Impossible Differential Attack on Reduced Round SPARX-64/128

Ahmed Abdelkhalek, Mohamed Tolba, and Amr M. Youssef

Concordia Institute for Information Systems Engineering
Concordia University, Montréal, Québec, Canada

Abstract. SPARX-64/128 is an ARX-based block cipher with 64-bit block size and 128-bit key. It was published in Asiacrypt 2016 as one of the instantiations of a family of ARX-based block ciphers with provable security against single-characteristic differential and linear cryptanalysis. In this work, we present 12 and 13-round impossible distinguishers on SPARX-64/128 that can be used to attack 15 and 16-round SPARX-64/128 with post-whitening keys, respectively. While the 15-round attack starts from round 0, the 16-round one, exploiting the key schedule, has to start from round 2.
Keywords: Block Ciphers, Impossible Differential, Miss-in-the-middle, SPARX.

1 Introduction

SPARX is a family of ARX-based block ciphers that was published in Asiacrypt 2016 [6]. It was designed with the goal of putting forward a general strategy for designing ARX-based symmetric-key primitives with provable security against single-characteristic differential and linear cryptanalysis. As a dual to the wide trail strategy [4] adopted by many S-box based block ciphers, the designers proposed the long trail strategy. This strategy promotes the use of a rather weak but large S-box, i.e., an ARX-based S-box, along with a very light linear layer. Fostering the existence of long trails, that involve an uninterrupted sequence of calls to the S-box interleaved with key additions, rather than having maximum diffusion in each linear layer is at the core of this proposed strategy. The long trail strategy allowed the designers to bound the maximum differential and linear probabilities for any number of rounds of a block cipher designed following such strategy. SPARX-64/128 is a member of this family of block ciphers following the long trail strategy with 64-bit block size and 128-bit key. The only cryptanalysis of SPARX was done by its designers as they presented a 13-round bit-based division property distinguisher that they used to launch an integral attack against 15-round SPARX-64/128 [5]. No other attacks were given in the short/full versions of the design paper.

Impossible differential cryptanalysis that was independently proposed by Biham *et al.* [3] and Knudsen [9] is one of the most powerful cryptanalytic techniques. Firstly, we try to find a certain input difference that propagates to a

specific output difference with zero probability resulting in an impossible differential distinguisher. In general, the input and output differences can be truncated. Then, after finding the longest possible impossible differential, it is used in a key recovery attack by prepending and/or appending a few additional rounds which are usually called the analysis rounds. The attack proceeds as follows: first, we collect pairs with certain plaintext and ciphertext differences. Then, we guess some bits of the key material involved in the analysis rounds and if one of the pairs satisfies the input and output differences of the impossible differential under some subkey bits, then these subkey bits must be wrong. Thus, we discard as many wrong keys as possible and do an exhaustive search on the surviving ones along with the rest of the key. The early abort technique [10] allows us to guess the involved key material on steps to discard the undesired pairs as early as possible and therefore reduce the time complexity of the attack.

In this paper, we present a 12-round truncated impossible differential on SPARX-64/128 that can be extended to a 13-round impossible differential with a specific input difference and a truncated output difference. We use the 12-round impossible differential to launch an impossible differential attack against 15-round SPARX-64/128 including the post-whitening key with data complexity of 2^{51} chosen plaintexts, time complexity of $2^{94.1}$ 15-round encryptions and memory complexity of $2^{43.5}$ 64-bit blocks. Then, we use the 13-round impossible differential to attack 16-round SPARX-64/128, including the post-whitening key, starting from round 2 with data, time and memory complexities of $2^{61.5}$ known plaintexts, 2^{94} 16-round encryptions, and $2^{61.5}$ 64-bit blocks, respectively.

The remainder of the paper is organized as follows. In Section 2, the notations used throughout the paper are given followed by the specification of SPARX-64/128. Our impossible differentials are presented in Section 3. Afterwards, in Section 4, we provide a detailed description of our impossible differential attacks on SPARX-64/128. Finally, Section 5 concludes the paper.

2 Description of SPARX-64/128

Notations. The following notations are used throughout the paper:

- K: The master key.
- k_i: The i^{th} 16-bit of the key state, where $0 \leq i \leq 7$.
- k_i^j: The i^{th} 16-bit of the key state after applying the key schedule permutation j times, where $0 \leq i \leq 7$ and $0 \leq j \leq 17$ for SPARX-64/128.
- $RK_{(a,i)}$: The 32-bit round key used at branch a of round i where $0 \leq i \leq 24$ and $a = 0$ (1) denotes the left (right) branch of SPARX-64/128.
- $X_{(a,i)}$ ($Y_{(a,i)}$): The left (right) 16-bit input at branch a of round i where $0 \leq i \leq 24$, $a = 0$ (1) denotes the left (right) branch of SPARX-64/128, and the LSB of either $X_{(a,i)}$ or $Y_{(a,i)}$ is on the right.
- w: The number of 32-bit words, i.e., $w = 2$ for a 64-bit block and $w = 4$ for a 128-bit master key.

– R^3: The iteration of 3 rounds of SPECKEY with their corresponding key additions.
– L_w: Linear mixing layer used in SPARX with w-word block size, thus L_2 represents the linear mixing layer used in SPARX-64/128.
– $\boxplus$: Addition mod 2^{16}.
– $\oplus$: Bitwise XOR.
– $\lll q$ ($\ggg q$): Rotation of a word by q bits to the left (right).
– $\|$: Concatenation of bits.
– $0xabcd$: A 16-bit number in hexadecimal representation.

2.1 Specifications of SPARX-64/128

SPARX [6,5] is a family of ARX-based Substitution-Permutation Network (SPN) block ciphers. It follows the SPN design construction while using ARX-based S-boxes instead of S-boxes based on look-up tables. ARX-based S-boxes form a specific category of S-boxes that rely solely on addition, rotation and XOR operations to provide both non-linearity and diffusion. The SPARX family adopts the 32-bit SPECKEY ARX-based S-box, shown in Fig. 1, which resembles one round of SPECK-32 [1,2] with only one difference, that is, the key is added to the whole 32-bit state instead of just half the state as in SPECK-32.

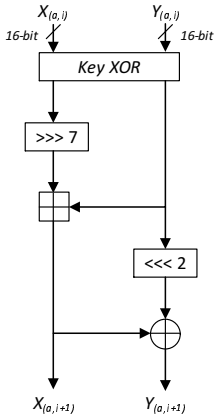


Fig. 1: The SPECKEY ARX-based S-box used in the SPARX family.

For a given member of the SPARX family whose block size is n bits, the plaintext is divided into $w = n/32$ words of 32 bits each. Then, the SPECKEY S-box (S), being applied to w words in parallel, is iterated r times interleaved by the addition of independent subkeys. Then, a linear mixing layer (L_w) is applied to ensure diffusion between the words. The structure made of a key addition followed by S is called a round while the structure made of r rounds followed by

L_w is called a step, as depicted in Fig. 2. Thus, the ciphertext corresponding to a given plaintext is generated by iterating such steps. The number of steps and the number of rounds in each step depend on both the block size of the cipher and the size of the key it utilizes.

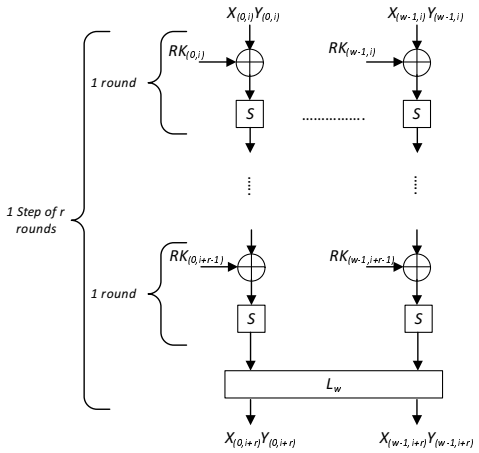
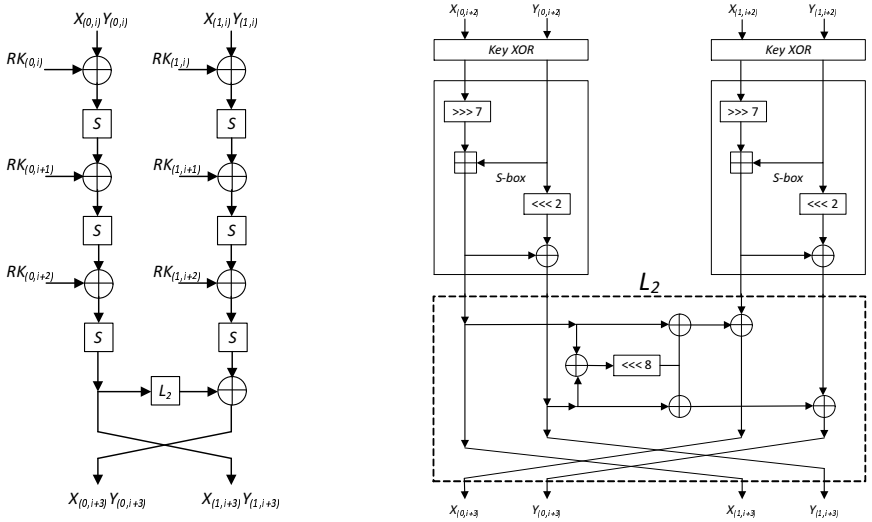


Fig. 2: SPARX structure

SPARX-64/128 is the lightest member of this family operating on 64-bit blocks using 128-bit keys. It uses 3 rounds in each step and iterates over 8 steps, i.e., the total number of rounds is 24. More precisely, in SPARX-64/128, 2 SPECKEY S-boxes (S) are iterated simultaneously 3-times, while being interleaved by the addition of the round keys and then a linear mixing layer (L_2) is applied, as shown in Fig. 3a. The structure of L_2 is depicted in the dotted square in Fig. 3b.

Key schedule. The 128-bit master key instantiates the key state, denoted by $k_0^0 \| k_1^0 \| k_2^0 \| k_3^0 \| k_4^0 \| k_5^0 \| k_6^0 \| k_7^0$. Then, the 3×32-bit round keys used in the left branch of the first step are extracted. Afterwards, the permutation illustrated in Fig. 4 is applied and then the 3×32-bit round keys used in the right branch of the first step are extracted. The application of the permutation and the extraction of the keys are interleaved untill all the round keys encompassing the post-whitening ones are generated. This means that, first, the round keys of a branch of a given step j are generated and then the key state is updated. The following observation on the key schedule is exploited in our attacks.

Observation: The last round key of a given step and the first round key of the subsequent step can be deduced from one another. To clarify this point, we consider the last round key of step 0 and the first round key of step 1. The 64-bit round key of the third round is $k_4^0 \| k_5^0, k_4^1 \| k_5^1$ and the 64-bit round key of



(a) step (b) round + linear layer

Fig. 3: SPARX-64/128 structure

the fourth round is $k_0^2 \| k_1^2, k_0^3 \| k_1^3$. According to the key schedule: $k_0^2 = k_6^1 = k_4^0$, $k_1^2 = k_7^1 \boxplus 2 = k_5^0 \boxplus 2$, $k_0^3 = k_6^2 = k_4^1$ and $k_0^3 = k_7^1 \boxplus 3 = k_5^1 \boxplus 3$.

Finally, it is to be noted that we measure the memory complexity of our attacks in number of 64-bit blocks and the time complexity in terms of the equivalent number of round-reduced encryptions.

3 Impossible Differentials of SPARX-64/128

A 12-round impossible differential is readily noticeable when considering SPARX-64/128 to be a twisted variant of a Feistel construction where the two halves undergo a keyed function before getting mixed and swapped. Indeed, as depicted in Fig. 5, if the left branch of SPARX-64/128 at round i has a zero difference while the right half has a nonzero difference, then after 2 steps (6 rounds), the input at the left branch must have a nonzero difference. From the other direction, if the input of the right branch of round $i+12$ has a nonzero difference, i.e, Γ and the input of the left branch at that round has a difference $L_2(\Gamma)$, then after the linear transformation, the right branch will have a zero difference which propagates unaltered for 2 complete steps (6 rounds) and contradicts with the forward differential at the left branch.

This 12-round truncated impossible differential can be extended to a 13-round distinguisher with a specific input difference and truncated output difference. This is feasible by exploiting the fact that there exist differentials with

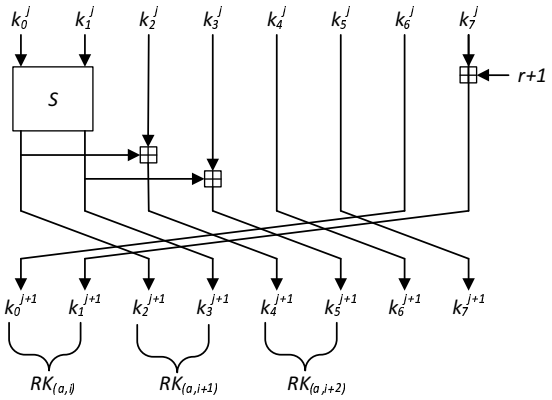


Fig. 4: SPARX-64/128 key schedule permutation, where the counter r is initialized to 0.

probability 1 for one SPECKEY round and one of these differentials is a fixed point of L_2. Particularly, if the input difference of the distinguisher is chosen to be $0x8000\ 0x8000$ then by propagating it backward through L_2 we have the same difference at both the right and left branches as an output for the S-box and this output difference corresponds to the input difference $0x0040\ 0x0000$ with probability 1. Hence, the input of the 13-round distinguisher is $0x0040\ 0x0000$ and $0x0040\ 0x0000$ while the output is still truncated in the form of $L_2(\Gamma)$ and Γ.

4 Impossible Differential Cryptanalysis of SPARX-64/128

The 12 and 13-round impossible distinguishers described above can be used to attack 15 and 16-round SPARX-64/128, respectively. Both attacks include the post-whitening key, however, the 16-round attack starts at round 2.

4.1 15-round Impossible Differential Attack on SPARX-64/128

In this attack, we have chosen to place the 12-round distinguisher at the top, end it with a specific difference that meets the constraint of $L_2(\Gamma)$ and Γ, and then append 3 rounds that have a high probability as shown in Fig. 6. That specific difference at the end of the distinguisher and the 3 analysis rounds were found using Mixed Integer Linear Programming (MILP). Specifically, we have followed the guidelines in [7] to create an MILP model that describes SPARX-64/128 and solved it using the publicly available MILP optimizer Gurobi [8]. The detailed procedure of the attack is described as follows.

Data Collection. We first choose 2^m structures of plaintexts where in each structure the left 32 bits of the plaintexts take a fixed value and the right 32 bits

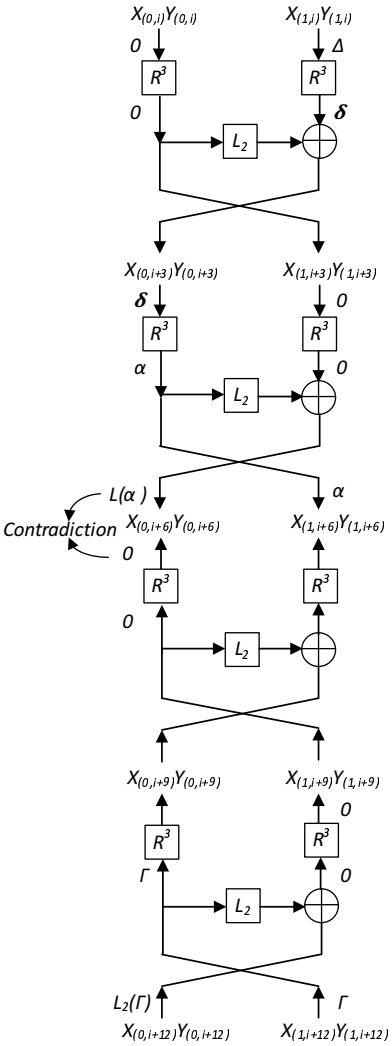


Fig. 5: 12-round impossible differential SPARX-64/128

take all the 2^{32} possible values. Each structure includes about $\binom{2^{32}}{2} \approx 2^{63}$ pairs of plaintexts, therefore we have $2^m \times 2^{63} = 2^{m+63}$ pairs of plaintexts in total. We encrypt these pairs and keep the ones whose ciphertext difference matches the difference shown in Fig. 6. The probability of such ciphertext difference is about 2^{-64}, therefore the expected number of remaining pairs after this phase is about $2^{m+63-64} = 2^{m-1}$.

Key Recovery. To verify if the pairs generated during the data collection phase follow our 12-round impossible differential, we need to guess $RK_{(0,15)}, RK_{(1,15)}$, $RK_{(0,14)}, RK_{(1,14)}$, and $RK_{(0,13)}$. However, as pointed out above, $RK_{(0,15)}$, $RK_{(1,15)}$ are related to $RK_{(0,14)}, RK_{(1,14)}$. This means that these round keys take 2^{96} values only. The details of this phase are as follows.

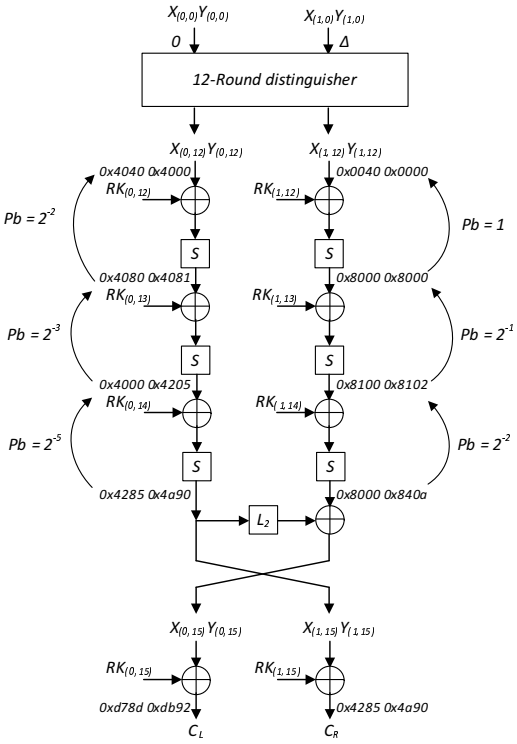


Fig. 6: 15-round impossible differential attack on SPARX-64/128

Step 1. For all the ciphertext pairs obtained in the data collection phase, we guess the 64-bit round keys $RK_{(0,15)}$ and $RK_{(1,15)}$, decrypt round 15 and check if the difference matches the one shown in Fig. 6. If it is not the case, the pair is discarded. The probability of this event is 2^{-7} and thus after this step the expected number of remaining pairs is about $2^{m-1-7} = 2^{m-8}$.

Step 2. We deduce $RK_{(0,14)}$ and $RK_{(1,14)}$ from the guessed $RK_{(0,15)}$ and $RK_{(1,15)}$, decrypt round 14 and check if the difference is the expected one according to Fig. 6. If it is not the case, the pair is discarded. The probability of this event is 2^{-4} and therefore the expected number of pairs surviving this step is about $2^{m-8-4} = 2^{m-12}$.

Step 3. We guess the 32-bit $RK_{(0,13)}$ and partially decrypt the left branch of round 13 and check if the difference meets the impossible differential difference. Once it is correct, we delete the 32-bit round key guesses of $RK_{(0,13)}$ since such a differential is impossible; each round key guess that proposes such a difference is a wrong key. After analyzing all the 2^{m-12} remaining pairs, we output the 96-bit round keys guess of $RK_{(0,15)}$, $RK_{(1,15)}$, and $RK_{(0,13)}$ as a candidate. The probability that the pairs pass this step is about 2^{-2}, therefore the time complexity of this step is the number of key guesses $\times$ 2 messages in each pair $\times$ the probability that the key guess is excluded after sequentially testing it against all the surviving pairs.

The steps of the key recovery phase are described in Table 1, whereas the second column gives the round keys to be guessed in the corresponding round for each attack step. The third column presents the number of surviving pairs after each step, and the fourth column is the time complexity of each step measured in 15-round encryption.

Table 1: Key recovery process of the attack on 15-round SPARX-64/128

Attack step	Guessed keys	# Surviving pairs	Time complexity
1	$RK_{(0,15)}$ $RK_{(1,15)}$	$2^{m-1-7} = 2^{m-8}$	$2^{64} \times 2 \times 2^{m-1} \times 1/15 \approx 2^{m+60.1}$
2	†	$2^{m-8-4} = 2^{m-12}$	$2^{64} \times 2 \times 2^{m-8} \times 1/15 \approx 2^{m+53.1}$
3	$RK_{(0,13)}$	–	$2^{96} \times 2 \times [1 + (1-2^{-2}) + (1-2^{-2})^2 + \cdots + (1-2^{-2})^{2^{m-12}}] \times 1/(2 \times 15)$

†: No additional key guesses needed, i.e., the round keys are deduced from the previously guessed ones.

Attack complexity. To balance the attack complexity between the different phases, we take $m = 19$. This means that after analyzing all the remaining pairs, there will be about $2^{96} \times (1-2^{-2})^{2^{m-12}} = 2^{96} \times (0.75)^{128} \approx 2^{42.9}$ remaining candidates for the 96-bit round keys. Then, we guess the 32-bit $RK_{(1,12)}$ which along with the surviving candidates allows us to recover the master key K via the key schedule. Afterwards, we test each one of these master key candidates using 2 plaintext/ciphertext pairs to find the correct master key. The time complexity of this exhaustive search step is $2 \times 2^{32} \times 2^{42.9} = 2^{75.9}$. Therefore the time complexity is dominated by step 3 of the attack and estimated to be $2^{96} \times 2 \times (1/2^{-2}) \times (1/30) \approx 2^{94.1}$. The data complexity of the attack is $2^{19+32} = 2^{51}$ chosen plaintexts. The memory complexity of the attack is dominated by the memory that is required to store the keys to be excluded, i.e., $2^{42.9} \times 96/64 \approx 2^{43.5}$ 64-bit blocks.

4.2 16-round Impossible Differential Attack on SPARX-64/128

Although each round of SPARX-64/128 uses a 64-bit round key, there exists 3 specific rounds that contain only 2^{96} bits of key information as exemplified by the ones exploited in the previous attack. Nonetheless, any 4 rounds contain at least 128 bits of key information. Therefore, our 16-round attack on SPARX-64/128 has to start from round 2 and in this case, we use the 13-round impossible differential and prepend 3 rounds on its top as shown in Fig. 7. Again, we have used the Gurobi optimizer to find these 3 rounds after creating the MILP model that describes them.

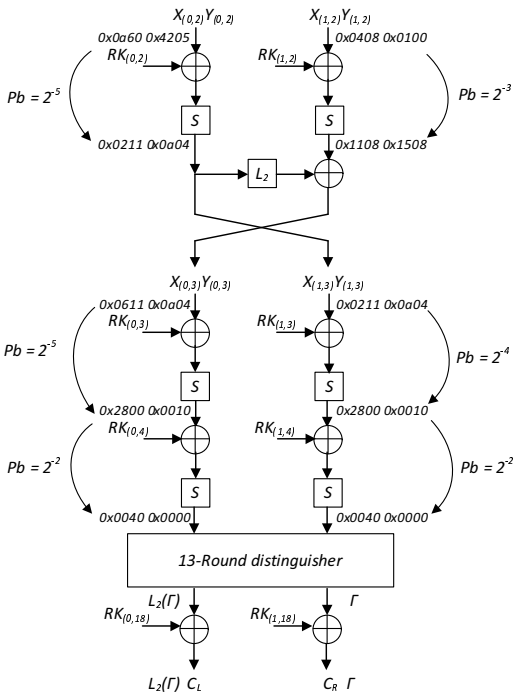


Fig. 7: 16-round impossible differential attack on SPARX-64/128

In this attack, we do not use data structures as they do not generate enough pairs to launch the attack. Instead, we use known plaintexts and generate the pairs we need probabilistically. Hence, if we have $2^{61.5}$ known plaintexts, these can generate $\binom{2^{61.5}}{2} \approx 2^{122}$ pairs. Out of these pairs, we would have $2^{122-64} = 2^{58}$ pairs that satisfy the plaintext difference shown in Fig. 7. Then, as the difference at the end of the distinguisher is the difference in the ciphertext, we have to filter the ciphertexts such as the right branch is a nonzero difference Γ and the left

branch difference is $L_2(\Gamma)$ which means that we have $2^{58-32} = 2^{26}$ proper pairs.

In the key recovery phase which we perform on these 2^{26} pairs, the 3 round keys take 2^{96} values only and they are guessed on steps to reduce the time complexity of the attack as listed in Table 2. It is to be noted that, according to the key schedule, $RK_{(0,3)}$, $RK_{(1,3)}$ are deduced from the guessed $RK_{(0,2)}$, $RK_{(1,2)}$ and that $RK_{(1,4)}$ is deduced from $RK_{(0,3)}$.

Table 2: Key recovery process of the attack on 16-round SPARX-64/128

Attack step	Guessed keys	# Surviving pairs	Time complexity
1	$RK_{(0,2)}$ $RK_{(1,2)}$	$2^{26-8} = 2^{18}$	$2^{64} \times 2 \times 2^{26} \times 1/16 = 2^{87}$
2	†	$2^{18-9} = 2^{9}$	$2^{64} \times 2 \times 2^{18} \times 1/16 = 2^{79}$
3	†	$2^{9-2} = 2^{7}$	$2^{64} \times 2 \times 2^{9} \times 1/(2 \times 16) = 2^{69}$
4	$RK_{(0,4)}$	–	$2^{96} \times 2 \times [1 + (1 - 2^{-2}) + (1 - 2^{-2})^2 + \cdots + (1 - 2^{-2})^{2^7}] \times 1/(2 \times 16)$

†: No additional key guesses needed, i.e., the round keys are deduced from the previously guessed ones.

After analyzing all the remaining pairs, there will be about $2^{96} \times (1-2^{-2})^{2^7} = 2^{96} \times (0.75)^{128} \approx 2^{42.9}$ remaining candidates for the 96-bit round keys. Then, we guess the remaining 32 bits of the master key and test each one of these master key candidates using 2 plaintext/ciphertext pairs to find the correct one. The time complexity of this exhaustive search step is $2 \times 2^{32} \times 2^{42.9} = 2^{75.9}$. Therefore the time complexity is dominated by step 4 of the attack (see Table 2) and estimated to be $2^{96} \times 2 \times (1/2^{-2}) \times (1/32) = 2^{94}$. The data complexity of the attack is $2^{61.5}$ known plaintexts. In this case, the memory complexity of the attack is dominated by the hash table [11] that is used to store the plaintexts while generating the required pairs, i.e., $2^{61.5}$ 64-bit blocks.

5 Conclusion

In this paper, we have analyzed SPARX-64/128 against the impossible differential attack. We have presented 12 and 13-round impossible differential distinguishers that are used to attack 15 and 16-round SPARX-64/128 with the post-whitening key, respectively. The (data complexity in chosen/known plaintexts, time complexity in 15/16-round encryptions, memory complexity in 64-bit blocks) of these attacks are $(2^{51}, 2^{94.1}, 2^{43.5})$ and $(2^{61}, 2^{94}, 2^{61.5})$, respectively.

References

1. Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers. The SIMON and SPECK Families of Lightweight Block Ciphers. Cryptology ePrint Archive, Report 2013/404, 2013. http://eprint.iacr.org/2013/404.
2. Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers. SIMON and SPECK: Block Ciphers for the Internet of Things. Cryptology ePrint Archive, Report 2015/585, 2015. http://eprint.iacr.org/2015/585.
3. Eli Biham, Alex Biryukov, and Adi Shamir. Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials. In Jacques Stern, editor, *Advances in Cryptology - EUROCRYPT 99*, volume 1592 of *Lecture Notes in Computer Science*, pages 12–23. Springer Berlin Heidelberg, 1999.
4. Joan Daemen and Vincent Rijmen. The Wide Trail Design Strategy. In Bahram Honary, editor, *Cryptography and Coding: 8th IMA International Conference Cirencester, UK, December 17–19, 2001 Proceedings*, pages 222–238. Springer Berlin Heidelberg, 2001.
5. Daniel Dinu, Léo Perrin, Aleksei Udovenko, Vesselin Velichkov, Johann Groschdl, and Alex Biryukov. Design Strategies for ARX with Provable Bounds: SPARX and LAX (Full Version). Cryptology ePrint Archive, Report 2016/984, 2016. http://eprint.iacr.org/2016/984.
6. Daniel Dinu, Léo Perrin, Aleksei Udovenko, Vesselin Velichkov, Johann Großschädl, and Alex Biryukov. Design Strategies for ARX with Provable Bounds: Sparx and LAX. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *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*, pages 484–513. Springer Berlin Heidelberg, 2016.
7. Kai Fu, Meiqin Wang, Yinghua Guo, Siwei Sun, and Lei Hu. MILP-Based Automatic Search Algorithms for Differential and Linear Trails for Speck. In Thomas Peyrin, editor, *Fast Software Encryption: 23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers*, pages 268–288. Springer Berlin Heidelberg, 2016.
8. Gurobi Optimization Inc. Gurobi Optimizer Reference Manual, 2016. http://www.gurobi.com.
9. Lars Knudsen. DEAL: A 128-bit block cipher. *Complexity*, 258(2), 1998. NIST AES Proposal.
10. Lu, Jiqiang and Kim, Jongsung and Keller, Nathan and Dunkelman, Orr. Improving the Efficiency of Impossible Differential Cryptanalysis of Reduced Camellia and MISTY1. In Tal Malkin, editor, *Topics in Cryptology – CT-RSA 2008: The Cryptographers' Track at the RSA Conference 2008, San Francisco, CA, USA, April 8-11, 2008. Proceedings*, pages 370–386. Springer Berlin Heidelberg, 2008.
11. Hamid Mala, Mohammad Dakhilalian, Vincent Rijmen, and Mahmoud Modarres-Hashemi. Improved Impossible Differential Cryptanalysis of 7-Round AES-128. In Guang Gong and Kishan Chand Gupta, editors, *Progress in Cryptology - INDOCRYPT 2010: 11th International Conference on Cryptology in India, Hyderabad, India, December 12-15, 2010. Proceedings*, pages 282–291. Springer Berlin Heidelberg, 2010.