Match Box Meet-in-the-Middle Attacks on the SIMON Family of Block Ciphers

Ling Song[1,2,3(✉)], Lei Hu[1,2], Bingke Ma[1,2,3], and Danping Shi[1,2,3]

[1] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
[2] Data Assurance and Communication Security Research Center, Chinese Academy of Sciences, Beijing 100093, China
[3] University of Chinese Academy of Sciences, Beijing 100049, China
{lsong,hu,bkma,dpsh}@is.ac.cn

Abstract. SIMON is a family of lightweight block ciphers designed by the U.S National Security Agency in 2013. In this paper, we analyze the resistance of the SIMON family of block ciphers against the recent match box meet-in-the-middle attack which was proposed in FSE 2014. Our attack particularly exploits the weaknesses of the linear key schedules of SIMON. Since the data available to the adversary is rather limited in many concrete applications, it is worthwhile to assess the security of SIMON against such low-data attacks.

Keywords: Lightweight block cipher · SIMON · Meet-in-the-middle attack · Match box

1 Introduction

Recent years have witnessed the rapid development of lightweight cryptography. In order to meet the uprising security demands in resource restrained environment such as RFID tags and wireless sensor networks, many lightweight block ciphers have been proposed as the fundamental cryptographic primitives, including PRESENT [6], KATAN & KTANTAN [9], PRINTcipher [14], LBlock [19], Piccolo [16], LED [11], SIMON & SPECK [3], to name but a few. In order to design a cipher satisfying resource constraints, the inner components should be simple and easy to implement, especially the key schedules. For example, KATAN, PRINCE and SIMON have linear key schedules and LED uses the master key directly without any key schedule.

These new design styles give rise to new cryptanalytic techniques. In particular, meet-in-the-middle attacks have developed a lot in the analysis of lightweight block ciphers [7,8]. New techniques, such as indirect matching, all-subkey recovery [17], bicliques [13], sieve-in-the-middle [8] and match box [10] have been proposed and make meet-in-the-middle attacks more powerful in cryptanalysis.

Meet-in-the-middle attacks normally can apply to ciphers with simple key schedules. As a unique instance, the recently proposed match box technique [10] aims particularly at block ciphers with linear key schedules.

© Springer International Publishing Switzerland 2015
T. Eisenbarth and E. Öztürk (Eds.): LightSec 2014, LNCS 8898, pp. 140–151, 2015.
DOI: 10.1007/978-3-319-16363-5_9

SIMON is a family of lightweight block ciphers designed by the U.S National Security Agency in 2013. It attracts many researchers since there is no internal security analysis of the cipher included in the specification document. In the literature, analyses of SIMON focus on differential cryptanalysis [4] and linear cryptanalysis [15]. Typical examples are [1,2,5,18] which belong to the sort of statistical cryptanalysis and thus require a great data complexity. However, in this paper we only concentrate on attacks with low data complexity.

Our contributions. In this paper, we analyze the resistance of the SIMON family of block ciphers against the match box meet-in-the-middle attack. Compared with classical statistical attacks such as differential and linear attacks [4,15], the match box meet-in-the-middle attack on SIMON requires much less and more reasonable amount of data. To the best of our knowledge, this is the first meet-in-the-middle type of attack on SIMON, which is both meaningful and attractive for many concrete protocols and applications, where only a small amount of plaintext/ciphertext pairs can be eavesdropped by the adversary. Our work enriches the analytical results on SIMON in the literature.

The remainder of this paper is organized as follows. The notations used in this paper are defined in Sect. 2; we recall the match box meet-in-the-middle attack in Sect. 3; Sect. 4 briefly describes the family of block ciphers SIMON, elaborates the match box meet-in-the-middle attack against the smallest version of SIMON and summarizes the results of other SIMON versions; we conclude the paper in the last section.

2 Notations

The following notations will be used throughout this paper:

P, C	: plaintext and ciphertext.
$E_{0-R}(K, P)$	: encryption of P from round 0 to round R with the secret key K.
$D_{R-R_1}(K, C)$	: decryption of C from round R to round R_1 with the secret key K.
X^i	: the i-th state word.
X^i_j	: the j-th bit of the word X^i.
$\|K\|$	: the bit size of K or the dimension of the linear space generated by K.

3 Match Box Meet-in-the-Middle Attack

In this section, we recall the details of the match box meet-in-the-middle attack [10] and some related techniques.

3.1 Basic Meet-in-the-Middle Attack

As depicted in Fig. 1, the basic meet-in-the-middle attack assumes that a fraction of the internal state v could be computed from a plaintext P with a portion K_1 of the master key K, and that v could also be computed from the corresponding ciphertext C with a portion K_2 of K.

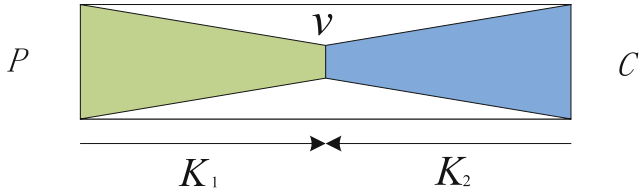

Fig. 1. Basic meet-in-the-middle attack

Assume $K_1 \cup K_2 = K$, $K_1 \cap K_2 = K_\cap$, $K_1' = K_1 - K_\cap$ and $K_2' = K_2 - K_\cap$. The basic meet-in-the-middle attack proceeds in two stages: a key filtering stage which sieves out the key candidates, followed by a verification stage that tests each candidate to derive the right key. The first stage with one pair of plaintex/ciphertext is as follows:

– For each $k_\cap \in K_\cap$:
  1. For each $k_1' \in K_1'$, v is computed, and store the corresponding pair (v, k_1') in a table indexed by v.
  2. For each $k_2' \in K_2'$, v is computed. From the previous table, retrieve the k_1's by matching v. The combination of k_1', k_2' and $k_\cap$ forms a candidate master key.

The right key is necessarily among the candidate keys, since it must lead to a match at any internal state. After the key filtering stage described above, the key space is reduced to $2^{|K|-|v|}$.

In the verification stage, each candidate key is tested. To find the right key, $N = \lceil \frac{|K|}{n} \rceil$ pairs of plaintext/ciphertext are needed, where n is the block size.

In total, the time complexity is:

$$\begin{aligned} T &= T_{\text{Filtering}} + T_{\text{Verifying}} \\ &= 2^{K_\cap} \cdot \left(2^{|K_1'|} \cdot \frac{R_1}{R} + 2^{|K_2'|} \cdot \frac{R_2}{R} \right) + \sum_{i=0}^{N-1} 2^{|K|-|v|-i*n} \end{aligned}$$

encryptions, where R_1 and R_2 are the number of rounds in forward computation and backward computation respectively, and $R = R_1 + R_2$ is the number of rounds attacked. The memory complexity is $\min\{2^{|K_1'|}, 2^{|K_2'|}\}$, and the data complexity is N known plaintext/ciphertext pairs.

A simple tweak can be utilized to reduce the complexity. Suppose t pairs of plaintext/ciphertext are used in the first stage. Then the complexity of the first stage rises t times, while in the second stage only $2^{|K|-t\cdot|v|}$ candidate keys need to be tested.

The indirect matching technique, which neglects the round key bits which have a linear impact on the matching point, can also be exploited to decrease the complexity. Suppose $v = E_{0-R_1}(K_1, P) = D_{R-R_1}(K_2, C)$ and $L(K_1), L(K_2)$

are the bits of K_1, K_2 that have a linear impact on v. Then this equation can be rewritten as

$$v = E_{0-R_1}(K_1 - L(K_1), P) \oplus L(K_1) = D_{R-R_1}(K_2 - L(K_2), C) \oplus L(K_2),$$

which is equivalent to

$$E_{0-R_1}(K_1 - L(K_1), P) \oplus D_{R-R_1}(K_2 - L(K_2), C) = L(K_1) \oplus L(K_2). \quad (1)$$

A correct guess of $K_1 - L(K_1), K_2 - L(K_2)$ makes Eq. (1) hold for different pairs of plaintext/ciphertext, say (P_1, C_1) and (P_2, C_2), that is,

$$\begin{aligned} &E_{0-R_1}(K_1 - L(K_1), P_1) \oplus D_{R-R_1}(K_2 - L(K_2), C_1) \\ =&E_{0-R_1}(K_1 - L(K_1), P_2) \oplus D_{R-R_1}(K_2 - L(K_2), C_2). \end{aligned}$$

In this way, if multiple pairs of plaintext/ciphertext are used, key bits in $L(K_1)$, $L(K_2)$ can be excluded and thus less bits need to be considered from both directions of computation in the first stage. Consequently, the complexity decreases accordingly.

3.2 Match Box Meet-in-the-Middle Attack

The match box technique was proposed in [10] which fits in the general sieve-in-the-middle framework [8]. As shown in Fig. 2, l is computed from a plaintext P with $k_1 \in K_1$ and r is computed from the corresponding ciphertext C with $k_2 \in K_2$. The match box is a precomputed table that stores all compatible (l, r)s under control of $k_3 \in K_3$, which has a small size. The simplest case is that $K_3 \cap K_1 = K_3 \cap K_2 = \varnothing$. In this case, once l and r are computed, the match box returns whether l and r are compatible. If so, the corresponding (k_1, k_2, k_3) is a candidate key. In more practical cases, K_3 may involve bits from both K_1 and K_2, which makes it difficult to construct a match box with a reasonable complexity.

In [10], the authors proposed a method to construct match boxes for block ciphers with linear key schedules. Following the previous notations, $K_1 \cap K_2 = K_\cap, K_1' = K_1 - K_\cap$ and $K_1 \cup K_2 = K$. K_3 is related to both K_1 and K_2. Let f be the key schedule function. Since $K = K_1' + K_2$, then $f(K) = f(K_1') \oplus f(K_2)$. On the right side of the match box, r and k_2 are known (which means $f(K_2)$ is also known) and considered as a whole $\vec{r} = (r, f(K_2))$. On the left side of the match box, (l, k_1') are known. l depends on k_1' and there are $2^{l \cdot |K_1'|}$ mappings of $g : k_1' \mapsto l$.

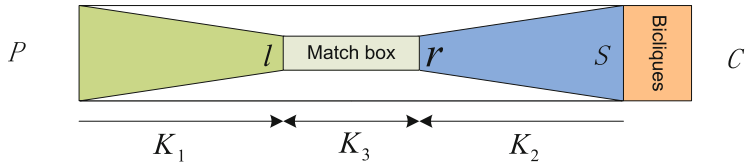

Fig. 2. Match box meet-in-the-middle attack

Given a mapping g, for each value of $\vec{r}$ we can precompute a list of k_1's (at most $2^{|K_1'|}$) leading to l such that (l, k_1') and $\vec{r}$ are compatible. There are $2^{|\vec{r}|}$ values of $\vec{r}$, and as a result $2^{|K_1'|+|\vec{r}|}$ computations should be stored for each mapping g. Enumerating all possible mappings, the match box is supposed to have

$$2^{|l|\cdot 2^{|K_1'|}+|K_1'|+|\vec{r}|} \tag{2}$$

entries in total[1].

During the attack, l is computed using k_1 in the forward computation and the mapping g is thus determined; in the backward computation, $\vec{r}$ is computed. To verify the compatibility between (l, k_1') and $\vec{r}$, we search for the list of k_1's using $\vec{r}$ among the items of the match box which is indexed by the mapping g, and check whether the k_1' used in forward computation is in that list.

The main limitation of this technique is the size of the match box table. More precisely, as specified in Eq. (2), the table becomes too large if $|K_1'|$ is not small enough.

On the key separations. Following the idea in [10], the master key can be regarded as a vector in $(\mathbb{Z}/2\mathbb{Z})^{|K|}$. The value of the master key corresponds to the coordinates of this vector along the canonical basis. Each round key is a linear combination of the master key bits, and then $|K_1|$ (resp. $|K_2|$) can be regarded as the dimension of the linear space generated by K_1 (resp. K_2). In this way, it becomes much easier to find independent separations of the round key bits, which results in more efficient meet-in-the-middle attacks for KATAN in [10]. This strategy is very likely to be applicable to other block ciphers with linear key schedules.

4 Match Box Meet-in-the-Middle Attack on SIMON

4.1 The SIMON Family of Block Ciphers

SIMON is a family of lightweight block ciphers designed by NSA which aims to provide an optimal hardware implementation performance [3]. SIMON supports a variety of word sizes $n = 16, 24, 32, 48$ and 64 bits. SIMON2n with m n-bit key words is denoted by SIMON2n/mn. Table 1 makes explicit the parameter choices for all versions of SIMON.

The design of SIMON follows a classical Feistel structure, operating on two n-bit halves in each round. The round function makes use of three n-bit operations: XOR ($\oplus$), AND ($\&$) and circular shift ($\lll$). Given a round key k it is defined on two inputs x and y as

$$R_k(x, y) = (y \oplus f(x) \oplus k, x),$$

where $f(x) = ((x \lll 1)\&(x \lll 8)) \oplus (x \lll 2)$. In this paper, (X^0, X^1) denotes the plaintext and (X^i, X^{i+1}) denotes the state after i rounds of encryption.

[1] We have confirmed from the authors of [10] that the complexity is not $2^{|l|^{|K_1'|}+|K_1'|+|\vec{r}|}$ as their paper describes, but $2^{|l|\cdot 2^{|K_1'|}+|K_1'|+|\vec{r}|}$.

Table 1. SIMON parameters.

Block size $2n$	Key size mn	Word size n	Key words m	Rounds T
32	64	16	4	32
48	72	24	3	36
48	96	24	4	36
64	96	32	3	42
64	128	32	4	44
96	96	48	2	52
96	144	48	3	54
128	128	64	2	68
128	192	64	3	69
128	256	64	4	72

The key schedules of SIMON are linear transformations, as depicted in Fig. 4. The m master key words are used as the first m round keys. Also, they are the inputs for the first iteration of the key schedules. Note that, the i-th round key is denoted as K^i. The key schedule differs slightly for different m. The constant c is $2^n - 4$ and z_js are 1-bit constant sequences. For more specifications of SIMON, please refer to [3] Fig. 3.

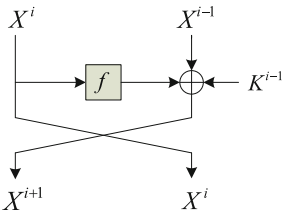

Fig. 3. The round function of SIMON

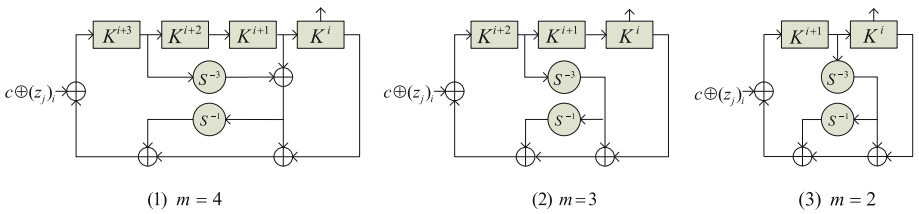

Fig. 4. The key schedules for $m = 4, 3, 2$. S^j operation denotes left circular shift by j bits.

4.2 Application of Match Box Meet-in-the-Middle Attack to SIMON32/64

SIMON32/64 is the version of SIMON with 16-bit words and 64-bit keys ($n = 16, m = 4$). As depicted in Fig. 5, we need to guarantee that $|K_1| < 64$ in the forward computation, such that the attack is faster than the brute-force attack. Following the algorithm in [10] that determines key dependencies by marking the key bits that enter the state bits and propagating the dependencies along the cipher, we also write a program to observe the dependency between internal state bits and key bits. According to our computation, after seven rounds of encryption each state bit are influenced by 63 round key bits, among which 61 bits have a nonlinear impact on the state bit. These 61 round key bits form a linear space of dimension 61.

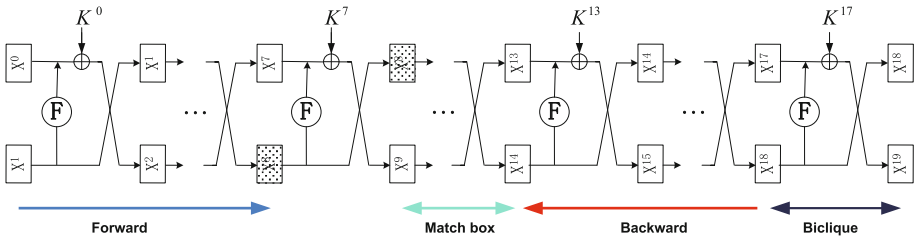

Fig. 5. Match box meet-in-the-middle attack on SIMON32/64

Similarly in the backward computation, 61 round key bits have a nonlinear impact on each state bit after seven rounds of decryption, which also form a linear space of dimension 61.

Setting any bit of state X^8 as the matching point, we are able to mount a basic meet-in-the-middle attack on 15-round SIMON32/64 using 3 plaintext/ciphertext pairs. The complexity is

$$3 \cdot \left(2^{61} \cdot \frac{7}{15} + 2^{61} \cdot \frac{7}{15}\right) + \sum_{i=0}^{2} 2^{61-i*32} = 2^{62.93}.$$

Extend the Basic Attack to More Rounds. Suppose the matching point is X_0^8, i.e. the least significant bit of X^8. According to the round function,

$$X_0^8 = X_{15}^9 \cdot X_8^9 \oplus X_{14}^9 \oplus X_0^{10} \oplus K_0^8. \tag{3}$$

By decomposing X_{15}^9 in Eq. (3), we get

$$X_0^8 = (X_{14}^{10} \cdot X_7^{10} \oplus X_{13}^{10} \oplus X_{15}^{11} \oplus K_{15}^9) \cdot X_8^9 \oplus X_{14}^9 \oplus X_0^{10} \oplus K_0^8.$$

Here we neglect the round key bits which have a linear impact on X_0^8, so K_0^8 is neglected. Besides K_0^8, several other round key bits may be neglected as

long as they impact on X_0^8 linearly. From the decryption side, to get the value of X_0^8, we just need to calculate the values of $X_{15}^{11}, X_0^{10}, X_7^{10}, X_{13}^{10}, X_{14}^{10}, X_8^9$ and X_{14}^9. Thus, the round key bit K_{15}^9 is isolated from the backward computation, and will be processed in the match box. As can be seen, more round key bits of K_2 will be isolated from the backward computation if the decomposition goes further, leading a reduction on the dimension of K_2. To keep $K_2 = 61$, more round key bits after the 15-th round can be added. In this way the number of rounds attacked may increase.

From the above analysis, we can derive the following principle: we can move the backward computation beyond the 15-th round by isolating more round key bits in the middle; however we need to keep the dimension of K_2 to be 61. In other words, the load of the backward computation keeps the same, but the overall number of rounds attacked may increase since more middle rounds can be covered with the match box.

In fact, the number of round key bits that can be isolated is determined by the construction of the match box. The round key bits in the middle involve information form both K_1 and K_2, and they can not be computed independently in any direction. Since SIMON adopts linear key schedules and $K = K_1' + K_2$, each round key bit can be split into two parts: $K_j^i = rk_j^i \oplus lk_j^i$, where rk_j^i denotes the part generated by K_2 and lk_j^i denotes the part generated by K_1'.

In this case, $|K_1'| = 3$ and $|l| = 3$ if three plaintext/ciphertext pairs are used; $\vec{r}$ consists of the state bits and rk_j^is, which is absorbed from the decomposed expression of X_0^8. According to Eq. (2), the match box requires a memory complexity of $M = 2^{27+|\vec{r}|}$.

Compression table. Normally, $|\vec{r}|$ should be smaller than 37 to guarantee $M < 2^{64}$. In order to cover as many middle rounds with the match box as possible, we utilize the so-called *compression table* technique of [8] to shorten $\vec{r}$ when $|K_1'|$ is small. This technique comes from the fact that the decomposed expression of X_0^8 can be expressed as a boolean polynomial in the bits of K_1'. It is obvious that the boolean polynomial in the bits of K_1' has no more than $2^{|K_1'|}$ coefficients, to which all bits in $\vec{r}$ can be mapped. Therefore, if $2^{|K_1'|} < 2^{|\vec{r}|}$, $|\vec{r}|$ bits from the right side of the match box can be equivalently expressed as at most $2^{|K_1'|}$ coefficients. Consequently, the memory complexity of the match box gets reduced.

However, the cost of building a compression table that transforms bits of $\vec{r}$ into coefficients of bits in K_1' cannot be neglected. Since the coefficients can be computed with $2^{|K_1'|}$ partial encryptions for each $\vec{r}$, the whole compression table is built with time complexity $2^{|\vec{r}|+|K_1'|}$ and memory complexity $2^{|\vec{r}|}$.

The decomposition shows X_8^0 can be represented with 27 state bits and 32 rk_j^is, and the attack can now be extended to 17 rounds. We have $|K_1'| = 3, |\vec{r}| = 27 + 32 = 55 > 2^3$ and a compression table of size 2^{55} will be built with time complexity of 2^{58}.

If we match three plaintext/ciphertext pairs simultaneously, the three corresponding 55-bit $\vec{r}$ are shorten to $8 \times 3 = 24$ bits using the compression table. This yields a match box of size $2^{27+24} = 2^{51}$.

Total complexity. We attack 17 rounds of SIMON32/64, and the overall time complexity is

$$\begin{aligned} T_{Total} &= T_{Compression} + T_{MatchBox} + T_{Filtering} + T_{Verifying} \\ &= 2^{58} + 2^{51} + 3 \cdot (2^{61} \cdot \frac{7}{17} + 2^{61} \cdot \frac{4}{17}) + \sum_{i=0}^{3-1} 2^{61-32*i} \approx 2^{62.57}. \end{aligned}$$

The round key bits in K_1, K_2 and $\vec{r}$ are provided in the Appendix. For all versions of SIMON, since the last round key is added to the state linearly, which means no overlapped nonlinear component, we can extend one round with bicliques [13] at the end of the cipher with no additional time complexity but 2^N chosen plaintext/ciphertext pairs, where N is 3 or 4. Finally, 18 rounds of SIMON32/64 can be attacked.

4.3 Application to Other Versions

For other versions of SIMON, similar attacks can be mounted. Table 2 summarizes the results of eight versions. We omit the results of the other two versions for $m = 2$, i.e. SIMON96/96 and SIMON128/128, because the match box meet-in-the-middle attack does not outperform the basic meet-in-the-middle attack which can attack 17 and 19 rounds of SIMON96/96 and SIMON128/128 respectively. The reason why the match box meet-in-the-middle attack does not work well for these two versions is due to the fact that any match box for them covering one round in the middle becomes too large. Note that, for the cases $m = 2$, the key size is as large as the block size, which makes it difficult to construct a match box with a reasonable complexity. For larger m, which means the key size is larger than block size, the match box meet-in-the-middle attack has been proved to be more efficient.

Table 2. Results for eight versions of SIMON

Version:	Rounds		Data	Time	$\|K_1\|$	$\|K_2\|$	Match	Compression
$2n/mn$	Total	Attacked					box	table
32/64	32	18	2^3	$2^{62.57}$	61	61	2^{51}	2^{55}
48/72	36	17	2^3	$2^{71.65}$	70	71	2^{18}	2^{47}
48/96	36	19	2^3	$2^{95.26}$	94	94	2^{18}	2^{72}
64/96	42	17	2^3	$2^{94.05}$	93	93	2^{35}	2^{65}
64/128	44	19	2^3	$2^{126.01}$	125	125	2^{35}	2^{76}
96/144	54	21	2^4	$2^{141.27}$	140	140	2^{132}	2^{96}
128/192	69	25	2^3	$2^{190.60}$	189	190	2^{51}	2^{97}
128/256	72	25	2^3	$2^{253.94}$	253	253	2^{35}	2^{116}

5 Conclusion

In this paper, we analyzed eight versions of the SIMON family of block ciphers with the match box meet-in-the-middle attack. These eight versions share a common feature that the key size is larger than the block size. Our work exploits the weaknesses of the linear key schedules of SIMON. Compared to the existing attacks based on statistical methods, our attack requires much less data, which is meaningful for many concrete situations where the data available to the adversary is rather limited.

Acknowledgement. The authors would like to thank anonymous reviewers for their helpful comments and suggestions. The work of this paper was supported by the National Key Basic Research Program of China (2013CB834203), the National Natural Science Foundation of China (Grants 61070172), the Strategic Priority Research Program of Chinese Academy of Sciences under Grant XDA06010702, and the State Key Laboratory of Information Security, Chinese Academy of Sciences.

A Details for the Attack on SIMON32/64

- K_1 involves 61 round key bits (dimension 61) as follows:
  $K_0^0, K_1^0, K_2^0, K_3^0, K_4^0, K_5^0, K_6^0, K_7^0, K_8^0, K_9^0, K_{10}^0, K_{11}^0, K_{12}^0, K_{13}^0, K_{14}^0, K_{15}^0$,
  $K_0^1, K_1^1, K_2^1, K_3^1, K_4^1, K_5^1, K_6^1, K_7^1, K_8^1, K_9^1, K_{10}^1, K_{11}^1, K_{12}^1, K_{13}^1, K_{14}^1, K_{15}^1$,
  $K_0^2, K_2^2, K_3^2, K_4^2, K_5^2, K_6^2, K_7^2, K_9^2, K_{10}^2, K_{11}^2, K_{12}^2, K_{13}^2, K_{14}^2$,
  $K_4^3, K_5^3, K_6^3, K_8^3, K_{11}^3, K_{12}^3, K_{13}^3, K_{14}^3, K_{15}^3$,
  $K_0^4, K_6^4, K_7^4, K_{13}^4, K_{14}^4$,
  K_8^5, K_{15}^5.
- The match box involves 29 round keys generated by $K2$:
  $rk_8^9, rk_{15}^9, rk_0^{10}, rk_6^{10}, rk_7^{10}, rk_{13}^{10}, rk_{14}^{10}, rk_4^{11}, rk_5^{11}, rk_6^{11}, rk_8^{11}, rk_{11}^{11}, rk_{12}^{11}$,
  $rk_{13}^{11}, rk_{14}^{11}, rk_{15}^{11}, rk_0^{12}, rk_5^{12}, rk_6^{12}, rk_7^{12}, rk_{11}^{12}, rk_{12}^{12}, rk_{13}^{12}, rk_{14}^{12}, rk_8^{13}, rk_{13}^{13}$,
  $rk_{15}^{13}, rk_{14}^{13}, rk_0^{14}$.
- K_2 involves 67 round key bits (dimension 61) as follows:
  $K_2^{12}, K_3^{12}, K_4^{12}, K_6^{12}, K_7^{12}, K_{10}^{12}$,
  $K_0^{13}, K_1^{13}, K_2^{13}, K_3^{13}, K_4^{13}, K_5^{13}, K_6^{13}, K_7^{13}, K_8^{13}, K_9^{13}, K_{10}^{13}, K_{11}^{13}, K_{12}^{13}$,
  $K_0^{14}, K_1^{14}, K_2^{14}, K_3^{14}, K_4^{14}, K_5^{14}, K_6^{14}, K_7^{14}, K_8^{14}, K_9^{14}, K_{10}^{14}, K_{11}^{14}, K_{12}^{14}, K_{13}^{14}$,
  K_{14}^{14}, K_{15}^{14},
  $K_0^{15}, K_1^{15}, K_2^{15}, K_3^{15}, K_4^{15}, K_5^{15}, K_6^{15}, K_7^{15}, K_8^{15}, K_9^{15}, K_{10}^{15}, K_{11}^{15}, K_{12}^{15}, K_{13}^{15}$,
  K_{14}^{15}, K_{15}^{15},
  $K_0^{16}, K_1^{16}, K_2^{16}, K_3^{16}, K_4^{16}, K_5^{16}, K_6^{16}, K_7^{16}, K_8^{16}, K_9^{16}, K_{10}^{16}, K_{11}^{16}, K_{12}^{16}, K_{13}^{16}$,
  K_{14}^{16}, K_{15}^{16}.

References

1. Abed, F., List, E., Wenzel, J., Lucks, S.: Differential cryptanalysis of round-reduced simon and speck. In: Cid, C., Rechberger, C. (eds.) FSE 2014. LNCS. Springer (2014, to appear)
2. Alizadeh, J., Alkhzaimi, H.A., Aref, M.R., Bagheri, N., Gauravaram, P., Kumar, A., Lauridsen, M.M., Sanadhya, S.K.: Cryptanalysis of SIMON variants with connections. In: Sadeghi, A.-R., Saxena, N. (eds.) RFIDSec 2014. LNCS, vol. 8651, pp. 90–107. Springer, Heidelberg (2014)
3. Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., Wingers, L.: The SIMON and SPECK Families of Lightweight Block Ciphers. Cryptology ePrint Archive, Report 2013/404 (2013). http://eprint.iacr.org/
4. Biham, E., Shamir, A.: Differential cryptanalysis of DES-like cryptosystems. In: Menezes, A., Vanstone, S.A. (eds.) CRYPTO 1990. LNCS, vol. 537, pp. 2–21. Springer, Heidelberg (1991)
5. Biryukov, A., Roy, A., Velichkov, V.: Differential analyis of block cipher SIMON and SPECK. In: Cid, C., Rechberger, C. (eds.) FSE 2014. LNCS. Springer (2014, to appear)
6. Bogdanov, A.A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M., Seurin, Y., Vikkelsoe, C.: PRESENT: an ultra-lightweight block cipher. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 450–466. Springer, Heidelberg (2007)
7. Bogdanov, A., Rechberger, C.: A 3-subset meet-in-the-middle attack: cryptanalysis of the lightweight block cipher KTANTAN. In: Biryukov, A., Gong, G., Stinson, D.R. (eds.) SAC 2010. LNCS, vol. 6544, pp. 229–240. Springer, Heidelberg (2011)
8. Canteaut, A., Naya-Plasencia, M., Vayssière, B.: Sieve-in-the-middle: improved MITM attacks. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part I. LNCS, vol. 8042, pp. 222–240. Springer, Heidelberg (2013)
9. De Cannière, C., Dunkelman, O., Knežević, M.: KATAN and KTANTAN — a family of small and efficient hardware-oriented block ciphers. In: Clavier, C., Gaj, K. (eds.) CHES 2009. LNCS, vol. 5747, pp. 272–288. Springer, Heidelberg (2009)
10. Fuhr, T., Minaud, B.: Match box meet-in-the-middle attack against KATAN. In: FSE 2014. Springer (2014, to appear)
11. Guo, J., Peyrin, T., Poschmann, A., Robshaw, M.: The LED block cipher. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 326–341. Springer, Heidelberg (2011)
12. Borghoff, J., Canteaut, A., Güneysu, T., Kavun, E.B., Knezevic, M., Knudsen, L.R., Leander, G., Nikov, V., Paar, C., Rechberger, C., Rombouts, P., Thomsen, S.S., Yalçın, T.: PRINCE – a low-latency block cipher for pervasive computing applications. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 208–225. Springer, Heidelberg (2012)
13. Khovratovich, D., Rechberger, C., Savelieva, A.: Bicliques for preimages: attacks on Skein-512 and the SHA-2 family. In: Canteaut, A. (ed.) FSE 2012. LNCS, vol. 7549, pp. 244–263. Springer, Heidelberg (2012)
14. Knudsen, L., Leander, G., Poschmann, A., Robshaw, M.J.B.: PRINTcipher: a block cipher for IC-printing. In: Mangard, S., Standaert, F.-X. (eds.) CHES 2010. LNCS, vol. 6225, pp. 16–32. Springer, Heidelberg (2010)
15. Matsui, M.: Linear cryptanalysis method for DES cipher. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 386–397. Springer, Heidelberg (1994)

16. Shibutani, K., Isobe, T., Hiwatari, H., Mitsuda, A., Akishita, T., Shirai, T.: *Piccolo*: an ultra-lightweight blockcipher. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 342–357. Springer, Heidelberg (2011)
17. Isobe, T., Shibutani, K.: All subkeys recovery attack on block ciphers: extending meet-in-the-middle approach. In: Knudsen, L.R., Wu, H. (eds.) SAC 2012. LNCS, vol. 7707, pp. 202–221. Springer, Heidelberg (2013)
18. Wang, N., Wang, X., Jia, K., Zhao, J.: Improved Differential Attacks on Reduced SIMON Versions. http://eprint.iacr.org/2014/448
19. Wu, W., Zhang, L.: LBlock: a lightweight block cipher. In: Lopez, J., Tsudik, G. (eds.) ACNS 2011. LNCS, vol. 6715, pp. 327–344. Springer, Heidelberg (2011)