Programming the Demirci-Selçuk Meet-in-the-Middle Attack with Constraints

Danping Shi, Siwei Sun, Patrick Derbez, Yosuke Todo, Bing Sun, Lei Hu

Programming the Demirci-Selçuk Meet-in-the-Middle Attack with Constraints

Danping Shi[1,2], Siwei Sun[1,2,3⋆], Patrick Derbez[4], Yosuke Todo[5]
Bing Sun[6], and Lei Hu[1,2,3]

[1] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, China
[2] Data Assurance and Communication Security Research Center, Chinese Academy of Sciences, China
[3] School of Cyber Security, University of Chinese Academy of Sciences, China
{shidanping, sunsiwei, hulei}@iie.ac.cn
[4] Univ Rennes, CNRS, IRISA, France patrick.derbez@irisa.fr
[5] NTT Secure Platform Laboratories, Japan todo.yosuke@lab.ntt.co.jp
[6] College of Liberal Arts and Sciences, National University of Defense Technology, China happy_come@163.com

Abstract. Cryptanalysis with SAT/SMT, MILP and CP has increased in popularity among symmetric-key cryptanalysts and designers due to its high degree of automation. So far, this approach covers differential, linear, impossible differential, zero-correlation, and integral cryptanalysis. However, the Demirci-Selçuk meet-in-the-middle ($\mathcal{DS}$-MITM) attack is one of the most sophisticated techniques that has not been automated with this approach. By an in-depth study of Derbez and Fouque's work on $\mathcal{DS}$-MITM analysis with dedicated search algorithms, we identify the crux of the problem and present a method for automatic $\mathcal{DS}$-MITM attack based on general constraint programming, which allows the cryptanalysts to state the problem at a high level without having to say how it should be solved. Our method is not only able to enumerate distinguishers but can also partly automate the key-recovery process. This approach makes the $\mathcal{DS}$-MITM cryptanalysis more straightforward and easier to follow, since the resolution of the problem is delegated to off-the-shelf constraint solvers and therefore decoupled from its formulation. We apply the method to SKINNY, TWINE, and LBlock, and we get the currently known best $\mathcal{DS}$-MITM attacks on these ciphers. Moreover, to demonstrate the usefulness of our tool for the block cipher designers, we exhaustively evaluate the security of $8! = 40320$ versions of LBlock instantiated with different words permutations in the F functions. It turns out that the permutation used in the original LBlock is one of the 64 permutations showing the strongest resistance against the $\mathcal{DS}$-MITM attack. The whole process is accomplished on a PC in less than 2 hours. The same process is applied to TWINE, and similar results are obtained.

Keywords: Demirci-Selçuk meet-in-the-middle attack, Automated cryptanalysis, Constraint programming, MILP

⋆ The corresponding author

1 Introduction

Cryptanalysis of block ciphers is a highly technical, time consuming and error-prone process. On the one hand, the attackers have to perform a variety of cryptanalytic techniques, including differential attack [1], linear attack [2], integral attack [3–5], etc., to see which technique leads to the best attack. On the other hand, the designers need to repeat all these different attacks again and again to identify the optimal choices of parameters and building blocks which meet the security and implementation requirements. Therefore, automatic tools are indispensable to the community, which significantly reduce the manual work and make a thorough exploration of the design/analysis space possible.

One paradigm for automatic symmetric-key cryptanalysis getting increasing popularity in recent years is to model the problem by means of constraints, which includes the methods based on SAT/SMT (satisfiability modulo theory) [6–8], MILP (mixed-integer linear programming) [9–13], and classical constraint programming [14, 15]. In this paper, these methods are collectively referred to as the general constraint programming (CP) based approach, or just CP based approach for short. So far, the CP based approach covers a wide range of symmetric-key cryptanalysis techniques. For instance, we can determine the minimum number of differentially or linearly active S-boxes of a block cipher with MILP [9]; we can search for actual differential characteristics, linear characteristics, and integral distinguishers with SAT/SMT, MILP or classical constraint programming [8, 10, 11, 14]; and we can search for impossible differentials and zero-correlation linear approximations [12, 16] in a similar way.

Compared with search algorithms implemented from scratch in general purpose programming languages [17–24], the CP based approach allows the cryptanalysts to state the problem very naturally, and at a high level without having to say how it should be solved. The resolution of the problem is delegated to generic solvers, and therefore decoupled from the formulation of the problem. As Eugene C. Freuder stated [25]: *Constraint programming represents one of the closest approaches computer science has yet made to the Holy Grail of programming : the user states the problem, the computer solves it.*

However, the Demirci-Selçuk meet-in-the-middle attack ($\mathcal{DS}$-MITM) attack [26], introduced by Demirci and Selçuk at FSE 2008 to attack the famous Advanced Encryption Standard (AES) [27], is one of the cryptanalytic techniques which has not been automated with general constraint programming due to its extraordinary sophistication. After a series of improvements of the attack with various creative techniques [28–32], the $\mathcal{DS}$-MITM attack reaches the best known attack on 7-round AES-128, 9-round AES-256 and 10-round AES-256 in the single-key model. The attack has been applied to several specific block ciphers [33–36] as well as on generic balanced Feistel constructions [37]. Most recently, Guo et al. show generic attacks on unbalanced Feistel ciphers based on the $\mathcal{DS}$-MITM technique which penetrate a large number of rounds of some specific class of unbalanced Feistels [38]. Note that despite sharing the same name with the traditional MITM attacks in some literature (the attacks on some block

ciphers [39, 40] and on a number of hash functions, e.g. [41, 42]), the $\mathcal{DS}$-MITM attack concerned in this paper follows a different and a more complex strategy.

Related work and our contribution. In [30, 31], Derbez and Fouque presented a tool implemented in C/C++ for finding the $\mathcal{DS}$-MITM attack with dedicated search algorithm. In this paper, we present the first CP-based tool for finding the $\mathcal{DS}$-MITM attack automatically. Our approach is based on a novel modelling technique in which we introduce several different types of variables for every input/output word of all operations, and impose constraints on these variables such that from a solution of these variables satisfying all the constraints we can deduce a $\mathcal{DS}$-MITM distinguisher or $\mathcal{DS}$-MITM attack.

Compared with Derbez and Fouque's tool [30, 31] which was implemented in the general purpose programming language C/C++, the CP based method allows the cryptanalysts to state the problem at a high level very naturally, without considering how to maintain the relationships between the variables explicitly with dedicated algorithms. Therefore, our tool should be very useful in fast prototyping in the process of block cipher design.

In [43], Lin et al. modeled the problem of searching for $\mathcal{DS}$-MITM distinguishers as an integer programming model. However, their integer programming model is incomplete and is solved by a dedicated search algorithm. Secondly, Lin et al. 's work only focuses on the distinguisher part. Our CP based approach can not only enumerate distinguishers but also partly automate the key-recovery process of the attack. Moreover, by applying our CP based approach to LBlock, the same cipher targeted in [43], we show it finds better distinguishers as well as better attacks. To demonstrate the effectiveness of our approach, we apply it to SKINNY [44], TWINE [45], and LBlock [46]. We produce so far the best $\mathcal{DS}$-MITM attacks on these well-known ciphers automatically.

For LBlock, we can not only find an 11-round $\mathcal{DS}$-MITM distinguisher which is 2 rounds longer than the one(s) presented in [43], but also construct the first $\mathcal{DS}$-MITM attack on 21-round LBlock. We also rediscover the same attack on TWINE-128 given in [34], and identify the first $\mathcal{DS}$-MITM attack on 20-round TWINE-80. In addition, we report the first concrete $\mathcal{DS}$-MITM analysis of SKINNY. A remarkable fact is that our tool identify an 10.5-round $\mathcal{DS}$-MITM distinguisher in a few seconds, while its designers expect an upper-bound of 10 rounds against such distinguishers in [44]. A summary of these results are given in Table 1.

We also show how helpful our tool can be in the block cipher design process by searching for the best choices of block shuffles in LBlock and TWINE. We scan over 40320 variants of LBlock, and 887040 variants of TWINE. We identify permutations which are potentially stronger than the permutations in the original designs. We make the source code of this work publicly available at

https://github.com/siweisun/MITM.

In addition, all supplementary materials referred later on are provided in an extended version of this paper at https://eprint.iacr.org/2018/813.

Table 1: A summary of the results. Though the focus of this paper is the $\mathcal{DS}$-MITM attack, we also list other types of attacks which achieve currently known best results against the ciphers targeted. For the $\mathcal{DS}$-MITM attack, the number of rounds attacked is presented in the form of $a+b$, where a shows how many rounds are covered by the underlying $\mathcal{DS}$-MITM distinguisher, while b is the number or outer rounds added when performing a key-recovery attack. Therefore, $b = 0$ indicates a distinguishing attack.

Target	Rounds	Time	Data	Memory	Method	Ref
LBlock	$11+10$	$2^{70.20}$	2^{48} CP	$2^{61.91}$	$\mathcal{DS}$-MITM	Sect. 7.2
	$9+0$	$2^{74.5}$	–	–	$\mathcal{DS}$-MITM Dist.	[43]
	23	$2^{74.5}$	$2^{59.5}$CP	$2^{74.3}$	ID	[47]
	23	$2^{75.36}$	2^{59}CP	2^{74}	ID	[48]
	23	2^{72}	$2^{62.1}$ Kp	2^{60}	MultiD ZC	[47]
	23	2^{76}	$2^{62.1}$ Kp	2^{60}	MultiD ZC	[49]
TWINE80	$11+9$	$2^{77.44}$	2^{32} CP	$2^{82.91}$	$\mathcal{DS}$-MITM	Sect. 7.3
	23	$2^{79.09}$	$2^{57.85}$ CP	$2^{84.06}$	ID	[50]
	23	2^{73}	$2^{62.1}$ KP	2^{60}	MultiD ZC	[47]
TWINE128	$11+14$	$2^{124.7}$	2^{48} CP	2^{109}	$\mathcal{DS}$-MITM*	[34]
	25	$2^{124.5}$	$2^{59.1}$ CP	$2^{78.1}$	ID	[34]
	25	2^{119}	$2^{62.1}$ KP	2^{60}	MultiD ZC	[47]
	25	$2^{122.12}$	$2^{62.1}$ KP	2^{60}	MultiD ZC	[49]
SKINNY-128-384	$10.5+11.5$	$2^{382.46}$	2^{96} CP	$2^{330.99}$	$\mathcal{DS}$-MITM	Sect. 7.1
	$11+11$	$2^{373.48}$	$2^{92.22}$ CP	$2^{147.22}$	ID	[51]

$\star$ We find the attacks with the same complexity.

Organization. In Sect. 2, we give the notations used in this paper. An introduction of the $\mathcal{DS}$-MITM attack is presented in Sect. 3. We show the general principle of how to model the $\mathcal{DS}$-MITM attack in Sect. 4, and subsequently in Sect. 5 the technical detail of the modelling method is given. Sect. 6 discusses how to use our method in practice. In Sect. 7, we apply our approach to SKINNY, TWINE, LBlock, AES, ARIA, and SIMON. In Sect. 8, we discuss how to use our tool to find high-quality building blocks (with respect to the $\mathcal{DS}$-MITM attack) in the process of block cipher design. Sect. 9 is the conclusion.

2 Notations

An n-bit state state with $n = cn_c$ is alternatively regarded as a sequence $(\mathsf{state}[0], \mathsf{state}[1], \cdots, \mathsf{state}[n_c - 1])$ of n_c c-bit words. Let $\mathcal{A} = [j_0, j_1, \cdots, j_{s-1}]$ be an ordered set of integers such that $0 \leq j_0 < \cdots < j_{s-1} < n_c$. Then $\mathsf{state}[\mathcal{A}]$ is used to represent $\mathsf{state}[j_0]||\cdots||\mathsf{state}[j_{s-1}]$, where $\mathsf{state}[j]$ is the j-th c-bit word of state and $||$ is the operation of bit string concatenation.

Definition 1. *A set* $\{P^0, \cdots, P^{N-1}\} \subseteq \mathbb{F}_2^{cn_c} = \mathbb{F}_2^n$ *of* $N = 2^{sc}$ *n-bit values for* state *is a* $\delta(\mathcal{A})$*-set for* state *with* $\mathcal{A} = [k_0, k_1, \cdots, k_{s-1}]$ *if* $P^0[\mathcal{A}] \oplus P^j[\mathcal{A}] = j$ $(1 \leq j < N)$*, and* $P^i[k] = P^j[k]$ *for all* $i, j \in \{0, \cdots, N-1\}$ *and* $k \notin \mathcal{A}$*. That is,* $\{P^0, \cdots, P^{N-1}\}$ *traverse the s c-bit words specified by* $\mathcal{A}$ *while share the same value in other word positions.*

An r-round iterative block cipher E with $r = r_0 + r_1 + r_2$, depicted in Fig. 1, is a keyed permutation which transforms an n-bit state state_0 into state_{2r} step by step with nonlinear and linear operations. In our indexing scheme, as illustrated in Fig. 1, state_{2k} is the input state of round k, state_{2k+1} is the output state of the nonlinear operation of round k, and $\mathsf{state}_{2(k+1)}$ is the output of round k or the input of round $k+1$ for $k \in \{0, \cdots, r_0+r_1+r_2-1\}$. For the sake of simplicity and concreteness, we will conduct the discussion based on Fig. 1, which visualizes the structure of a common SP cipher. Without loss of generality, we assume that the key addition is performed after the linear layer L as illustrated in Fig. 1. The basic rule is that we should always introduce a new state for the direct input to the nonlinear layer. For example, if the key addition is performed in between state_{2i} and the NL operation, then a new state (representing the direct input to NL) should be introduced in between the key addition and the NL operation, and the original state may be omitted (regarding the new state as an output obtained by masking the output of the previous round with the subkey).

Note that though our discussion are based on a SP cipher illustrated in Fig. 1, the ideas and techniques presented in this paper are general enough to be applied to other structures, such as Feistel and Generalized Feistel structures.

For convenience, a $\delta(\mathcal{A})$-set $\{P^0, \cdots, P^{N-1}\}$ is denoted by $\mathbb{P}_{\delta(\mathcal{A})}$, and let $\Delta_E(\mathbb{P}_{\delta(\mathcal{A})}, \mathcal{B})$ be the sequence $[C^0[\mathcal{B}] \oplus C^1[\mathcal{B}], \cdots, C^0[\mathcal{B}] \oplus C^{N-1}[\mathcal{B}]]$, where $C^i = E(P^i)$ and $\mathcal{B} = [j_0, \cdots, j_{t-1}]$ such that $0 \leq j_0 < \cdots < j_{t-1} < n_c$.

Let P, $P^{'} \in \mathbb{F}_2^n$ be two values of state_0 shown in Fig. 1, which are often regarded as plaintexts since state_0 is the input of the encryption algorithm. The value P creates a series of intermediate values during the encryption process. We define $P(\mathsf{state}_i)$ as the intermediate value at state_i created by the partial encryption of P. Sometimes we only care about the value of $P(\mathsf{state}_i)$ at some specified word positions indexed by an ordered set $\mathcal{I}$, which is denoted by $P(\mathsf{state}_i[\mathcal{I}])$. We define $P \oplus P^{'}(\mathsf{state}_i)$ and $P \oplus P^{'}(\mathsf{state}_i[\mathcal{I}])$ to be the intermediate differences $P(\mathsf{state}_i) \oplus P^{'}(\mathsf{state}_i)$ and $P(\mathsf{state}_i[\mathcal{I}]) \oplus P^{'}(\mathsf{state}_i[\mathcal{I}])$ respectively. Let C and $C^{'}$ be the ciphertexts of P and $P^{'}$. An intermediate value can also be regarded as the result of a partial decryption of the ciphertext C. Therefore, we define $C(\mathsf{state}_i)$, $C(\mathsf{state}_i[\mathcal{I}])$, $C \oplus C^{'}(\mathsf{state}_i)$, and $C \oplus C^{'}(\mathsf{state}_i[\mathcal{I}])$ similarly. Note that in the above notations, the intermediate values or differences of intermediate values are specified with respect to some plaintexts or ciphertexts. We may as well specify them with respect to some intermediate values, say $Q = P(\mathsf{state}_j)$ and $Q^{'} = P^{'}(\mathsf{state}_j)$. Hence, we may have notations such as $Q(\mathsf{state}_i)$, $Q(\mathsf{state}_i[\mathcal{I}])$, $Q \oplus Q^{'}(\mathsf{state}_i)$, and $Q \oplus Q^{'}(\mathsf{state}_i[\mathcal{I}])$, whose meanings should be clear from the context.

To make the notation succinct, if not stated explicitly, we always assume that $\mathcal{A} = [k_0, \cdots, k_{s-1}]$, $\mathcal{B} = [j_0, \cdots, j_{t-1}]$, and a state state is viewed as a

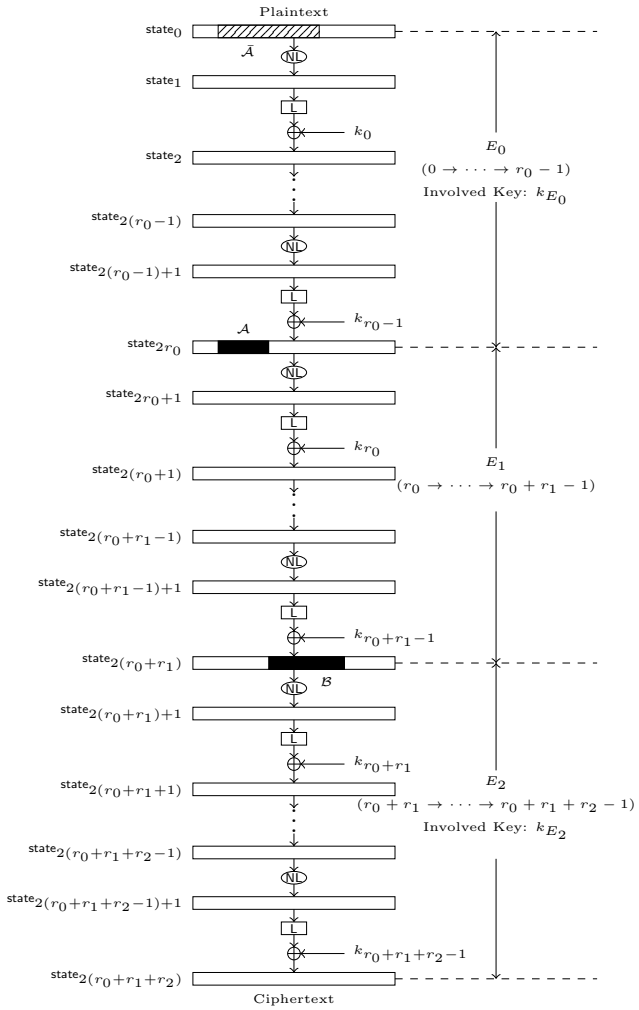


Fig. 1: An r-round SP block cipher $E = E_2 \circ E_1 \circ E_0$ with $r = r_0 + r_1 + r_2$, whose round function consists of a layer of nonlinear operation and a layer of linear operation. A $\mathcal{DS}$-MITM key-recovery attack is performed based on a $\mathcal{DS}$-MITM distinguisher placed at E_1. A more detailed explanation of this figure will be given in Sect. 3.2.

a sequence of n bits or a sequence of n_c c-bit words. Moreover, we make the following assumption which is very natural for a block cipher.

Assumption 1 *Let the nonlinear layer in Fig. 1 be a parallel application of n_c $c \times c$ invertible S-boxes, and $\mathcal{I} = [j : Q \oplus Q'(\mathsf{state}_{2k}[j]) \neq 0,\ 0 \leq j < n_c]$ be an ordered set, where Q and Q' are two values for state_{2k}. If we know the value of $Q(\mathsf{state}_{2k}[\mathcal{I}])$, then we can derive the value of $Q \oplus Q'(\mathsf{state}_{2k+1})$*

with the knowledge of $Q \oplus Q'(\mathsf{state}_{2k}[\mathcal{I}])$. *Similarly, we can derive the value of* $Q \oplus Q'(\mathsf{state}_{2k})$ *with the knowledge of* $Q(\mathsf{state}_{2k+1}[\mathcal{I}])$ *and* $Q \oplus Q'(\mathsf{state}_{2k+1}[\mathcal{I}])$. *In other words, we can derive the value of the output/input differences if we know the value of input/output values and differences at the active positions.*

3 The Demirci-Selçuk Meet-in-the-Middle Attack

3.1 The $\mathcal{DS}$-MITM Distinguisher

The $\mathcal{DS}$-MITM attack relies on a special differential-type distinguisher. Compared with ordinary differential distinguishers, the $\mathcal{DS}$-MITM distinguishers generally lead to much stronger filters.

Let F be a keyed permutation, and $\mathbb{Q}_{\delta(\mathcal{A})} = \{Q^0, \cdots, Q^{N-1}\}$ be a $\delta(\mathcal{A})$-set for the input state of F. If F is a random permutation, then it can be shown that there are $(2^{ct})^{2^{cs}-1}$ possibilities for $\Delta_F(\mathbb{Q}_{\delta(\mathcal{A})}, \mathcal{B})$. But for a block cipher F, it is possible that the sequence $\Delta_F(\mathbb{Q}_{\delta(\mathcal{A})}, \mathcal{B})$ can be fully determined with the knowledge of d c-bit words. For instance, from the values of one internal state and the master key one can derive the values for all the internal states. Therefore, given $\mathbb{Q}_{\delta(\mathcal{A})}$, we can get at most 2^{cd} possible cases of $\Delta_F(\mathbb{Q}_{\delta(\mathcal{A})}, \mathcal{B})$ by traversing the d c-bit words. We call d the $(\mathcal{A}, \mathcal{B})$-degree of F, which is denoted by $\mathsf{Deg}_F(\mathcal{A}, \mathcal{B})$, or simply $\mathsf{Deg}(\mathcal{A}, \mathcal{B})$ if F can be inferred from the context. If $\mathsf{Deg}_F(\mathcal{A}, \mathcal{B}) = d$ is small enough such that $\lambda = 2^{cd}/(2^{ct})^{2^{cs}-1} = 2^{c(d-t\cdot(2^{cs}-1))} < 1$, or $d < t \cdot (2^{cs} - 1)$, then we can use this property as a distinguisher and construct a key-recovery attack on F. Therefore, a $\mathcal{DS}$-MITM distinguisher of a keyed permutation F can be regarded as a tuple $(\mathcal{A}, \mathcal{B}, \mathsf{Deg}_F(\mathcal{A}, \mathcal{B}))$.

3.2 Key Recovery Attack based on $\mathcal{DS}$-MITM Distinguisher

We now describe how a key-recovery attack can be performed with a $\mathcal{DS}$-MITM distinguisher. This part should be read while referring to Fig. 1.

As shown in Fig. 1, we divide the target cipher E into 3 parts: E_0, E_1, and E_2, where E_i is a keyed permutation with r_i rounds. As depicted in Fig. 1, E_0 covers rounds $(0 \to \cdots \to r_0 - 1)$, E_1 covers rounds $(r_0 \to \cdots \to r_0 + r_1 - 1)$, and E_2 covers rounds $(r_0 + r_1 \to \cdots \to r_0 + r_1 + r_2 - 1)$. According to our indexing scheme, as illustrated in Fig. 1, state_0 is the input state of E_0; state_{2r_0} is the output state of E_0 which is also the input state of E_1; $\mathsf{state}_{2(r_0+r_1)}$ is the output of E_1 or the input of E_2; finally, $\mathsf{state}_{2(r_0+r_1+r_2)}$ is the output of E_2.

In the attack, we place a $\mathcal{DS}$-MITM distinguisher $(\mathcal{A}, \mathcal{B}, \mathsf{Deg}_{E_1}(\mathcal{A}, \mathcal{B}))$ at E_1, and prepare a $\delta(\bar{\mathcal{A}})$-set $\mathbb{P}_{\delta(\bar{\mathcal{A}})}$ of chosen plaintexts for state_0, where $\bar{\mathcal{A}}$ is the ordered set of integers k $(0 \le k < n_c)$ such that $V^0 \oplus V^j(\mathsf{state}_0[k]) \neq 0$ for some $\delta(\mathcal{A})$-set $\mathbb{V}_{\delta(\mathcal{A})} = \{V^0, \cdots, V^{N-1}\}$ for state_{2r_0} (the input state of E_1) and some $j \in \{0, \cdots, N-1\}$. Note that $\bar{\mathcal{A}}$ can be obtained by propagating the differences created by $\mathbb{V}_{\delta(\mathcal{A})}$ for state_{2r_0} (the input of E_1) reversely against E_0.

Then we select an arbitrary plaintext P^0 from $\mathbb{P}_{\delta(\bar{\mathcal{A}})}$, and guess the secret key information $k_{E_0} \in \mathbb{F}_2^{e_0}$ with which we can find $P^1, \cdots, P^{N-1}$ in $\mathbb{P}_{\delta(\bar{\mathcal{A}})}$ such

that $\mathbb{Q}_{\delta(\mathcal{A})} = \{Q^0, \cdots, Q^{N-1}\}$ where $Q^j = E_0(P^j)$ forms a $\delta(\mathcal{A})$-set for state_{2r_0}. Finally, we guess the secret key information $k_{E_2} \in \mathbb{F}_2^{e_2}$ involved in E_2 with which we can determine the sequence

$$\Delta_{E_1}(\mathbb{Q}_{\delta(\mathcal{A})}, \mathcal{B}) = [C^0 \oplus C^1(\mathsf{state}_{2(r_0+r_1)}[\mathcal{B}]), \cdots, C^0 \oplus C^{N-1}(\mathsf{state}_{2(r_0+r_1)}[\mathcal{B}])]$$

by partial decryption with E_2, where $C^j = E(P^j)$.

If the resulting sequence is not one of the possible $\Delta_{E_1}(\mathbb{Q}_{\delta(\mathcal{A})}, \mathcal{B})$ sequences which can be determined with the $\mathsf{Deg}_{E_1}(\mathcal{A}, \mathcal{B}) = d$ c-bit parameters, the guesses of k_{E_0} and k_{E_2} are certainly incorrect and therefore rejected. Similar to [52], we adopt the notion of $|k_{E_0} \cup k_{E_2}|$ to represent the log of the entropy of the involved secret key bits in the outer rounds from an information theoretical point of view.

3.3 Complexity Analysis

Offline phase. Store all the 2^{cd} possibilities of the sequence $\Delta_{E_1}(\mathbb{Q}_{\delta(\mathcal{A})}, \mathcal{B})$ in a hash table. The time complexity is $2^{cd} \cdot 2^{cs} \cdot \rho_{E_1} C_E$, and the memory complexity is $(2^{cs} - 1) \cdot ct \cdot 2^{cd}$ bits, where C_E is the time complexity of one encryption with E, and ρ_{E_1} is typically computed in literature as $\mathsf{Deg}(\mathcal{A}, \mathcal{B})$ divided by the total number of S-boxes in E.

Online phase. For each of the $2^{|k_{E_0} \cup k_{E_2}|}$ possible guesses, if the resulting sequence $\Delta_{E_1}(\mathbb{Q}_{\delta(\mathcal{A})}, \mathcal{B})$ is not in the hash table precomputed, then the guess under consideration is certainly not correct and is discarded. The time complexity of this step is $2^{|k_{E_0} \cup k_{E_2}|} \cdot 2^{sc} \cdot \rho_{E_0 \cup E_2} C_E$, where $\rho_{E_0 \cup E_2}$ is typically computed as the number of S-boxes involved in the outer rounds divided by the total number of S-boxes in E. After this step, the $2^{|k_{E_0} \cup k_{E_2}|}$ key space is reduced approximately to $\lambda \cdot 2^{|k_{E_0} \cup k_{E_2}|}$, where $\lambda = 2^{c(d - t \cdot (2^{cs} - 1))}$.

4 Modelling the $\mathcal{DS}$-MITM Attack with Constraints: A High Level Overview

In this section, we give a high level overview of our modelling method with the aid of Fig. 1 and Fig. 2, which serves as a road map for the next section (Sect. 5), where the technical details are presented. To model the attack with constraint programming (CP) for the cipher $E = E_2 \circ E_1 \circ E_0$ shown in Fig. 1, we proceed as the following steps.

Step 1. Modelling the distinguisher part

- Introduce three types (X, Y, and Z) of 0-1 variables for each word of the states $\mathsf{state}_{2r_0}, \cdots, \mathsf{state}_{2(r_0+r_1)}$ involved in E_1. We denote the sets of all type-X, type-Y and type-Z variables by $\mathsf{Vars}(X)$, $\mathsf{Vars}(Y)$ and $\mathsf{Vars}(Z)$, respectively.
- Introduce a set of constraints over $\mathsf{Vars}(X)$ to model the propagation of the *forward differential*, and introduce a set of constraints over $\mathsf{Vars}(Y)$ to model the *backward determination relationship*.
- Impose a set of constraints on $\mathsf{Vars}(Z)$ such that a type-Z variable for $\mathsf{state}_i[j]$ is 1 if and only if the type-X and type-Y variables for $\mathsf{state}_i[j]$ are 1 simultaneously.

Remark 1. Under the above configuration, every instantiation of the variables in $\mathsf{Vars}(X)$, $\mathsf{Vars}(Y)$, and $\mathsf{Vars}(Z)$ corresponds to a potential $\mathcal{DS}$-MITM distinguisher. Therefore, all distinguishers can be enumerated with the above model. Also note that the key addition can be omitted while searching for distinguishers if it does not affect the propagation of the forward differential and backward determination relationship. This is the case for all the examples presented in this paper, where key additions are only involved in computing the actual complexities.

Step 2. Modelling the outer rounds

- Introduce a type-M variable for each word of the states $\mathsf{state}_0, \cdots, \mathsf{state}_{2r_0}$ involved in E_0, and impose a set of constraints over $\mathsf{Vars}(M)$ to model the *backward differential*. Note that there are both type-X and type-M variables for state_{2r_0}. We require that the corresponding type-X and type-M variables for each of the n_c words of state_{2r_0} are equal.
- Introduce a type-W variable for each word of the states $\mathsf{state}_{2(r_0+r_1)}$, $\cdots$, $\mathsf{state}_{2(r_0+r_1+r_2)}$ involved in E_2, and impose a set of constraints over $\mathsf{Vars}(W)$ to model the *forward determination relationship*. Note that there are both type-Y and type-W variables for $\mathsf{state}_{2(r_0+r_1)}$. We require that the corresponding type-Y and type-W variables for each of the n_c words of $\mathsf{state}_{2(r_0+r_1)}$ are equal.

Remark 2. Every solution of $\mathsf{Vars}(M)$ and $\mathsf{Vars}(W)$ helps us to identify the information that needs to be guessed in the outer rounds, which will be clearer in the following.

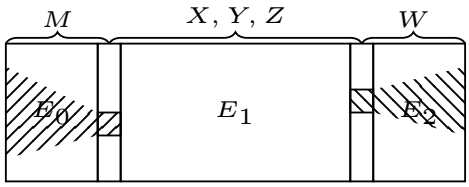


Fig. 2: A high level overview of the modelling method for $\mathcal{DS}$-MITM attack

The overall modelling strategy is depicted in Fig. 2. In summary, given a full solution of the variables such that all constraints are fulfilled, we can extract the following information

- $\mathcal{A}$: The variables in $\mathsf{Vars}(X)$ for state_{2r_0} whose values are 1 indicate $\mathcal{A}$;
- $\mathcal{B}$: The variables in $\mathsf{Vars}(Y)$ for $\mathsf{state}_{2(r_0+r_1)}$ whose values are 1 indicate $\mathcal{B}$;
- $\mathsf{Deg}_{E_1}(\mathcal{A}, \mathcal{B})$: The variables in $\mathsf{Vars}(Z)$ for state_{2j}, $r_0 \leq j < r_0 + r_1$ whose values are 1 indicate $\mathsf{Deg}_{E_1}(\mathcal{A}, \mathcal{B})$;
- $\bar{\mathcal{A}}$ and guessed materials in E_0 : The variables in $\mathsf{Vars}(M)$ whose values are 1 indicate $\bar{\mathcal{A}}$ and guessed materials in E_0 which tells us how to prepare the plaintexts leading a $\delta(\mathcal{A})$ set at state_{2r_0};

- Guessed materials in E_2 : The variables in $\mathsf{Vars}(W)$ whose values are 1 indicate the Guessed materials in E_2 with which we can derive the sequence of differences at $\mathsf{state}_{2(r_0+r_1)}$ from the ciphertexts.

Together this information forms a $\mathcal{DS}$-MITM attack on E. Note that the guessed materials in E_0 and E_2 still need to be converted to guessed key materials, which can be done manually or automatically fairly straightforwardly.

According to the semantics of $\mathsf{Vars}(Z)$, if we draw the propagation patterns of $\mathsf{Vars}(X)$ and $\mathsf{Vars}(Y)$ in two figures, then the propagation pattern of $\mathsf{Vars}(Z)$ can be obtained by superposition of the two figures. Therefore, the key to understand the details of the modelling of $\mathcal{DS}$-MITM attack is the so-called *forward/backward differential* and *forward/backward determination relationship*. To make the description succinct and without loss of generality, we introduce the concepts based on a 5-round keyed permutation shown in Fig. 4 and Fig. 6. We will also give two concrete examples of the forward differential and backward determination of a 3-round toy SPN block cipher with 32-bit (4-byte) block size. The round function shown in Fig. 3 of the toy cipher consists of an S-box layer (a parallel application of four 8×8 Sboxes), and a linear layer L with $y_i = \bigoplus_{j \in \{0,1,2,3\}-\{i\}} x_j$ for $i \in \{0, 1, 2, 3\}$.

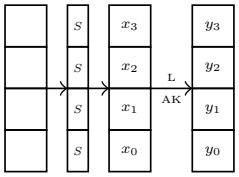


Fig. 3: The round function of the toy cipher

4.1 Forward Differential and Backward Differential

As shown in Fig. 4, given a set $\mathbb{Q}_{\delta(\mathcal{A})}$ of N values $\{Q^0, \cdots, Q^{N-1}\}$ for state_4 which forms a $\delta(\mathcal{A})$ set for the input state of round 2. For every word $\mathsf{state}_i[j]$ $(4 \leq i \leq 10, 0 \leq j < n_c)$, we introduce a 0-1 variable $X_i[j]$. We say that the set of 0-1 variables $\{X_i[j] : 4 \leq i \leq 10, 0 \leq j < n_c\}$ models the *forward differential* of $\mathbb{Q}_{\delta(\mathcal{A})}$ in rounds $(2 \to 3 \to 4)$ if the following conditions are satisfied.

- Conditions for state_4 (the starting point of the forward differential, which is also the input of round 2) : $\forall j \in \mathcal{A}$, $X_4[j] = 1$ and $\forall j \notin \mathcal{A}$, $X_4[j] = 0$
- Conditions for rounds $(2 \to 3 \to 4)$: $X_i[j] = 0$ $(5 \leq i \leq 10, 0 \leq j < n_c)$ if and only if $\forall Q^k \in \mathbb{Q}_{\delta(\mathcal{A})}$, $Q^0 \oplus Q^k(\mathsf{state}_i[j]) = 0$

Similarly, as depicted in Fig. 4, we say that the set of variables $\{X_i[j] : 0 \leq i \leq 4, 0 \leq j < n_c\}$ models the *backward differential* of $\mathbb{Q}_{\delta(\mathcal{A})}$ in rounds $(1 \to 0)$ if the following conditions are satisfied.

- Conditions for state_4 (the starting point of the backward differential, which is also the output of round 1): $\forall j \in \mathcal{A}$, $X_4[j] = 1$ and $\forall j \notin \mathcal{A}$, $X_4[j] = 0$
- Conditions for rounds $(1 \to 0)$: $X_i[j] = 0$ $(0 \leq i < 4, 0 \leq j < n_c)$ if and only if $\forall Q^k \in \mathbb{Q}_{\delta(\mathcal{A})}$, $Q^0 \oplus Q^k(\mathsf{state}_i[j]) = 0$

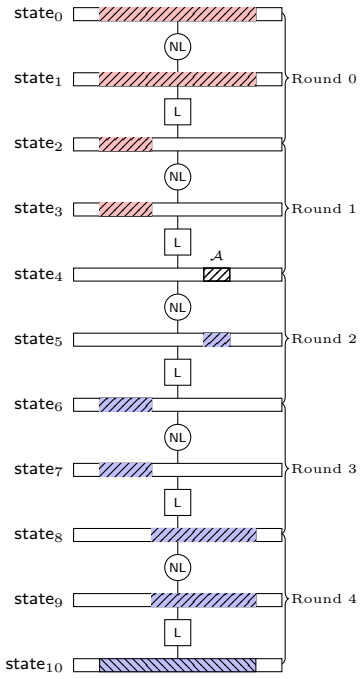


Fig. 4: Forward/backward differential illustrated on a 5-round keyed permutation

Let us give a concrete example. Let $\mathcal{A} = [3]$ and $\mathbb{Q}_{\delta(\mathcal{A})} = \{(0, 0, 0, x) \in (\mathbb{F}_2^8)^4 : x \in \mathbb{F}_2^8\}$. Then the set of variables $X_i[j]$ with $0 \leq i \leq 6$ and $0 \leq j < 4$ shown in Fig. 5 models forward differential of $\mathbb{Q}_{\delta(\mathcal{A})}$ in rounds $(0 \to 1 \to 2)$ if we impose the following constraints on $X_i[j]$. Since the values in $\mathbb{Q}_{\delta(\mathcal{A})}$ are active at the third byte, we have $X_0[0] = X_0[1] = X_0[2] = 0, X_0[3] = 1$. For the S-layers in the toy cipher, we have $X_{2i}[j] = X_{2i+1}[j], 0 \leq i \leq 2, 0 \leq j < 4$. For the linear layers, we enforce $3X_{2(i+1)}[j] - X_{2i+1}[j+1] - X_{2i+1}[j+2] - X_{2i+1}[j+3] \geq 0$ to ensure that $X_{2(i+1)}[j]$ will be equal to 1 when any one of $X_{2i+1}[j+1]$, $X_{2i+1}[j+2]$, $X_{2i+1}[j+3]$ is 1. We also add the constraint

$$X_{2i+1}[j+1] + X_{2i+1}[j+2] + X_{2i+1}[j+3] - X_{2(i+1)}[j] \geq 0$$

to dictate that $X_{2(i+1)}[j]$ must be 0 when all of $X_{2i+1}[j+1]$, $X_{2i+1}[j+2]$, $X_{2i+1}[j+3]$ are 0, where $0 \leq i \leq 2$, $0 \leq j < 4$ and the indexes are computed modulo 4. With these constraints, the $X_i[j]$ variables propagate in a pattern depicted in Fig. 5.

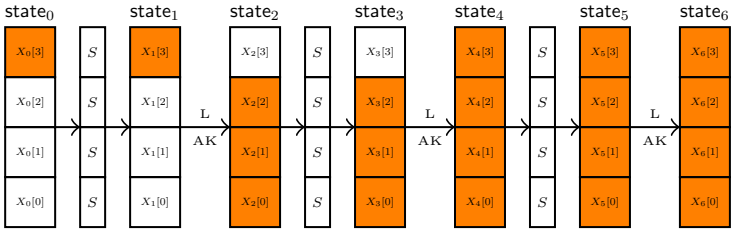


Fig. 5: The forward differential of a 3-round toy cipher

4.2 Forward Determination and Backward Determination

As shown in Fig. 6, given a set $\mathbb{Q} = \{Q^0, \cdots, Q^{N-1}\}$ of N values for state_6 and an ordered set $\mathcal{B}$ of indices, we say that the set of variables $\{Y_i[j] : 6 \leq i \leq 10, 0 \leq j < n_c\}$ models the *forward determination relationship* of $\{Q^0(\mathsf{state}_6[\mathcal{B}]), \cdots, Q^{N-1}(\mathsf{state}_6[\mathcal{B}])\}$ in rounds $(3 \rightarrow 4)$ if the following conditions hold.

- Conditions for state_6 (the starting point of the forward determination relationship, which is also the input of round 3) : $\forall j \in \mathcal{B}$, $Y_6[j] = 1$ and $\forall j \notin \mathcal{B}$, $Y_6[j] = 0$
- Conditions for rounds $(3 \rightarrow 4)$: For $6 \leq i < 10$, $\forall k \in \{0, \cdots, N-1\}$, with the knowledge of $Q^0 \oplus Q^k(\mathsf{state}_{i+1}[\mathcal{B}_{i+1}])$ (and $Q^0(\mathsf{state}_{i+1}[\mathcal{B}_{i+1}])$ if state_{i+1} is an output state of a nonlinear layer) one can deduce the value $Q^0 \oplus Q^k(\mathsf{state}_i[\mathcal{B}_i])$, where $\mathcal{B}_{i+1} = [j : Y_{i+1}[j] = 1,\ 0 \leq j < n_c]$ for $6 \leq i < 10$ and $\mathcal{B}_6 = \mathcal{B}$.

Similarly, as shown in Fig. 6, we say that the set of 0-1 variables $\{Y_i[j] : 0 \leq i \leq 6, 0 \leq j < n_c\}$ models the *backward determination relationship* of $\{Q^0(\mathsf{state}_6[\mathcal{B}]), \cdots, Q^{N-1}(\mathsf{state}_6[\mathcal{B}])\}$ in rounds $(2 \rightarrow 1 \rightarrow 0)$ if the following conditions hold.

- Conditions for the state_6 (the starting point of the backward determination relationship, which is also the output of round 2): $\forall j \in \mathcal{B}$, $Y_6[j] = 1$ and $\forall j \notin \mathcal{B}$, $Y_6[j] = 0$
- Conditions for rounds $(2 \rightarrow 1 \rightarrow 0)$: For $0 < i \leq 6$, $\forall k \in \{0, \cdots, N-1\}$ from the knowledge of the values $Q^0 \oplus Q^k(\mathsf{state}_{i-1}[\mathcal{B}_{i-1}])$, (and $Q^0(\mathsf{state}_{i-1}[\mathcal{B}_{i-1}])$ if state_{i-1} is an input state of a nonlinear layer), one can determine the value $Q^0 \oplus Q^k(\mathsf{state}_i[\mathcal{B}_i])$, where $\mathcal{B}_{i-1} = [j : Y_{i-1}[j] = 1, 0 \leq j < n_c]$ for $0 < i \leq 6$, and $\mathcal{B}_6 = \mathcal{B}$.

Now we show a concrete example. Assume that we have a set $\{Q^0, \cdots, Q^{255}\} = \{(0, 0, 0, x) \in (\mathbb{F}_2^8)^4 : x \in \mathbb{F}_2^8\}$ of 2^8 values for state_0, as depicted in Fig. 7. After the 3-round encryption of the toy cipher, we get a set $\{C^0, \cdots C^{255}\}$ of 2^8 values for state_6. Let $\mathcal{B} = [3]$. The set of variables $Y_i[j]$ with $0 \leq i \leq 6$ and $0 \leq j < 4$ shown in Fig. 7 models backward determination of $\{C^0, \cdots C^{255}\}$ in rounds $(2 \rightarrow 1 \rightarrow 0)$ if we impose the following constraints on $Y_i[j]$.

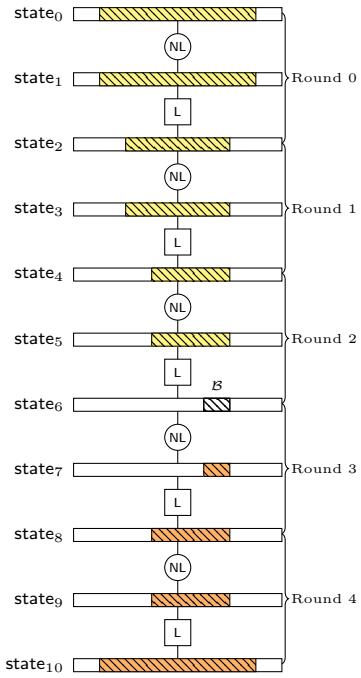


Fig. 6: The forward/backward determination relationship illustrated on a 5-round keyed permutation

Since $\mathcal{B} = [3]$, we have $Y_6[0] = Y_6[1] = Y_6[2] = 0, Y_6[3] = 1$. For the S layers in the toy cipher, we have $Y_{2i}[j] = Y_{2i+1}[j], 0 \leq i \leq 2, 0 \leq j < 4$. For the linear layers, we add $3Y_{2i+1}[j] - Y_{2(i+1)}[j+1] - Y_{2(i+1)}[j+2] - Y_{2(i+1)}[j+3] \geq 0$ to ensure that $Y_{2i+1}[j]$ must be 1 when any one of $Y_{2(i+1)}[j+1]$, $Y_{2(i+1)}[j+2]$, $Y_{2(i+1)}[j+3]$ is 1, and $Y_{2(i+1)}[j+1] + Y_{2(i+1)}[j+2] + Y_{2(i+1)}[j+3] - Y_{2i+1}[j] \geq 0$ to dictate that $Y_{2i+1}[j]$ must be 0 when all of $Y_{2(i+1)}[j+1]$, $Y_{2(i+1)}[j+2]$, $Y_{2(i+1)}[j+3]$ are 0, where the indexes are computed modulo 4. With these constraints, the $Y_i[j]$ variables propagate in a pattern depicted in Fig. 7.

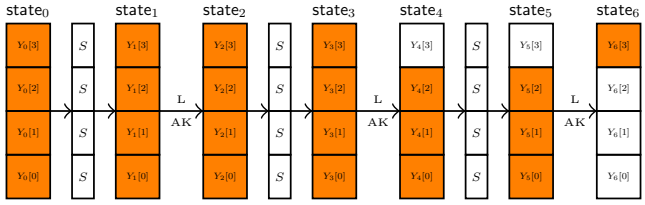


Fig. 7: The backward determination of a 3-round toy cipher

Note that the concepts introduced in this section are generic and not limited to SP ciphers. For instance, we depicted the propagation patterns of the forward differential and backward determination of a Feistel cipher with 8-bit block size and 4×4 S-box in Fig. 8a and Fig. 8b respectively.

(a) Forward differential (b) Backward determination

Fig. 8: The forward differential and backward determination of a 3-round toy cipher with Feistel structure

5 Modelling the $\mathcal{DS}$-MITM Attack with Constraints: The Technical Details

Given a cipher $E = E_2 \circ E_1 \circ E_0$, we show how to model the distinguisher part (E_1), and subsequently the key-recovery part (E_0 and E_2). These models for E_1, E_0 and E_2 jointly lead to a model for $\mathcal{DS}$-MITM attack on E. Note that this part of the paper should be read while referring to Fig. 1.

5.1 CP Model for E_1: The Distinguisher Part.

We introduce 2 sets of variables $\mathsf{Vars}(X) = \{X_i[j] : 2r_0 \leq i \leq 2(r_0 + r_1),\ 0 \leq j < n_c\}$ and $\mathsf{Vars}(Y) = \{Y_i[j] : 2r_0 \leq i \leq 2(r_0 + r_1),\ 0 \leq j < n_c\}$ for all the words of the states $\{\mathsf{state}_i[j] : 2r_0 \leq i \leq 2(r_0 + r_1),\ 0 \leq j < n_c\}$ involved in the r_1 rounds of E_1 as shown in Fig. 1.

We then impose a set of constraints on $\mathsf{Vars}(X)$ such that $\mathsf{Vars}(X)$ models the forward differential of a $\delta(\mathcal{A})$-set $\mathbb{Q}_{\delta(\mathcal{A})} = \{Q^0, \cdots, Q^{N-1}\}$ for state_{2r_0} with $\mathcal{A} = [j : X_{2r_0}[j] = 1,\ 0 \leq j < n_c]$ in rounds $(r_0 \to r_0 + 1 \to \cdots \to r_0 + r_1 - 1)$. Also, another set of constraints is imposed on $\mathsf{Vars}(Y)$ such that $\mathsf{Vars}(Y)$ models the backward determination relationship of

$$\{Q^0(\mathsf{state}_{2(r_0+r_1)}[\mathcal{B}]), \cdots, Q^{N-1}(\mathsf{state}_{2(r_0+r_1)}[\mathcal{B}])\}$$

with $\mathcal{B} = [j : Y_{2(r_0+r_1)}[j] = 1, 0 \leq j < n_c]$ in rounds $(r_0 + r_1 - 1 \to \cdots \to r_0)$. Finally, we introduce a new set of variables $\mathsf{Vars}(Z) = \{Z_i[j] : 2r_0 \leq i \leq 2(r_0 + r_1),\ 0 \leq j < n_c\}$ and impose a set of constraints on $\mathsf{Vars}(Z)$ such that $Z_i[j] = 1$ if and only if $X_i[j] = Y_i[j] = 1$. The variables in $\mathsf{Vars}(X)$, $\mathsf{Vars}(Y)$, and $\mathsf{Vars}(Z)$ together with the constraints imposed on them form a CP model.

Then we have the following observations which can be easily derived from the Assumption 1 made at the end of Sect. 2 and the definition of forward/backward differential and forward/backward determination relationship.

Observation 1 *If* $\mathsf{Vars}(X)$ *models the forward differential of a* $\delta(\mathcal{A})$*-set*

$$\mathbb{Q}_{\delta(\mathcal{A})} = \{Q^0, \cdots, Q^{N-1}\}$$

for state_{2r_0} *(Fig. 1) with* $\mathcal{A} = [j : X_{2r_0}[j] = 1,\ 0 \leq j < n_c]$ *in rounds* $(r_0 \rightarrow r_0 + 1 \rightarrow \cdots \rightarrow r_0 + r_1 - 1)$*, then for an arbitrary ordered set* $\mathcal{B}$ *of indices, we can determine the sequence of differences*

$$\Delta_{E_1}(\mathbb{Q}_{\delta(\mathcal{A})}, \mathcal{B}) = [Q^0 \oplus Q^1(\mathsf{state}_{2(r_0+r_1)}[\mathcal{B}]), \cdots, Q^0 \oplus Q^{N-1}(\mathsf{state}_{2(r_0+r_1)}[\mathcal{B}])]$$

from the knowledge of the following set of intermediate values of Q^0*.*

$$\{Q^0(\mathsf{state}_{2i}[j]) : X_{2i}[j] = 1, r_0 \leq i < r_0 + r_1, 0 \leq j < n_c\}.$$

Observation 2 *Let* $\mathbb{Q}_{\delta(\mathcal{A})} = \{Q^0, \cdots, Q^{N-1}\}$ *be a* $\delta(\mathcal{A})$ *set for* state_{2r_0} *for an arbitrary* $\mathcal{A}$*. If* $\mathsf{Vars}(Y)$ *models the backward determination relationship of*

$$\{Q^0(\mathsf{state}_{2(r_0+r_1)}[\mathcal{B}]), \cdots, Q^{N-1}(\mathsf{state}_{2(r_0+r_1)}[\mathcal{B}])\}$$

with $\mathcal{B} = [j : Y_{2(r_0+r_1)}[j] = 1, 0 \leq j < n_c]$ *in rounds* $(r_0 + r_1 - 1 \rightarrow \cdots \rightarrow r_0)$*, then we can determine the sequence of differences*

$$\Delta_{E_1}(\mathbb{Q}_{\delta(\mathcal{A})}, \mathcal{B}) = [Q^0 \oplus Q^1(\mathsf{state}_{2(r_0+r_1)}[\mathcal{B}]), \cdots, Q^0 \oplus Q^{N-1}(\mathsf{state}_{2(r_0+r_1)}[\mathcal{B}])]$$

from the knowledge of the following set of intermediate values of Q^0

$$\{Q^0(\mathsf{state}_{2i}[j]) : Y_{2i}[j] = 1, r_0 \leq i < r_0 + r_1, 0 \leq j < n_c\}.$$

Note that Observation 1 and Observation 2 are stated with an arbitrary ordered set $\mathcal{A}$ and $\mathcal{B}$ respectively. Therefore, if we know the intermediate values of $Q^0(\mathsf{state}[j])$ such that $X_{2i}[j]$ and $Y_{2i}[j]$ are equal to 1 simultaneously, we can determine the sequence $\Delta_{E_1}(\mathbb{Q}_{\delta(\mathcal{A})}, \mathcal{B})$ with the specific $\mathcal{A}$ and $\mathcal{B}$ corresponding to the underlying values of $\mathsf{Vars}(X)$ and $\mathsf{Vars}(Y)$.

Observation 3 *Let* $\mathcal{A} = [j : X_{2r_0}[j] = 1,\ 0 \leq j < n_c]$*,* $\mathcal{B} = [j : Y_{2(r_0+r_1)}[j] = 1,\ 0 \leq j < n_c]$*, and* $\mathbb{Q}_{\delta(\mathcal{A})} = \{Q^0, \cdots, Q^{N-1}\}$ *be a* $\delta(\mathcal{A})$ *set for* state_{2r_0}*. Then from the knowledge of the following* $\sum_{i=r_0}^{r_0+r_1-1} \sum_{j=0}^{n_c-1} Z_{2i}[j]$ *c-bit words*

$$\{Q^0(\mathsf{state}_{2i}[j]) : Z_{2i}[j] = 1, r_0 \leq i < r_0 + r_1, 0 \leq j < n_c\},$$

we can determine the value of the sequence of differences

$$\Delta_{E_1}(\mathbb{Q}_{\delta(\mathcal{A})}, \mathcal{B}) = [Q^0 \oplus Q^1(\mathsf{state}_{2(r_0+r_1)}[\mathcal{B}]), \cdots, Q^0 \oplus Q^{N-1}(\mathsf{state}_{2(r_0+r_1)}[\mathcal{B}])].$$

From the above observations, it is easy to see that any solution of $\mathsf{Vars}(X)$, $\mathsf{Vars}(Y)$, and $\mathsf{Vars}(Z)$ corresponds to a $\mathcal{DS}$-MITM distinguisher $(\mathcal{A}, \mathcal{B}, \mathsf{Deg}_{E_1}(\mathcal{A}, \mathcal{B}))$ with $\mathcal{A} = [j : X_{2r_0}[j] = 1,\ 0 \leq j < n_c]$, $\mathcal{B} = [j : Y_{2(r_0+r_1)}[j] = 1,\ 0 \leq j < n_c]$, and $\mathsf{Deg}_{E_1}(\mathcal{A}, \mathcal{B}) = \sum_{i=r_0}^{r_0+r_1-1} \sum_{j=0}^{n_c-1} Z_{2i}[j]$.

5.2 CP model for the outer rounds E_0 and E_2

The CP model for E_0. As discussed in Sect. 3, the attacker needs to prepare a set $\mathbb{P}_{\delta(\bar{\mathcal{A}})}$ of chosen plaintexts based on the distingusher $(\mathcal{A}, \mathcal{B}, \mathsf{Deg}_{E_1}(\mathcal{A}, \mathcal{B}))$ placed at E_1. According to the definition of $\bar{\mathcal{A}}$, there must be $P^1, \cdots, P^{N-1}$ in $\mathbb{P}_{\delta(\bar{\mathcal{A}})}$ such that $\mathbb{Q}_{\delta(\mathcal{A})} = \{Q^0, \cdots, Q^{n-1}\}$ forms a $\delta(\mathcal{A})$-set for state_{2r_0}, where $Q^j = E_0(P^j)$.

For E_0 we introduce a set of 0-1 variables $\mathsf{Vars}(M) = \{M_i[j] : 0 \le i \le 2r_0,\ 0 \le j < n_c\}$ and impose a set of constraints on $\mathsf{Vars}(M)$ such that $\mathsf{Vars}(M)$ models the backward differential of the $\delta(\mathcal{A})$-set $\mathbb{Q}_{\delta(\mathcal{A})}$ with $\mathcal{A} = \{j : X_{2r_0}[j] = 1,\ 0 \le j < n_c\}$ in rounds $(r_0 - 1 \to \cdots \to 0)$. Then according to the definition of backward differential and assumption 1, we have the following observation.

Observation 4 *Given $P^0 \in \mathbb{P}_{\delta(\bar{\mathcal{A}})}$, the set*

$$\mathsf{Guess}(E_0) = \{P^0(\mathsf{state}_{2i}[j]) : M_{2i}[j] = 1, 0 < i < r_0, 0 \le j < n_c\}$$

of $\sum_{i=1}^{r_0-1} \sum_{j=0}^{n_c-1} M_{2i}[j]$ c-bit words needs to be guessed to find $P^1, \cdots, P^{N-1}$ in $\mathbb{P}_{\delta(\bar{\mathcal{A}})}$.

The CP model for E_2. After the guess of $\mathsf{Guess}(E_0)$, we obtain a set $\{P^0, \cdots, P^{N-1}\} \subseteq \mathbb{P}_{\delta(\bar{\mathcal{A}})}$ such that $\mathbb{Q}_{\delta(\mathcal{A})} = \{Q^0, \cdots, Q^{N-1}\}$ with $Q^j = E_0(P^j)$ forms a $\delta(\mathcal{A})$ set for state_{2r_0} (under the guess). Let $C^j = E(P^j)$, $0 \le j < N$. Then we want to get the sequence

$$\Delta_{E_1}(\mathbb{Q}_{\delta(\mathcal{A})}, \mathcal{B}) = \{Q^0(\mathsf{state}_{2(r_0+r_1)}[\mathcal{B}]), \cdots, Q^{N-1}(\mathsf{state}_{2(r_0+r_1)}[\mathcal{B}])\}$$

by decrypting $\{C^0, \cdots, C^{N-1}\}$ with E_2.

For E_2 we introduce a set of 0-1 variables $\mathsf{Vars}(W) = \{W_i[j] : 2(r_0 + r_1) \le i \le 2(r_0 + r_1 + r_2),\ 0 \le j < n_c\}$ and impose a set of constraints on $\mathsf{Vars}(W)$ such that $\mathsf{Vars}(W)$ models the forward determination of the set $\{Q^0(\mathsf{state}_{2(r_0+r_1)}[\mathcal{B}]), \cdots, Q^{N-1}(\mathsf{state}_{2(r_0+r_1)}[\mathcal{B}])\}$ with $\mathcal{B} = \{j : Y_{2(r_0+r_1)}[j] = 1,\ 0 \le j < n_c\}$ in rounds $(r_0 + r_1 \to \cdots \to r_0 + r_1 + r_2 - 1)$.

Observation 5 *Given $\{C^0, \cdots, C^{N-1}\}$, the set*

$$\mathsf{Guess}(E_2) = \{Q^0(\mathsf{state}_{2i}[j]) : W_{2i}[j] = 1, r_0 + r_1 \le i < r_0 + r_1 + r_2, 0 \le j < n_c\}$$

of $\sum_{i=r_0+r_1}^{r_0+r_1+r_2-1} \sum_{j=0}^{n_c-1} W_{2i}[j]$ c-bit words needs to be guessed to determine the sequence

$$\Delta_{E_1}(\mathbb{Q}_{\delta(\mathcal{A})}, \mathcal{B}) = [C^0 \oplus C^1(\mathsf{state}_{2(r_0+r_1)}[\mathcal{B}]), \cdots, C^0 \oplus C^{N-1}(\mathsf{state}_{2(r_0+r_1)}[\mathcal{B}])].$$

Remark. There is still a gap between $\mathsf{Guess}(E_i)$ and k_{E_i} for $i \in \{0, 2\}$. To perform the attack (see Sect. 3), we need to identify k_{E_i} rather than $\mathsf{Guess}(E_i)$. As we will show in Sect. 7.1, Sect. 7.2 and Sect. 7.3, it is fairly straightforward to convert $\mathsf{Guess}(E_i)$ to k_{E_i}.

6 How to Use the Modelling Technique in Practice?

The modelling technique for $\mathcal{DS}$-MITM attack can be applied in several scenarios. In the following, we identify two of them and give a discussion of possible extensions.

6.1 Enumeration of $\mathcal{DS}$-MITM Distinguishers

In Sect. 5, the descriptions of the modelling of E_1 (the distinguisher part) and the outer rounds (E_0 and E_2) are intentionally separated to have a method whose only purpose is to search for $\mathcal{DS}$-MITM distinguishers.

When we target a cipher with $\mathcal{DS}$-MITM attack, probably the first that come into mind is to identify a $\mathcal{DS}$-MITM distinguisher covering as many rounds as possible. To this end, we can build a model with the method presented in Sect. 5 for k rounds of the target cipher, and add one more constraint dictating that

$$\mathsf{Deg}(\mathcal{A},\mathcal{B}) = \sum_{i=r_0}^{r_0+r_1-1} \sum_{j=0}^{n_c-1} Z_{2i}[j] < |K|_c$$

to prevent the complexity of the offline phase from being too high, where $|K|_c$ is the number of c-bit words in the master key of the target cipher. Then we can enumerate all solutions using a constraint solver. If the solutions of the model lead to valid distinguishers, we can increase k and try to find distinguishers covering more rounds.

6.2 Fast Prototyping for $\mathcal{DS}$-MITM Attacks

Given a keyed permutation $E = E_2 \circ E_1 \circ E_0$, it is difficult to determine which $\mathcal{DS}$-MITM distinguisher covering E_1 will lead to the best attack, though intuitively a distinguisher $(\mathcal{A},\mathcal{B},\mathsf{Deg}(\mathcal{A},\mathcal{B}))$ with smaller $\mathsf{Deg}(\mathcal{A},\mathcal{B})$ is preferred. In this situation, we can set up a model for the whole $E_2 \circ E_1 \circ E_0$ with the constraints

$$\begin{cases} \mathsf{Deg}(\mathcal{A},\mathcal{B}) = \sum_{i=r_0}^{r_0+r_1-1} \sum_{j=0}^{n_c-1} Z_{2i}[j] < |K|_c \\ \sum_{i=1}^{r_0-1} \sum_{j=0}^{n_c-1} M_{2i}[j] + \sum_{i=r_0+r_1}^{r_0+r_1+r_2-1} \sum_{j=0}^{n_c-1} W_{2i}[j] < |K|_c \end{cases}$$

The resolution of the model leads to both a distinguisher covering E_1 and an attack based on the distinguisher simultaneously, which should be very useful in fast prototyping of $\mathcal{DS}$-MITM attack in the analysis and design of block ciphers. Note that the output of the tool is a distinguisher $(\mathcal{A},\mathcal{B},\mathsf{Deg}(\mathcal{A},\mathcal{B}))$ and the secret information $\mathsf{Guess}(E_0)$ and $\mathsf{Guess}(E_2)$, which needs to be converted to k_{E_0} and k_{E_2} automatically or manually. Then the so-called key-bridging technique [29, 47] can be applied to give an estimation of $|k_{E_0} \cup k_{E_2}|$.

Another strategy is to find all k-round distinguishers $(\mathcal{A},\mathcal{B},\mathsf{Deg}(\mathcal{A},\mathcal{B}))$ with $\mathsf{Deg}(\mathcal{A},\mathcal{B}) < d$ for some integer d. Then various generic or dedicated optimization techniques [29] (some of which may be unknown at present) can be applied based on these distinguishers to see which one leads to the best attack.

7 Applications

7.1 Application to SKINNY

In this section, we apply our method to SKINNY-128-384 (the TK3 version with 128-bit block size, 384-bit key, and 0-bit tweak) to have a concrete example demonstrating the method presented in Sect. 4. The specification of SKINNY can be found in [44], and we omit it from this paper due to space restrictions.

The indexing scheme we used for analyzing SKINNY is illustrated in Fig. 9, which is essentially the same as Fig. 1, except that the states are drawn as 4×4 squares and the NL layer is composed of a parallel application of 16 Sboxes and a shift row operation.

To model an r-round $\mathcal{DS}$-MITM distinguisher, we introduce 3 sets $\mathsf{Vars}(X)$, $\mathsf{Vars}(Y)$, and $\mathsf{Vars}(Z)$ of variables for all the states involved in rounds (k, $k+1$, $\cdots$, $k+r-1$), where $\mathsf{Vars}(X) = \{X_i[j] : 2k \leq i \leq 2(k+r), 0 \leq j < n_c\}$ models the forward differential, $\mathsf{Vars}(Y) = \{Y_i[j] : 2k \leq i \leq 2(k+r), 0 \leq j < n_c\}$ models the backward determination relationship, and $\mathsf{Vars}(Z) = \{Z_i[j] : 2k \leq i \leq 2(k+r), 0 \leq j < n_c\}$ such that $Z_i[j] = 1$ if and only if $X_i[j] = Y_i[j] = 1$. Note that the logical statement of $Z_i[j]$ can be converted into allowed tuples of $(Z_i[j], X_i[j], Y_i[j])$, that is $(Z_i[j], X_i[j], Y_i[j]) \in \{(0,0,0), (0,0,1), (0,1,0), (1,1,1)\}$, which can be modeled in CP or MILP trivially [14, 10]. So the only question left is what kind of constraints should be imposed on $\mathsf{Vars}(X)$ and $\mathsf{Vars}(Y)$ such that they model the intended properties.

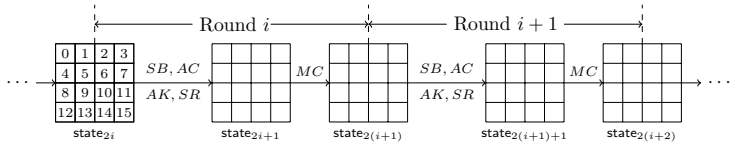


Fig. 9: The indexing scheme used for the rounds, states, and words of SKINNY

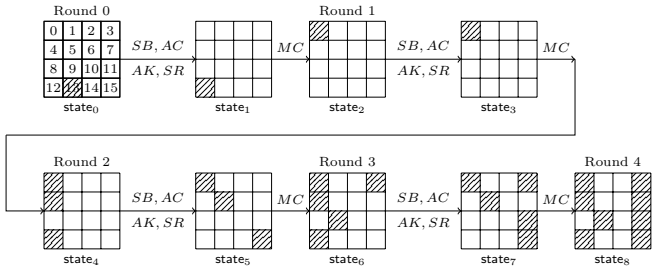


Fig. 10: Forward differential of a $\delta(\mathcal{A})$ set for state_0 in rounds $(0 \to 1 \to 2 \to 3)$ with $\mathcal{A} = [13]$

The constraints imposed on $\mathsf{Vars}(X)$. Firstly, according to the definition of forward differential and the SB, AC, AK, SR operations of SKINNY, we have $X_{2i+1}[4a+b] = X_{2i}[4a+(b-a) \mod 4]$ for $k \leq i < k+r$, where $a, b \in \{0,1,2,3\}$ are used to index the rows and columns of a state respectively. Secondly, for every column $b \in \{0,1,2,3\}$ and $k \leq i < k+r$, we impose the following constraints due to the MC operation

- $X_{2(i+1)}[b] = 0$ if and only if $X_{2i+1}[b] = X_{2i+1}[b+8] = X_{2i+1}[b+12] = 0$;
- $X_{2(i+1)}[b+4] = X_{2i+1}[b]$;
- $X_{2(i+1)}[b+8] = 0$ if and only if $X_{2i+1}[b+4] = X_{2i+1}[b+8] = 0$;
- $X_{2(i+1)}[b+12] = 0$ if and only if $X_{2i+1}[b] = X_{2i+1}[b+8] = 0$.

Note that all constraints given in the above can be converted to allowed tuples of some variables and therefore can be easily modeled by the CP approach. An example solution of a set of variables modelling the forward differential of 4-round SKINNY is visualized in Fig. 10.

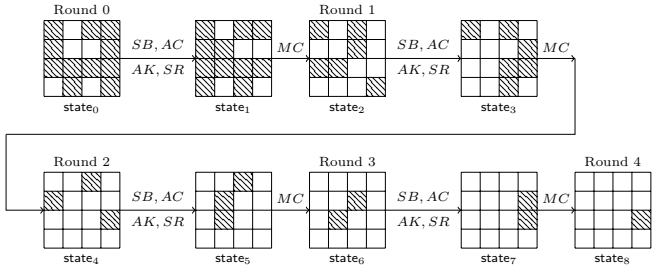


Fig. 11: The backward determination relationship of $\{Q^0(\mathsf{state}_8[\mathcal{B}]), \cdots, Q^{N-1}(\mathsf{state}_8[\mathcal{B}])\}$ for state_8 in rounds $(3 \to 2 \to 1 \to 0)$ with $\mathcal{B} = [11]$

The constraints imposed on $\mathsf{Vars}(Y)$. Similarly, according to the definition of backward determination relationship and the SB, AC, AK, SR operations of SKINNY, we have $Y_{2i+1}[4a+b] = Y_{2i}[4a+(b-a) \mod 4]$ for $k \leq i < k+r$ and $a, b \in \{0,1,2,3\}$. In addition, for every column $b \in \{0,1,2,3\}$ and $k \leq i < k+r$, we impose the following constraints

- $Y_{2i+1}[b] = 0$ if and only if $Y_{2(i+1)}[b] = Y_{2(i+1)}[b+4] = Y_{2(i+1)}[b+12] = 0$;
- $Y_{2i+1}[b+4] = Y_{2(i+1)}[b+8]$;
- $Y_{2i+1}[b+8] = 0$ if and only if $Y_{2(i+1)}[b] = Y_{2(i+1)}[b+8] = Y_{2(i+1)}[b+12] = 0$;
- $Y_{2i+1}[b+12] = Y_{2(i+1)}[b]$.

An example solution of a set of variables modelling the backward determination relationship of 4-round SKINNY is visualized in Fig. 11. According to the constraints imposed on $\mathsf{Vars}(Z)$, if $\mathsf{Vars}(X)$ and $\mathsf{Vars}(Y)$ are assigned to values as illustrated in Fig. 10 and Fig. 11 respectively, then we can derive the values of $\mathsf{Vars}(Z)$ by superposition of Fig. 10 and Fig. 11, as depicted in Fig. 12.

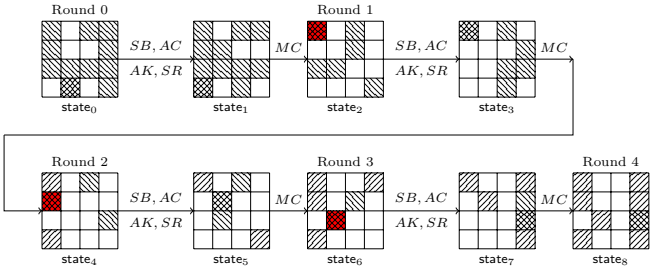


Fig. 12: A visualization of an instantiation of $\mathsf{Vars}(Z)$ according to the values assigned to $\mathsf{Vars}(X)$ and $\mathsf{Vars}(Y)$, which can be regarded as a superposition of Fig. 10 and Fig. 11

Additional constraints. We require $\sum X_i[j] \neq 0$, $\sum Y_i[j] \neq 0$, and $\sum Z_i[j] \neq 0$ to exclude the trivial solution where all variables are assigned to 0. Also, to make the time complexity of the offline phase not exceeding the complexity of the exhaustive search attack, we require $\sum Z_{2i}[j] \leq |K|_c = 384/8 = 48$.

Objective functions. The objective function is to minimize $\sum_{i=k}^{k+r-1} \sum_{j=0}^{15} Z_{2i}[j]$ to make $\mathsf{Deg}(\mathcal{A}, \mathcal{B})$ as small as possible.

Cipher-specific constraints. For SKINNY, we can reduce the number of guessed parameters by exploiting the properties of its linear transformation. According to the MC operation of SKINNY, for an intermediate value Q and $b \in \{0, 1, 2, 3\}$, we have

$$\begin{cases} Q(\mathsf{state}_{2(i+1)}[b]) & = Q(\mathsf{state}_{2i+1}[b]) + Q(\mathsf{state}_{2i+1}[b+8]) + Q(\mathsf{state}_{2i+1}[b+12]) \\ Q(\mathsf{state}_{2(i+1)}[b+4]) & = Q(\mathsf{state}_{2i+1}[b]) \\ Q(\mathsf{state}_{2(i+1)}[b+8]) & = Q(\mathsf{state}_{2i+1}[b+4]) + Q(\mathsf{state}_{2i+1}[b+8]) \\ Q(\mathsf{state}_{2(i+1)}[b+12]) & = Q(\mathsf{state}_{2i+1}[b]) + Q(\mathsf{state}_{2i+1}[b+8]) \end{cases}$$

Hence, the tuple $(Q(\mathsf{state}_{2i+1}[b+8]), Q(\mathsf{state}_{2(i+1)}[b+4]), Q(\mathsf{state}_{2(i+1)}[b+12]))$ can be fully determined when any two of the three entries are known. Similarly, the tuple $(Q(\mathsf{state}_{2i+1}[b+12]), Q(\mathsf{state}_{2(i+1)}[b]), Q(\mathsf{state}_{2(i+1)}[b+12]))$ can be fully determined when any two of the three entries are known. To take these facts into account, we introduce two new sets $\{\phi_i : k \leq i < k+r\}$ and $\{\varphi_i : k \leq i < k+r\}$ of 0-1 variables , and include the following constraints for $b \in \{0, 1, 2, 3\}$

- $\phi_i = 1$ if and only if $Z_{2i+1}[b+8] + Z_{2(i+1)}[b+4] + Z_{2(i+1)}[b+12] = 3$;
- $\psi_i = 1$ if and only if $Z_{2i+1}[b+12] + Z_{2(i+1)}[b] + Z_{2(i+1)}[b+12] = 3$;

We also need to set the objective function to minimize

$$\sum_{i=k}^{k+r-1} \sum_{j=0}^{15} Z_{2i}[j] - \sum_{i=k}^{k+r-1} (\phi_i + \psi_i).$$

Using the above model, we can find a $\mathcal{DS}$-MITM distinguisher for 10.5-round SKINNY-128-384 in 2 seconds. In [44], the designers of SKINNY expected that there should be no $\mathcal{DS}$-MITM distinguisher covering more than 10 rounds of SKINNY since partial-matching can work at most $(6-1)+(6-1)=10$ rounds. Hence, our result concretize the 10-round distinguisher, and actually our tool found $\mathcal{DS}$-MITM distinguishers of SKINNY covering more than 10 rounds. An enumeration of all $\mathcal{DS}$-MITM distinguishers covering 10.5-round SKINNY with $40 \leq \mathsf{Deg}(\mathcal{A},\mathcal{B}) \leq 48$ is performed and the results are listed in Table. 2. Note that distinguishers with $\mathsf{Deg}(\mathcal{A},\mathcal{B}) > 48$ are ineffective for an attack. We then try to get an attack on SKINNY by modelling E_1 (the distinguisher part), E_0 and E_2 (the outer rounds) as a whole with the method presented in Sect. 4. We omit the detailed description of the constraints for $\mathsf{Vars}(M)$ and $\mathsf{Vars}(W)$ introduced for E_0 and E_2 since they are similar to the constraints imposed on $\mathsf{Vars}(X)$ and $\mathsf{Vars}(Y)$ given previously. As a result, we identify a $\mathcal{DS}$-MITM attack on 22-round SKINNY-128-384 based on a distinguisher $(\mathcal{A},\mathcal{B},\mathsf{Deg}(\mathcal{A},\mathcal{B}))$ with $\mathcal{A}=$ [14], $\mathcal{B}=$ [7], and $\deg(\mathcal{A},\mathcal{B})=40$, which is shown in Fig. 17 in [supplementary material]. The secret intermediate values $\mathsf{Guess}(E_0)$ and $\mathsf{Guess}(E_2)$ created by P^0 in the outer rounds are presented in Fig. 18 in [supplementary material A]. To perform the attack, we still need to convert $\mathsf{Guess}(E_0)$ and $\mathsf{Guess}(E_2)$ into the secret information of subkeys manually, which is visualized in Fig. 19 in [supplementary material A]. Then we perform the key-bridging technique [29, 47] on k_{in} and k_{out}, and find that $|k_{in} \cup k_{out}| \leq 376$.

Complexity analysis. According to the discussion of Sect. 3.3, in the offline phase, the time complexity is $2^{8\times 40} \times 2^{8\times 1} \times \frac{40}{16\times 22} C_E \approx 2^{324.86} C_E$, and the memory complexity is $(2^8-1)\times 8 \times 1 \times 2^{8\times 40} \approx 2^{330.99}$ bits. In the online phase, the time complexity is $2^{47\times 8} \times 2^{8\times 1} \times \frac{57+64}{22\times 16} C_E \approx 2^{382.46} C_E$. The data complexity of the attack is $2^{8\times 12} = 2^{96}$, which can be obtained from the input state of Fig. 18 in [supplementary material A].

7.2 Application to LBlock

The indexing scheme we used for analyzing LBlock is shown in Fig. 13, where the AK is the subkey xor operation, SB is a parallel application of 8 4×4 S-boxes, and LN is a permutation permuting j to LN[j].

To model an r-round $\mathcal{DS}$-MITM distinguisher of LBlock, we introduce 3 sets $\mathsf{Vars}(X)$, $\mathsf{Vars}(Y)$, and $\mathsf{Vars}(Z)$ of variables for all the states involved in rounds $(k, k+1, \cdots, k+r-1)$, where $\mathsf{Vars}(X) = \{X_i^L[j], X_i^R[j] : k \leq i \leq k+r, 0 \leq j < n_c\} \cup \{X_i^S[j], X_i^M[j] : k \leq i < k+r, 0 \leq j < n_c\}$ models the forward differential, $\mathsf{Vars}(Y) = \{Y_i^L[j], Y_i^R[j] : k \leq i \leq k+r, 0 \leq j < n_c\} \cup \{Y_i^S[j], Y_i^M[j] : k \leq i < k+r, 0 \leq j < n_c\}$ models the backward determination relationship, and $\mathsf{Vars}(Z) = \{Z_i^L[j], Z_i^R[j] : k \leq i \leq k+r, 0 \leq j < n_c\} \cup \{Z_i^S[j], Z_i^M[j] : k \leq i < k+r, 0 \leq j < n_c\}$ such that

- $Z_i^L[j] = 1$ if and only if $X_i^L[j] = Y_i^L[j] = 1$
- $Z_i^R[j] = 1$ if and only if $X_i^R[j] = Y_i^R[j] = 1$

Table 2: An enumeration of all $\mathcal{DS}$-MITM distinguishers for 10.5-round SKINNY-128-384 with $40 \leq \mathsf{Deg}(\mathcal{A}, \mathcal{B}) \leq 48$.

No.	$\mathcal{A}$	$\mathcal{B}$	$\mathsf{Deg}(\mathcal{A}, \mathcal{B})$	No.	$\mathcal{A}$	$\mathcal{B}$	$\mathsf{Deg}(\mathcal{A}, \mathcal{B})$	No.	$\mathcal{A}$	$\mathcal{B}$	$\mathsf{Deg}(\mathcal{A}, \mathcal{B})$
1	[15]	[4]	40	21	[13]	[6, 4]	45	41	[13]	[5]	46
2	[12]	[5]	40	22	[14]	[7, 5]	45	42	[12]	[4]	46
3	[13]	[6]	40	23	[13]	[6, 4]	45	43	[14]	[6]	46
4	[14]	[7]	40	24	[15]	[4, 6]	45	44	[15]	[7]	46
5	[15]	[5]	42	25	[13]	[5]	45	51	[13]	[4, 6]	47
6	[12]	[6]	42	26	[15]	[6]	45	52	[12]	[7, 5]	47
7	[13]	[7]	42	27	[14]	[4]	45	53	[14]	[5, 7]	47
8	[14]	[4]	42	28	[13]	[4]	45	54	[15]	[6, 4]	47
9	[13]	[5]	43	29	[14]	[5]	45	49	[13]	[6]	47
10	[14]	[6]	43	30	[14]	[6]	45	50	[13]	[6]	47
11	[12]	[4]	43	31	[12]	[4]	45	51	[14]	[7]	47
12	[15]	[7]	43	32	[15]	[5]	45	52	[12]	[5]	47
13	[12]	[7]	44	33	[13]	[7]	45	53	[12]	[5]	47
14	[13]	[4]	44	34	[12]	[6]	45	54	[14]	[7]	47
15	[12]	[7]	44	35	[15]	[7]	45	55	[15]	[4]	47
16	[13]	[4]	44	36	[12]	[7]	45	56	[15]	[4]	47
17	[13]	[4]	44	37	[14]	[4, 6]	46	57	[15]	[7, 5]	48
18	[14]	[5]	44	38	[13]	[7, 5]	46	58	[14]	[6, 4]	48
19	[14]	[5]	44	39	[15]	[5, 7]	46	59	[12]	[4, 6]	48
20	[13]	[4]	44	40	[12]	[6, 4]	46	60	[13]	[5, 7]	48

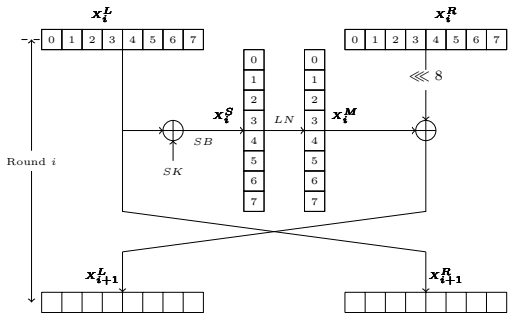


Fig. 13: The indexing scheme used for LBlock

- $Z_i^S[j] = 1$ if and only if $X_i^S[j] = Y_i^S[j] = 1$
- $Z_i^M[j] = 1$ if and only if $X_i^M[j] = Y_i^M[j] = 1$

Note that the logical statement of $\mathsf{Vars}(Z)$ can be converted into allowed tuples, e.g. $(Z_i^L[j], X_i^L[j], Y_i^L[j]) \in \{(0,0,0), (0,0,1), (0,1,0), (1,1,1)\}$, which can be modeled in CP or MILP trivially [10, 14]. So the only question left is what kind of constraints should be imposed on $\mathsf{Vars}(X)$ and $\mathsf{Vars}(Y)$ such that they model the intended properties.

The constraints imposed on $\mathsf{Vars}(X)$. According to the definition of forward differential and the AK, SB, LN, $\lll 8$, XOR operations of LBlock, we have the following constraints

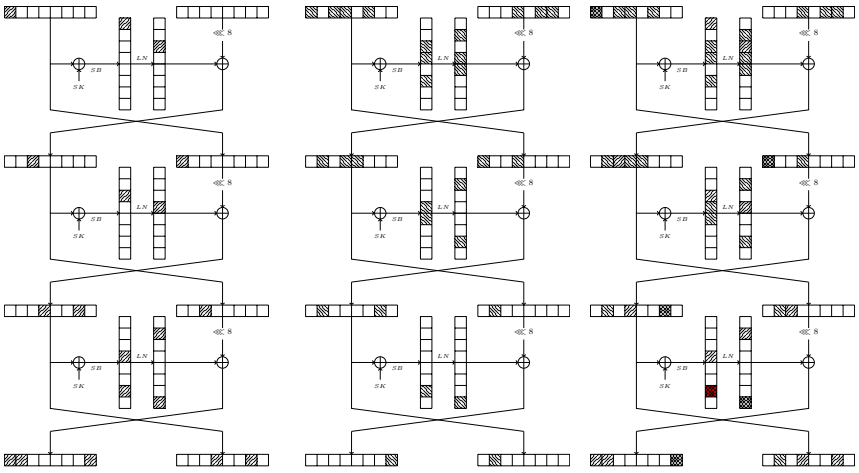
- $X_i^L[j] = X_i^S[j] = X_{i+1}^R[j]$, for $k \leq i < k + r$ and $0 \leq j \leq 7$;
- $X_i^M[LN[j]] = X_i^S[j]$, for $k \leq i < k + r$ and $0 \leq j \leq 7$;

- $X_{i+1}^L[j] = 0$ if and only if $X_i^R[(j+2) \mod 8] = X_i^M[j] = 0$, for $k \leq i < k+r$ and $0 \leq j \leq 7$.

The constraints imposed on $\mathsf{Vars}(Y)$. Similarly, according to the definition of the backward determination relationship and the AK, SB, LN, $\lll 8$, XOR operations of LBlock, we have the following constraints

- For $k \leq i < k+r$ and $0 \leq j \leq 7$, $Y_i^L[j] = 0$ if and only if $Y_{i+1}^R[j] = Y_i^S[j] = 0$;
- $Y_i^M[LN[j]] = Y_i^S[j]$, for $k \leq i < k+r$ and $0 \leq j \leq 7$;
- For XOR and SR operations: $Y_i^M[j] = Y_i^R[(j+2) \mod 8] = Y_{i+1}^L[j]$

According to the constraints imposed on $\mathsf{Vars}(Z)$, if $\mathsf{Vars}(X)$ and $\mathsf{Vars}(Y)$ are assigned to values as illustrated in Fig. 14a and Fig. 14b, then we can derive the values of $\mathsf{Vars}(Z)$ by superposition of Fig. 14a and Fig. 14b, which is depicted in Fig. 14c.



(a) Forward differential (b) Backward determination (c) $\mathsf{Vars}(Z)$

Fig. 14: An instantiation of the $\mathsf{Vars}(X)$, $\mathsf{Vars}(Y)$ and $\mathsf{Vars}(Z)$

Additional constraints. We require $\sum X_k^L[j] + \sum X_k^R[j] \neq 0$, $\sum Y_{k+r}^L[j] + \sum Y_{k+r}^R[j] \neq 0$, to exclude the trivial solution where all variables are assigned to 0. Also, to make the time complexity of the offline phase not exceeding the complexity of the exhaustive search, we require $\sum Z_i^S[j] < |K|_c = 80/4 = 20$.

Objective functions. The objective function is to minimize $\sum_{i=k}^{k+r-1} \sum_{j=0}^{7} Z_i^S[j]$ to make $\mathsf{Deg}(\mathcal{A}, \mathcal{B})$ as small as possible.

By integrating the above model with the models of E_0 and E_2 with some simple tweak, we identify a $\mathcal{DS}$-MITM attack on 21-round LBLOCK. The distinguisher used in the attack is an 11-round $\mathcal{DS}$-MITM distinguisher $(\mathcal{A}, \mathcal{B}, \mathsf{Deg}(\mathcal{A}, \mathcal{B}))$

with $\mathcal{A} = [12]$, $\mathcal{B} = [12]$, and $\deg(\mathcal{A}, \mathcal{B}) = 14$, which is shown in Fig. 20 in [supplementary material]. The secret intermediate values $\mathsf{Guess}(E_0)$ and $\mathsf{Guess}(E_2)$ created by P^0 in the outer rounds are presented in Fig. 21 in [supplementary material] marked with red color. To perform the attack, we convert $\mathsf{Guess}(E_0)$ and $\mathsf{Guess}(E_2)$ into the secret information of subkeys manually, which is visualized in Fig. 22 in [supplementary material], where there are 22 nibbles in k_{in} and 12 nibbles in k_{out}. Then we perform the key-bridging technique [29, 47] on k_{in} and k_{out}, and find that $|k_{in} \cup k_{out}| \leq 69$, which is illustrated in Fig. 23 in [supplementary material].

Complexity analysis. According to the discussion of Sect. 3.3, in the offline phase, the time complexity is $2^{4\times14} \times 2^{4\times1} \times \frac{14}{21\times8} C_E \approx 2^{56.42} C_E$, and the memory complexity is $(2^4 - 1) \times 4 \times 1 \times 2^{4\times14} \approx 2^{61.91}$ bits. In the online phase, the time complexity is $2^{69} \times 2^{4\times1} \times \frac{12+12}{21\times8} C_E \approx 2^{70.20} C_E$. The data complexity of the attack is $2^{4\times12} = 2^{48}$, which can be obtained from input state (Round 0) of Fig. 21 in [supplementary material].

7.3 Application to TWINE-80

With the method presented in Sect. 4, we find a $\mathcal{DS}$-MITM attack on 20-round TWINE-80 based on a distinguisher $(\mathcal{A}, \mathcal{B}, \mathsf{Deg}(\mathcal{A}, \mathcal{B}))$ with $\mathcal{A} = [3]$, $\mathcal{B} = [9, 13]$, and $\deg(\mathcal{A}, \mathcal{B}) = 19$, which is shown in Fig. 24 in [supplementary material]. The secret intermediate values $\mathsf{Guess}(E_0)$ and $\mathsf{Guess}(E_2)$ created by P^0 in the outer rounds are presented in Fig. 25 in [supplementary material]. To perform the attack, we convert $\mathsf{Guess}(E_0)$ and $\mathsf{Guess}(E_2)$ into the secret information of subkeys manually, which is visualized in Fig. 26 in [supplementary material]. Then we perform the key-bridging technique [29, 47] on k_{in} and k_{out}, and find that $|k_{in} \cup k_{out}| \leq 76$, which is illustrated in Fig. 27 in [supplementary material].

Complexity analysis. According to the discussion of Sect. 3.3, in the offline phase, the time complexity is $2^{4\times19} \times 2^{4\times1} \times \frac{19}{20\times8} C_E \approx 2^{76.93} C_E$, and the memory complexity is $(2^4 - 1) \times 4 \times 2 \times 2^{4\times19} \approx 2^{82.91}$ bits. In the online phase, the time complexity is $2^{76} \times 2^{4\times1} \times \frac{7+20}{20\times8} C_E \approx 2^{77.44} C_E$. The data complexity of the attack is $2^{4\times8} = 2^{32}$, which can be obtained from input state (Round 0) of Fig. 25 in [supplementary material].

7.4 Applications to AES, ARIA, and SIMON

We also apply our method to AES, ARIA, and SIMON. However, no better result is obtained. Still, We would like to provide some information about our analysis for the sake of completeness.

For AES, our tool can recover the base $\mathcal{DS}$-MITM attacks behind all attacks (including the best ones) presented in [28–30, 53, 54]. However, currently known best attacks on AES exploit the *differential enumeration technique* [28] which our tool cannot take into account automatically. To deal with this, we use a 2-step approach. First, we list all the distinguishers that may lead to a valid

attack using the fact that, at best, the differential enumeration technique can decrease the memory complexity by a factor strictly less than 2^n, where n is the state size. For AES-128 we would only add the constraint dictating that two consecutive states cannot be fully active in the distinguisher. Then in a second step, we can obtain the concrete complexities of the attacks derived from the distinguishers by applying known techniques. Usually, the distinguisher leading to the best attack has the lowest number of active bytes. But some manual work is inevitable to really optimize the attacks. Actually, during our analysis, our code generates figures based on the distinguishers automatically, which greatly facilitates subsequent manual analysis and the checking of correctness. Note that the first step alone can be used to get an upper bound on the number of rounds one may attack (independent of any tricks involving manual work): if there is no distinghuisher then there is no attack.

For ARIA, we obtain the same result presented in [55]. Unlike the other targets presented in the paper which are modeled using MILP, we also provide a Choco [56] implementation for finding the $\mathcal{DS}$-MITM distinguishers of the ARIA cipher to show that we can choose from MILP/SAT/SMT/CP as the modeling language freely. This fact is important since the solvers are being improved constantly, and thus we can expect the resolution of more difficult instances in the future. We also try our tool on bit-oriented ciphers like SIMON. For SIMON32/64, only an 8-round $\mathcal{DS}$-MITM distinguisher is identified, which is far less than the rounds can be penetrated by differential attacks.

8 Applications in the Process of Block Cipher Design

In the design process, the designer typically first fixes the general structure of the block cipher. Then she or he tries to identify the optimal local components in terms of security, efficiency, power consumption etc. by a tweaking-and-analysis style iterative approach. Therefore, it is important to have efficient tools at hands such that a thorough exploration of the design space can be performed. In this section, we show that our tool can be applied in this situation by tweaking the block ciphers LBlock and TWINE. Note that unlike Ivica's tool [57], where nature-inspired meta-heuristics are employed, our method essentially performs an $\mathcal{DS}$-MITM distinguishing attack for each possible instantiation of the target cipher, and pick the optimal ones according to the results.

For LBlock-80, we tweak the 8-nibble to 8-nibble permutation. We exhaustively search for the 11-round $\mathcal{DS}$-MITM distinguishers with the lowest $\mathsf{Deg}(\mathcal{A}, \mathcal{B})$ for the $8! = 40320$ cases. The distribution of the 40320 cases in terms of $\mathsf{Deg}(\mathcal{A}, \mathcal{B})$ is shown in Fig. 15. According to Fig. 15, we can make several interesting observations. Firstly, there are many very weak permutations with very low $\deg(\mathcal{A}, \mathcal{B})$ which obviously should be avoided. In extreme cases, there are 12560 permutations with $\mathsf{Deg}(\mathcal{A}, \mathcal{B}) = 0$. Secondly, the number of permutations with high resistance against $\mathcal{DS}$-MITM attack is small. There are 64 permutations among the 40320 ones with $\mathsf{Deg}(\mathcal{A}, \mathcal{B}) = 14$, and actually the original permutation of LBlock is chosen from these good permutations.

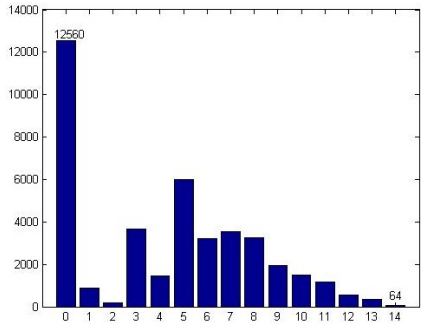


Fig. 15: The horizontal axis shows $\mathsf{Deg}(\mathcal{A}, \mathcal{B})$ of the 11-round distinguisher (N/A means there is no valid distinguisher found), while the vertical axis indicates the corresponding numbers of permutations

For TWINE-80, we tweak the word shuffle of 16 nibbles. There are totally $16! \approx 2^{44.25}$ possibilities, which is out of reach of our computational power. However, according to [58], we only need to consider the $8! \times 8!$ *even-odd* shuffles. Let $P = (P_0, P_1)$, be the word shuffle where P_0 is the shuffle of all even positions while P_1 is the shuffle of all odd positions. Then it can be shown that (P_0, P_1) is equivalent to $(Q \circ P_1 \circ Q^{-1}, Q \circ P_2 \circ Q^{-1})$, where Q is an arbitrary word shuffle. Therefore, the number of cases can be further reduced since the $8! \times 8!$ shuffles can be divided into $22 \times 8! = 887040$ equivalent classes with respect to the $\mathcal{DS}$-MITM attack. We exhaustively search for the 11-round $\mathcal{DS}$-MITM distinguishers with the lowest $\mathsf{Deg}(\mathcal{A}, \mathcal{B})$ for the 887040 cases. The distribution of the 887040 cases in terms of $\mathsf{Deg}(\mathcal{A}, \mathcal{B})$ is shown in Fig. 16. According to

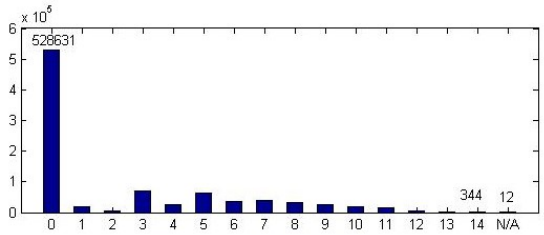


Fig. 16: The horizontal axis shows $\mathsf{Deg}(\mathcal{A}, \mathcal{B})$ of the 11-round distinguisher (N/A means there is no valid distinguisher found), while the vertical axis indicates the corresponding numbers of permutations

Fig. 16, we can make several interesting observations. Firstly, there are many very weak permutations with very low $\deg(\mathcal{A}, \mathcal{B})$ which obviously should be avoided. In extreme cases, there are 528631 permutations with $\mathsf{Deg}(\mathcal{A}, \mathcal{B}) = 0$. Secondly, the number of permutations with high resistance against $\mathcal{DS}$-MITM

attack is small. There are only 344 permutations among the 887040 ones with $\mathsf{Deg}(\mathcal{A}, \mathcal{B}) = 14$, and actually the original permutation of TWINE is chosen from these good permutations. Finally, we identify a set of 12 permutations for which we can not find any 11-round distinguisher, indicating that they are stronger than the original permutation in TWINE-80 with respect to the $\mathcal{DS}$-MITM attack.

Since both the $\mathcal{DS}$-MITM attack in this paper and the word-oriented truncated impossible differential attack are structure attacks whose effectiveness is not affected by the details of the underlying S-boxes, we are wondering whether there is a set of strongest word shuffles with respect to the $\mathcal{DS}$-MITM attack and impossible differential attack simultaneously. We exhaustively analysis the 887040 TWINE variants. It turns out that for any variant there is a 14-round impossible differential, and there are 144 variants with no 15-round impossible differential. Finally, we identify a set of 12 word shuffles with no 15-round impossible differential and no 11-round $\mathcal{DS}$-MITM distinguisher (listed in Table. 4 in [supplementary material]). Note that the word shuffle used in TWINE is not in this set. Therefore, it is potentially better to use one from these 12 word shuffles.

9 Conclusion and Discussion

In this paper, we present the first tool for automatic Demirci-Selçuk meet-in-the-middle analysis based on constraint programming. In our approach, the formulation and resolution of the model are decoupled. Hence, the only thing needs to do by the cryptanalysts is to specify the problem in some modeling language, and the remaining work can be done with any open-source or commercially available constraint solvers. This approach should be very useful in fast prototyping block cipher designs. Finally, we would like to identify a set of limitations of our approach, overcoming which is left for future work.

Limitations. First of all, some important techniques for improving the $\mathcal{DS}$-MITM attack have not been integrated into our framework yet, including (but not limited to) the differential enumeration technique, and using several distinguishers in parallel. Secondly, we cannot guarantee the optimality of the attacks produced by our tool, due to the heuristic natures of the key-recovery process, and the lack of automatically considering cipher specific properties. Finally, we do not know how to apply our method to ARX based constructions.

Acknowledgments. The authors thank the anonymous reviewers for many helpful comments, and Gaëtan Leurent for careful reading and shepherding our paper. The work is supported by the Chinese Major Program of National Cryptography Development Foundation (Grant No. MMJJ20180102), the National Natural Science Foundation of China (61732021, 61802400, 61772519, 61802399), the Youth Innovation Promotion Association of Chinese Academy of Sciences, and the Institute of Information Engineering, CAS (Grant No. Y7Z0251103). Patrick Derbez is supported by the French Agence Nationale de la Recherche through the CryptAudit project under Contract ANR-17-CE39-0003.

References

1. Biham, E., Shamir, A.: Differential cryptanalysis of DES-like cryptosystems. Journal of Cryptology **4**(1) (1991) 3–72
2. Matsui, M.: Linear cryptanalysis method for DES cipher. In: Advances in Cryptology–EUROCRYPT 1993, Springer (1994) 386–397
3. Daemen, J., Knudsen, L.R., Rijmen, V.: The block cipher square. In: Fast Software Encryption, 4th International Workshop, FSE '97, Haifa, Israel, January 20-22, 1997, Proceedings. (1997) 149–165
4. Knudsen, L.R., Wagner, D.: Integral cryptanalysis. In: Fast Software Encryption, 9th International Workshop, FSE 2002, Leuven, Belgium, February 4-6, 2002, Revised Papers. (2002) 112–127
5. Todo, Y.: Structural evaluation by generalized integral property. In: Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I. (2015) 287–314
6. Liu, Y., Wang, Q., Rijmen, V.: Automatic search of linear trails in ARX with applications to SPECK and chaskey. In: Applied Cryptography and Network Security - 14th International Conference, ACNS 2016, Guildford, UK, June 19-22, 2016. Proceedings. (2016) 485–499
7. Mouha, N., Preneel, B.: Towards finding optimal differential characteristics for ARX: Application to Salsa20. IACR Cryptology ePrint Archive, Report 2013/328 (2013) http://eprint.iacr.org/2013/328.
8. Kölbl, S., Leander, G., Tiessen, T.: Observations on the SIMON block cipher family. In: Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I. (2015) 161–185
9. Mouha, N., Wang, Q., Gu, D., Preneel, B.: Differential and linear cryptanalysis using mixed-integer linear programming. In: Information Security and Cryptology –ISC 2012, Springer (2012) 57–76
10. Sun, S., Hu, L., Wang, P., Qiao, K., Ma, X., Song, L.: Automatic security evaluation and (related-key) differential characteristic search: Application to SIMON, PRESENT, LBlock, DES(L) and other bit-oriented block ciphers. In: Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I. (2014) 158–178
11. Xiang, Z., Zhang, W., Bao, Z., Lin, D.: Applying MILP method to searching integral distinguishers based on division property for 6 lightweight block ciphers. In: Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I. (2016) 648–678
12. Sasaki, Y., Todo, Y.: New impossible differential search tool from design and cryptanalysis aspects - revealing structural properties of several ciphers. In: Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part III. (2017) 185–215
13. Fu, K., Wang, M., Guo, Y., Sun, S., Hu, L.: MILP-Based Automatic Search Algorithms for Differential and Linear Trails for Speck. In: Fast Software Encryption - 23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers. (2016) 268–288

14. Gerault, D., Minier, M., Solnon, C.: Constraint programming models for chosen key differential cryptanalysis. In: Principles and Practice of Constraint Programming - 22nd International Conference, CP 2016, Toulouse, France, September 5-9, 2016, Proceedings. (2016) 584–601
15. Sun, S., Gerault, D., Lafourcade, P., Yang, Q., Todo, Y., Qiao, K., Hu, L.: Analysis of AES, SKINNY, and others with constraint programming. IACR Trans. Symmetric Cryptol. **2017**(1) (2017) 281–306
16. Cui, T., Jia, K., Fu, K., Chen, S., Wang, M.: New automatic search tool for impossible differentials and zero-correlation linear approximations. IACR Cryptology ePrint Archive **2016** (2016) 689
17. Matsui, M.: On correlation between the order of S-boxes and the strength of DES. In: Advances in Cryptology–EUROCRYPT 1994, Springer (1995) 366–375
18. Dobraunig, C., Eichlseder, M., Mendel, F.: Heuristic tool for linear cryptanalysis with applications to CAESAR candidates. In: Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II. (2015) 490–509
19. Biryukov, A., Velichkov, V.: Automatic search for differential trails in ARX ciphers. In: Topics in Cryptology–CT-RSA 2014. Springer (2014) 227–250
20. Biryukov, A., Nikolić, I.: Search for related-key differential characteristics in DES-like ciphers. In: Fast Software Encryption – FSE 2011, Springer (2011) 18–34
21. Fouque, P.A., Jean, J., Peyrin, T.: Structural Evaluation of AES and Chosen-Key Distinguisher of 9-Round AES-128. In: Advances in Cryptology–CRYPTO 2013. Springer (2013) 183–203
22. Bouillaguet, C., Derbez, P., Fouque, P.A.: Automatic search of attacks on round-reduced AES and applications. In: CRYPTO 2011, Springer (2011) 169–187
23. Dobraunig, C., Eichlseder, M., Mendel, F.: Analysis of SHA-512/224 and SHA-512/256. In: Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II. (2015) 612–630
24. Mella, S., Daemen, J., Assche, G.V.: New techniques for trail bounds and application to differential trails in Keccak. IACR Trans. Symmetric Cryptol. **2017**(1) (2017) 329–357
25. Freuder, E.C.: In pursuit of the holy grail. Constraints **2**(1) (1997) 57–61
26. Demirci, H., Selçuk, A.A.: A meet-in-the-middle attack on 8-round AES. In: Fast Software Encryption, 15th International Workshop, FSE 2008, Lausanne, Switzerland, February 10-13, 2008, Revised Selected Papers. (2008) 116–126
27. Daemen, J., Rijmen, V.: The Design of Rijndael: AES - The Advanced Encryption Standard. Information Security and Cryptography. Springer (2002)
28. Dunkelman, O., Keller, N., Shamir, A.: Improved single-key attacks on 8-round AES-192 and AES-256. In: Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings. (2010) 158–176
29. Derbez, P., Fouque, P., Jean, J.: Improved key recovery attacks on reduced-round AES in the single-key setting. In: Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings. (2013) 371–387
30. 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. (2013) 541–560
31. 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. (2016) 157–184
32. Li, R., Jin, C.: Meet-in-the-middle attacks on 10-round AES-256. Des. Codes Cryptography **80**(3) (2016) 459–471
33. Derbez, P., Perrin, L.: Meet-in-the-middle attacks and structural analysis of round-reduced PRINCE. In: Fast Software Encryption - 22nd International Workshop, FSE 2015, Istanbul, Turkey, March 8-11, 2015, Revised Selected Papers. (2015) 190–216
34. Biryukov, A., Derbez, P., Perrin, L.: Differential analysis and meet-in-the-middle attack against round-reduced TWINE. In: Fast Software Encryption - 22nd International Workshop, FSE 2015, Istanbul, Turkey, March 8-11, 2015, Revised Selected Papers. (2015) 3–27
35. Li, L., Jia, K., Wang, X., Dong, X.: Meet-in-the-middle technique for truncated differential and its applications to CLEFIA and camellia. In: Fast Software Encryption - 22nd International Workshop, FSE 2015, Istanbul, Turkey, March 8-11, 2015, Revised Selected Papers. (2015) 48–70
36. Dong, X., Li, L., Jia, K., Wang, X.: Improved attacks on reduced-round camellia-128/192/256. In: Topics in Cryptology - CT-RSA 2015, The Cryptographer's Track at the RSA Conference 2015, San Francisco, CA, USA, April 20-24, 2015. Proceedings. (2015) 59–83
37. Guo, J., Jean, J., Nikolic, I., Sasaki, Y.: Meet-in-the-middle attacks on generic Feistel constructions. In: Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I. (2014) 458–477
38. Guo, J., Jean, J., Nikolic, I., Sasaki, Y.: Meet-in-the-Middle Attacks on Classes of Contracting and Expanding Feistel Constructions. IACR Trans. Symmetric Cryptol. **2016**(2) (2016) 307–337
39. Diffie, W., Hellman, M.E.: Special feature exhaustive cryptanalysis of the NBS data encryption standard. IEEE Computer **10**(6) (1977) 74–84
40. Bogdanov, A., Rechberger, C.: A 3-subset meet-in-the-middle attack: Cryptanalysis of the lightweight block cipher KTANTAN. In: Selected Areas in Cryptography - 17th International Workshop, SAC 2010, Waterloo, Ontario, Canada, August 12-13, 2010, Revised Selected Papers. (2010) 229–240
41. Aoki, K., Sasaki, Y.: Meet-in-the-middle preimage attacks against reduced SHA-0 and SHA-1. In: Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings. (2009) 70–89
42. Guo, J., Ling, S., Rechberger, C., Wang, H.: Advanced meet-in-the-middle preimage attacks: First results on full Tiger, and improved results on MD4 and SHA-2. In: Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings. (2010) 56–75
43. 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. (2013) 203–223

44. 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. (2016) 123–153
45. Suzaki, T., Minematsu, K., Morioka, S., Kobayashi, E.: TWINE : A lightweight block cipher for multiple platforms. In: Selected Areas in Cryptography, 19th International Conference, SAC 2012, Windsor, ON, Canada, August 15-16, 2012, Revised Selected Papers. (2012) 339–354
46. Wu, W., Zhang, L.: LBlock: A lightweight block cipher. In: Applied Cryptography and Network Security - 9th International Conference, ACNS 2011, Nerja, Spain, June 7-10, 2011. Proceedings. (2011) 327–344
47. Lin, L., Wu, W., Zheng, Y.: Automatic search for key-bridging technique: Applications to lblock and TWINE. In: Fast Software Encryption - 23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers. (2016) 247–267
48. Boura, C., Minier, M., Naya-Plasencia, M., Suder, V.: Improved impossible differential attacks against round-reduced lblock. IACR Cryptology ePrint Archive **2014** (2014) 279
49. Wang, Y., Wu, W.: Improved multidimensional zero-correlation linear cryptanalysis and applications to lblock and TWINE. In: Information Security and Privacy ACISP. (2014) 1–16
50. Zheng, X., Jia, K.: Impossible differential attack on reduced-round TWINE. In: Information Security and Cryptology - ICISC. (2013) 123–143
51. Tolba, M., Abdelkhalek, A., Youssef, A.M.: Impossible differential cryptanalysis of reduced-round SKINNY. In: Progress in Cryptology - AFRICACRYPT. (2017) 117–134
52. Boura, C., Naya-Plasencia, M., Suder, V.: Scrutinizing and Improving Impossible Differential Attacks: Applications to CLEFIA, Camellia, LBlock and Simon. In: Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I. (2014) 179–199
53. 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. (2014) 127–146
54. Li, R., Jin, C.: Meet-in-the-middle attacks on 10-round AES-256. Des. Codes Cryptography **80**(3) (2016) 459–471
55. Akshima, Chang, D., Ghosh, M., Goel, A., Sanadhya, S.K.: Improved meet-in-the-middle attacks on 7 and 8-round ARIA-192 and ARIA-256. In: INDOCRYPT. (2015) 198–217
56. Prud'homme, C., Fages, J.G., Lorca, X.: Choco Documentation. TASC - LS2N CNRS UMR 6241, COSLING S.A.S. (2017)
57. Nikolic, I.: How to use metaheuristics for design of symmetric-key primitives. In: Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part III. (2017) 369–391
58. Suzaki, T., Minematsu, K.: Improving the generalized feistel. In: Fast Software Encryption, 17th International Workshop, FSE 2010, Seoul, Korea, February 7-10, 2010, Revised Selected Papers. (2010) 19–39

A Visualization of the $\mathcal{DS}$-MITM Distinguishers and Attacks [supplementary material]

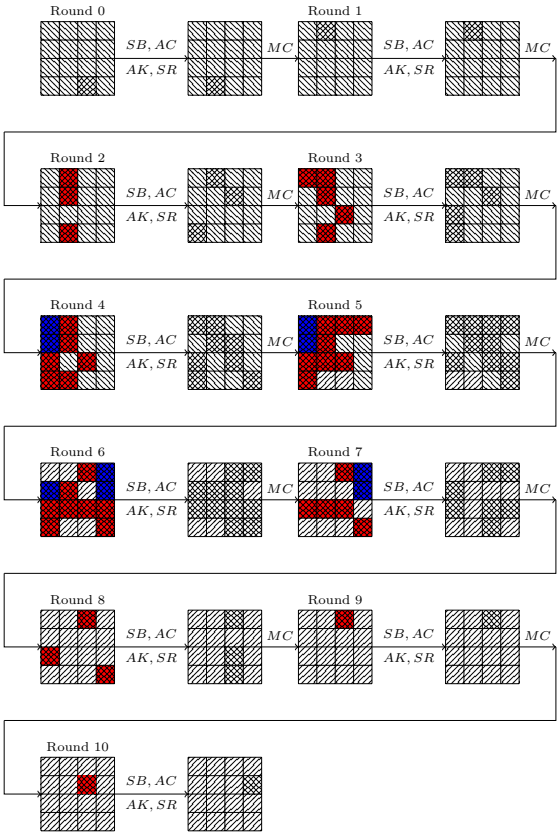


Fig. 17: A $\mathcal{DS}$-MITM distinguisher $(\mathcal{A}, \mathcal{B}, \mathsf{Deg}(\mathcal{A}, \mathcal{B}))$ for 10.5-round SKINNY-128-384 with $\mathcal{A} = [14]$ (the nibble marked with crosshatch in the input state of round 0), $\mathcal{B} = [7]$ (the nibbles marked with crosshatch in the input state of round 11), and $\mathsf{Deg}(\mathcal{A}, \mathcal{B}) = 40$ (the nibbles before the SB, AC, AK, and SR operations mark with red color). The nibble marked with blue color are those redundant bytes required to be guessed if we do not impose the cipher-specific constraints presented in Sect. 7.1.

Table 3: The 64 permutations for which we can not find 12-round distinguishers and have high resistance against 11-round $\mathcal{DS}$-MITM attack, where $(7, 6, 5, 4, 3, 2, 1, 0)$ means that 0 is permuted to 7, 1 is permuted to 6, and so on.

No.	Permutation	No.	Permutation	No.	Permutation	No.	Permutation
1	(7,6,5,3,4,1,2,0)	17	(7,6,2,4,0,5,1,3)	33	(7,5,6,3,4,2,1,0)	49	(7,5,2,0,3,1,6,4)
2	(7,2,4,1,3,6,0,5)	18	(7,2,0,5,3,6,4,1)	34	(7,1,5,4,0,2,6,3)	50	(7,1,6,0,3,5,2,4)
3	(6,4,7,5,2,0,3,1)	19	(6,3,5,0,2,7,1,4)	35	(6,3,1,4,2,7,5,0)	51	(6,0,3,5,2,4,7,1)
4	(5,7,4,6,1,3,0,2)	20	(5,4,3,2,0,1,6,7)	36	(5,4,2,3,0,1,7,6)	52	(5,3,0,6,1,7,4,2)
5	(5,1,3,7,4,0,2,6)	21	(5,1,6,2,4,0,7,3)	37	(5,0,2,7,1,4,6,3)	53	(5,0,6,3,1,4,2,7)
6	(4,6,2,7,3,5,1,0)	22	(4,6,1,3,0,2,5,7)	38	(4,5,3,2,1,0,6,7)	54	(4,5,2,3,1,0,7,6)
7	(4,2,5,3,0,6,1,7)	23	(4,2,1,0,7,5,6,3)	39	(4,1,5,7,3,2,6,0)	55	(4,1,7,2,0,5,3,6)
8	(4,1,3,6,0,5,7,2)	24	(4,1,2,0,7,6,5,3)	40	(4,0,7,3,5,1,6,2)	56	(4,0,2,6,5,1,3,7)
9	(3,6,4,1,7,2,0,5)	25	(3,6,0,5,7,2,4,1)	41	(3,5,2,4,7,1,6,0)	57	(3,5,1,0,4,6,2,7)
10	(3,2,6,0,4,1,5,7)	26	(3,2,1,7,0,5,6,4)	42	(3,1,2,7,0,6,5,4)	58	(3,1,6,4,7,5,2,0)
11	(2,4,7,1,6,0,3,5)	27	(2,7,5,0,6,3,1,4)	43	(2,7,1,4,6,3,5,0)	59	(2,0,3,1,6,4,7,5)
12	(1,5,2,6,0,4,3,7)	28	(1,5,7,3,0,4,6,2)	44	(1,4,6,3,5,0,2,7)	60	(1,4,2,7,5,0,6,3)
13	(1,3,0,2,5,7,4,6)	29	(1,7,4,2,5,3,0,6)	45	(1,0,7,6,4,5,2,3)	61	(1,0,6,7,4,5,3,2)
14	(0,6,5,4,3,1,2,7)	30	(0,6,1,7,4,2,5,3)	46	(0,5,6,4,3,2,1,7)	62	(0,5,3,6,4,1,7,2)
15	(0,5,1,3,7,6,2,4)	31	(0,5,7,2,4,1,3,6)	47	(0,4,6,2,1,5,7,3)	63	(0,4,3,7,1,5,2,6)
16	(0,2,5,7,4,6,1,3)	32	(0,2,6,3,7,1,5,4)	48	(0,1,6,7,5,4,3,2)	64	(0,1,7,6,5,4,2,3)

Table 4: The 12 strongest word shuffles with respect to both $\mathcal{DS}$-MITM attack and impossible differential attack.

No.	Permutation	No.	Permutation
1	(15,2,13,4,11,6,3,8,1,10,5,0,7,12,9,14)	7	(7,2,13,4,15,6,1,8,5,10,3,0,11,12,9,14)
2	(15,2,9,4,1,6,11,8,3,10,13,0,7,12,5,14)	8	(7,2,11,4,9,6,1,8,15,10,13,0,3,12,5,14)
3	(13,2,15,4,11,6,3,8,1,10,5,0,9,12,7,14)	9	(7,2,11,4,9,6,1,8,13,10,15,0,5,12,3,14)
4	(13,2,9,4,1,6,11,8,3,10,15,0,5,12,7,14)	10	(7,2,15,4,13,6,1,8,5,10,3,0,9,12,11,14)
5	(9,2,7,4,11,6,15,8,13,10,5,0,1,12,3,14)	11	(5,2,9,4,13,6,15,8,3,10,7,0,1,12,11,14)
6	(9,2,7,4,11,6,13,8,15,10,5,0,3,12,1,14)	12	(5,2,9,4,15,6,13,8,3,10,7,0,11,12,1,14)

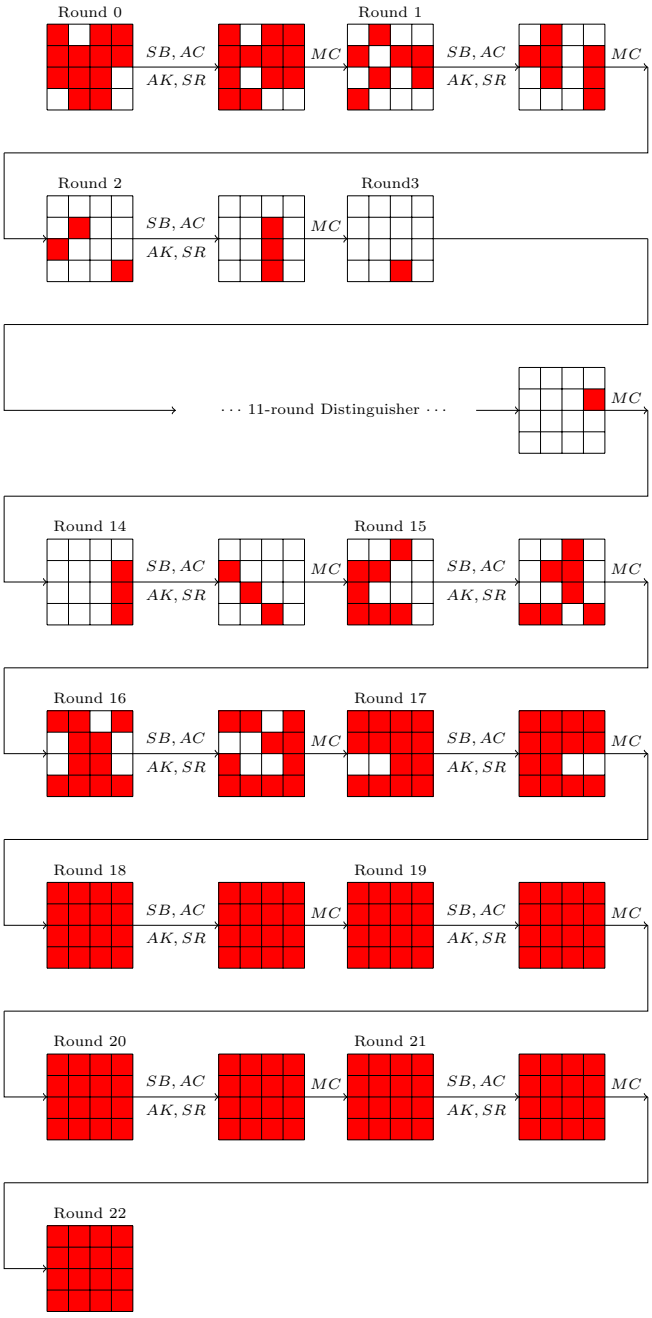


Fig. 18: An illustration of the backward differential and forward determination relationship in the outer rounds of SKINNY with the $\mathcal{DS}$-MITM distinguisher presented in Fig. 17 placed at E_1. The bytes in $\mathsf{Guess}(E_0)$ and $\mathsf{Guess}(E_2)$ are marked with red color. From the input state of round 0, we know that $\bar{\mathcal{A}} = [0, 2, 3, 4, 5, 6, 7, 8, 9, 10, 13, 14]$. Hence, the data complexity is $2^{8\times 12} = 2^{96}$.

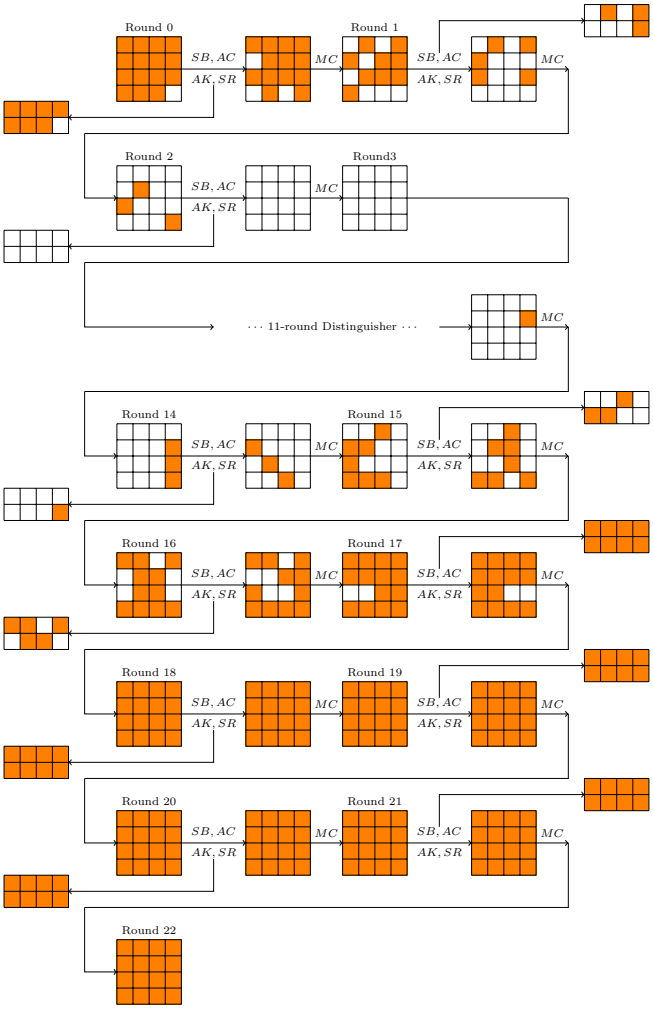


Fig. 19: Derive k_{E_0} and k_{E_2} from Fig. 18. The subkey nibbles marked with orange color are the secret-key information we need to guess. With the knowledge of these bytes, we can derive the values of all the bytes of P^0 marked with orange color in this figure, from which we can determine all the bytes in $\mathsf{Guess}(E_0)$ and $\mathsf{Guess}(E_2)$ shown in Fig. 18.

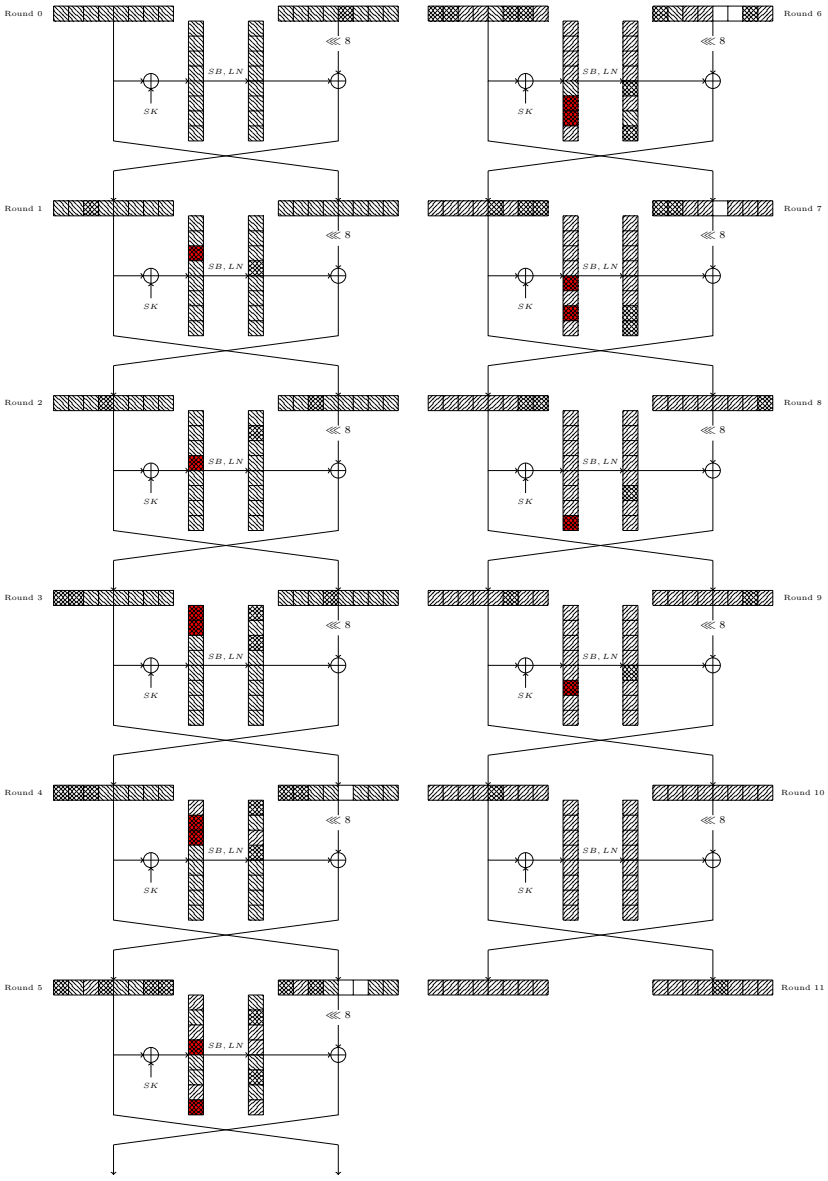


Fig. 20: A $\mathcal{DS}$-MITM distinguisher $(\mathcal{A}, \mathcal{B}, \mathsf{Deg}(\mathcal{A}, \mathcal{B}))$ for 11-round LBlock with $\mathcal{A} = [12]$ (the nibble marked with crosshatch in round 0), $\mathcal{B} = [12]$ (the nibbles marked with crosshatch in round 11), and $\mathsf{Deg}(\mathcal{A}, \mathcal{B}) = 14$ (the nibbles before the S-box operations mark with red).

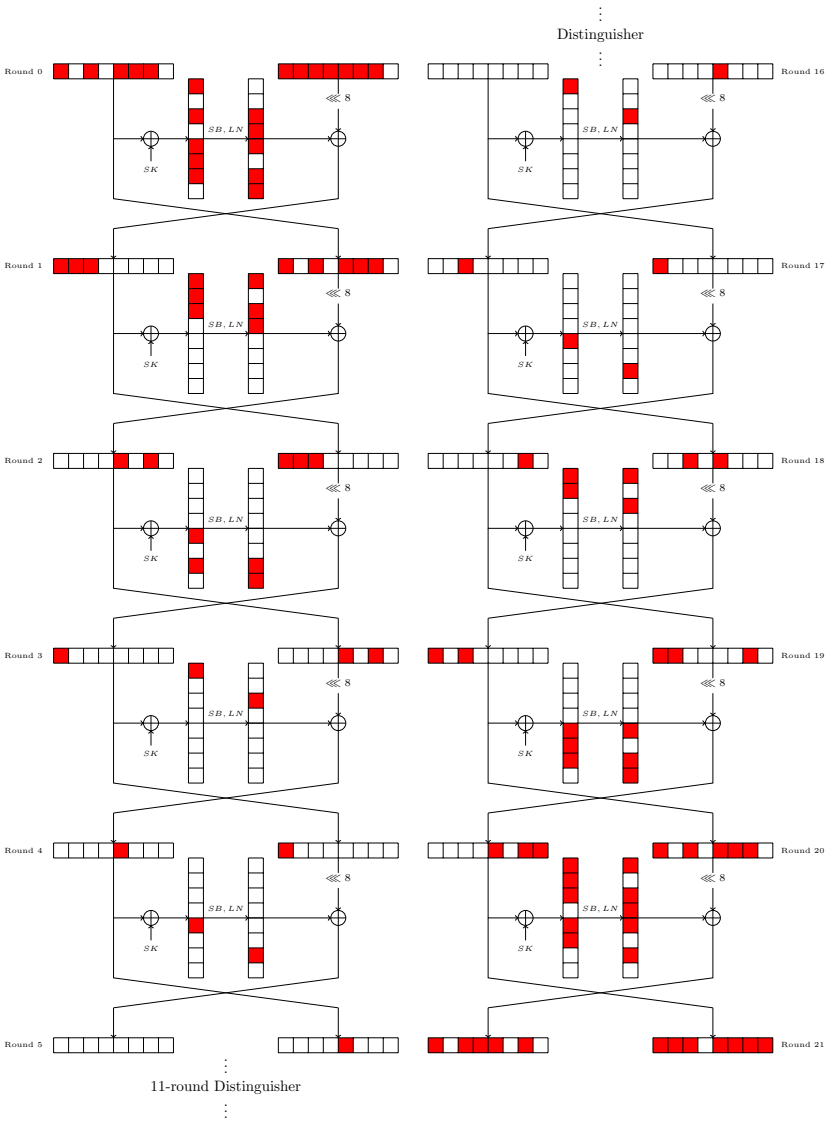


Fig. 21: An illustration of the backward differential and forward determination relationship in the outer rounds of LBlock with the $\mathcal{DS}$-MITM distinguisher presented in Fig. 20 placed at E_1. Nibbles in $\mathsf{Guess}(E_0)$ and $\mathsf{Guess}(E_2)$ are marked with red.

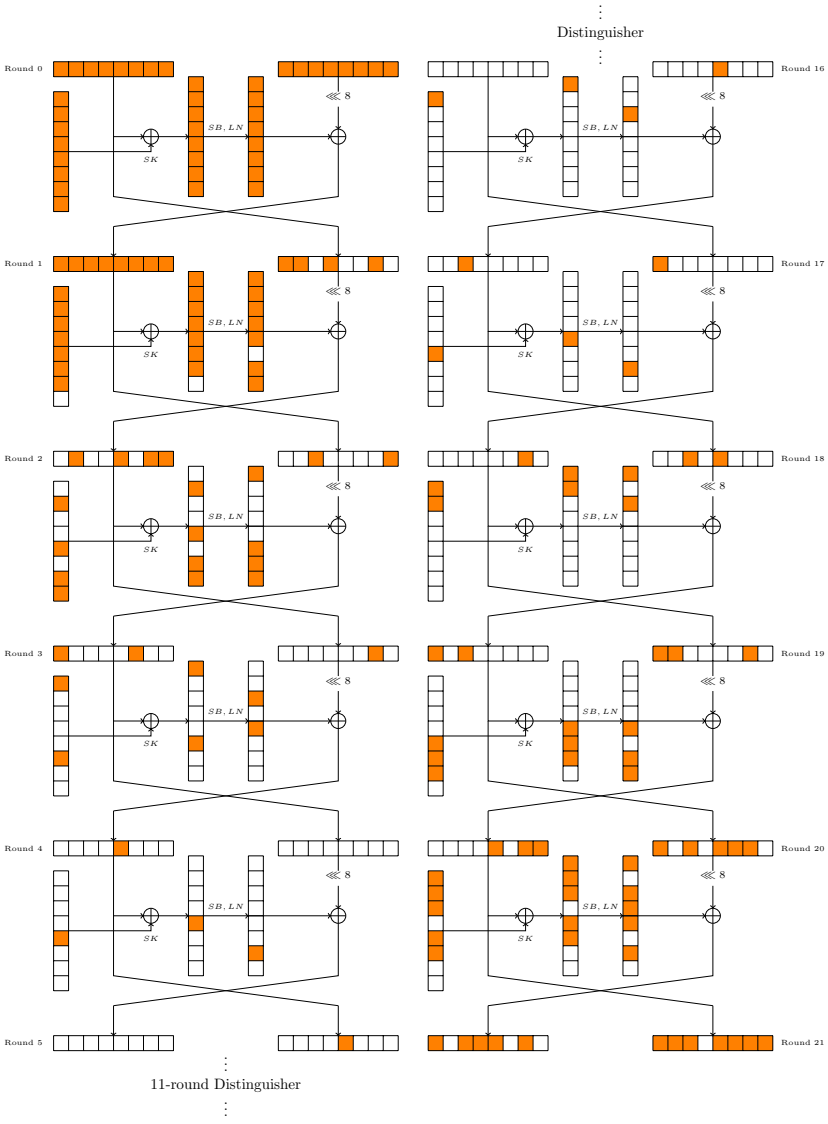


Fig. 22: Derive k_{E_0} and k_{E_2} from Fig. 21. The subkey nibbles marked with orange are the secret-key information we need to guess.

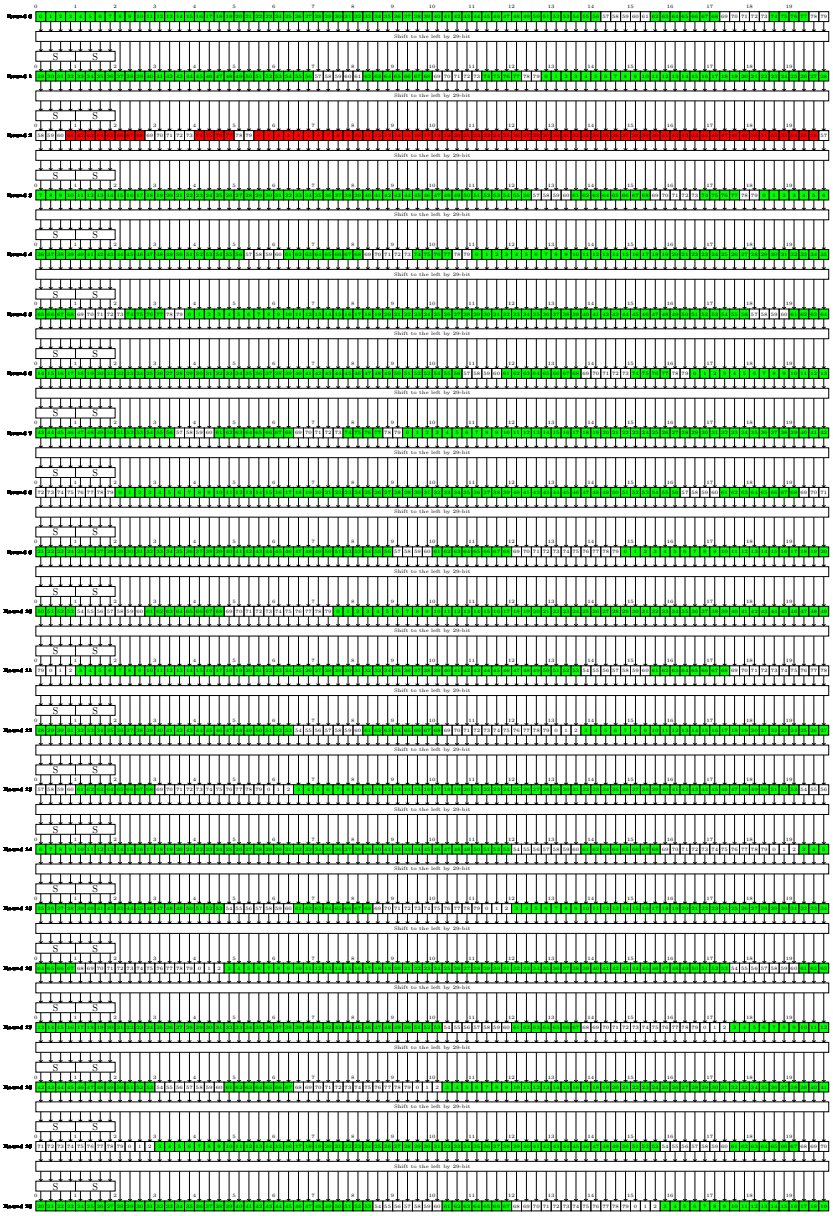


Fig. 23: An illustration that if we know the nibbles marked with red in round 2 of the key schedule algorithm of LBlock, we can derive the values of all the nibbles marked with green in the other rounds. The reader can verify that all the nibbles marked with orange in Fig. 22 are included in the colored nibbles of this figure.

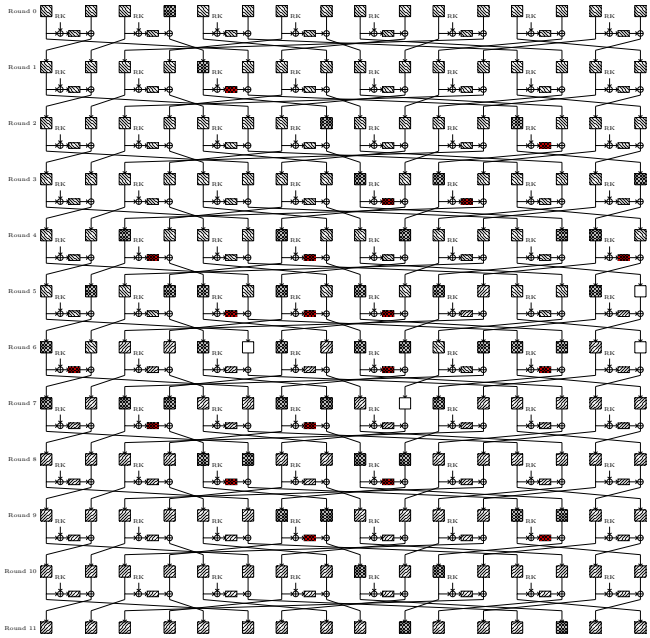


Fig. 24: A $\mathcal{DS}$-MITM distinguisher $(\mathcal{A}, \mathcal{B}, \mathsf{Deg}(\mathcal{A}, \mathcal{B}))$ for 11-round TWINE with $\mathcal{A} = [3]$ (the nibble marked with crosshatch in round 0), $\mathcal{B} = [9, 13]$ (the nibbles marked with crosshatch in round 11), and $\mathsf{Deg}(\mathcal{A}, \mathcal{B}) = 19$ (the nibbles before the S-box operations mark with red).

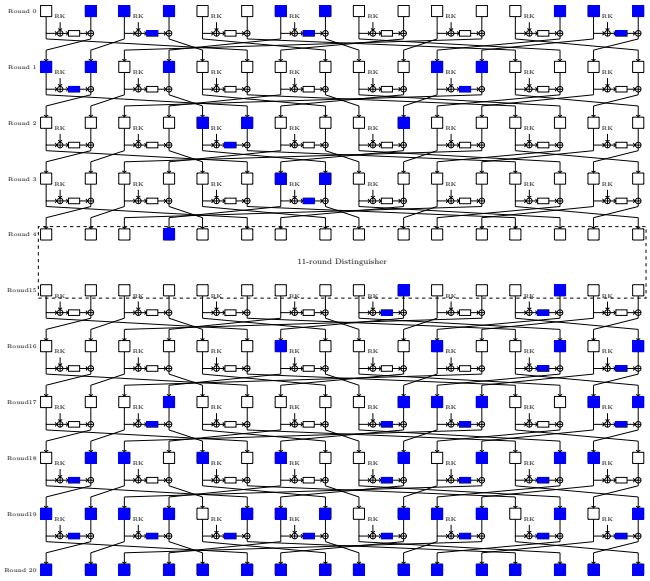


Fig. 25: An illustration of the backward differential and forward determination relationship in the outer rounds of TWINE with the $\mathcal{DS}$-MITM distinguisher presented in Fig. 24 placed at E_1. Nibbles in $\mathsf{Guess}(E_0)$ and $\mathsf{Guess}(E_2)$ are marked with blue.

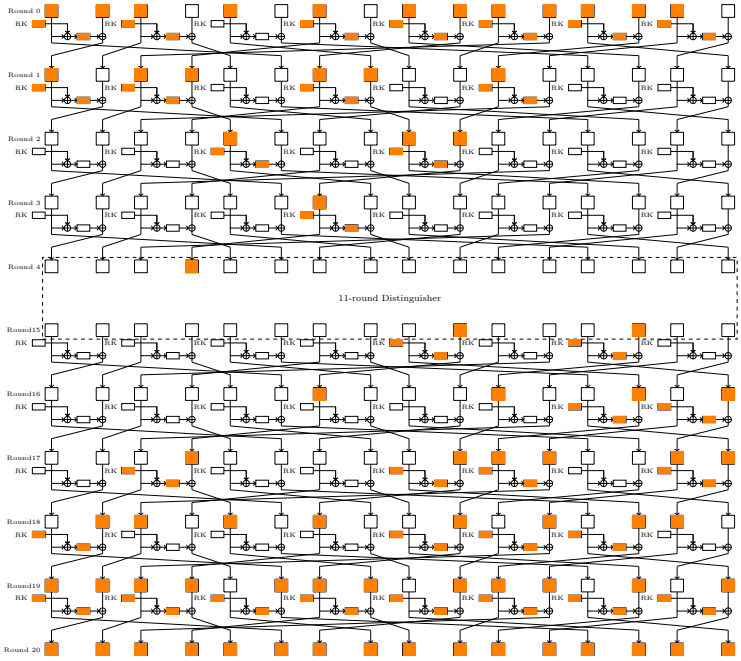


Fig. 26: Derive k_{E_0} and k_{E_2} from Fig. 25. The subkey nibbles marked with orange are the secret-key information we need to guess.

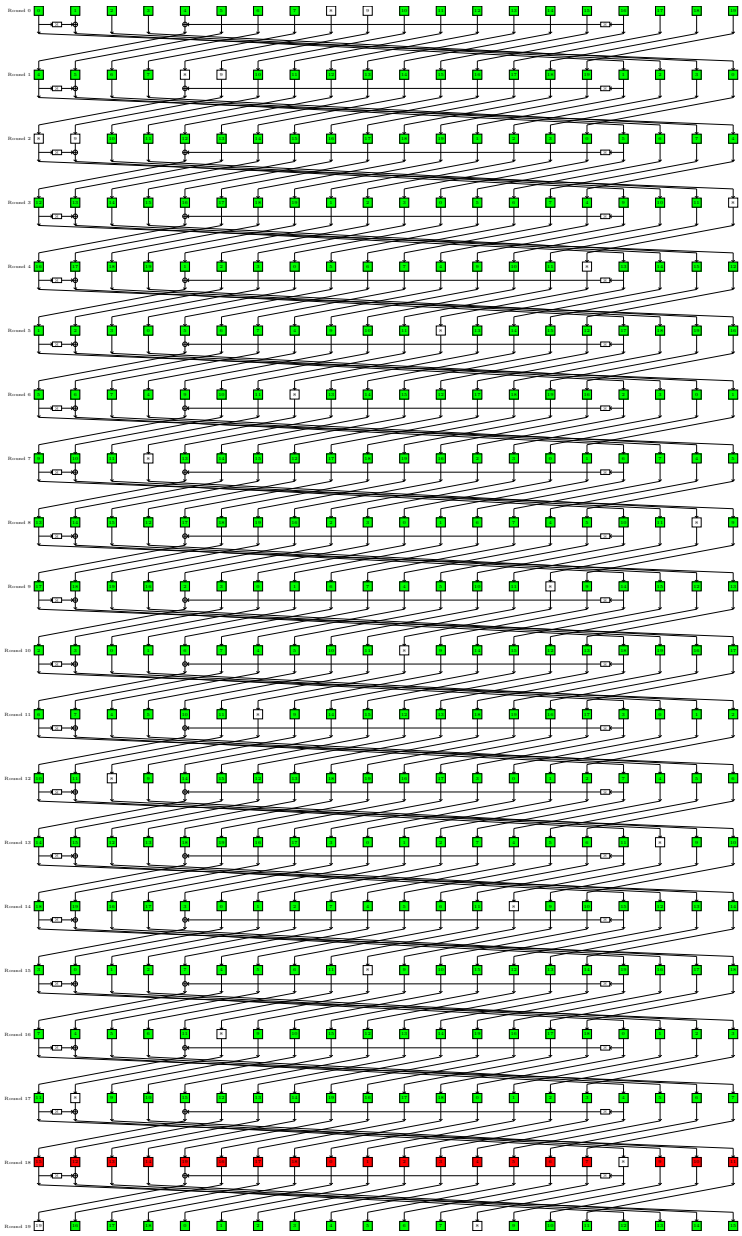


Fig. 27: A illustration that if we know the nibbles marked with red in round 18 of the key schedule algorithm of TWINE, we can derive the values of all the nibbles marked with green in the other rounds. The reader can verify that all the nibbles marked with orange in Fig. 26 are included in the colored nibbles of this figure.