Meet-in-the-Middle Attacks on Round-reduced Khudra

Mohamed Tolba, Ahmed Abdelkhalek, and Amr M. Youssef

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

Abstract. Khudra is a hardware-oriented lightweight block cipher that is designed to run efficiently on Field Programmable Gate Arrays. It employs an 18-rounds Generalized type-2 Feistel Structure with a 64-bit block length and an 80-bit key. In this paper, we present Meet-in-the-Middle (MitM) attacks on 13 and 14 round-reduced Khudra. These attacks are based on finding a distinguisher that is evaluated offline independently of the key. Then in an online phase, some rounds are appended before and after the distinguisher and the correct key candidates for these rounds are checked whether they verify the distinguisher property or not. Using this technique, we find two 6-round distinguishers and use them to attack 13 and 14 rounds of Khudra with time complexity of $2^{66.11}$ and $2^{66.19}$, respectively. Both attacks require the same data and memory complexities of 2^{51} chosen plaintexts and $2^{64.8}$ 64-bit blocks, respectively.

Keywords: Cryptanalysis, Meet-in-the-middle attacks, Generalized type-2 Feistel Structure.

1 Introduction

Recently, the design and cryptanalysis of lightweight block ciphers have received a lot of attention due to the demand for cryptographic protection in the increasingly used resource constrained devices such as RFIDs and wireless sensor networks. Designing an efficient hardware-oriented lightweight block cipher is a challenging task. Therefore, novel design criteria were proposed such as the use of a simple round function along with a simple key schedule. Examples of lightweight block ciphers that use these new techniques are HIGHT [16], PRESENT [3], KATAN/KTANTAN [5], KLEIN [12], Zorro [11], TWINE [19], and Khudra [17]. With such simple design, lightweight block ciphers should be deeply scrutinized in order to guarantee their security.

Unlike Application-Specific Integrated Circuits (ASICs), low cost Field Programmable Gate Arrays (FPGAs) are reconfigured and upgraded easily and therefore are now used extensively in numerous network applications. Consequently, lightweight block ciphers have to be designed with the goal of being integrated with FPGA applications in order to guarantee their security. Khudra is a new lightweight block cipher that was proposed by Kolay and Mukhopadhyay

at SPACE 2014 [17] in order to address the issue of efficient lightweight block ciphers that operate on FPGAs. To have an efficient lightweight block cipher for deployment on FPGAs, a new design criterion, namely, recursive structure was proposed. Khudra has a 64-bit block length and employs an 80-bit key. Its structure inherits the Generalized type-2 Feistel Structure (GFS) that was proposed by Hoang and Rogaway [15]. In particular, it uses 4 branches each of 16-bit length.

In 1977, Diffie and Hellman proposed the MitM attack to be used in the cryptanalysis of Data Encryption Standard (DES) [9]. The MitM attack is one of the major attacks on block ciphers as it requires low data complexity. Its time complexity is, however, very close to that of an optimized exhaustive key search. Hence, enhancing its time complexity and increasing the number of attacked rounds have always been hot topics in cryptanalysis. For example, Bogdanov and Rechberger proposed the 3-Subset MitM attack and applied it to the full KTANTAN cipher [4]. Zhu and Guang presented multidimensional MitM against KATAN32/48/64 [20]. Demirci and Selçuk attacked 8 rounds of both AES-192 and AES-256 using MitM techniques [6]. The main drawback of their attack is the high memory requirement. To tackle the high memory requirement issue, Dunkelman, Keller and Shamir put forward a couple of new ideas. Particularly, they presented the concepts of differential enumeration and multisets [10] that have drastically decreased the high memory requirement of the attack of Demirci and Selçuk. Later on, Derbez *et al.* enhanced the attack and decreased the memory requirement even further which made it possible on AES-128 [7]. The MitM techniques, which were developed to attack AES and Substitution Permutation Network (SPN) based block ciphers such as Hierocrypt-3 [1] and mCrypton [14], were also proven to be equally powerful on Feistel constructions, as exemplified by the generic work done by Guo *et al.* [13] and Lin *et al.* [18]. Finally, at FSE 2015, two MitM attacks based on the Demirci and Selçuk approach were presented on the SPN structure PRINCE [8] and the Feistel construction TWINE [2].

In this paper, we present MitM attacks on 13 and 14 rounds of Khudra. In the attack on 13 rounds, we first construct a 6-round distinguisher, append three rounds at the top and four rounds at the bottom. To attack 14 rounds, the same distinguisher would require the whole key to be guessed, therefore we construct a different 6-round distinguisher, and append three rounds at the top and five rounds at the bottom. The time complexities of these attacks are $2^{66.11}$ to attack 13 rounds and $2^{66.19}$ to attack 14 rounds, respectively. Both attacks require the same data and memory complexities of 2^{51} chosen plaintext and $2^{64.8}$ 64-bit blocks. To the best of our knowledge, these are the best attacks on Khudra so far.

The rest of the paper is organized as follows. In section 2, we provide a brief description of Khudra and the notations used throughout the paper. Our attacks are presented in section 3 and the paper is concluded in section 4.

2 Specifications of Khudra

Khudra is an iterated lightweight block cipher that operates on 64-bit blocks using an 80-bit key and employs a Generalized Feistel Structure (GFS). It has four branches of 16-bit each, i.e., the state is divided into four words and each word is 16-bit long. The cipher iterates over 18 rounds where in every round, an unkeyed 16×16-bit F-function is applied on two words. This unkeyed F-function, designed to be efficient when deploying Khudra on FPGAs, uses a 6-round GFS as depicted in the right side of Figure 1. Each round of these 6-round GFS has two 4×4-bit SBoxes identical to the SBox used in PRESENT [3]. After applying the F-functions of round i, two 16-bit round keys RK_{2i} and RK_{2i+1} are xored to the state along with the other two words to generate the two new words of round $i+1$ for $i = 0, 1, \cdots, 17$. Additionally, two pre-whitening keys WK_0 and WK_1 are xored with the plaintext before the first round and two other post-whitening keys WK_2 and WK_3 are xored with the internal state after the last round and before generating the ciphertext.

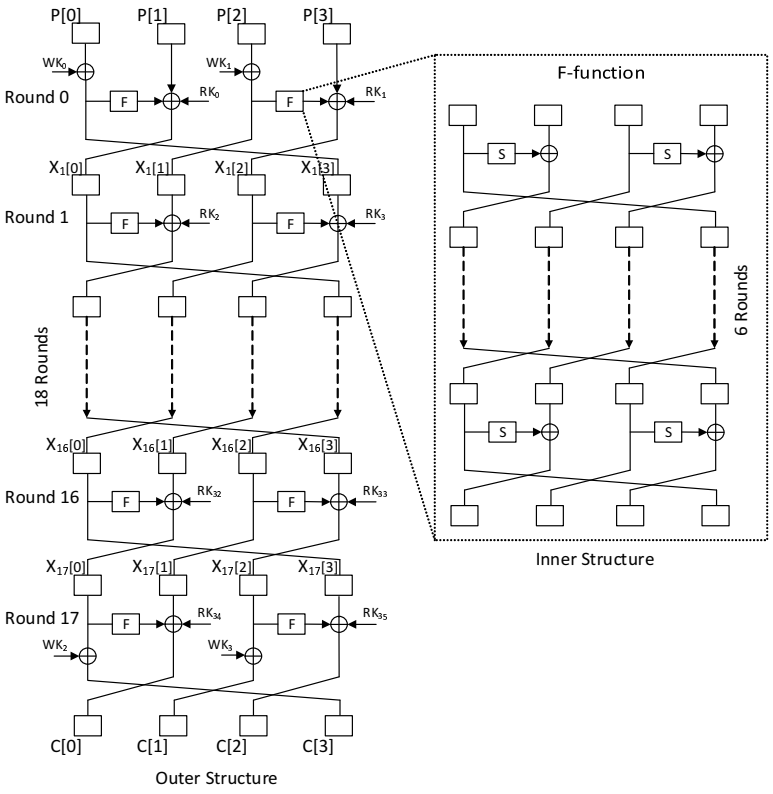

Fig. 1. Structure of Khudra

The key schedule of Khudra takes an 80-bit master key K and splits it into five keys k_i of 16-bit each where $K = k_0||k_1||k_2||k_3||k_4$. Then, it generates 16-bit 36 round keys $RK_i, 0 \leqslant i < 36$, two per round, and four 16-bit whitening keys $WK_i, 0 \leqslant i < 4$, as shown in Algorithm 1.

Data: Key Scheduling$(k_0, k_1, k_2, k_3, k_4)$
Result: $WK_i, 0 \leqslant i < 4$ and $RK_i, 0 \leqslant i < 36$
$WK_0 \leftarrow k_0, WK_1 \leftarrow k_1, WK_2 \leftarrow k_3, WK_3 \leftarrow k_4$;
for $i \leftarrow 0$ *to* 35 **do**
  $RC_i \leftarrow 0||i_{(6)}||00||i_{(6)}||0$;
  $RK_i \leftarrow k_{i \bmod 5} \oplus RC_i$;
end

Algorithm 1: The Key Schedule employed in Khudra [17]

2.1 Notations

The following notations will be used throughout the rest of the paper:

- K: The master key.
- k_i: The i^{th} 16-bit of K, where $0 \leq i < 5$.
- RK_i: The 16-bit key used in round $\lfloor i/2 \rfloor$.
- WK_i: The 16-bit whitening key, where $0 \leq i < 4$.
- X_i: The 64-bit input to round i, where $0 \leq i \leq 18$, X_0 is the plaintext P and X_{18} is the ciphertext C.
- $X_i[l]$: The l^{th} 16-bit word of X_i, where $0 \leq l < 4$.
- $\Delta X_i, \Delta X_i[l]$: The difference at state X_i and word $X_i[l]$, respectively.
- X_i^j: The j^{th} state of the 64-bit input to round i.
- $X_i^j[l]$: The l^{th} 16-bit word of the j^{th} state of the 64-bit input to round i.

We measure the memory complexity of our attacks in number of 64-bit Khudra blocks and the time complexity in terms of the equivalent number of round-reduced Khudra encryptions.

3 MitM Attacks on round-reduced Khudra

In our MitM attacks, Khudra is split into three sub-ciphers such that $E_K(P) = E_{K_2} \circ E_{dis} \circ E_{K_1}(P)$, where E_{dis} is the middle part which exhibits a distinguishing property. In the offline phase, that particular property is evaluated independently of the keys used in the middle rounds. Then in the online phase, correct K_1 and K_2 key candidates are checked whether they verify this distinguishing property or not.

The *b-δ-set* concept [13], as captured by Definition 1, is used to build our distinguisher. Using a *b-δ-set* enables us to reduce the memory and data complexities of our distinguisher.

Definition 1. *(b-δ-set, [13]). A b-δ-set is a set of 2^b state values that are all different in b state bits (the active bits) and are all equal in the remaining state bits (the inactive bits).*

In the following subsections, we demonstrate our attacks on 13 and 14 rounds of Khudra in details.

3.1 A MitM Attack on 13-Round Khudra

A *b-δ-set* is employed in our MitM attack where we set $b = 3$, i.e., 3 active bits. b is chosen in order to reduce the memory and data requirements of the attack without increasing its time complexity. In our 13-round attack, the active word is $P[1]$, i.e., the second word. The 3 active bits can take any position in this 16-bit word. Such 3-*δ-set* enables us to build a 6-round distinguisher, as depicted in Figure 2, by the following proposition:

Proposition 1. *Consider the encryption of* 3-*δ-set* $\{P^0, P^1, ..., P^7\}$ *through six rounds of Khudra. The ordered sequence* $[X_6^0[3] \oplus X_6^1[3], X_6^0[3] \oplus X_6^2[3], ..., X_6^0[3] \oplus X_6^7[3]]$ *is fully determined by the following 4 16-bit parameters,* $X_1^0[0]$, $X_2^0[0]$, $X_3^0[0]$ *and* $X_4^0[0]$.

The above proposition means that we have $2^{4\times16} = 2^{64}$ ordered sequences out of the $2^{(2^3-1)\times16} = 2^{112}$ theoretically possible ones.

Proof. The knowledge of the 3-*δ-set* $= \{P^0, P^1, \cdots, P^7\}$ allows us to determine $[P^0 \oplus P^1, P^0 \oplus P^2, \cdots, P^0 \oplus P^7]$. In what follows we show how the knowledge of the 4 16-bit parameters mentioned in proposition 1 is enough to compute the ordered sequence of the differences at $X_6[3]$. As there is no F-function involved in the first round, the difference $\Delta P[1]$ is propagated through the first round as is. The knowledge of $X_1^0[0]$ enables us to bypass the F-function of the second round to compute $\Delta X_2[0]$. Then, the knowledge of $X_2^0[0]$ enables us to bypass the F-function of the third round to compute $\Delta X_3[0]$ and the previous steps are repeated until we compute $\Delta X_6[3]$. It is to be noted that after the third (resp. fourth) round, $X_3[3]$ (resp. $X_4[3]$) should have non-zero difference because $X_2[0]$ (resp. $X_3[0]$) has non-zero difference. However, these differences are omitted from Figure 2 since they do not impact the ordered sequence at $X_6[3]$.

The previous proposition is utilized to attack 13-round Khudra by appending 3 rounds on top of it and 4 rounds below it, as illustrated in Figure 3. The attack has two phases and proceeds as follows:

Offline Phase. Build the distinguisher property by determining all the 2^{64} ordered sequences as illustrated in Proposition 1 and save them in a hash table H.

Online Phase. As illustrated in Figure 3, the online phase advances as follows:

1. A plaintext P^0 is chosen to act as a reference to all the differences in the 3-δ-set.

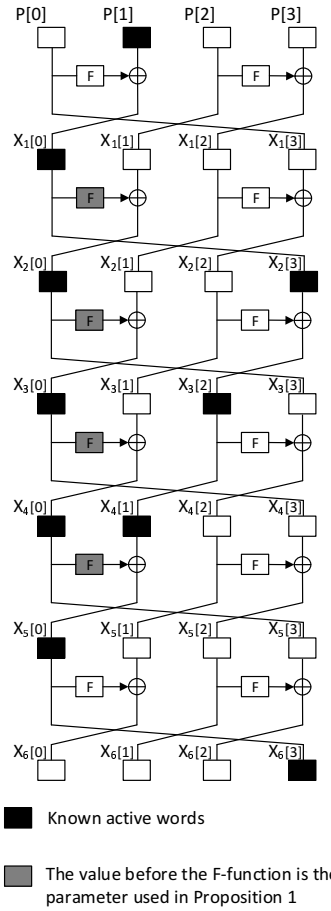

Fig. 2. 6-Round distinguisher to attack 13-round Khudra

2. The 3-δ-set P^0, P^1, $\cdots$, P^7 is determined by guessing the state variables $X_1^0[3]$, $X_1^0[1]$, $X_1^0[0]$, $X_2^0[2]$ to decrypt the 3-δ-set differences $[X_3^0[1] \oplus X_3^1[1], X_3^0[1] \oplus X_3^2[1], \cdots, X_3^0[1] \oplus X_3^7[1]]$.
3. The corresponding ciphertexts C^0, C^1, $\cdots$,C^7 are requested.
4. The differences in $[X_9^0[3] \oplus X_9^1[3], X_9^0[3] \oplus X_9^2[3], \cdots, X_9^0[3] \oplus X_9^7[3]]$ are determined by guessing the state variables $X_9^0[2]$, $X_{10}^0[0]$, $X_{11}^0[0]$, $X_{11}^0[2]$, $X_{12}^0[0]$, $X_{12}^0[2]$ that are required to decrypt the ciphertext differences $[C^0 \oplus C^1, C^0 \oplus C^2, \cdots, C^0 \oplus C^7]$.
5. The guessed state variables are filtered by checking if the computed ordered sequence exists in H or not.

Steps 2 and 4 require the guessing of 10 words and the attack time complexity would then exceed the exhaustive key search. Therefore, we investigate

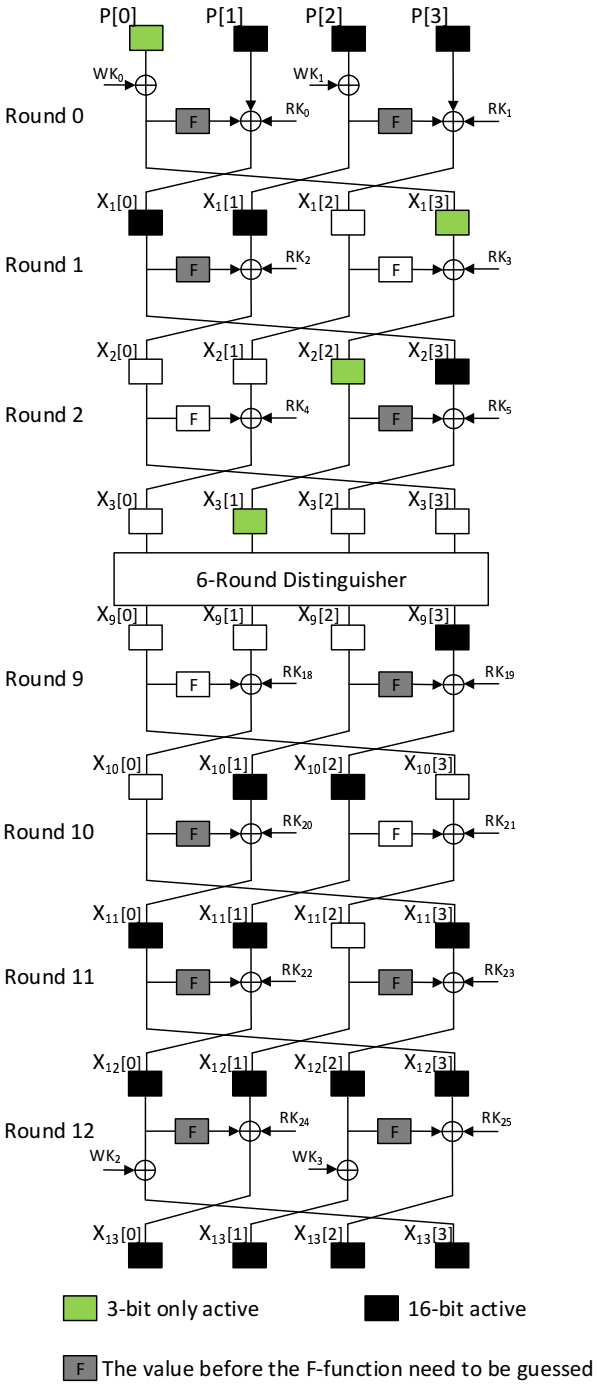

Fig. 3. 13-Round attack on Khudra

the key schedule aiming to find relations between the round keys and thus reduce the number of guessed words. Indeed, we find that by guessing k_0, k_1, k_3, and with the knowledge of P^0, we can compute $X_1^0[3]$, $X_1^0[1]$, $X_1^0[0]$, $X_2^0[2]$ and by guessing k_0, k_3, k_4, and with the knowledge of C^0, C^1, $\cdots$, C^7 and $[C^0 \oplus C^1, C^0 \oplus C^2, \cdots, C^0 \oplus C^7]$, we can compute $X_9^0[2]$, $X_{10}^0[0]$, $X_{11}^0[0]$, $X_{11}^0[2]$, $X_{12}^0[0]$, $X_{12}^0[2]$. Therefore, instead of guessing 10 words, only 4 key words k_0, k_1, k_3, k_4 are to be guessed. The probability of a wrong key resulting in an ordered sequence in H is $2^{64-(7\times16)} = 2^{-48}$. As we have 2^{64} key guesses, we expect that only $2^{64-48} = 2^{16}$ keys will remain. Hence, we guess k_2 to fully recover the master key and test it using two plaintext/ciphertext pairs.

Attack Complexity. The memory complexity of the attack is determined by the memory required to store the hash table H in the offline phase. This table has 2^{64} entries where each entry contains seven 16-bit words, i.e., 112 bits. Therefore, the memory complexity is given by $2^{64} \times 112/64 = 2^{64.8}$ 64-bit blocks. The data complexity is determined from step 2. As shown in Figure 3, after the decryption of step 2, three words are fully active, i.e., they assume all the 2^{16} possible values while the fourth word has only three active bits, i.e., assumes 2^3 possible values only in correspondence to the 3-δ-set. Therefore, the data complexity of the attack is upper bounded by 2^{51} chosen plaintext. The time complexity of the offline phase is determined by the time required to build the hash table H and is estimated to be $2^{64} \times 8 \times 4/(2 \times 13) = 2^{64.3}$. The complexity of the online phase includes the time required to filter the key space and is estimated to be $2^{64} \times 8 \times (4+6)/(2 \times 13) = 2^{65.62}$. It also includes the time to exhaustively search through the remaining key candidates along with the guess of k_2 using two plaintext/ciphertext pairs and is estimated to be $2 \times 2^{(64-48)} \times 2^{16} = 2^{33}$. Therefore, the overall time complexity of the attack is estimated to be $2^{64.3} + 2^{65.62} + 2^{33} \approx 2^{66.11}$ 13-round Khudra encryptions.

3.2 A MitM Attack on 14-Round Khudra

Reusing the same distinguisher to extend our attack by one round requires guessing the 5 words of the key. Therefore, we construct another distinguisher, depicted in Figure 4, to attack 14-round reduced Khudra without the post-whitening keys. The active word in this new distinguisher is $P[3]$. It is built according to proposition 2 below and, as in the previous attack, b is set to 3.

Proposition 2. *Consider the encryption of 3-δ-set $\{P^0, P^1, \cdots, P^7\}$ through six rounds of Khudra. The ordered sequence $[X_6^0[1] \oplus X_6^1[1], X_6^0[1] \oplus X_6^2[1], \cdots, X_6^0[1] \oplus X_6^7[1]]$ is fully determined by the following 4 16-bit parameters $X_1^0[2]$, $X_2^0[2]$, $X_3^0[2]$ and $X_4^0[2]$.*

By appending three rounds on top of this new distinguisher and five rounds beneath it, we are able to attack 14-round Khudra. The attack proceeds as the previous one, as illustrated in Figure 5, with the exception that the active word

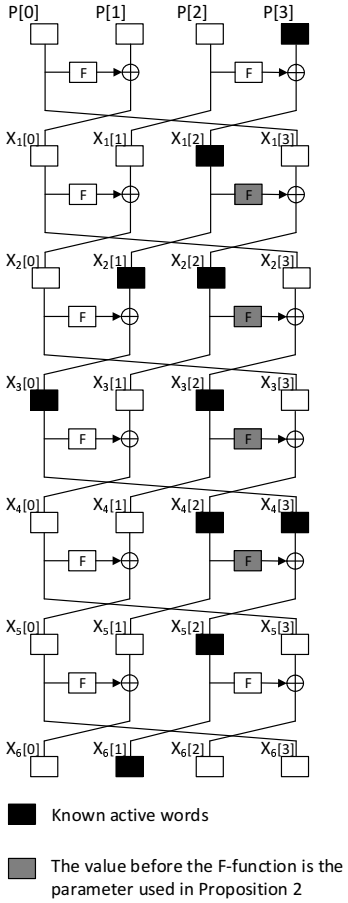

Fig. 4. 6-Round distinguisher to attack 14-round Khudra

is $X_3[3]$ rather than $X_3[1]$ in the 13-round attack and the ordered sequence is calculated at $X_9[1]$ instead of $X_9[3]$. Guessing k_0, k_1, k_2 with the knowledge of P^0 enables us to compute the state variables needed to determine the 3-δ-set. In order to determine the ordered sequence, we need to guess k_0, k_1, k_2, k_4. Therefore, guessing the four key words, k_0, k_1, k_2, k_4 allows us to mount an attack on 14-round Khudra.

Attack Complexity. The memory and data complexities of this attack are similar to the previous one, i.e., $2^{64.8}$ 64-bit blocks and 2^{51} chosen plaintext, respectively. The time complexity is $2^{64}\times 8\times 4/(2\times 14)+2^{64}\times 8\times(4+8)/(2\times 14)+ 2\times 2^{(64-48)}\times 2^{16} = 2^{64.19}+2^{65.78}+2^{33}\approx 2^{66.19}$ 14-round Khudra encryptions.

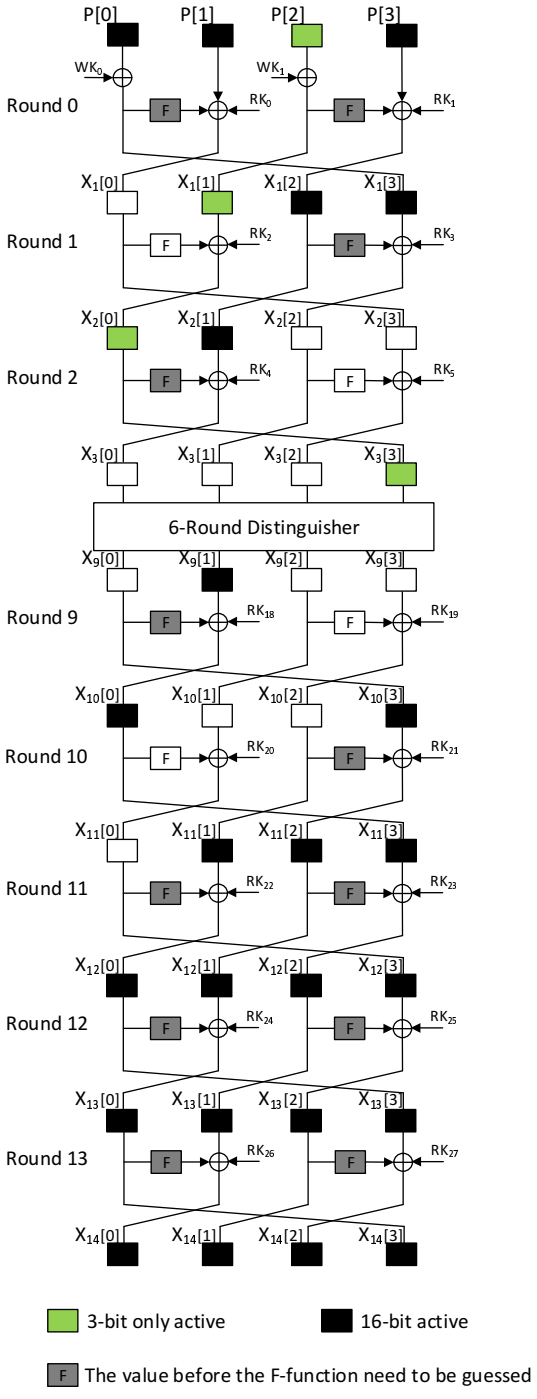

Fig. 5. 14-Round attack on Khudra

4 Conclusion and Discussion

We presented MitM attacks on Khudra. The time complexities of the attacks are given by $2^{66.11}$ and $2^{66.19}$ for the 13-round and 14-round reduced cipher, respectively. Both attacks have the same data and memory complexities of 2^{51} chosen plaintext and $2^{64.8}$ 64-bit blocks, respectively. To the best of our knowledge, these are the best known attacks on Khudra.

Finally, we briefly discuss why we did not use the notion of differential enumeration. In the attack of Dunkelman *et al.* [10], the differential enumeration technique helps reduce the number of parameters by using the differential property of the SBox over one round. In Feistel constructions, the differential property of the SBox can be utilized over at least two rounds and can be extended further depending on the specific structure of the scheme. However, in the case of Khudra, differential enumeration does not help reduce the number of parameters because propagating the difference backward requires a set of parameters that is different than the set of parameters needed to compute the ordered sequence. In other words, using the differential enumeration technique reduces the number of parameters by using the differential property of the SBox but at the same time, incurs additional parameters to be guessed in order to propagate the difference backward. Since Khudra has an 80-bit key, the number of parameters is limited to 4 16-bit parameters. Using the differential enumeration technique, the best 6-round distinguisher that we are able to construct requires 6 parameters which renders the attack worse than exhaustive key search.

References

1. Ahmed Abdelkhalek, Riham Altawy, Mohamed Tolba, and Amr M. Youssef. Meet-in-the-Middle Attacks on Reduced-Round Hierocrypt-3. In *LatinCrypt*. Lecture Notes in Computer Sience, Springer, *to appear*, 2015.
2. Alex Biryukov, Patrick Derbez, and Léo Paul Perrin. Differential Analysis and Meet-in-the-Middle Attack against Round-Reduced TWINE. Fast Software Encryption 2015. *to appear*.
3. Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. Robshaw, Yannick Seurin, and C. Vikkelsoe. PRESENT: An Ultra-Lightweight Block Cipher. In Pascal Paillier and Ingrid Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems - CHES 2007*, volume 4727 of *Lecture Notes in Computer Science*, pages 450–466. Springer Berlin Heidelberg, 2007.
4. Andrey Bogdanov and Christian Rechberger. A 3-subset meet-in-the-middle attack: Cryptanalysis of the lightweight block cipher KTANTAN. In Alex Biryukov, Guang Gong, and Douglas R. Stinson, editors, *Selected Areas in Cryptography*, volume 6544 of *Lecture Notes in Computer Science*, pages 229–240. Springer Berlin Heidelberg, 2010.
5. Christophe Cannière, Orr Dunkelman, and Miroslav Knežević. KATAN and KTANTAN - A family of small and efficient hardware-oriented block ciphers. In Christophe Clavier and Kris Gaj, editors, *Cryptographic Hardware and Embedded Systems - CHES 2009*, volume 5747 of *Lecture Notes in Computer Science*, pages 272–288, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

6. Hüseyin Demirci and Ali Aydn Selçuk. A Meet-in-the-Middle Attack on 8-Round AES. In Kaisa Nyberg, editor, *Fast Software Encryption*, volume 5086 of *Lecture Notes in Computer Science*, pages 116–126. Springer Berlin Heidelberg, 2008.
7. Patrick Derbez, Pierre-Alain Fouque, and Jérémy Jean. Improved Key Recovery Attacks on Reduced-Round AES in the Single-Key Setting. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 371–387. Springer Berlin Heidelberg, 2013.
8. Patrick Derbez and Léo Perrin. Meet-in-the-Middle Attacks and Structural Analysis of Round-Reduced PRINCE. Fast Software Encryption 2015. *to appear*.
9. Whitfield Diffie and Martin E Hellman. Special feature exhaustive cryptanalysis of the NBS data encryption standard. *Computer*, 10(6):74–84, June 1977.
10. Orr Dunkelman, Nathan Keller, and Adi Shamir. Improved Single-Key Attacks on 8-Round AES-192 and AES-256. In Masayuki Abe, editor, *Advances in Cryptology - ASIACRYPT 2010*, volume 6477 of *Lecture Notes in Computer Science*, pages 158–176. Springer Berlin Heidelberg, 2010.
11. Benoit Gèrard, Vincent Grosso, Maria Naya-Plasencia, and François-Xavier Standaert. Block ciphers that are easier to mask: How far can we go? In Guido Bertoni and Jean-Sèbastien Coron, editors, *Cryptographic Hardware and Embedded Systems - CHES 2013*, volume 8086 of *Lecture Notes in Computer Science*, pages 383–399. Springer Berlin Heidelberg, 2013.
12. Zheng Gong, Svetla Nikova, and Yee Wei Law. KLEIN: A new family of lightweight block ciphers. In Ari Juels and Christof Paar, editors, *RFID. Security and Privacy*, volume 7055 of *Lecture Notes in Computer Science*, pages 1–18. Springer Berlin Heidelberg, 2011.
13. Jian Guo, Jérémy Jean, Ivica Nikolić, and Yu Sasaki. Meet-in-the-Middle Attacks on Generic Feistel Constructions. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology ASIACRYPT 2014*, volume 8873 of *Lecture Notes in Computer Science*, pages 458–477. Springer Berlin Heidelberg, 2014.
14. Yonglin Hao, Dongxia Bai, and Leibo Li. A Meet-in-the-Middle Attack on Round-Reduced mCrypton Using the Differential Enumeration Technique. In ManHo Au, Barbara Carminati, and C.-C.Jay Kuo, editors, *Network and System Security*, volume 8792 of *Lecture Notes in Computer Science*, pages 166–183. Springer International Publishing, 2014.
15. Viet Tung Hoang and Phillip Rogaway. On Generalized Feistel Networks. In Tal Rabin, editor, *Advances in Cryptology CRYPTO 2010*, volume 6223 of *Lecture Notes in Computer Science*, pages 613–630. Springer Berlin Heidelberg, 2010.
16. Deukjo Hong, Jaechul Sung, Seokhie Hong, Jongin Lim, Sangjin Lee, Bon-Seok Koo, Changhoon Lee, Donghoon Chang, Jesang Lee, Kitae Jeong, Hyun Kim, Jongsung Kim, and Seongtaek Chee. HIGHT: A new block cipher suitable for low-resource device. In Louis Goubin and Mitsuru Matsui, editors, *Cryptographic Hardware and Embedded Systems - CHES 2006*, volume 4249 of *Lecture Notes in Computer Science*, pages 46–59, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.
17. Souvik Kolay and Debdeep Mukhopadhyay. Khudra: A New Lightweight Block Cipher for FPGAs. In Rajat Subhra Chakraborty, Vashek Matyas, and Patrick Schaumont, editors, *Security, Privacy, and Applied Cryptography Engineering*, volume 8804 of *Lecture Notes in Computer Science*, pages 126–145. Springer International Publishing, 2014.

18. Li Lin and Wenling Wu. Improved Meet-in-the-Middle Distinguisher on Feistel Schemes. IACR Cryptology ePrint Archive, 2015/051, 2015. https://eprint.iacr.org/2015/051.pdf.
19. Tomoyasu Suzaki, Kazuhiko Minematsu, Sumio Morioka, and Eita Kobayashi. TWINE: A lightweight block cipher for multiple platforms. In LarsR. Knudsen and Huapeng Wu, editors, *Selected Areas in Cryptography*, volume 7707 of *Lecture Notes in Computer Science*, pages 339–354. Springer Berlin Heidelberg, 2013.
20. Bo Zhu and Guang Gong. Multidimensional meet-in-the-middle attack and its applications to KATAN32/48/64. IACR Cryptology ePrint Archive, 2011/619, 2011. https://eprint.iacr.org/2011/619.pdf.