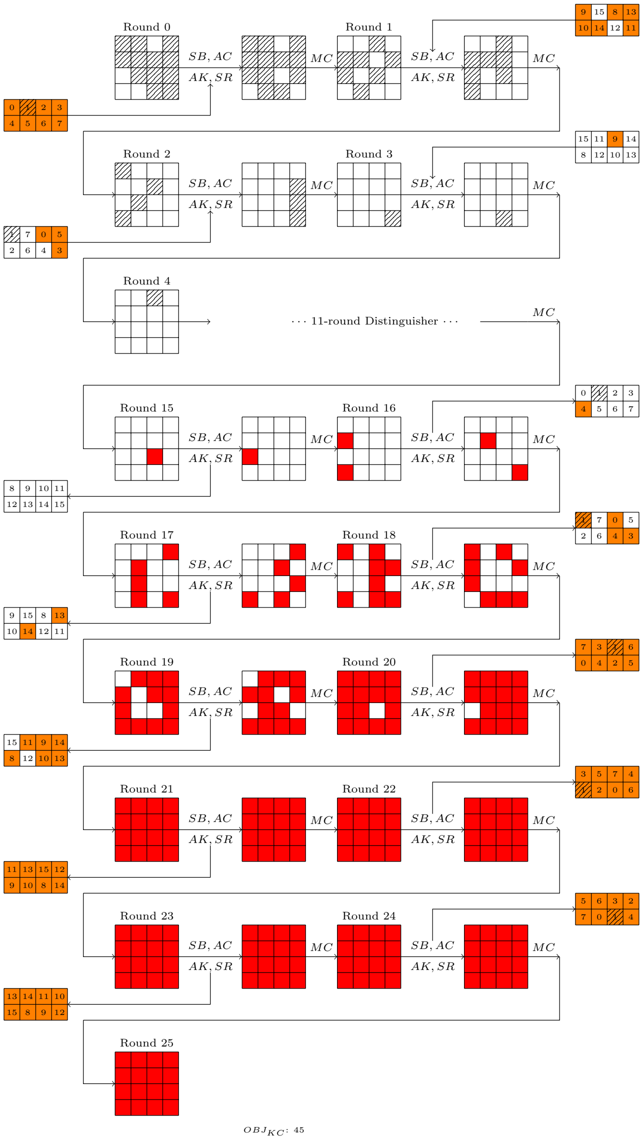


Fig. 11. 25-round attack on SKINNY-128-384

▨ describe a backward differential trail to determine the plaintext structure that will identify a $\delta(\mathcal{A})$-set for the distinguisher. All internal parameters of P in ▨ cells of $\mathbf{S}_i, 1 \leq i \leq 4$ can be deduced from plaintext by guessing values of involved 🟧 round-keys. The output sequence can deduced from the ciphertext difference by guessing values of the involved 🟧 round-keys. Two guesses for $\{TKz[12], TKz[15] : z \in \{1, 2, 3\}\}$ are saved.

step 2 A structure of plaintext-tweak combinations $\{(P^i, TW^i) : i = 0, \dots, N-1\}$ satisfying that $\{P^0, \dots, P^{N-1}\}$ is a $\delta(\mathcal{A})$-set ($\mathcal{A} = [\mathbf{S}_4[2]]$) with $P^i[\mathbf{S}_4[2]] \oplus P^0[\mathbf{S}_4[2]] = i$ and $TW^i[RK_4[2]] = P^i[\mathbf{S}_4^{\mathbb{SB}}[2]] \oplus P^0[\mathbf{S}_4^{\mathbb{SB}}[2]]$ can be constructed from above internal parameters in following ways. Firstly, deduce $TW^i[RK_4[2]]$, which is $P^i \oplus P^0[\mathbf{S}_4^{\mathbb{SB}}[2]]$ deduced from $P^i \oplus P^0[\mathbf{S}_4[2]] = i$ and $P^0[\mathbf{S}_4[2]]$, then TW^i loaded in $TK[1]$ is determined by the tweakey schedule. Secondly, $\{P^i \oplus P^0[\mathbf{S}_3^{\mathbb{SR}}[j]] : j \in [0, \dots, 15]\}$ can be deduced from $\{P^i \oplus P^0[\mathbf{S}_4[j]] : j \in [0, \dots, 15]\}$ as MC is a linear transformation, and $P^i \oplus P^0[\mathbf{S}_3^{\mathbb{SR}}[j]] = 0, \forall j \neq 14$. Then $P^i[\mathbf{S}_3] \oplus P^0[\mathbf{S}_3]$ can be uniquely determined from parameter $P^0[\mathbf{S}_3[15]]$ in the active cell (deduced in step 1) through the inverse of ShiftRows. Iterate the process, $P^i \oplus P^0$ can be uniquely determined by these internal parameters (▨) deduced in step 1.

step 3 Obtain the ciphertext $\{C^0, \dots, C^{N-1}\}$ by querying the plaintext-tweak combinations.

step 4 The output difference $P^i \oplus P^0$ at $\mathbf{S}_{15}[10]]$ can be obtained by partially decrypting the ciphertext difference by values of active round-keys (🟧).

step 5 Check whether the output sequence in the lookup table $Tab_{\Delta\mathcal{E}_{11}(\delta(\mathcal{A}))[\mathcal{B}]}$ constructed in precomputation phase, obtain the candidate of guessed round-keys that past the text.

- *Key-bridging technique.* The key-bridging technique can be utilized to reduce the guessed number of involved round-keys. If more than t/n cells in these round-key cells that can be uniquely determined by $\{TKz[j] : z \in \{1, \dots, t/n\}\}$, guess the values of the master keys $\{TKz[j] : z \in \{1, \dots, t/n\}\}$ directly. Otherwise, guess the values of round-key cells directly. Thus in this attack, guessing values of the master keys $\{TKz[1] : z \in \{2, 3\}\} \cup \{TKz[j] : z \in \{1, 2, 3\}, j \notin [1, 12, 15]\}$, two round-key cells $\{RK_{21}[3], RK_{23}[7]\}$ updated from $\{TKz[12] : z \in \{1, 2, 3\}\}$, and two round-key cells $\{RK_{21}[2], RK_{23}[4]\}$ updated from $\{TKz[15] : z \in \{1, 2, 3\}\}$ is sufficient to obtain all values of involved round keys (🟧). We also say that two cell guesses for $\{TKz[j] : z \in \{1, 2, 3\}, j \in [12, 15]\}$ are saved in the full key space.

Complexity. The time complexity is $N \cdot 2^{OBJ_{KC} \cdot c} \cdot \rho_1$, where c is the length of the cell, ρ_1 is typically computed by number of active S-box (▨) divided by total number of S-box, which is $43 \cdot 2^{45 \cdot 8} \cdot \frac{132}{16 \cdot 25} \approx 2^{363.83}$. The data complexity is $2^{8 \cdot 12} = 2^{96}$.

6 Discussions.

- We also apply our method to AES. Our tool can recover the previous best DS-MITM attacks. However no better result is obtained.
- Different results in the single-key chosen-tweak setting and single-tweak are listed in Table 2 and Table 3 respectively, which illustrate that most of key-recovery attacks are not extended from the best distinguishers.

- What is more, the best key-recovery attacks on SKINNY are produced without utilizing the differential enumeration technique that has been included in the model, while the best distinguishers are produced by utilizing this technique. The best 13-round distinguisher of SKINNY-128-384 (Fig. 42) in the single-key single-tweak setting is presented in Sect. P, which improves the previous best 10.5-round distinguisher by 2.5 rounds, and can not be extended to the best attack. We guess the reason is the design of the linear layer. The backward differential and backward determination trails through the linear layer of SKINNY from the same input are different, while they are the same through the linear layer of AES. Then the involved round-keys for finding a pair of plaintexts conforming to a truncated differential trail and for constructing a $\delta(\mathcal{A})$-set are different.
- Interestingly, the time of searching for the best attack is less than that of searching for the best distinguisher sometimes. For example, the best 25-round key-recovery attack on SKINNY-128-384 in the single-key chosen-tweak setting is produced in 331 seconds, while the best 13-round distinguisher in the single-key single-tweak setting is produced in 1012 seconds.

Acknowledgments

We thank anonymous reviewers for their valuable comments. This research is supported by the National Key R&D Program of China (Grants No. 2022YFB2701900, 2018YFA0704704), the National Natural Science Foundation of China (Grants No. 62172410, 62022036, 62132008, 62032014, 62202460), the Youth Innovation Promotion Association of Chinese Academy of Sciences, and the Fundamental Research Funds for the Central Universities.

References

1. Ankele, R., Dobraunig, C., Guo, J., Lambooij, E., Leander, G., Todo, Y.: Zero-correlation attacks on tweakable block ciphers with linear tweakey expansion. IACR Trans. Symmetric Cryptol. **2019**(1), 192–235 (2019). https://doi.org/10.13154/tosc.v2019.i1.192-235
2. Bao, Z., Guo, J., Shi, D., Tu, Y.: Superposition meet-in-the-middle attacks: Updates on fundamental security of aes-like hashing. In: Dodis, Y., Shrimpton, T. (eds.) Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part I. Lecture Notes in Computer Science, vol. 13507, pp. 64–93. Springer (2022). https://doi.org/10.1007/978-3-031-15802-5_3
3. Beierle, C., Jean, J., Kölbl, S., Leander, G., Moradi, A., Peyrin, T., Sasaki, Y., Sasdrich, P., Sim, S.M.: The SKINNY family of block ciphers and its low-latency variant MANTIS. In: Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II. pp. 123–153 (2016). https://doi.org/10.1007/978-3-662-53008-5_5

4. Chen, Q., Shi, D., Sun, S., Hu, L.: Automatic demirci-selçuk meet-in-the-middle attack on SKINNY with key-bridging. In: Zhou, J., Luo, X., Shen, Q., Xu, Z. (eds.) Information and Communications Security 2019. Lecture Notes in Computer Science, vol. 11999, pp. 233–247. Springer (2019). https://doi.org/10.1007/978-3-030-41579-2_14
5. Daemen, J., Rijmen, V.: The Design of Rijndael - The Advanced Encryption Standard (AES), Second Edition. Information Security and Cryptography, Springer (2020). https://doi.org/10.1007/978-3-662-60769-5
6. Demirci, H., Selçuk, A.A.: A meet-in-the-middle attack on 8-round AES. In: Fast Software Encryption, 15th International Workshop, FSE 2008. pp. 116–126 (2008). https://doi.org/10.1007/978-3-540-71039-4_7
7. Demirci, H., Taskin, I., Çoban, M., Baysal, A.: Improved meet-in-the-middle attacks on AES. In: Roy, B.K., Sendrier, N. (eds.) Progress in Cryptology - INDOCRYPT 2009, 10th International Conference on Cryptology in India, New Delhi, India, December 13-16, 2009. Proceedings. Lecture Notes in Computer Science, vol. 5922, pp. 144–156. Springer (2009). https://doi.org/10.1007/978-3-642-10628-6_10
8. Derbez, P., Fouque, P.: Exhausting demirci-selçuk meet-in-the-middle attacks against reduced-round AES. In: Fast Software Encryption - 20th International Workshop, FSE 2013, Singapore, March 11-13, 2013. Revised Selected Papers. pp. 541–560 (2013). https://doi.org/10.1007/978-3-662-43933-3_28
9. Derbez, P., Fouque, P.: Automatic search of meet-in-the-middle and impossible differential attacks. In: Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II. pp. 157–184 (2016). https://doi.org/10.1007/978-3-662-53008-5_6
10. Derbez, P., Fouque, P., Jean, J.: Improved key recovery attacks on reduced-round AES in the single-key setting. In: Johansson, T., Nguyen, P.Q. (eds.) Advances in Cryptology - EUROCRYPT 2013. Proceedings. Lecture Notes in Computer Science, vol. 7881, pp. 371–387. Springer (2013). https://doi.org/10.1007/978-3-642-38348-9_23
11. Dong, X., Hua, J., Sun, S., Li, Z., Wang, X., Hu, L.: Meet-in-the-middle attacks revisited: Key-recovery, collision, and preimage attacks. In: Malkin, T., Peikert, C. (eds.) Advances in Cryptology - CRYPTO 2021, Proceedings, Part III. Lecture Notes in Computer Science, vol. 12827, pp. 278–308. Springer (2021). https://doi.org/10.1007/978-3-030-84252-9_10
12. Dunkelman, O., Huang, S., Lambooij, E., Perle, S.: Biased differential distinguisher - cryptanalysis of reduced-round SKINNY. Inf. Comput. **281**, 104796 (2021). https://doi.org/10.1016/j.ic.2021.104796
13. Dunkelman, O., Keller, N., Shamir, A.: Improved single-key attacks on 8-round AES-192 and AES-256. In: Advances in Cryptology - ASIACRYPT 2010. Proceedings. pp. 158–176 (2010). https://doi.org/10.1007/978-3-642-17373-8_10
14. Hadipour, H., Eichlseder, M.: Autoguess: A tool for finding guess-and-determine attacks and key bridges. In: Ateniese, G., Venturi, D. (eds.) Applied Cryptography and Network Security - 20th International Conference, ACNS 2022, Rome, Italy, June 20-23, 2022, Proceedings. Lecture Notes in Computer Science, vol. 13269, pp. 230–250. Springer (2022). https://doi.org/10.1007/978-3-031-09234-3_12
15. Hadipour, H., Sadeghi, S., Eichlseder, M.: Finding the impossible: Automated search for full impossible differential, zero-correlation, and integral attacks. IACR Cryptol. ePrint Arch. p. 1147 (2022), https://eprint.iacr.org/2022/1147

16. Li, L., Jia, K., Wang, X.: Improved single-key attacks on 9-round AES-192/256. In: Fast Software Encryption - 21st International Workshop, FSE 2014, London, UK, March 3-5, 2014. Revised Selected Papers. pp. 127–146 (2014). https://doi.org/10.1007/978-3-662-46706-0_7
17. Li, R., Jin, C.: Meet-in-the-middle attacks on 10-round AES-256. Des. Codes Cryptography **80**(3), 459–471 (2016). https://doi.org/10.1007/s10623-015-0113-3
18. Li, R., Jin, C.: Meet-in-the-middle attacks on round-reduced tweakable block cipher deoxys-bc. IET Inf. Secur. **13**(1), 70–75 (2019). https://doi.org/10.1049/iet-ifs.2018.5091
19. Lin, L., Wu, W., Wang, Y., Zhang, L.: General model of the single-key meet-in-the-middle distinguisher on the word-oriented block cipher. In: Information Security and Cryptology - ICISC 2013 - 16th International Conference, Seoul, Korea, November 27-29, 2013, Revised Selected Papers. pp. 203–223 (2013). https://doi.org/10.1007/978-3-319-12160-4_13
20. Shi, D., Sun, S., Derbez, Todo, Y., Sun, B., Hu, L.: Programming the demirci-selçuk meet-in-the-middle attack with constraints. In: Advances in Cryptology - ASIACRYPT 2018, Proceedings, Part II. Lecture Notes in Computer Science, vol. 11273, pp. 3–34. Springer (2018). https://doi.org/10.1007/978-3-030-03329-3_1
21. Shi, D., Sun, S., Song, L., Hu, L., Yang, Q.: Exploiting non-full key additions: Full-fledged automatic demirci-selcuk meet-in-the-middle cryptanalysis of skinny. IACR Cryptol.ePrint Arch. p. 255 (2023), https://eprint.iacr.org/2023/255
22. Tolba, M., Abdelkhalek, A., Youssef, A.M.: Impossible differential cryptanalysis of reduced-round SKINNY. In: Joye, M., Nitaj, A. (eds.) Progress in Cryptology - AFRICACRYPT 2017, Proceedings. Lecture Notes in Computer Science, vol. 10239, pp. 117–134 (2017). https://doi.org/10.1007/978-3-319-57339-7_7
23. Yang, D., Qi, W., Chen, H.: Impossible differential attacks on the SKINNY family of block ciphers. IET Inf. Secur. **11**(6), 377–385 (2017). https://doi.org/10.1049/iet-ifs.2016.0488

Supplementary Material

Table 2. Overall Results of Attacks on SKINNY in single-key chosen-tweak setting

Cipher(Target)	R_{attack}	R_{Dis}	OBJ_{KC}	OBJ_{Dis}	N_{data}
SKINNY-128-384	25	11	45	42	12
	25	12	45	43	12
SKINNY-128-256	21	9	29	20	13
	21	10	29	29	8
	21	10	29	21	13
	21	11	29	29	12
	21	11	29	30	8
SKINNY-64-192	23	12	45	45	13
	23	11	43	43	12
	23	11	43	38	15
	23	10	43	42	12
	23	9	43	42	15
SKINNY-64-128	21	9	29	29	15
	21	10	30	28	13

[1] All numbers for $OBJ_{KC}, OBJ_{Dis}, N_{data}$ in table represent the number of cells.

[2] A R_{attack} rounds attack is extended from a R_{Dis}-round distinguisher. $\Delta\mathcal{E}_{R_{Dis}}(\delta(\mathcal{A}))[\mathcal{B}]$ sequence can take at most $2^{c\cdot OBJ_{Dis}}$ possible values. The number of guessed round-keys to construct plaintexts set identifying a $\delta(\mathcal{A})$-set for the distinguisher and obtain the output sequence by partially decrypting the associated $\delta(\mathcal{A})$-set is OBJ_{KC}. The data complexity is $2^{c\cdot N_{data}}$. c is length of the cell, which is 4 and 8 for 64 block size and 128 block size respectively. The OBJ_{KC} and OBJ_{Dis} should be less than the size of the master key.

A 21-round DS-MITM attack on SKINNY-128-256(248-bit key, 8-bit tweak)

1. Load 8-bit tweak material in $TK1[1]$. $\mathcal{A} = [\mathbf{S}_4[2]], \mathcal{B} = [\mathbf{S}_{13}[11]]$
2. Precomputation Phase (Fig. 12). Construct a $\delta(\mathcal{A})$-set with $|\delta(\mathcal{A})| = 24$, the output difference sequence at $\mathbf{S}_{13}[11]$ can be uniquely determined by 22-internal parameters (🟩), which can be used to distinguish from a random 23-byte sequence. The time complexity for building a lookup table to save all possible values that the output sequence may take is $24 \cdot 2^{8\cdot 22} \cdot \frac{22}{16\cdot 21} \approx 2^{176.65}$. The memory complexity is $23 \cdot 8 \cdot 2^{22\cdot 8} \approx 2^{183.52}$.

Table 3. Overall Results of Attacks on SKINNY in single-key single-tweak setting

Cipher(Target)	R_{attack}	R_{Dis}	OBJ_{KC}	OBJ_{Dis}	N_{data}	mode
SKINNY-128-384	23	8	47	41	12	Multiset
	23	9	46	42	12	Multiset
	23	10	46	42	12	Multiset
	23	11	46	43	12	Multiset
	23	9	46	43	12	Sequence
	23	10	46	44	12	Sequence
	23	11	46	45	12	Sequence
SKINNY-128-256	20	9	31	30	12	Multiset
	20	10	31	30	12	Multiset
	19	8	29	24	12	Multiset
	19	9	28	30	12	Multiset
	19	9	30	29	7	Multiset
	19	10	28	31	12	Multiset
	19	10	30	30	7	Multiset
	19	8	29	26	12	Sequence
	19	9	29	27	12	Sequence
	19	9	30	30	7	Sequence
	19	10	29	31	12	Sequence
	19	10	30	31	7	Sequence
SKINNY-128-128	17	7	15	13	12	Multiset
	17	8	15	13	12	Multiset
	17	7	15	15	12	Sequence
	17	8	15	15	12	Sequence
SKINNY-64-192	21	10	45	46	8	Sequence
	21	10	44	47	11	Sequence
SKINNY-64-128	19	7	31	30	14	Multiset
	19	7	31	31	14	Multiset
	19	6	31	26	12	Sequence
	19	7	30	30	13	Sequence
	19	7	31	28	12	Sequence
	19	8	30	29	15	Sequence
	19	8	30	31	13	Sequence
SKINNY-64-64	17	7	15	14	12	Sequence

[1] All numbers for $OBJ_{KC}, OBJ_{Dis}, N_{data}$ in table represent the number of cells.

[2] A R_{attack} rounds attack is extended from a R_{Dis} rounds distinguisher. The output sequence $\Delta\mathcal{E}_{R_{Dis}}(\delta(\mathcal{A}))[\mathcal{B}]$ of the distinguisher can take at most $2^{c\cdot OBJ_{Dis}}$ possible values. The number of guessed round-keys to construct a $\delta(\mathcal{A})$-set and obtain the output sequence by partially decrypting the associated $\delta(\mathcal{A})$-set is OBJ_{KC}. The data complexity is $2^{c\cdot N_{data}}$. c is length of the cell, which is 4 and 8 for 64 block size and 128 block size respectively. The OBJ_{KC} and OBJ_{Dis} should be less than the size of the master key.

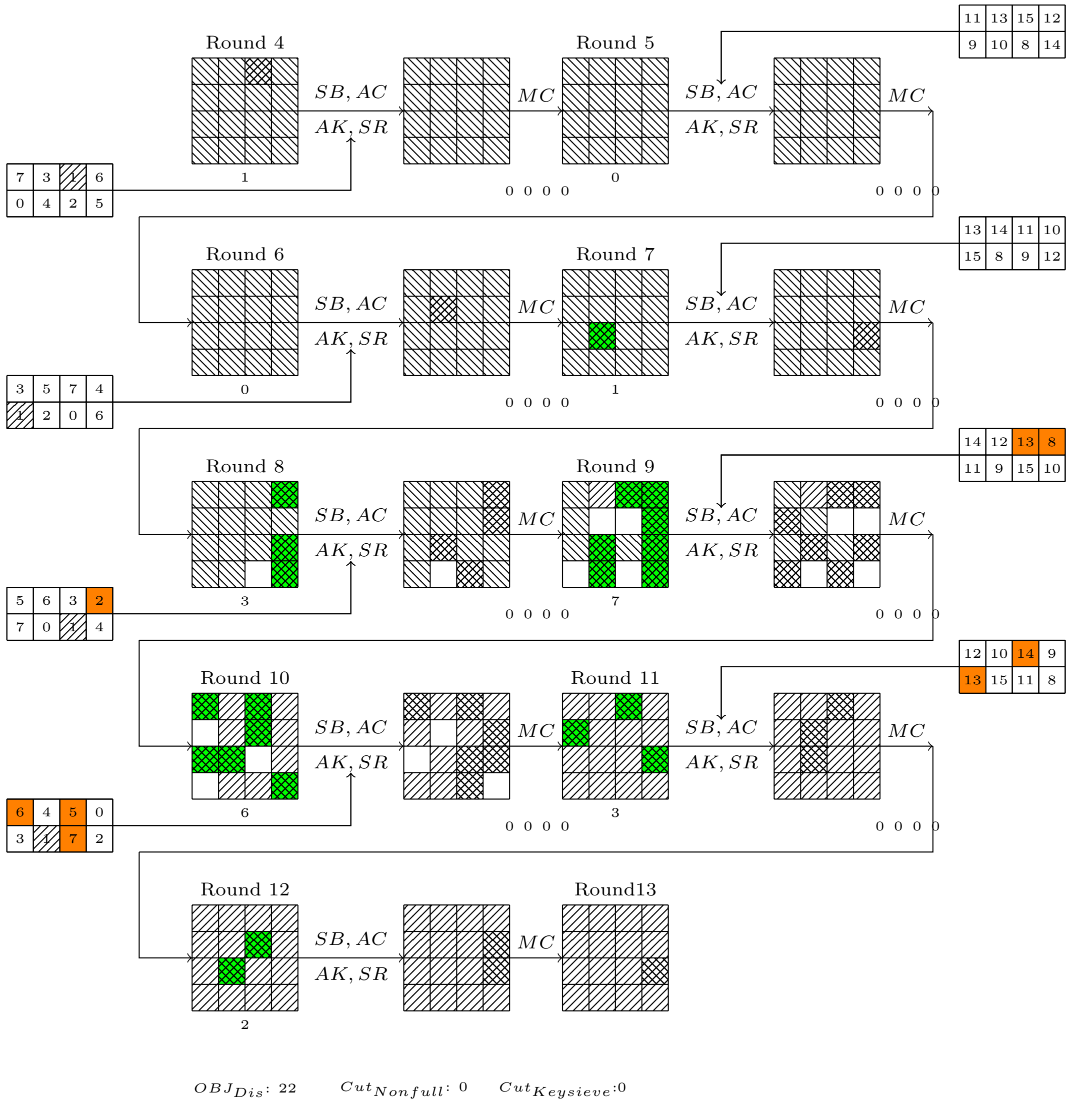


Fig. 12. 9-round Distinguisher of SKINNY-128-256

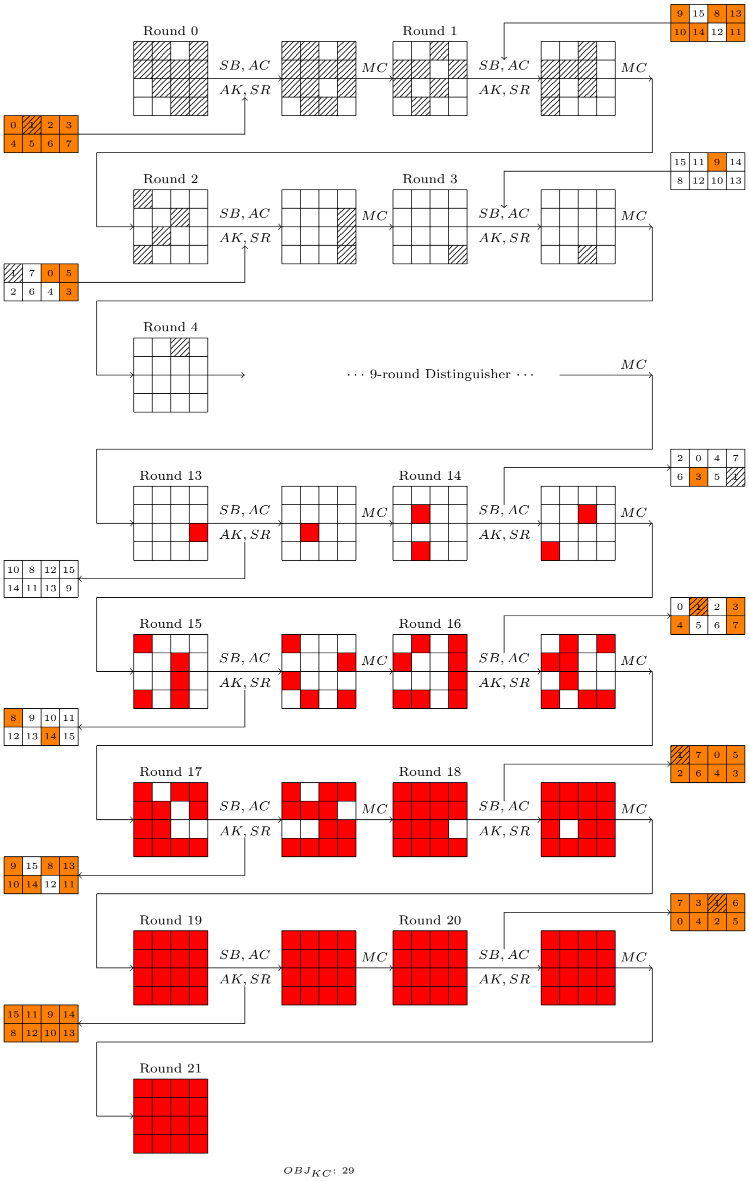


Fig. 13. 21-round Attack on SKINNY-128-256

3. Online phase (Fig. 13). All round-keys marked by 🟧 are guessed to construct a plaintext structure that identifies a $\delta(\mathcal{A})$-set and get the value of the output difference at $\mathbf{S}_{13}[11]$ by partially decrypting associated $\delta(\mathcal{A})$-set. Among all these 31-cell master key, 2-cell guesses for $\{TKz[12,15] : z \in \{1,2\}\}$ are saves. Thus the time complexity is $24 \cdot 2^{8 \cdot 29} \cdot \frac{100}{16 \cdot 21} \approx 2^{234.84}$. The data complexity is 2^{96}.

B 21-round DS-MITM attack on SKINNY-128-256(248-bit key, 8-bit tweak)

1. Load 8-bit tweak material in $TK1[1]$. $\mathcal{A} = [\mathbf{S}_4[3]], \mathcal{B} = [\mathbf{S}_{13}[8]]$.
2. Precomputation Phase (Fig. 14). Construct a $\delta(\mathcal{A})$-set with $|\delta(\mathcal{A})| = 21$, the output difference sequence at $\mathbf{S}_{13}[8]$ can be uniquely determined by 19-internal values (🟩), which can be used to distinguish from a random 20-byte sequence. The time complexity for building a lookup table to save all possible values that the output sequence may take is $21 \cdot 2^{8 \cdot 19} \cdot \frac{19}{16 \cdot 21} \approx 2^{152.25}$. The memory complexity is $20 \cdot 8 \cdot 2^{19 \cdot 8} \approx 2^{159.32}$.
3. Online phase (Fig. 15). All round-keys marked by 🟧 are guessed to construct a plaintext structure that identifies a $\delta(\mathcal{A})$-set and get the value of the output difference at $\mathbf{S}_{13}[11]$ by partially decrypting associated $\delta(\mathcal{A})$-set. Among all these 31-cell master key, 2-cell guesses can be saved for $\{TKz[8], TKz[11] : z \in \{1,2\}\}$. Thus the time complexity is $21 \cdot 2^{8 \cdot 29} \cdot \frac{102}{16 \cdot 21} \approx 2^{234.67}$. The data complexity is 2^{104}.

C 21-round DS-MITM attack on SKINNY-128-256(248-bit key, 8-bit tweak)

1. Load 8-bit tweak material in $TK1[14]$. $\mathcal{A} = [\mathbf{S}_3[3]], \mathcal{B} = [\mathbf{S}_{13}[11]]$.
2. Precomputation Phase (Fig. 16). Construct a $\delta(\mathcal{A})$-set with $|\delta(\mathcal{A})| = 30$, the output difference sequence at $\mathbf{S}_{13}[11]$ can be uniquely determined by 30-internal parameters (🟩). Among the 30-internal parameters, 1-cell can be reduced by utilizing the non-full key-addition technique on $\{\mathbf{S}_7[14], \mathbf{S}_8[1], \mathbf{S}_8[13]\}$. 1-cell can be reduced by relations on round-keys updated from $TK2[14]$ by utilizing the key-dependent-sieve technique. Thus the output difference sequence can be uniquely determined by 28 internal parameters, which can be used to distinguish from a random 29-byte sequence. The time complexity for building a lookup table to save all possible values that the output sequence may take is $30 \cdot 2^{8 \cdot 28} \cdot \frac{30}{16 \cdot 21} \approx 2^{225.42}$. The memory complexity is $29 \cdot 8 \cdot 2^{28 \cdot 8} \approx 2^{231.86}$.
3. Online phase (Fig. 17). All round-key marked by 🟧 are guessed to construct a plaintext structure that identifies a $\delta(\mathcal{A})$-set and get the value of the output difference at $\mathbf{S}_{13}[11]$ by partially decrypting associated $\delta(\mathcal{A})$-set. Among all these 31-cell master key, 2-cell guesses can be saved for $\{TKz[12], TKz[15] : z \in \{1,2\}\}$. Thus the time complexity is $30 \cdot 2^{8 \cdot 29} \cdot \frac{89}{16 \cdot 21} \approx 2^{234.99}$. The data complexity is 2^{64}.

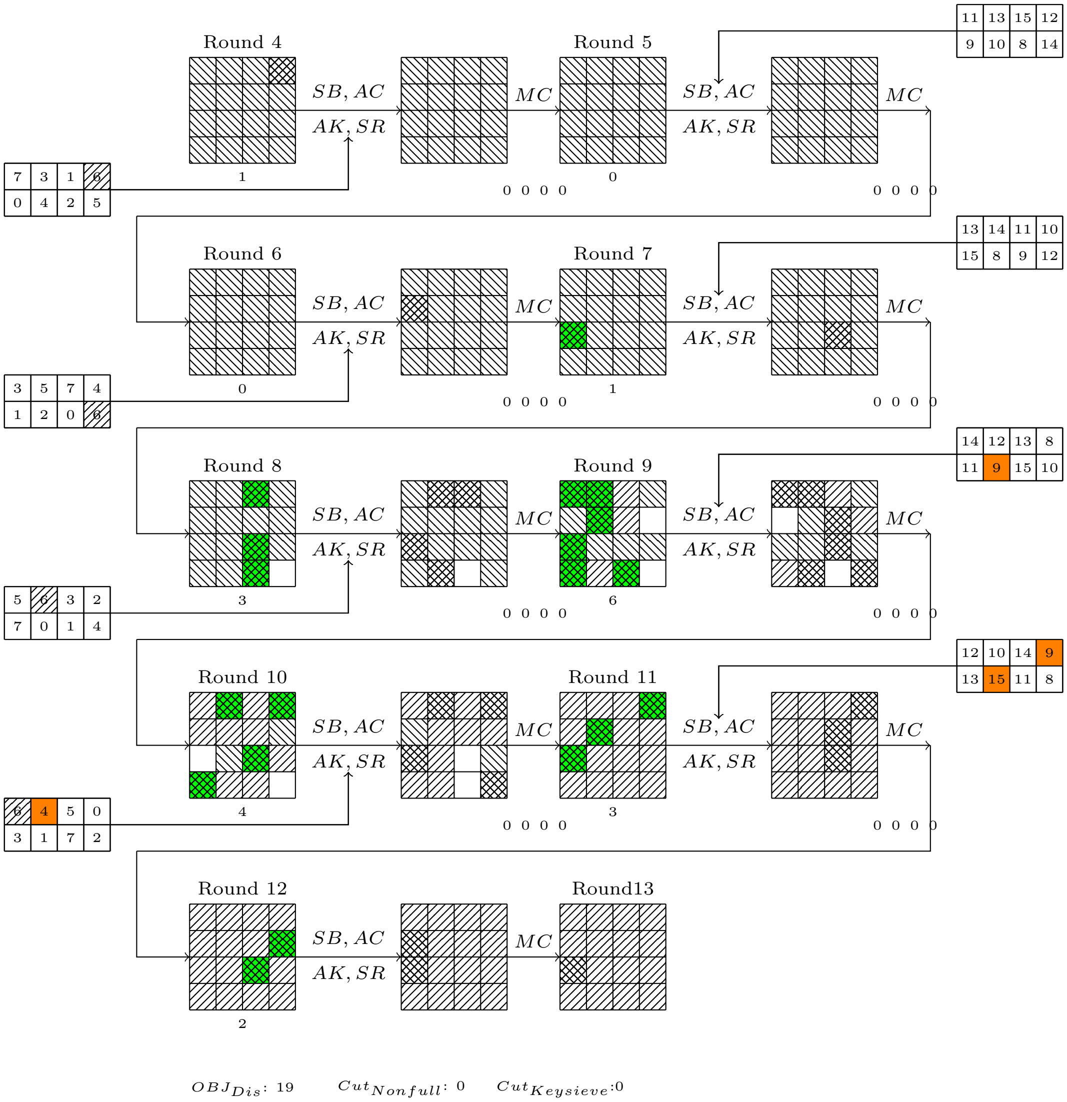


Fig. 14. 9-round Distinguisher of SKINNY-128-256

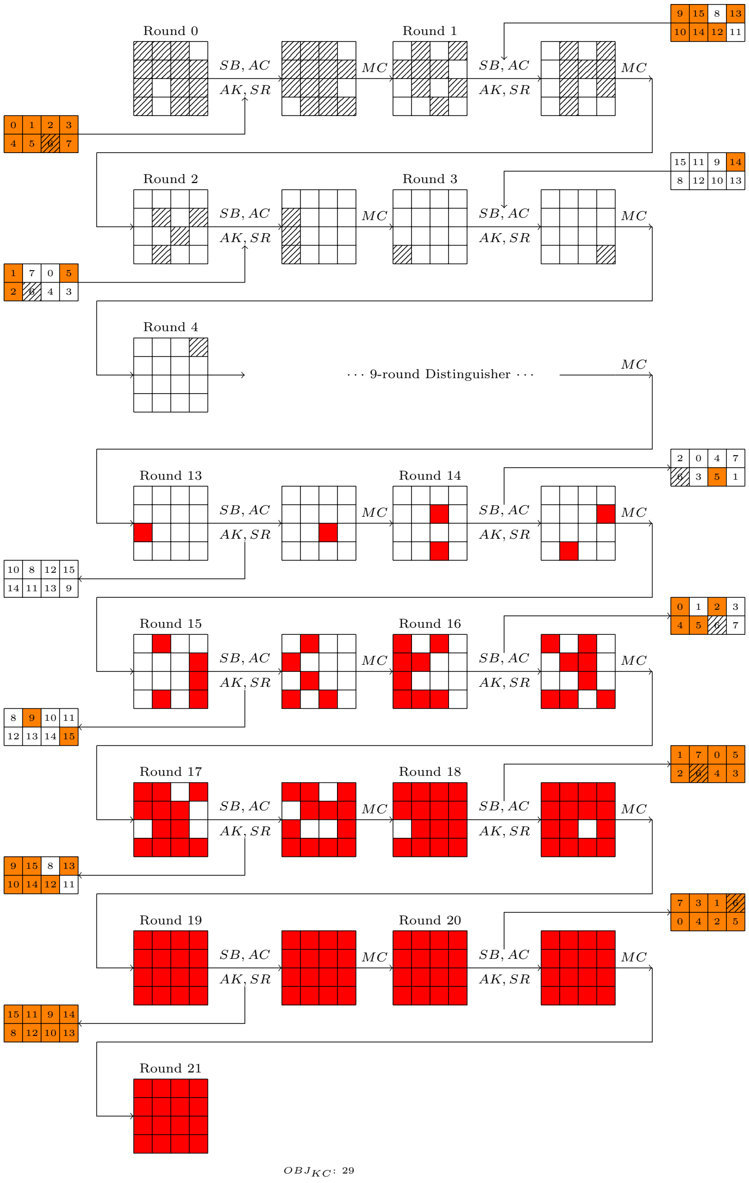


Fig. 15. 21-round Attack on SKINNY-128-256

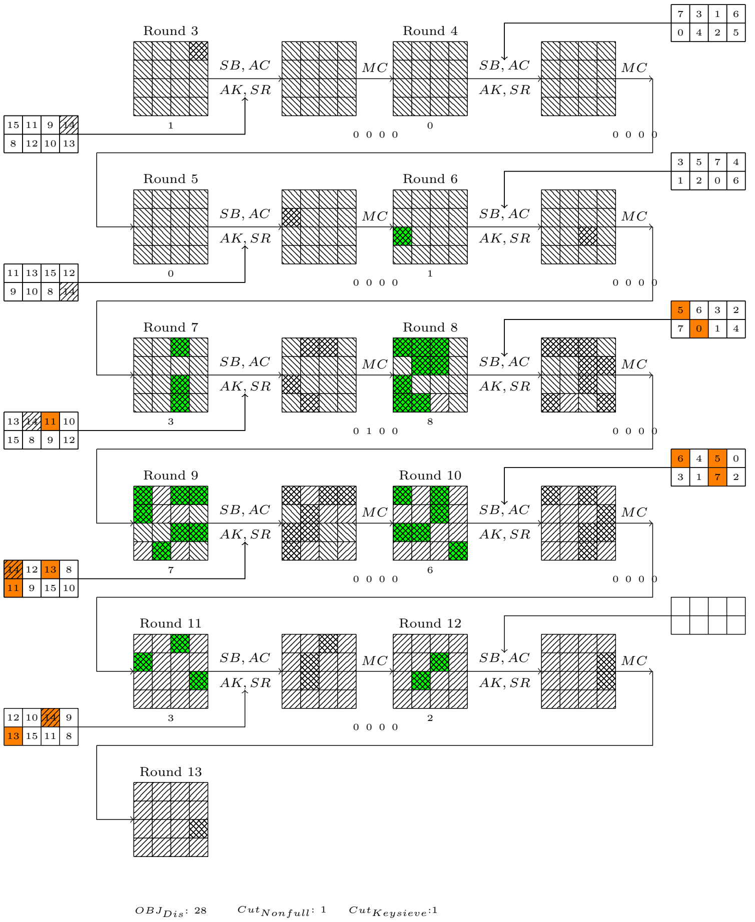


Fig. 16. 10-round Distinguisher of SKINNY-128-256

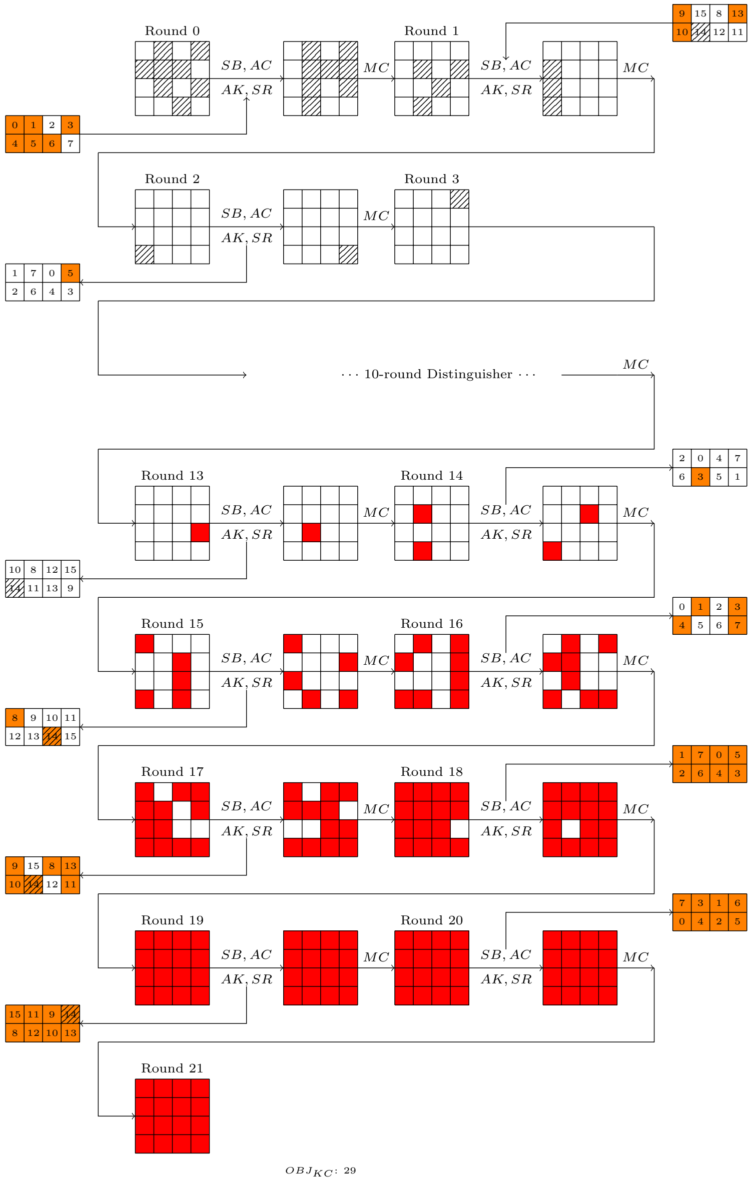


Fig. 17. 21-round Attack on SKINNY-128-256

D 21-round DS-MITM attack on SKINNY-64-128(120-bit key, 8-bit tweak)

1. Load 8-bit tweak material in $TK1[0], TK1[1]$. $\mathcal{A} = [\mathbf{S}_4[2], \mathbf{S}_4[4]], \mathcal{B} = [\mathbf{S}_{13}[11]]$.
2. Precomputation Phase (Fig. 18). Construct a $\delta(\mathcal{A})$-set with $|\delta(\mathcal{A})| = 29$, the output difference sequence can be uniquely determined by 28 internal parameters (🟩). Among the 28 internal parameters, 1-cell can be reduced by utilizing the non-full key-addition technique on $\{\mathbf{S}_8[9], \mathbf{S}_9[7], \mathbf{S}_9[14]\}$. Thus the output difference sequence can be uniquely determined by 27 internal parameters, which can be used to distinguish from a random 28-cell sequence. The time complexity for building a lookup table to save all possible values that the output sequence may take is $29 \cdot 2^{4 \cdot 27} \cdot \frac{28}{16 \cdot 21} \approx 2^{109.27}$. The memory complexity is $29 \cdot 4 \cdot 2^{27 \cdot 4} \approx 2^{114.81}$.
3. Online phase (Fig. 19). All round-keys marked by 🟧 are guessed to construct a plaintext structure that identifies a $\delta(\mathcal{A})$-set and get the value of the output difference at $\mathbf{S}_{13}[11]$ by partially decrypting the plaintext structure. Among all these 30-cell master key, 1-cell guess can be saved for $\{TK1[12], TK2[12]\}$. Thus the time complexity is $29 \cdot 2^{4 \cdot 29} \cdot \frac{116}{16 \cdot 21} \approx 2^{119.32}$. The data complexity is 2^{60}.

E 23-round DS-MITM attack on SKINNY-64-192 (176-bit key, 16-bit tweak)

1. Load 16-bit tweak material in $\{TK1[j] : j \in [1, 5, 13, 4]\}$. $\mathcal{A} = [\mathbf{S}_3[3], \mathbf{S}_3[7]], \mathcal{B} = [\mathbf{S}_{13}[11]]$.
2. Precomputation Phase (Fig. 20). Construct a $\delta(\mathcal{A})$-set with $|\delta(\mathcal{A})| = 45$, the output difference sequence at $\mathbf{S}_{13}[11]$ can be uniquely determined by 61-internal values (🟩). Among the 61 internal parameters, 15-cell can be reduced by utilizing the non-full key-addition technique (marked by number under the operation MC). 3-cell can be reduced by relations on round-keys generated from $\{TKz[j] : z \in \{2, 3\}, j \in [5, 13, 14]\}$ by utilizing the key-dependent-sieve technique. Thus the output difference sequence can be uniquely determined by 43 internal parameters, which can be used to distinguish from a random 44-cell sequence. The time complexity for building a lookup table to save all possible values that the output sequence may take is $45 \cdot 2^{4 \cdot 43} \cdot \frac{61}{16 \cdot 23} \approx 2^{174.9}$. The memory complexity is $44 \cdot 4 \cdot 2^{4 \cdot 43} \approx 2^{179.46}$.
3. Online phase (Fig. 21). All round-key marked by 🟧 are enough to construct a plaintext structure that identifies a $\delta(\mathcal{A})$-set and get the value of output difference at $\mathbf{S}_{13}[11]$ by partially decrypting the plaintext structure. Among all these 44 cells master key, 2-cell guesses can be saved for $\{TKz[12], TKz[15] : z \in \{1, 2, 3\}\}$. Thus the time complexity is $45 \cdot 2^{4 \cdot 42} \cdot \frac{137}{16 \cdot 23} \approx 2^{172.07}$. The data complexity is 2^{56}.

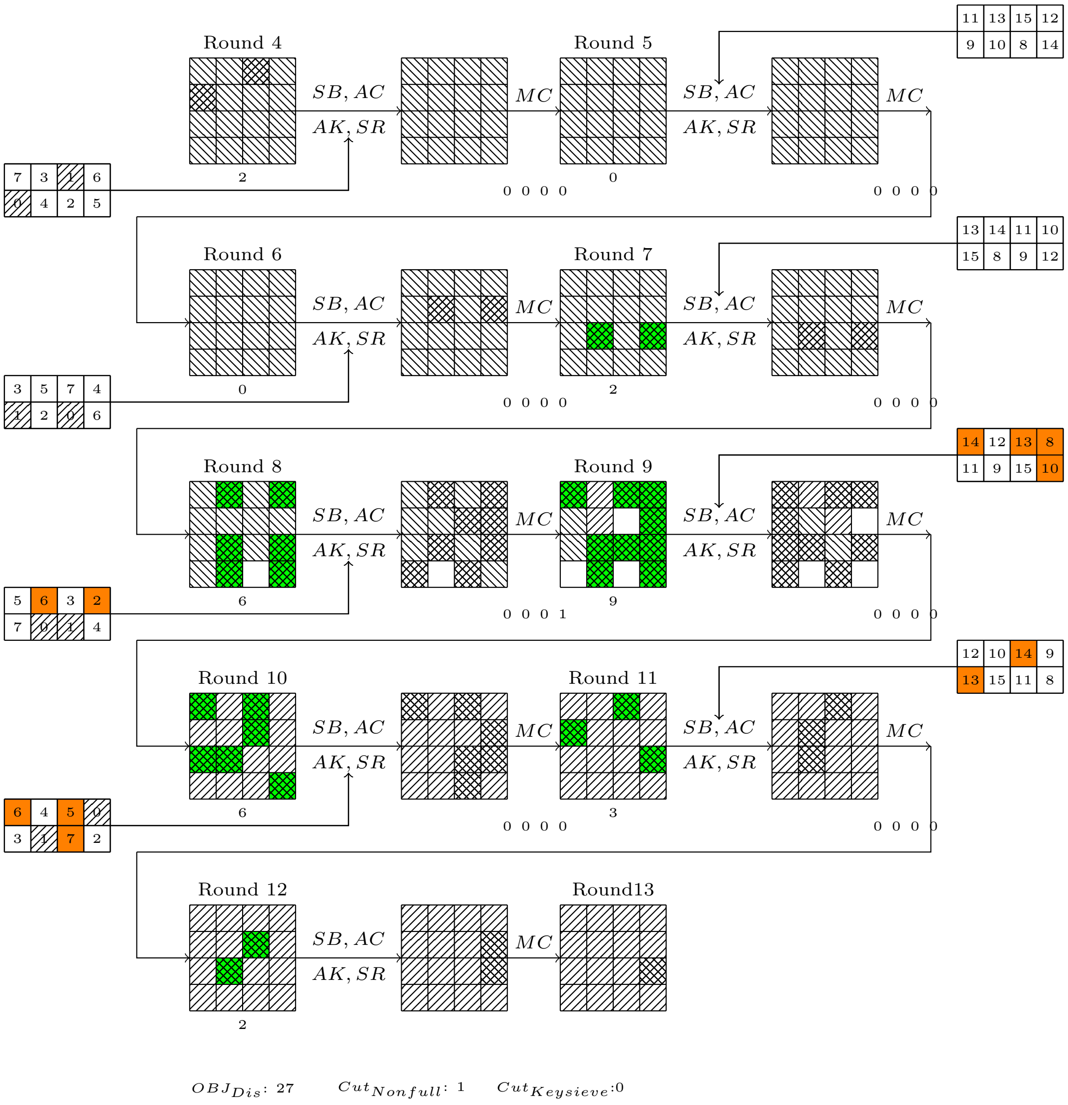


Fig. 18. 9-round Distinguisher of SKINNY-64-128

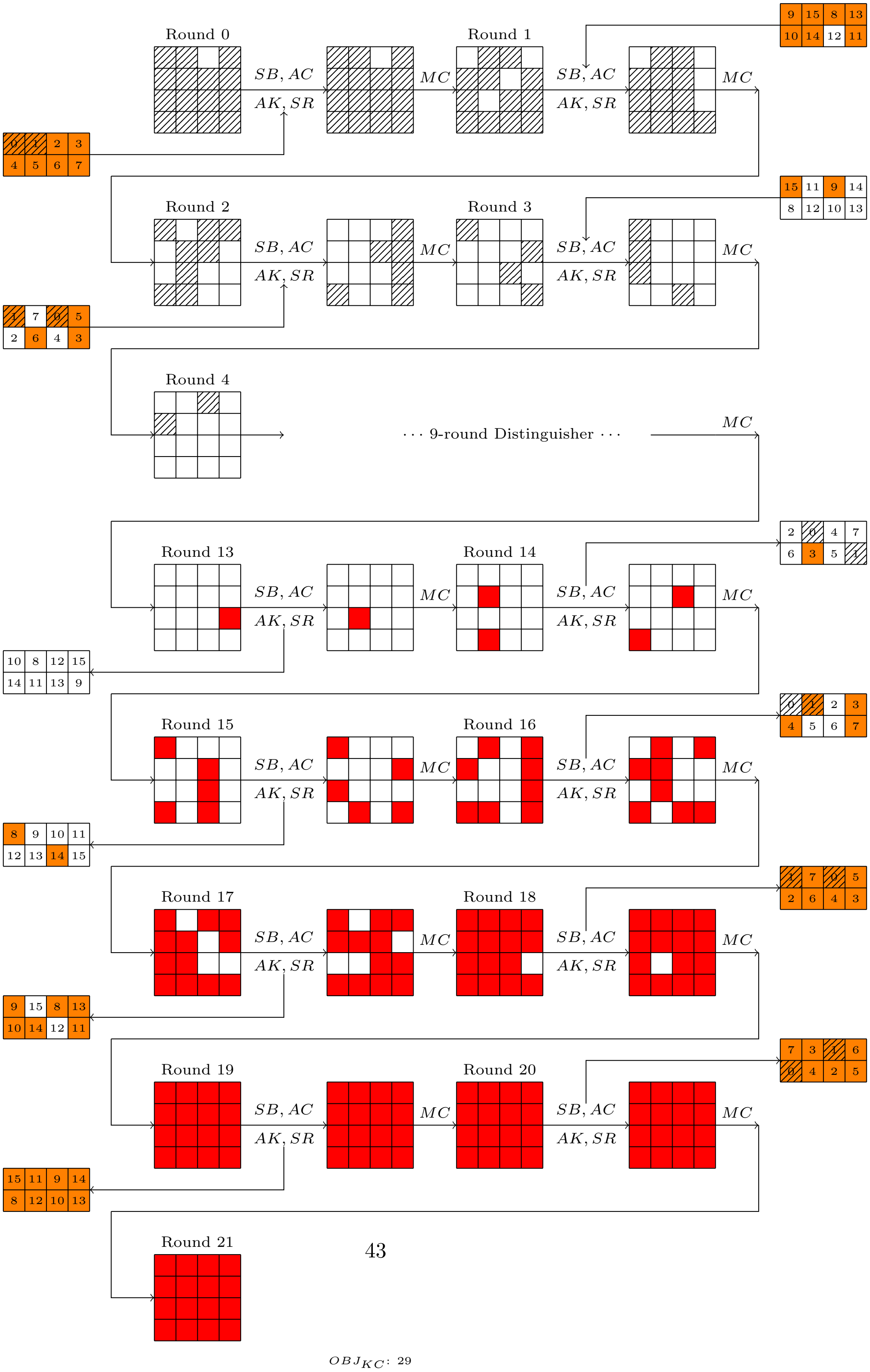
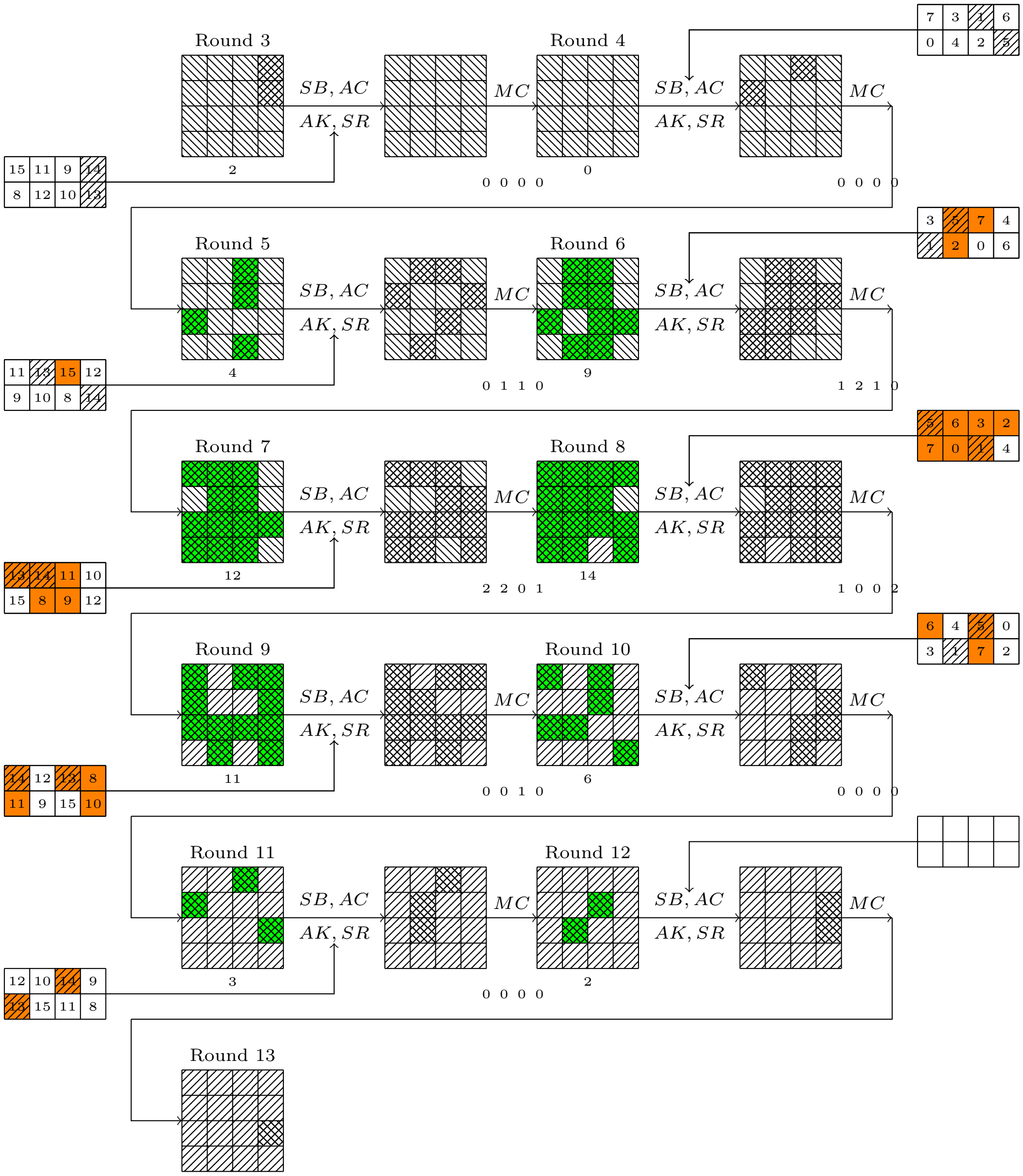


Fig. 19. 21-round Attack on SKINNY-64-128



OBJ_{Dis}: 43 $Cut_{Nonfull}$: 15 $Cut_{Keysieve}$:3

Fig. 20. 10-round Distinguisher of SKINNY-64-192

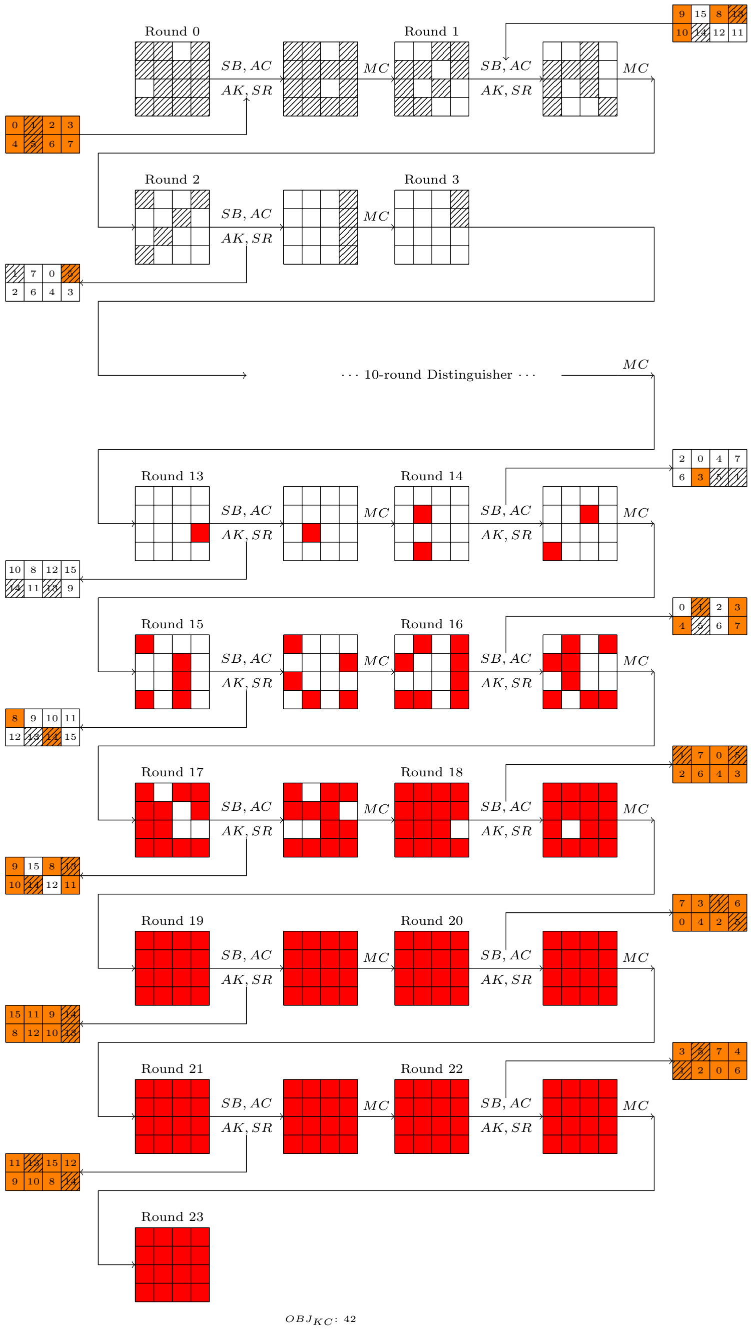


Fig. 21. 23-round Attack on SKINNY-64-192

F 23-round DS-MITM attack on SKINNY-64-192 (184-bit key, 8-bit tweak)

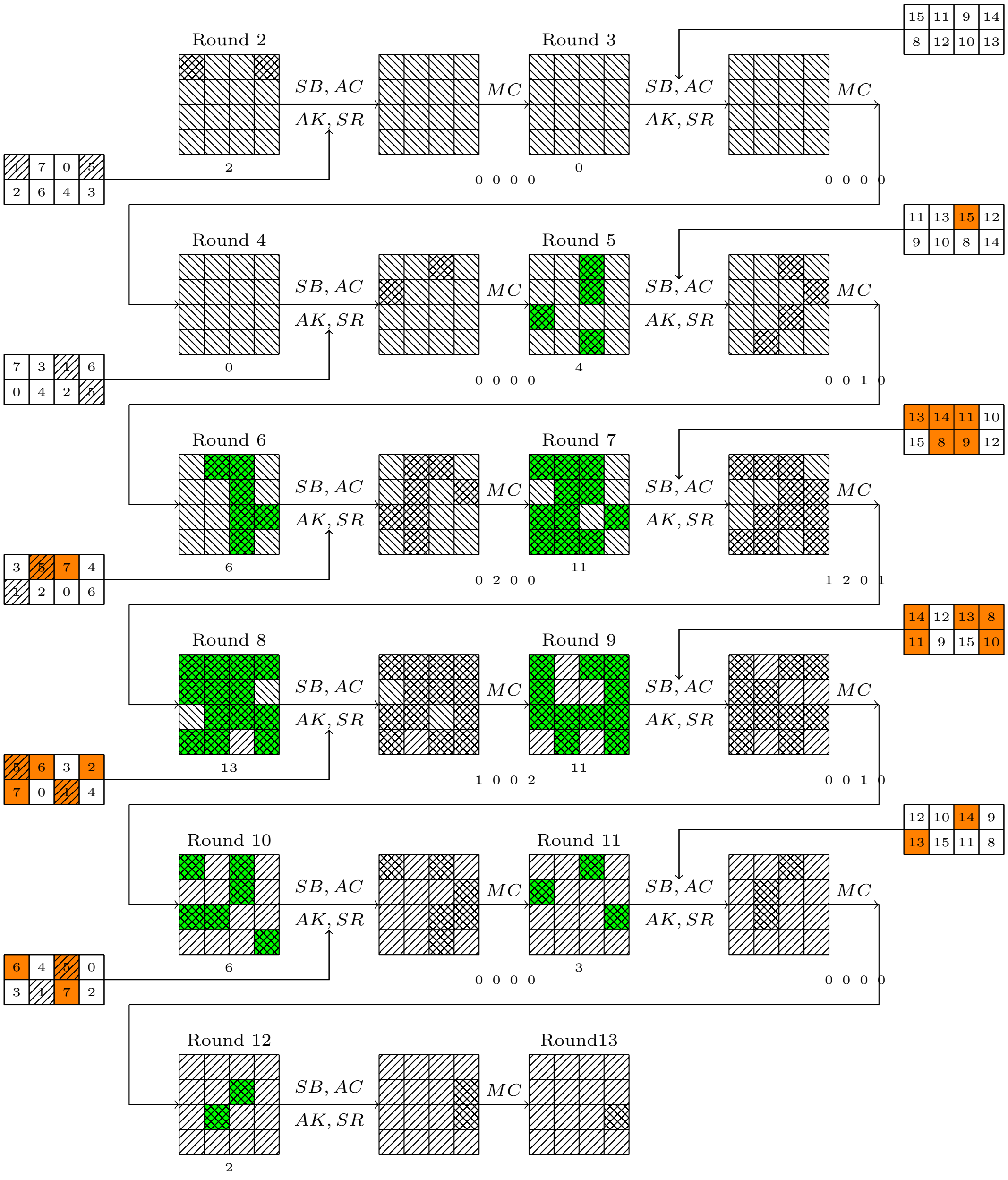
1. Load 8-bit tweak material in $TK1[1], TK1[5]$. $\mathcal{A} = [\mathbf{S}_2[0], \mathbf{S}_2[3]], \mathcal{B} = [\mathbf{S}_{13}[11]]$.
2. Precomputation Phase (Fig. 22). Construct a $\delta(\mathcal{A})$-set with $|\delta(\mathcal{A})| = 46$, the output difference sequence at $\mathbf{S}_{13}[11]$ can be uniquely determined by 56 internal parameters (🟩). Among the 56 internal parameters, 11-cell can be reduced by utilizing the non-full key-addition technique (marked by number under the operation MC). 1-cell can be reduced on relations on round-keys generated from $\{TKz[5] : z \in \{2, 3\}\}$ by utilizing the key-dependent-sieve technique. Thus the output difference sequence can be uniquely determined by 44 internal parameters, which can be used to distinguish from a random 45-cell sequence. The time complexity for building a lookup table to save all possible values that the output sequence may take is $46 \cdot 2^{4 \cdot 44} \cdot \frac{56}{16 \cdot 23} \approx 2^{178.81}$. The memory complexity is $45 \cdot 4 \cdot 2^{4 \cdot 46} \approx 2^{183.49}$.
3. Online phase (Fig. 23). All round-key marked by 🟧 are guessed to construct a plaintext structure that identifies a $\delta(\mathcal{A})$-set and get the value of the output difference at $\mathbf{S}_{13}[11]$ by partially decrypting the plaintext structure. Among all these 46-cell master key, 2-cell guesses can be saved for $\{TKz[12, 15] : z \in \{1, 2, 3\}\}$. Thus the time complexity is $46 \cdot 2^{4 \cdot 44} \cdot \frac{119}{16 \cdot 23} \approx 2^{179.9}$. The data complexity is 2^{32}.

G 23-round DS-MITM attack on SKINNY-128-384(384-bit key, 0-bit tweak)

1. $\mathcal{A} = [\mathbf{S}_4[3]], \mathcal{B} = [\mathbf{S}_{13}[8]]$.
2. Precomputation Phase (Fig. 24). Construct a $\delta(\mathcal{A})$-set with $|\delta(\mathcal{A})| = 45$, the output difference sequence can be uniquely determined by 50 internal parameters (🟩). Among the 50 internal parameters, 7 cells can be reduced by utilizing the non-full key-addition technique (marked by number under the operation MC). Thus the output difference sequence can be uniquely determined by 43 internal parameters, which can be used to distinguish from a random 44-byte sequence. The time complexity for building a lookup table to save all possible values that the output sequence may take is $45 \cdot 2^{8 \cdot 43} \cdot \frac{50}{16 \cdot 23} \approx 2^{346.61}$. The memory complexity is $44 \cdot 8 \cdot 2^{43 \cdot 8} \approx 2^{352.46}$.
3. Online phase (Fig. 25). All round-key marked by 🟧 are guessed to construct a plaintext structure that identifies a $\delta(\mathcal{A})$-set and get the value of output difference at $\mathbf{S}_{13}[8]$ by partially decrypting the plaintext structure. Among all these 48-cell master key, 2-cell guesses can be saved for $\{TKz[8], TKz[11] : z \in \{1, 2, 3\}\}$. Thus the time complexity is $45 \cdot 2^{8 \cdot 46} \cdot \frac{131}{16 \cdot 23} \approx 2^{372}$. The data complexity is 2^{96}.

H 20-round DS-MITM Attack on SKINNY-128-256(256-bit key, 0-bit tweak)

1. $\mathcal{A} = [\mathbf{S}_3[12]], \mathcal{B} = [\mathbf{S}_{12}[10]]$.



OBJ_{Dis}: 44 $Cut_{Nonfull}$: 11 $Cut_{Keysieve}$:1

Fig. 22. 11-round Distinguisher of SKINNY-64-192

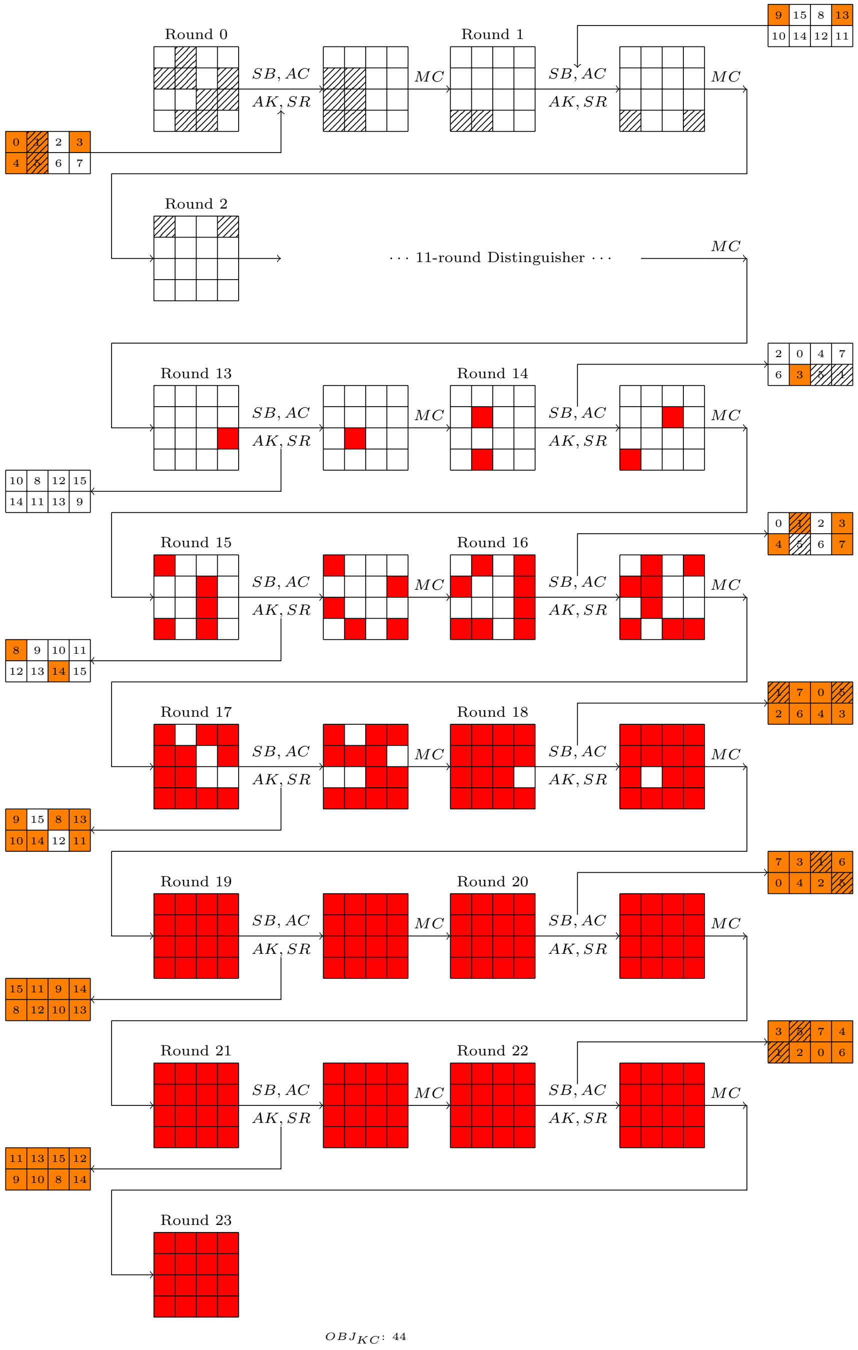
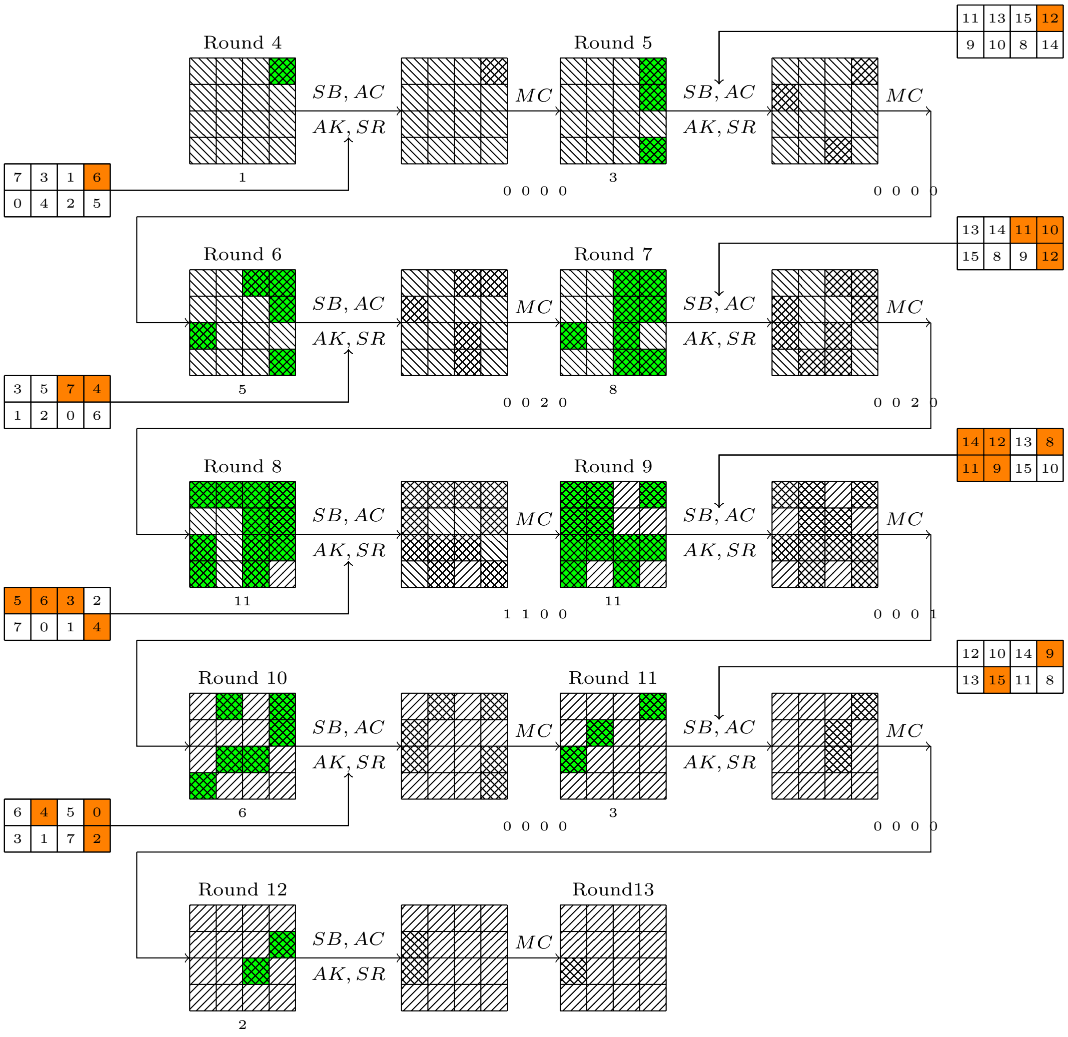


Fig. 23. 23-round Attack on SKINNY-64-192



OBJ_{Dis}: 43 $Cut_{Nonfull}$: 7 $Cut_{Keysieve}$:0

Fig. 24. 9-round Distinguisher of SKINNY-128-384

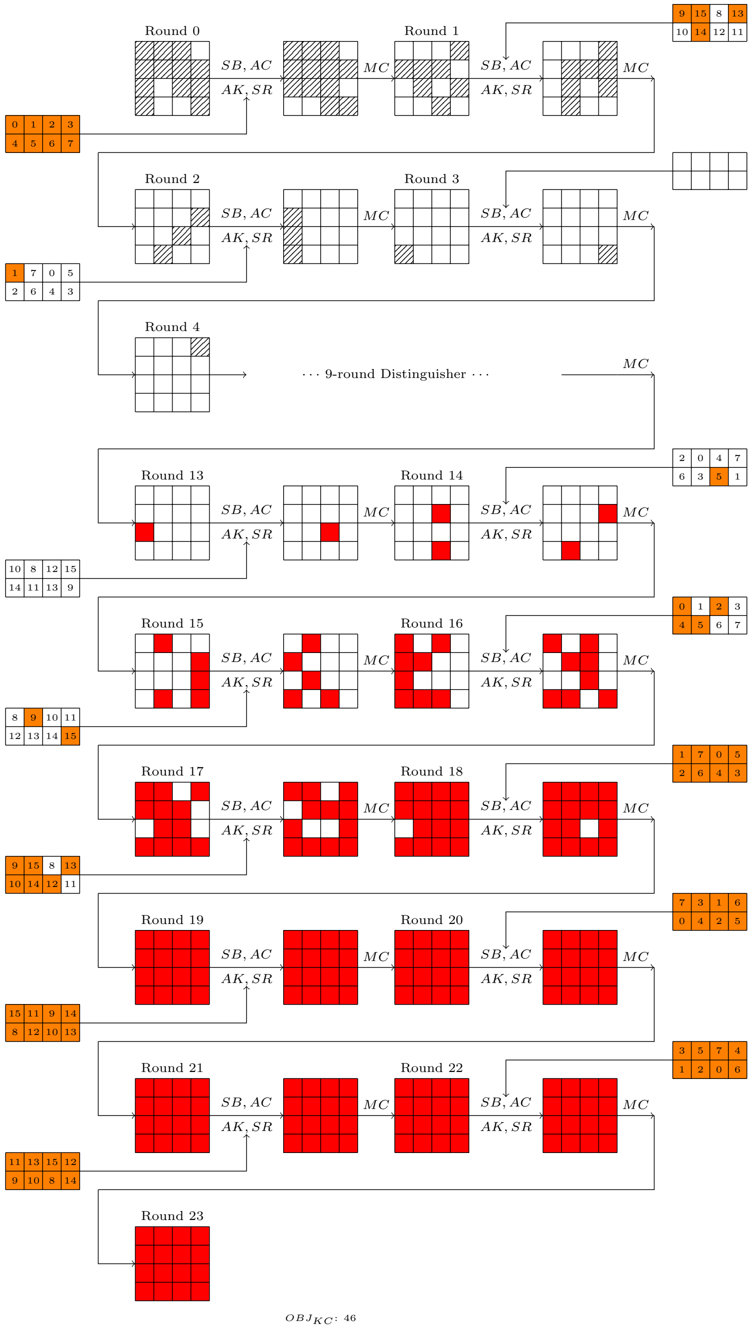


Fig. 25. 23-round Attack on SKINNY-128-384

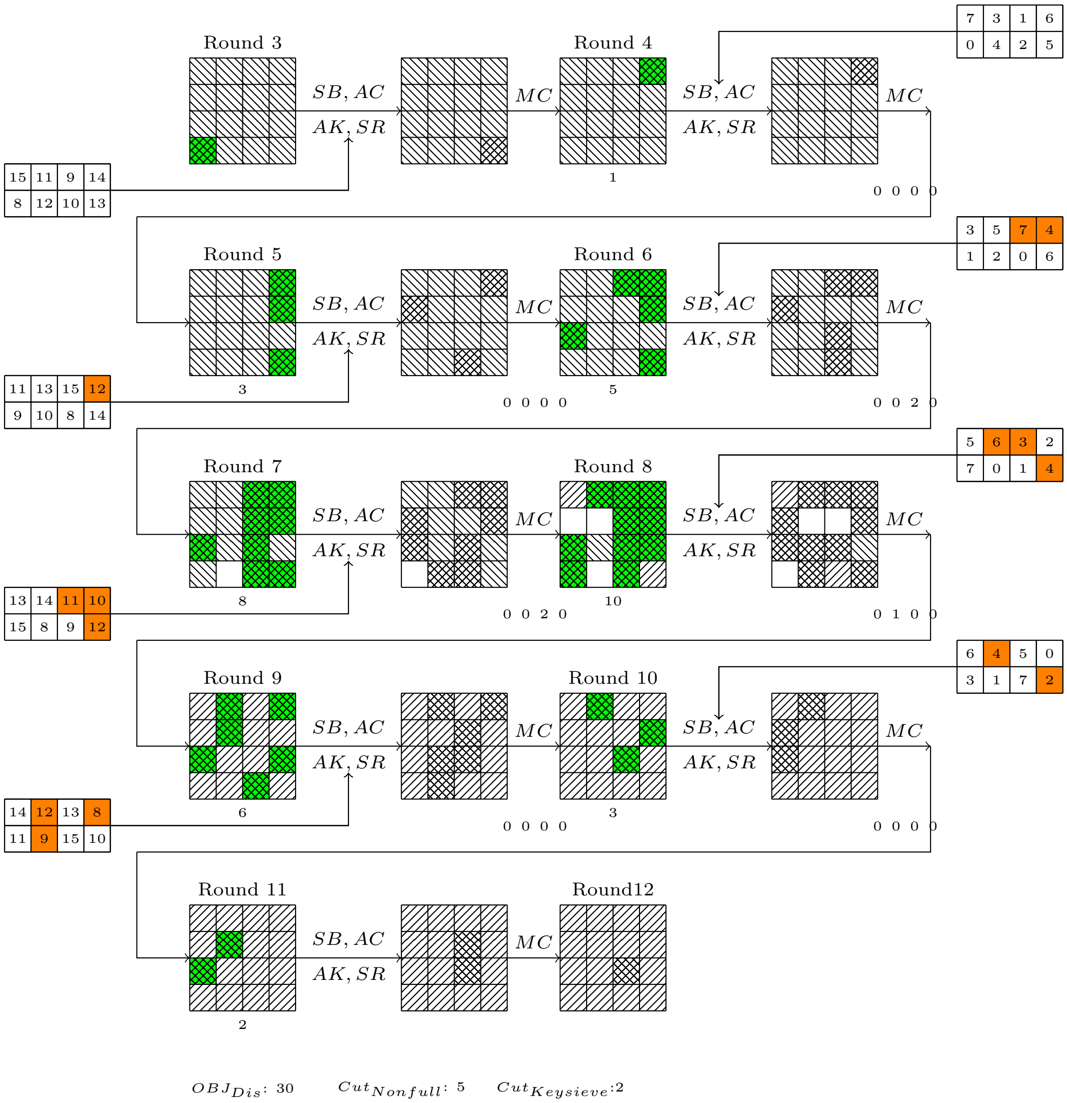


Fig. 26. 9-round Distinguisher of SKINNY-128-256

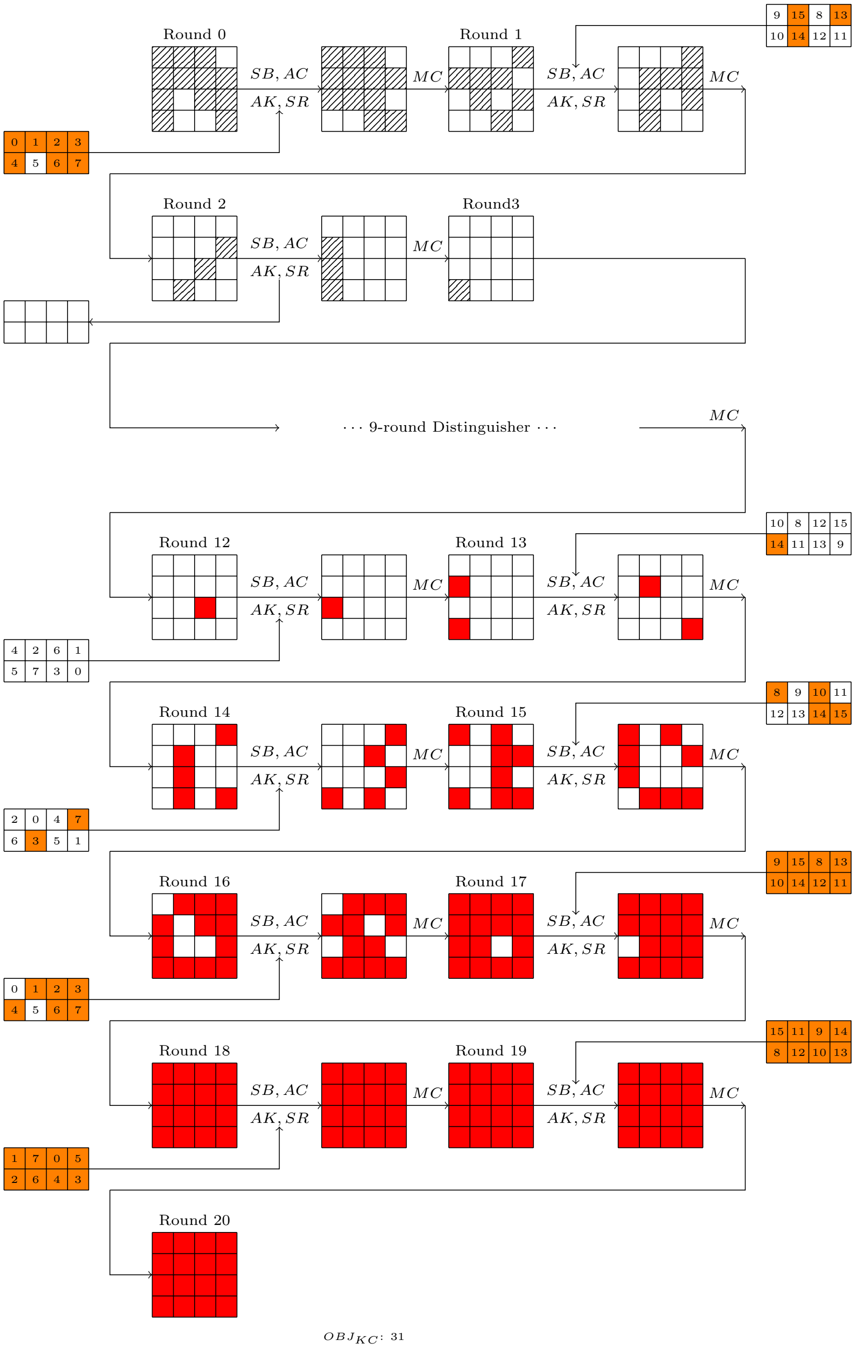


Fig. 27. 20-round Attack on SKINNY-128-256

2. *Precomputation Phase* (Fig. 26). Construct a $\delta(\mathcal{A})$-set with $|\delta(\mathcal{A})| = 2^8$, the output difference multiset at $\mathbf{S}_{12}[10]$ can be uniquely determined by 37 internal cells (🟩). Among the 37 internal parameters, 5 cells can be reduced by utilizing the non-full key-addition technique (marked by number under the operation MC). 2 cells can be reduced by relations on round-key generated by $\{TKz[4], TKz[12] : z \in \{1,2\}\}$. Thus the output difference multiset can be fully determined by 30 internal cells, which can be used to distinguish from a random 255-byte multiset. The time complexity for constructing a lookup table to save all possible values that output difference may take is $2^8 \cdot 2^{8\cdot 30} \cdot \frac{37}{16\cdot 20} \approx 2^{244.89}$. The memory complexity is $255 \cdot 8 \cdot 2^{8\cdot 30} \approx 2^{250.99}$.
3. *Online phase* (Fig. 27). All round-key marked by 🟧 are guessed to construct a plaintext structure that identifies a $\delta(\mathcal{A})$-set and get the value of the output multiset at $\mathbf{S}_{12}[10]$ by partially decrypting the plaintext structure. Among all these 32 cells of master key, 1-cell guesses can be saved for $\{TKz[5] : z \in \{1,2\}\}$. Thus the time complexity is $2^8 \cdot 2^{8\cdot 31} \cdot \frac{97}{16\cdot 20} \approx 2^{254.28}$. The data complexity is 2^{96}.

I 19-round DS-MITM Attack on SKINNY-128-256(256-bit key, 0-bit tweak)

1. $\mathcal{A} = [\mathbf{S}_4[0]], \mathcal{B} = [\mathbf{S}_{11}[13]]$.
2. *Precomputation Phase* (Fig. 28). Construct a $\delta(\mathcal{A})$-set with $|\delta(\mathcal{A})| = 27$, the output difference sequence at $\mathbf{S}_{11}[13]$ can be uniquely determined by 27 internal cells (🟩). Among the 27 internal parameters, 2 cells can be reduced by utilizing the non-full key-addition technique (marked by number under the operation MC). Thus the output difference sequence can be fully determined by 25 internal cells, which can be used to distinguish from a random 26-byte sequence. The time complexity for building a lookup table to save all possible values that the output sequence may take is $27 \cdot 2^{8\cdot 25} \cdot \frac{27}{16\cdot 19} \approx 2^{201.26}$. The memory complexity is $26 \cdot 8 \cdot 2^{8\cdot 29} \approx 2^{207.7}$.
3. *Online phase* (Fig. 29). All round-key marked by 🟧 are guessed to construct a plaintext structure that identifies a $\delta(\mathcal{A})$-set and get the value of the output difference at $\mathbf{S}_{11}[13]$ by partially decrypting the plaintext structure. Among all these 32 cells of master key, 2-cell guesses can be saved for $\{TKz[j] : z \in \{1,2\}, j \in [10,11,13]\}$ by utilizing the key-bridging technique. Thus the time complexity is $27 \cdot 2^{8\cdot 29} \cdot \frac{85}{16\cdot 19} \approx 2^{235.05}$. The data complexity is 2^{96}.

J 21-round DS-MITM Attack on SKINNY-64-192(192-bit key, 0-bit tweak)

1. $\mathcal{A} = [\mathbf{S}_4[0]], \mathcal{B} = \mathbf{S}_{11}[13]$.
2. *Precomputation Phase* (Fig. 30). Construct a $\delta(\mathcal{A})$-set with $|\delta(\mathcal{A})| = 36$, the output difference sequence at $\mathbf{S}_{12}[13]$ can be uniquely determined by 41 internal cells (🟩). Among the 41 internal parameters, 7 cells can be reduced

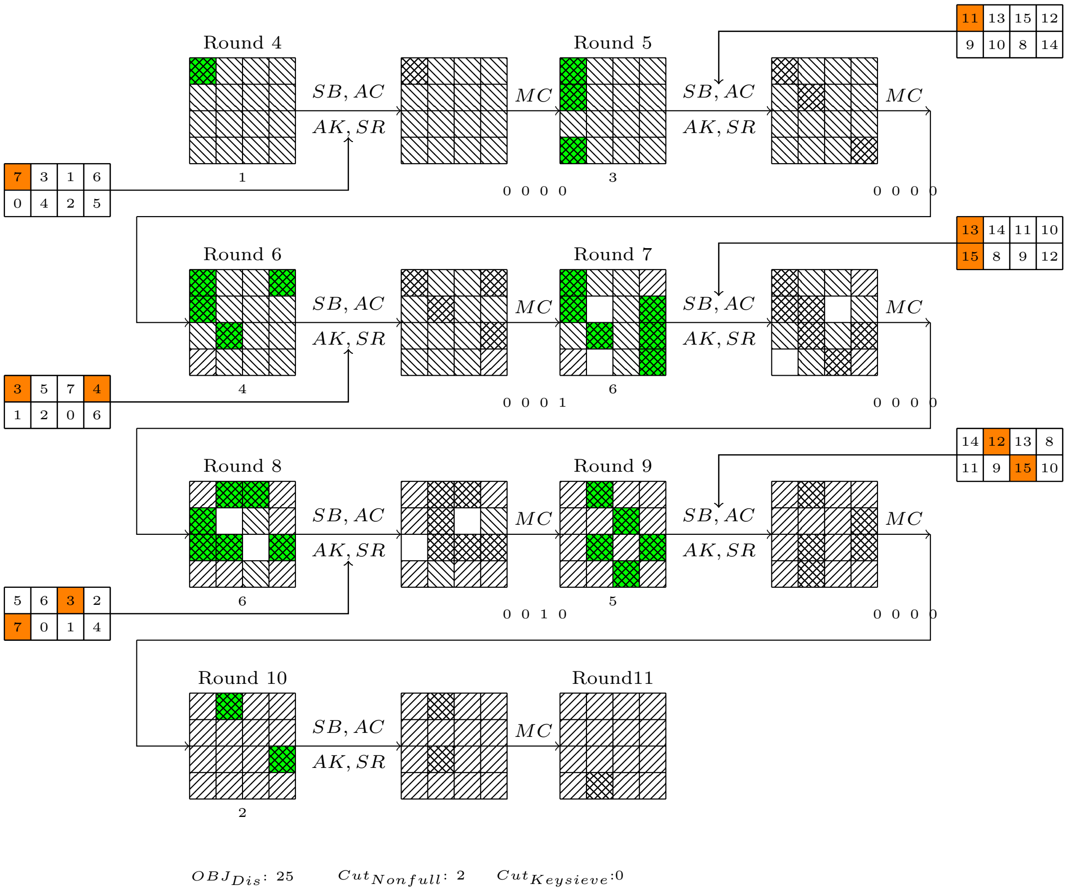


Fig. 28. 7-round Distinguisher of SKINNY-128-256

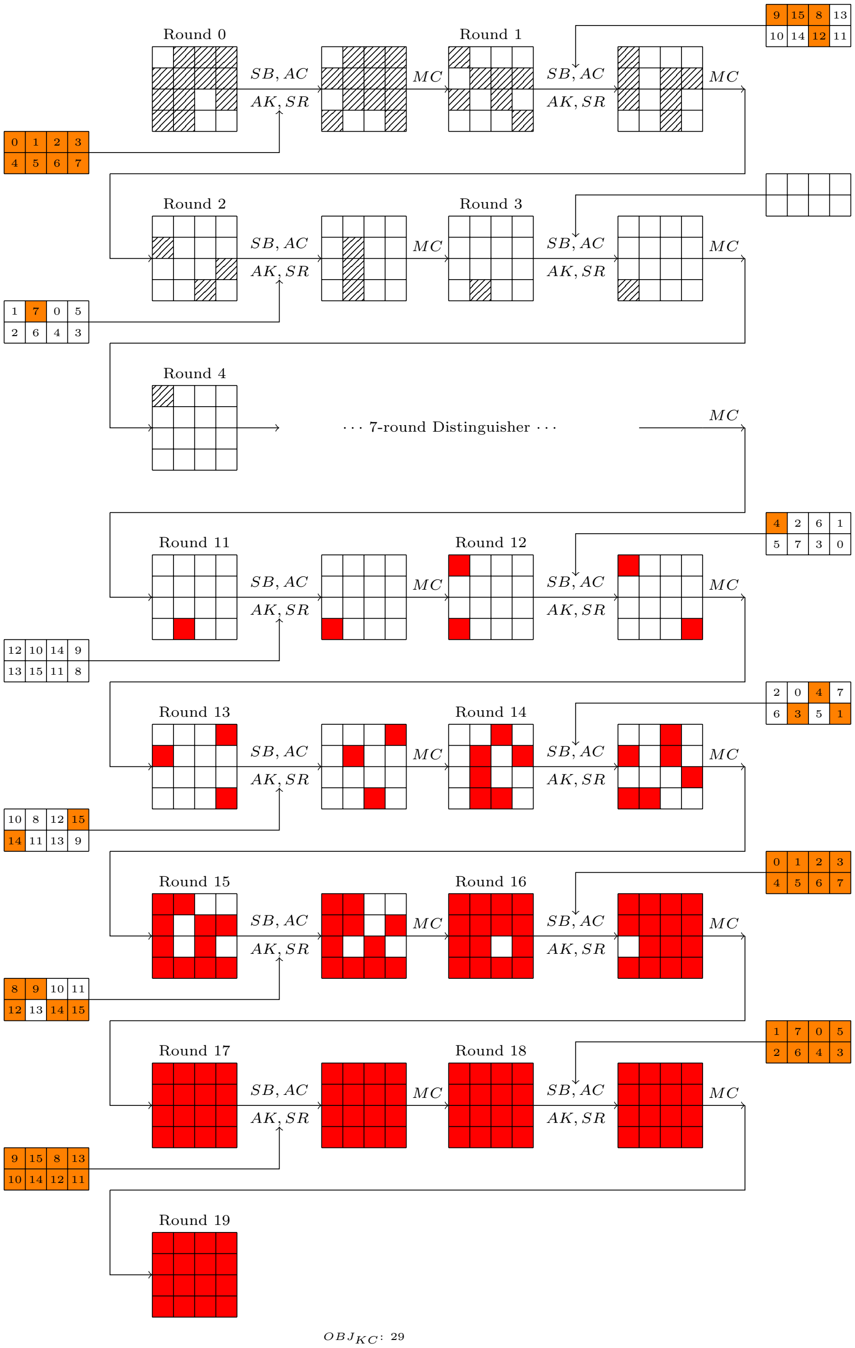
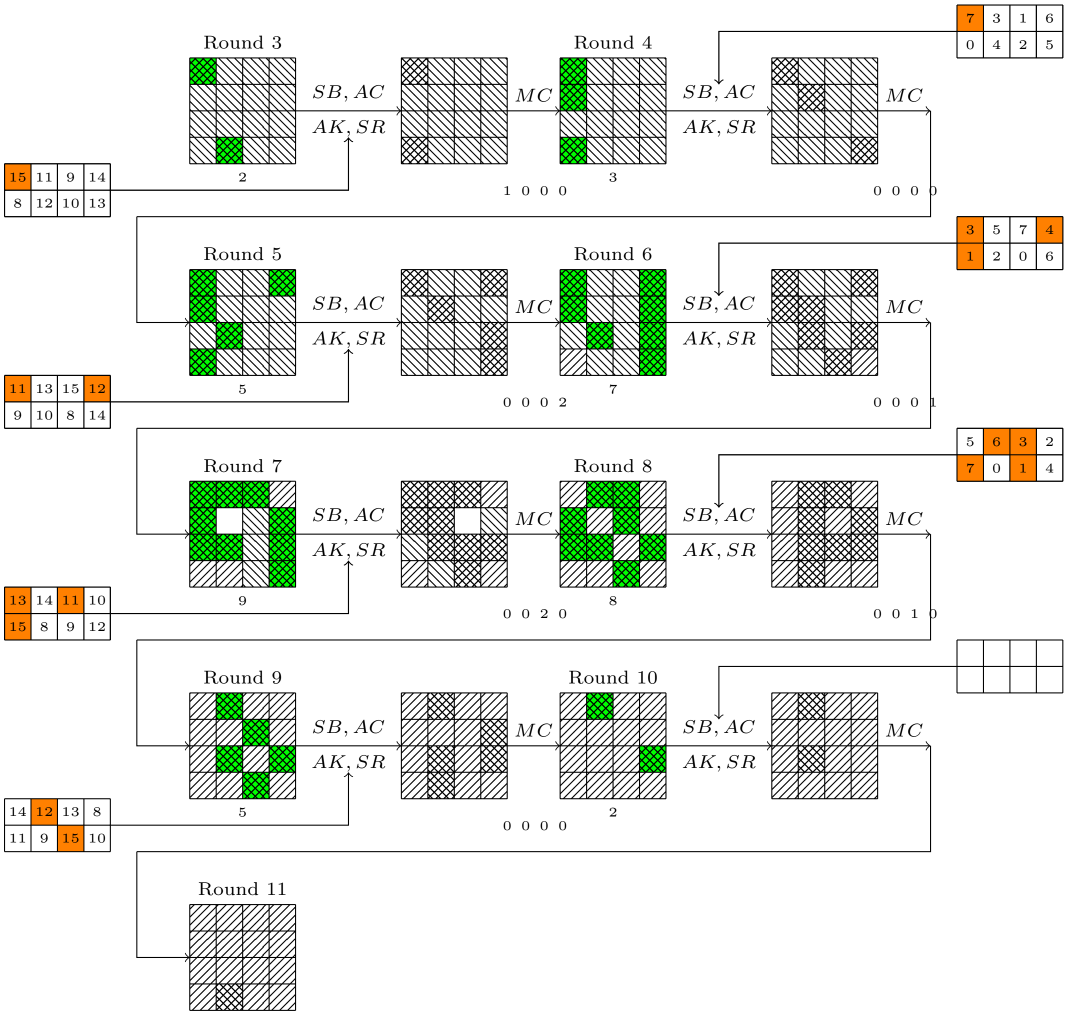


Fig. 29. 19-round Attack on SKINNY-128-256



OBJ_{Dis}: 34 $Cut_{Nonfull}$: 7 $Cut_{Keysieve}$:0

Fig. 30. 9-round Distinguisher of SKINNY-64-192

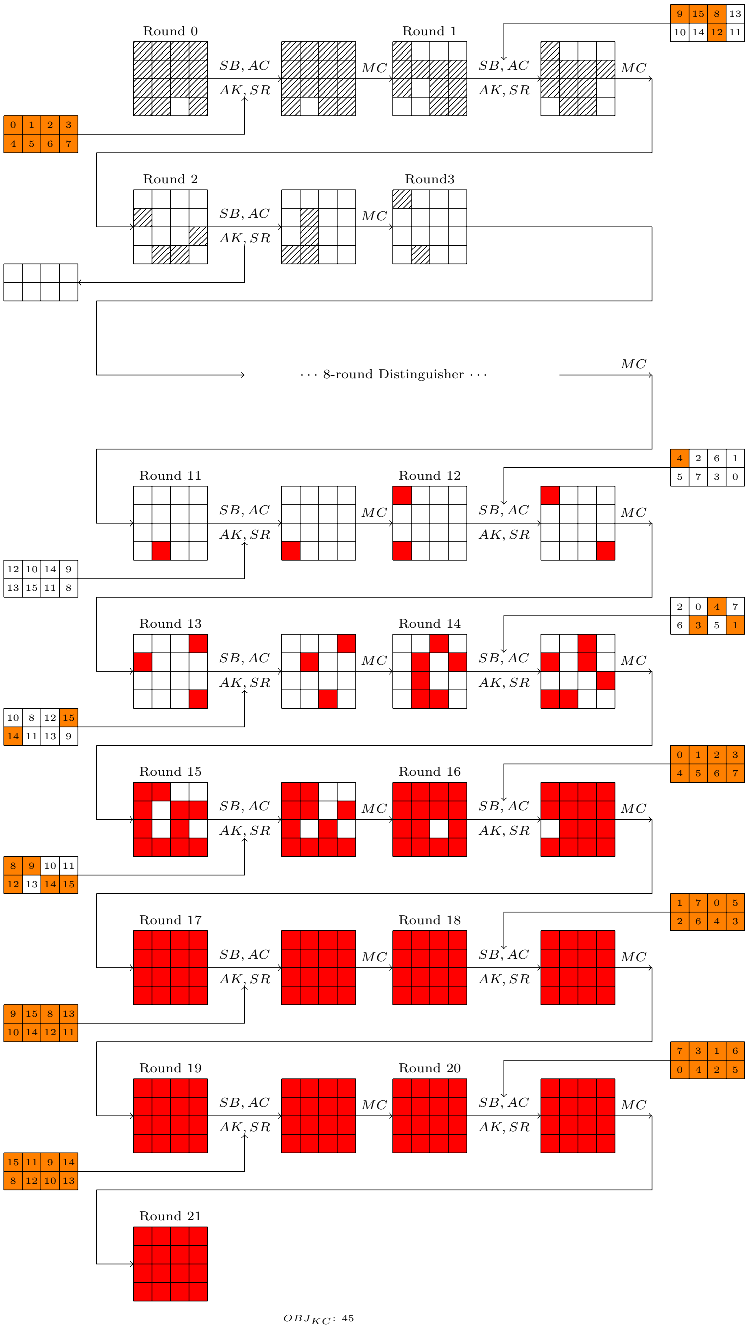


Fig. 31. 21-round Attack on SKINNY-64-192

by utilizing the non-full key-addition technique (marked by number under the operation MC). Thus the output difference sequence can be fully determined by 34 internal cells, which can be used to distinguish from a random 35-cell sequence. The time complexity for building a lookup table to save all possible values that the output sequence may take is $36 \cdot 2^{4 \cdot 34} \cdot \frac{41}{16 \cdot 21} \approx 2^{138.14}$. The memory complexity is $35 \cdot 4 \cdot 2^{4 \cdot 34} \approx 2^{143.13}$.

3. *Online phase* (Fig. 31). All round-key marked by 🟧 are guessed to construct a plaintext structure that identifies a $\delta(\mathcal{A})$-set and get the output difference at $\mathbf{S}_{11}[13]$ by partially decrypting the plaintext structure. Among all these 48 cells of master key, 3-cell guesses can be saved from $\{TKz[j] : z \in \{1,2,3\}, j \in [10,11,13]\}$ by utilizing the key-bridging technique. Thus the time complexity is $36 \cdot 2^{4 \cdot 45} \cdot \frac{131}{16 \cdot 21} \approx 2^{183.81}$. The data complexity is 2^{60}.

K 21-round DS-MITM Attack on SKINNY-64-192(192-bit key, 0-bit tweak)

1. $\mathcal{A} = [\mathbf{S}_3[3], \mathbf{S}_3[4]], \mathcal{B} = [\mathbf{S}_{12}[9]]$.
2. *Precomputation Phase* (Fig. 32). Construct a $\delta(\mathcal{A})$-set with $|\delta(\mathcal{A})| = 48$, the output difference sequence at $\mathbf{S}_{12}[9]$ can be uniquely determined by 58 internal cells (🟩). Among the 58 internal parameters, 11 cells can be reduced by utilizing the non-full key-addition technique (marked by number under the operation MC). 1 cells can be reduced by relation on round-key generated by $\{TKz[6] : z \in \{1,2,3\}\}$. Thus the output difference sequence can be fully determined by 46 internal cells, which can be used to distinguish from a random 47-cell sequence. The time complexity for constructing a hash table to save all possible values that the output difference may take is $48 \cdot 2^{4 \cdot 46} \cdot \frac{58}{16 \cdot 21} \approx 2^{187.05}$. The memory complexity is $47 \cdot 4 \cdot 2^{4 \cdot 46} \approx 2^{191.55}$.
3. *Online phase* (Fig. 33). All round-key marked by 🟧 are guessed to construct a plaintext structure that identifies a $\delta(\mathcal{A})$-set and get the output difference at $\mathbf{S}_{12}[9]$ by partially decrypting the plaintext structure. Among all these 48 cells of master key, 4-cell guesses can be saved for $\{TKz : z \in \{1,2,3\}, j \in [8,10,12,15]\}$ by utilizing the key-bridging technique. Thus the time complexity is $48 \cdot 2^{4 \cdot 44} \cdot \frac{113}{16 \cdot 21} \approx 2^{180.01}$. The data complexity is 2^{44}.

L 17-round DS-MITM Attack on SKINNY-128-128(128-bit key, 0-bit tweak)

1. $\mathcal{A} = [\mathbf{S}_4[7]], \mathcal{B} = [\mathbf{S}_{11}[11]]$.
2. *Precomputation Phase* (Fig. 34). Construct a $\delta(\mathcal{A})$-set with $|\delta(\mathcal{A})| = 16$, the output difference sequence at $\mathbf{S}_{11}[11]$ can be uniquely determined by 15 internal cells (🟩). Among the 15 internal parameters, 1 cells can be reduced from round-key generated by $\{TK1[11]\}$. Thus the output difference sequence can be fully determined by 14 internal cells, which can be used to distinguish from a random 15-byte sequence. The time complexity for constructing a

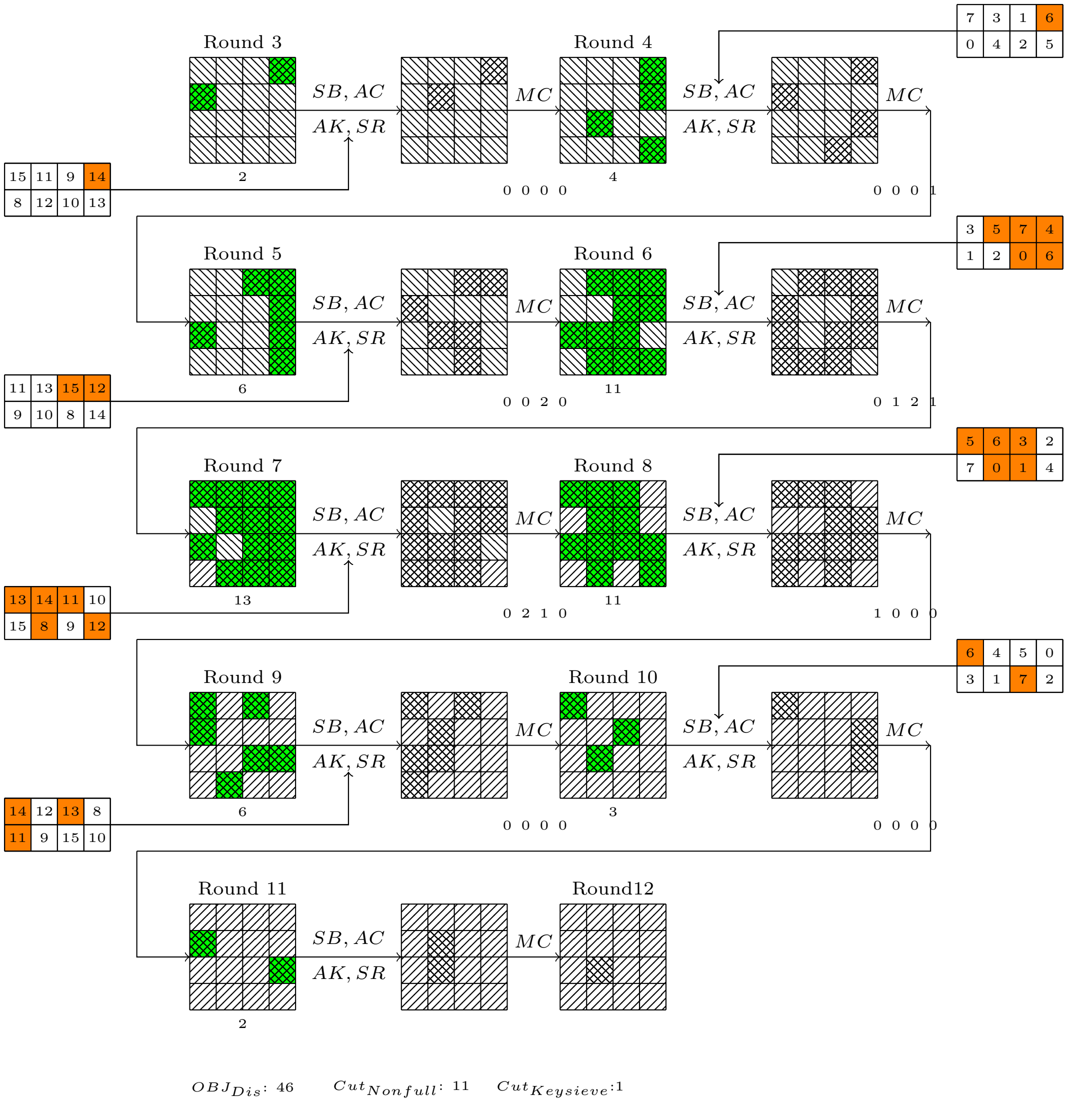


Fig. 32. 9-round Distinguisher of SKINNY-64-192

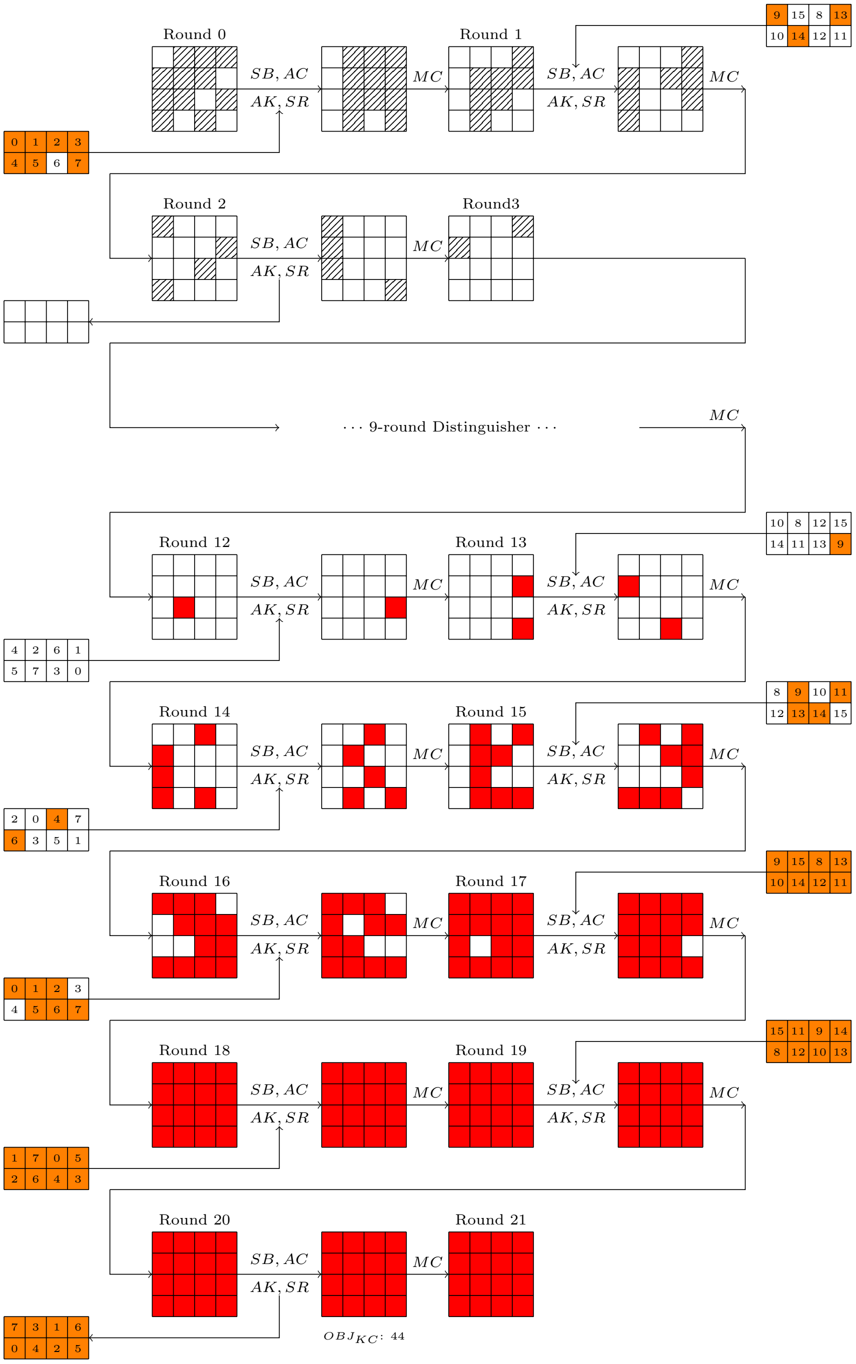


Fig. 33. 21-round Attack on SKINNY-64-192

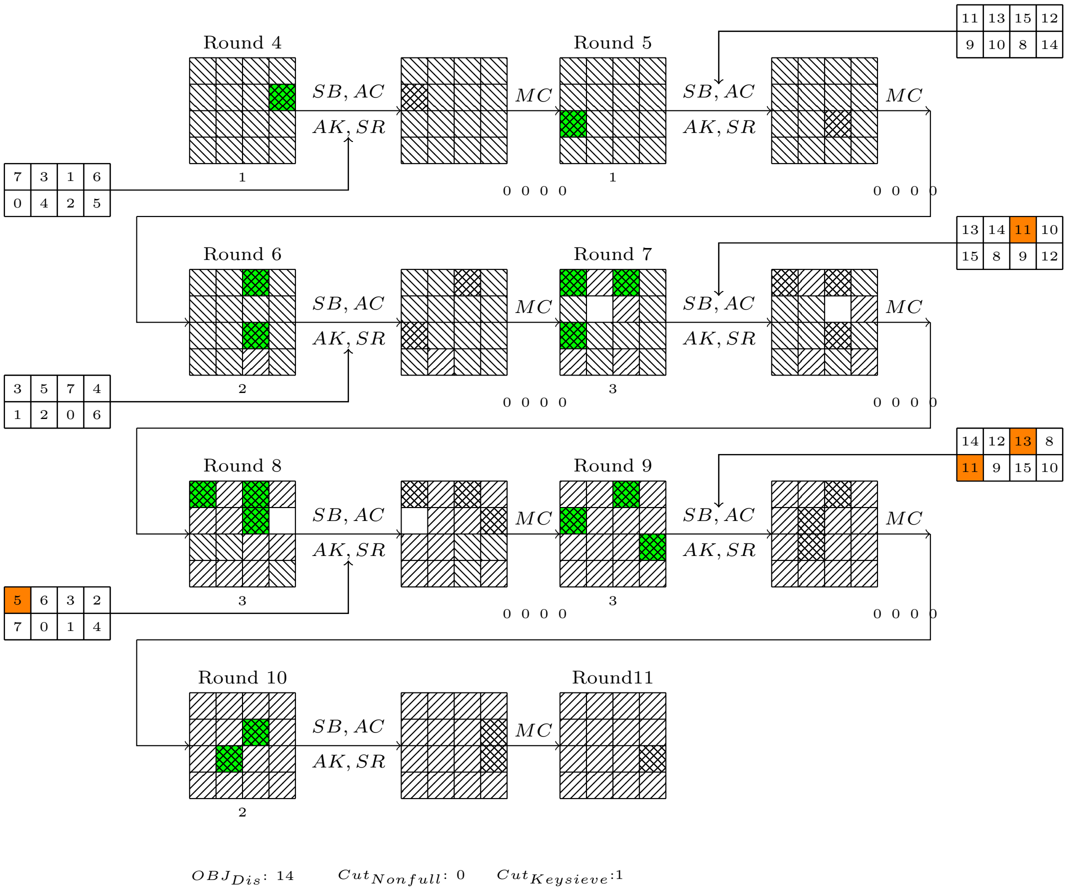


Fig. 34. 7-round Distinguisher of SKINNY-128-128

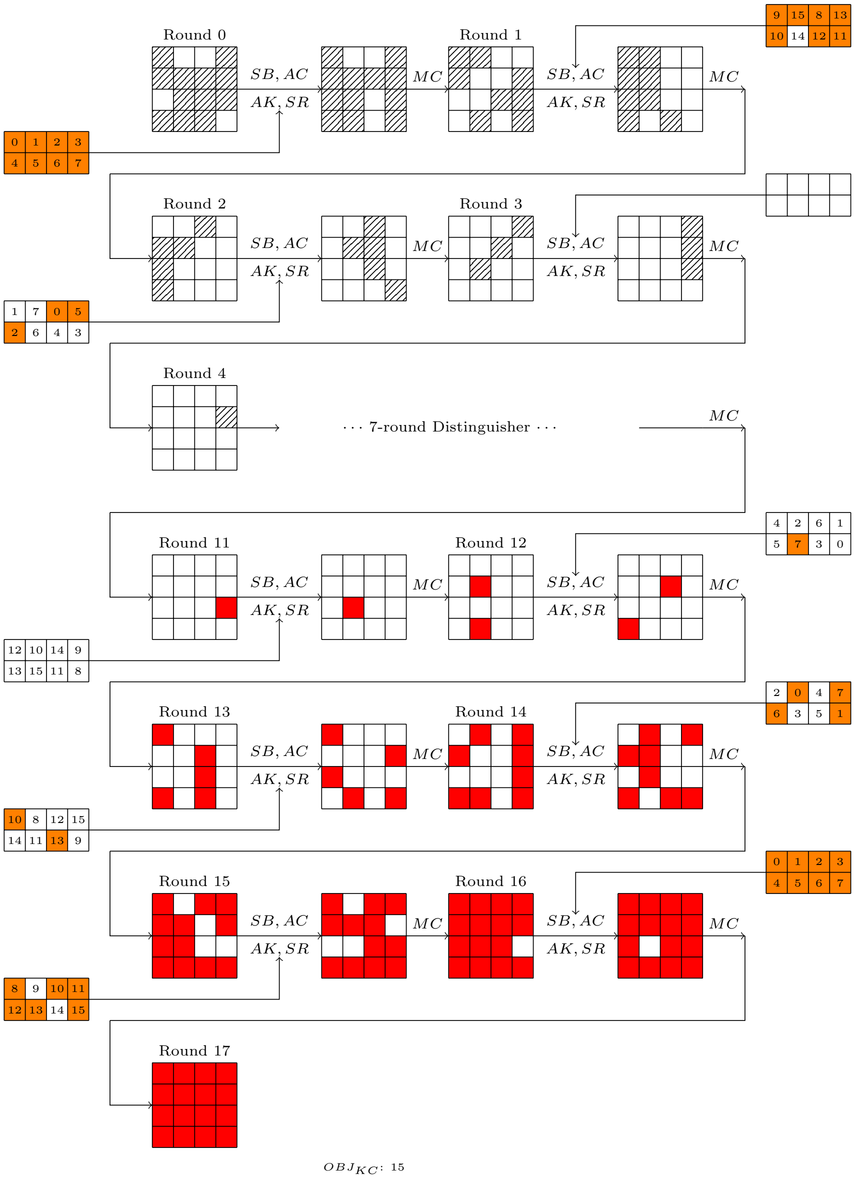


Fig. 35. 17-round Attack on SKINNY-128-128

hash table to save all possible values that the output difference may take is $16 \cdot 2^{8\cdot 14} \cdot \frac{15}{16\cdot 17} \approx 2^{111.82}$. The memory complexity is $15 \cdot 8 \cdot 2^{8\cdot 14} \approx 2^{118.91}$.

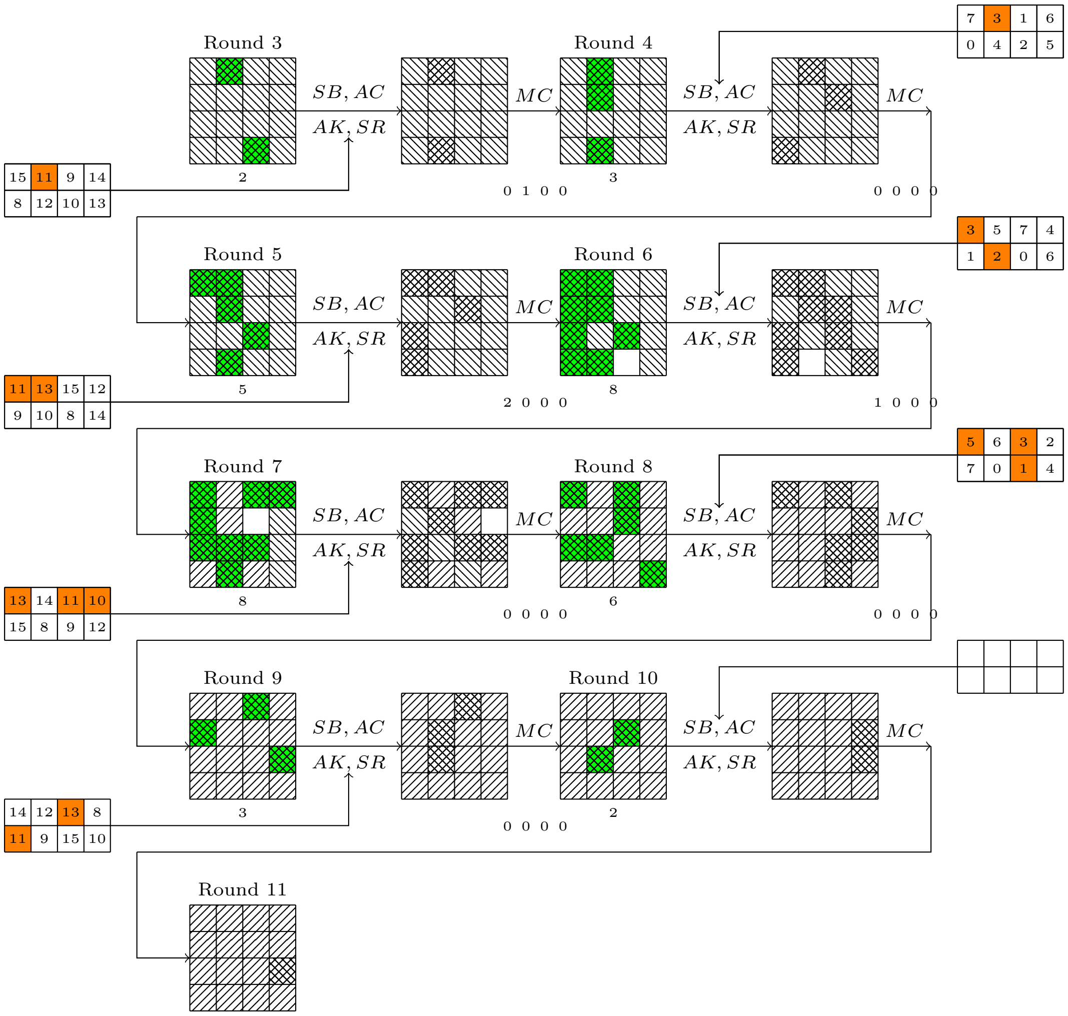
3. *Online phase* (Fig. 35). All round-key marked by 🟧 are guessed to construct a plaintext structure that identifies a $\delta(\mathcal{A})$-set and get the value of output difference at $\mathbf{S}_{11}[11]$ by partially decrypting the plaintext structure. Among all these 16 cells of master key, 1-cell guesses can be saved for $\{TK1[14]\}$. Thus the time complexity is $16 \cdot 2^{8\cdot 15} \cdot \frac{71}{16\cdot 17} \approx 2^{122.06}$. The data complexity is 2^{96}.

M 19-round DS-MITM attack on SKINNY-64-128(128-bit key, 0-bit tweak)

1. $\mathcal{A} = [\mathbf{S}_3[1], \mathbf{S}_3[14]], \mathcal{B} = [\mathbf{S}_{11}[11]]$.
2. Precomputation Phase (Fig. 36). Construct a $\delta(\mathcal{A})$-set with $|\delta(\mathcal{A})| = 31$, the output difference sequence can be uniquely determined by 37 internal cells (🟩). Among the 37 internal parameters, 4 cells can be reduced by utilizing the non-full key-addition technique (marked by number under the operation MC). 4 cells can be reduced by relations on round-key generated by $\{TKz[j] : z \in \{1,2\}, j \in [3,11,13]\}$. Thus the output difference sequence can be fully determined by 29 internal cells, which can be used to distinguish from a random 30-cell sequence. The time complexity for building a lookup table to save all possible values that the output sequence may take is $31 \cdot 2^{4\cdot 29} \cdot \frac{37}{16\cdot 19} \approx 2^{117.92}$. The memory complexity is $30 \cdot 4 \cdot 2^{29\cdot 4} \approx 2^{122.91}$.
3. Online phase (Fig. 37). All round-key marked by 🟧 are guessed to construct a plaintext structure that identifies a $\delta(\mathcal{A})$-set at $\mathbf{S}_4$ and get the output difference at $\mathbf{S}_{11}[11]$ by partially decrypting the plaintext structure. Among all these 32-cell master key, 2-cell guesses can be saved for $\{TKz[j] : z \in \{1,2\}, j \in [9,14]\}$. Thus the time complexity is $31 \cdot 2^{4\cdot 30} \cdot \frac{104}{16\cdot 19} \approx 2^{123.41}$. The data complexity is 2^{60}.

N 19-round DS-MITM attack on SKINNY-64-128(128-bit key, 0-bit tweak)

1. $\mathcal{A} = [\mathbf{S}_4[1], \mathbf{S}_4[11]], \mathcal{B} = [\mathbf{S}_{11}[11]]$.
2. Precomputation Phase (Fig. 38). Construct a $\delta(\mathcal{A})$-set with $|\delta(\mathcal{A})| = 32$, the output difference sequence can be uniquely determined by 35 internal cells (🟩). Among the 35 internal parameters, 3 cells can be reduced by utilizing the non-full key-addition technique (marked by number under the operation MC) and 2 cells can be reduced by relations on round-keys generated by $\{TKz[3], TKz[13] : z \in \{1,2\}\}$. Thus the output difference sequence can be fully determined by 30 internal cells, which can be used to distinguish from a random 31-cell sequence. The time complexity for building a lookup table to save all possible values that the output sequence may take is $32 \cdot 2^{4\cdot 30} \cdot \frac{35}{16\cdot 19} \approx 2^{121.88}$. The memory complexity is $31 \cdot 4 \cdot 2^{30\cdot 4} \approx 2^{126.95}$.



OBJ_{Dis}: 29 $Cut_{Nonfull}$: 4 $Cut_{Keysieve}$:4

Fig. 36. 8-round Distinguisher of SKINNY-64-128

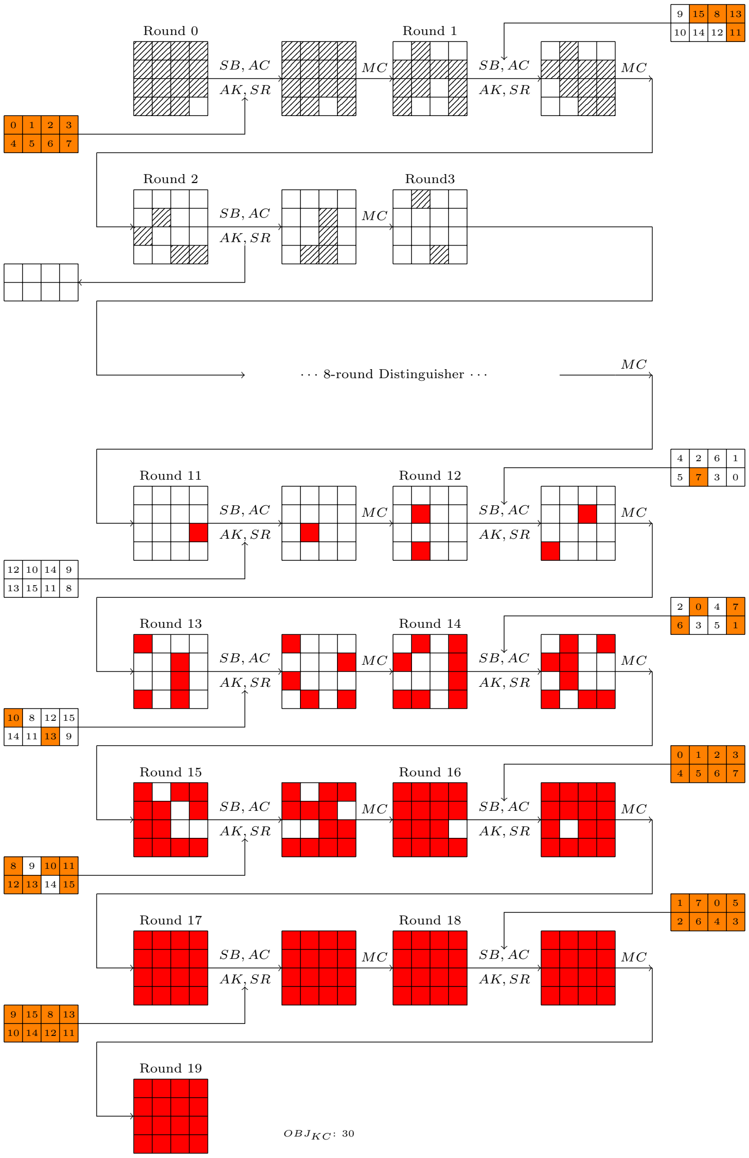


Fig. 37. 19-round Attack on SKINNY-64-128

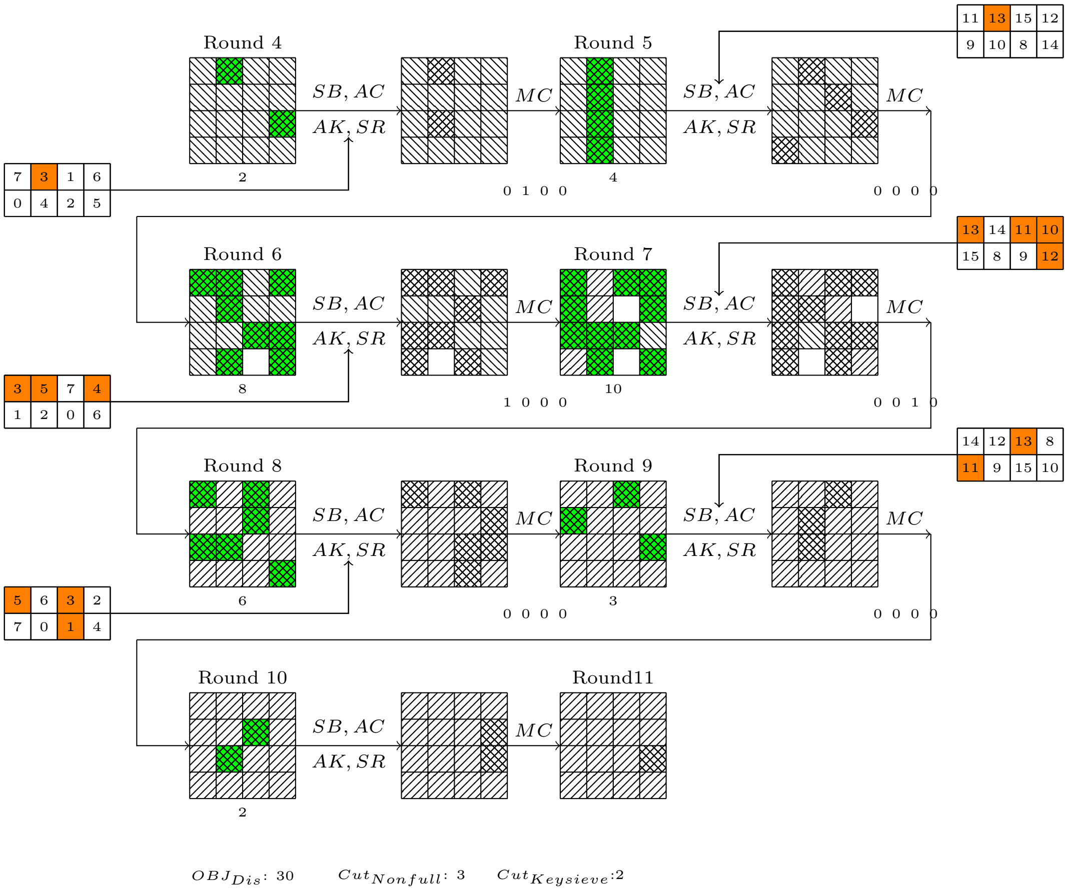


Fig. 38. 7-round Distinguisher of SKINNY-64-128

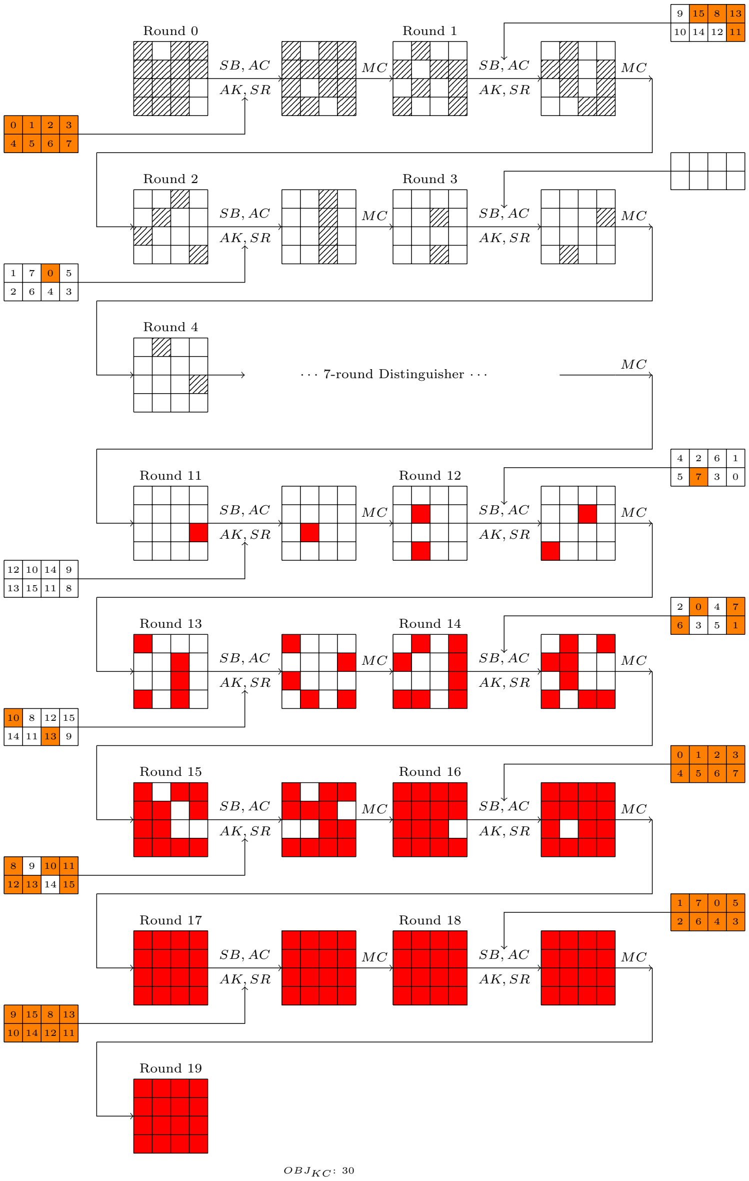


Fig. 39. 19-round Attack on SKINNY-64-128

3. Online phase (Fig. 39). All round-keys marked by 🟧 are guessed to construct a plaintext structure that identifies a $\delta(\mathcal{A})$-set and get the value of the output difference at $\mathbf{S}_{11}[11]$ by partially decrypting the plaintext structure. Among all these 32-cell master key, 2-cell guesses can be saved for $\{TKz[9], TKz[14] : z_1\{1,2\}\}$. Thus the time complexity is $32 \cdot 2^{4 \cdot 30} \cdot \frac{102}{16 \cdot 19} \approx 2^{123.43}$. The data complexity is 2^{52}.

O 17-round DS-MITM attack on SKINNY-64-64

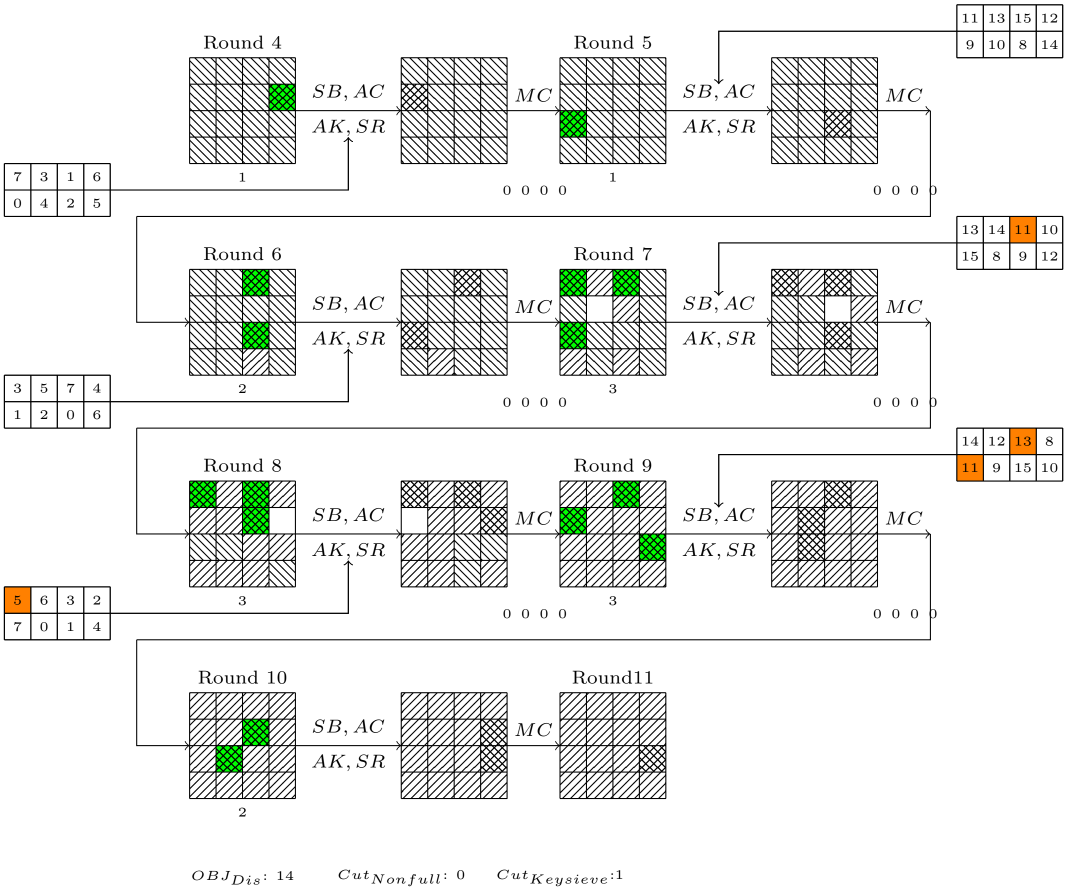


Fig. 40. 7-round Distinguisher of SKINNY-64-64

1. $\mathcal{A} = [\mathbf{S}_4[7]], \mathcal{B} = [\mathbf{S}_{11}[11]]$.

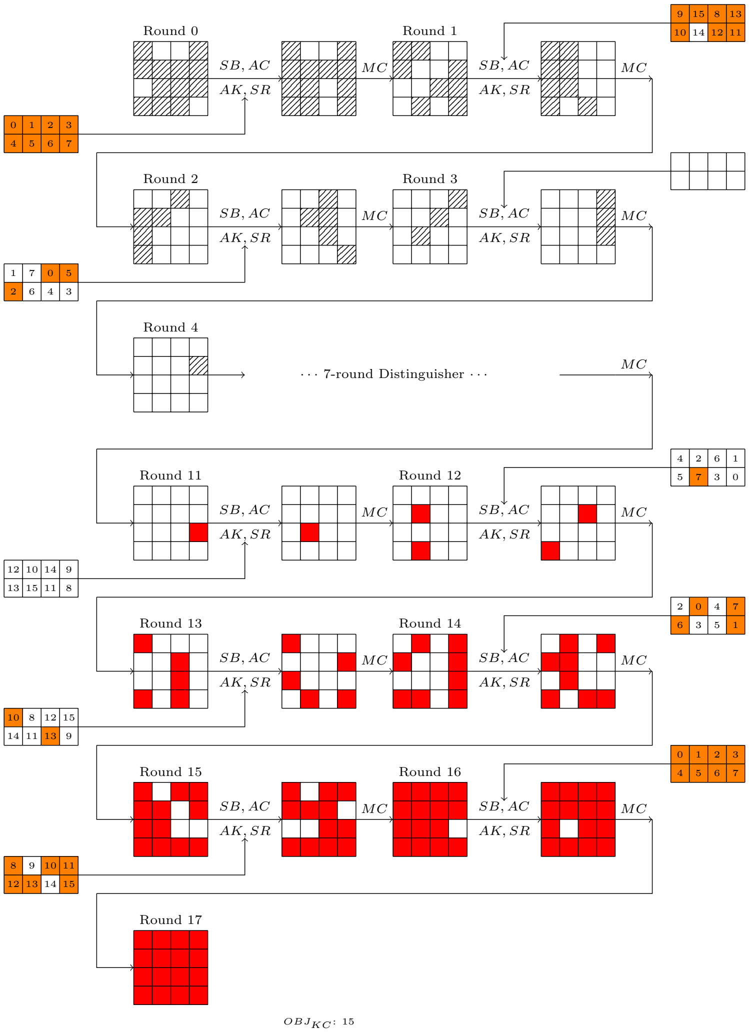


Fig. 41. 17-round Attack on SKINNY-64-64

2. Precomputation Phase (Fig. 40). Construct a $\delta(\mathcal{A})$-set with $|\delta(\mathcal{A})| = 16$, the output difference sequence can be uniquely determined by 15 internal cells (🟩). Among these 15 internal parameters, 1 cell can be reduced by relations on round-key generated from $TK1[11]$ by utilizing the key-sieve-dependent technique. Thus the output difference sequence can be fully determined by 14 internal cells, which can be used to distinguish from a random 15-cell sequence. The time complexity for building a lookup table to save all possible values that the output sequence may take is $16 \cdot 2^{14 \cdot 4} \cdot \frac{15}{16 \cdot 17} \approx 2^{55.82}$. The memory complexity is $15 \cdot 4 \cdot 2^{14 \cdot 4} \approx 2^{61.91}$.
3. Online phase (Fig. 41). All round-key marked by 🟧 are guessed to construct a plaintext structure that identifies a $\delta(\mathcal{A})$-set and get the value of the output difference at $\mathbf{S}_{11}[11]$ by partially decrypting the plaintext structure. Among all these 16-cell master key, 1-cell guesses can be saved for $\{TK1[14]\}$ by utilizing the key-bridging technique. Thus the time complexity is $16 \cdot 2^{15 \cdot 4} \cdot \frac{80}{16 \cdot 17} \approx 2^{62.06}$. The data complexity is 2^{48}.

P 13-round Distinguisher on SKINNY-128-384

1. Assume (P^0, P') conforms to the truncated differential trail described by ▣ (Fig. 42). Let $\mathcal{A} = [\mathbf{S}_0[13]], \mathcal{B} = [\mathbf{S}_{12}[5] \oplus \mathbf{S}_{12}[9] \oplus \mathbf{S}_{12}[13]]$. Construct a $\delta(\mathcal{A})$-set from P^0 with $|\delta(\mathcal{A})| = 49$. The output difference sequence can be uniquely determined by 77 internal parameters ($P \oplus P'[\mathbf{S}_1[3]], P \oplus P'[\mathbf{S}_{13}[9]], P \oplus P'[\mathbf{S}_{13}[14]]$, 🟥 and 🟩 of $\mathbf{S}_r, r \neq 6$) from proposition 3 with $R_M = 6$. 26 cells can be reduced by utilizing the non-full key-addition technique (listed below of operation MC), and 4 cells can be reduced by relations on round-keys generated by $\{TKz[j] : j \in [6, 9, 12, 14]\}$. Thus the output difference sequence can be uniquely determined by 47 internal parameters, which can be used to distinguish from a random 48-byte sequence. The time complexity for building a lookup table to save all possible values that the output sequence may take is $49 \cdot 2^{47 \cdot 8} \cdot \frac{93}{16 \cdot 12} \approx 2^{380.57}$. The memory complexity is $48 \cdot 8 \cdot 2^{47 \cdot 8} \approx 2^{384.59}$.

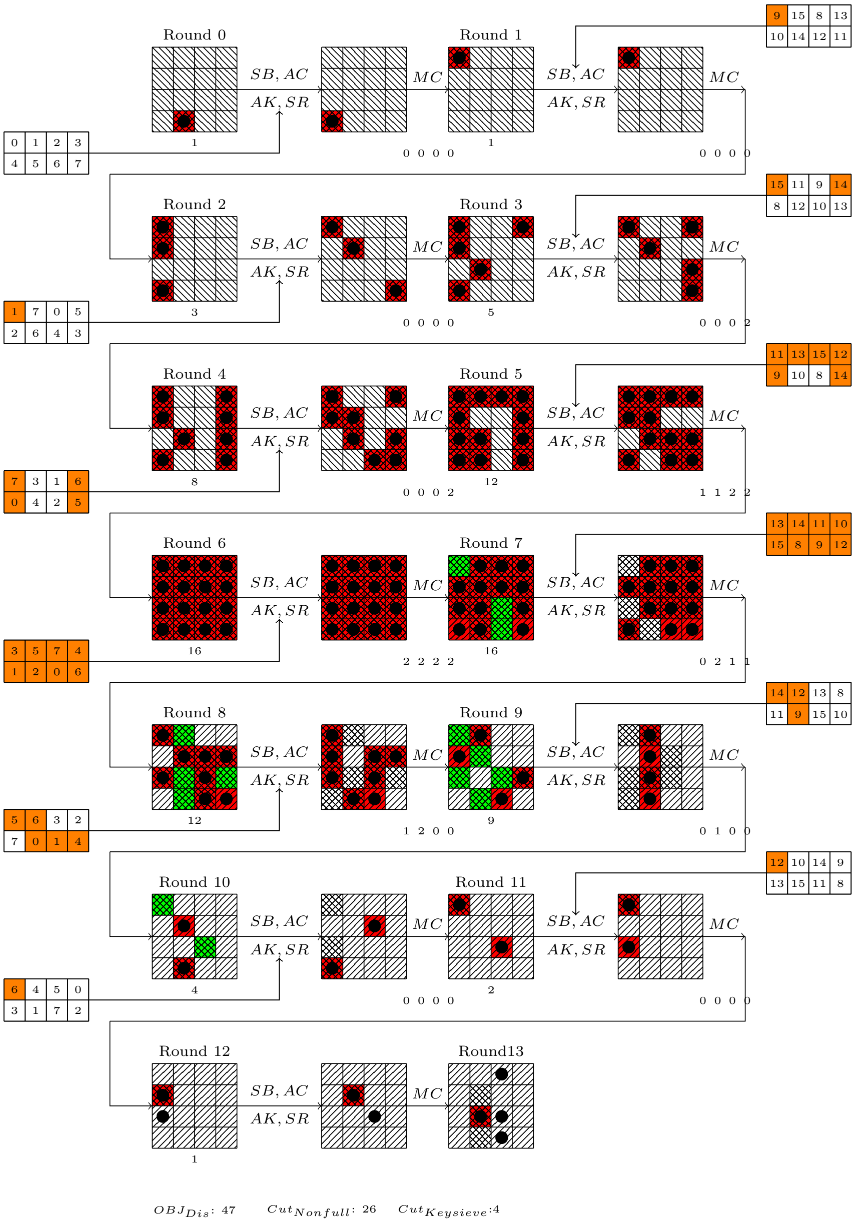


Fig. 42. 13-round Distinguisher on SKINNY-128-384