Journal of Information Security and Applications 41 (2018) 117–123

Contents lists available at ScienceDirect

Journal of Information Security and Applications

journal homepage: www.elsevier.com/locate/jisa

Single key MITM attack and biclique cryptanalysis of full round Khudra

Prakash Dey[a], Raghvendra Singh Rohit[b], Avishek Adhikari[a,*]

[a] Department of Pure Mathematics, University of Calcutta, Kolkata-700019, West Bengal, India
[b] Department of Electrical & Computer Engineering, University of Waterloo, 200 University Avenue West, Waterloo, ON, N2L 3G1 Canada

ARTICLE INFO

Article history:
Available online 11 July 2018

Keywords:
Cryptanalysis
Block cipher
Meet-in-the-middle attack
Biclique attack

ABSTRACT

Khudra, an efficient lightweight block cipher for FPGAs, requires at least around 45% less slices and 29% less AT product compared to round wise implementation of any of the contemporary lightweight block cipher. Though a few attacks on Khudra, mostly on meet-in-the-middle (MITM) attack, rectangle attack etc. with reduced rounds having high data complexity, are reported in the literature of cryptanalysis, none have been reported for single key attack on full round Khudra. In this paper, up to the best of our knowledge, we for the first time, present two single-key attacks on full round Khudra. In the first attack, we propose MITM attack on full round Khudra having time complexity $2^{79.61}$ and data complexity 2. Next we find a 3-round biclique in the plaintext side which we utilize to attack full round Khudra with a reduced time complexity of $2^{79.36}$.

1. Introduction

Block ciphers are one of the most important cryptographic primitives. Block ciphers operate on fixed length groups of plaintext-bits (plaintext blocks) using a fixed length secret key. Individually they provide confidentiality but can be used as fundamental building blocks for pseudo-random number generators, stream ciphers, MACs and hash functions.

Lightweight block ciphers are a special class of block cipher primitives that are capable of running on devices with very low computing power. To achieve lightness, generally block ciphers rely on elementary operations like binary XOR or AND etc., extremely simplified key schedule and to avoid large hardware footprint uses no or small S-Boxes. To complement this the required number of rounds are increased. Compact implementation of AES and DES, Kasumi, mCrypton, HIGHT, DESL and DESXL, CLEFIA, PRESENT, Puffin, MIBS, KATAN, Klein, TWINE, LED and Piccolo etc are some examples of lightweight block ciphers.

Khudra [1], a block cipher designed by Kolay and Mukhopadhyay, is the first reported work in designing a new lightweight cipher specifically for the growingly popular Field Programmable Gate Array (FPGA) platforms. Although ASICs are popular choice for lightweight cryptography, recent low cost FPGAs make them an alternative for battery powered devices [1]. FPGAs have the advantage over ASIC chips as FPGAs can be reconfigured or upgraded after manufacture.

Khudra operates on 64 bit blocks with 80 bits of key. To the best of our knowledge [2–6] are the only external analyses on Khudra. Tolba et al. [5] identified two 6-round distinguishers and used 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 $2^{64.8}$ many 64-bit blocks, respectively. In [3], Ma et al. propose a related-key rectangle attack on the 16-round Khudra without whitening key by constructing a related-key rectangle distinguisher for 12-round Khudra with a probability of $2^{-23.82}$. Yang et al. [6] proposed an attack which says that based on related-key impossible differentials for 32 related keys, one can launch an attack on the full Khudra with data complexity of 2^{63} related-key chosen-plaintexts, time complexity of about $2^{68.46}$ encryptions and memory complexity of 2^{64}. Ozen et al. [4] showed that the effective round key length is 16-bit. By the help of this observation, they improve the 14-round MITM attack proposed by Tolba et al. [5] by reducing the memory complexity from $2^{64.8}$ to $2^{32.8}$. Dai et al. [2] considered the security of Khudra against the related-key attack. By utilizing the observations on F-function and the searching algorithm, they launch related-key differential attacks on 16-round Khudra and full Khudra without whitening keys. Along with that they build a 13-round related-key rectangle distinguisher and proposed an attack on 16-round Khudra, which requires 2^{53} chosen plaintexts and $2^{64.08}$ encryptions.

* Corresponding author.
E-mail addresses: pdprakashdey@gmail.com (P. Dey), rsrohit@uwaterloo.ca (R.S. Rohit), avishek.adh@gmail.com (A. Adhikari).

https://doi.org/10.1016/j.jisa.2018.06.005
2214-2126/

Table 1
Summary of attacks on Khudra.

Paper	Type of attack		Round	Data	Complexity
[5]	Single-key	Meet-in-the-Middle Attack	13-rounds	2^{51}	$2^{66.11}$
	Single-key	Meet-in-the-Middle Attack	14-rounds	2^{51}	$2^{66.19}$
[3]	Related-Key	Rectangle Attack	16-round without whitening key	$2^{57.82}$	$2^{78.68}$
[6]	Related-Key	Impossible differential	Full cipher	2^{63}	$2^{68.46}$
[4]	Single-key	Guess-and-Determine Attack	14-rounds	2	2^{64}
[2]	Related-key		16-round	2^{53}	$2^{64.08}$
Current Work	Single-key	Meet-in-the-Middle Attack	Full cipher	2	$2^{79.61}$
	Single-key	Biclique Cryptanalysis	Full cipher	2^{36}	$2^{79.36}$

CONTRIBUTION OF THE PAPER. Though a few attacks on Khudra are proposed in the literature of cryptanalysis, none have been reported for single key attack on full round Khudra. In this paper, up to the best of our knowledge, we for the first time, present two single-key attacks on full round Khudra. First we show that full round Khudra is susceptible under MITM attack. The (known plaintext) attack utilizes matching with precomputation technique and has data complexity 2 and time complexity $2^{79.61}$. Next we show that full round Khudra can be attacked with a reduced time complexity using biclique cryptanalysis. We find a 3-round biclique in the plaintext side and show that by using precomputation and recomputation techniques the secret key can be recovered faster than brute-force. This attack has data complexity 2^{36} and time complexity $2^{79.36}$. Table 1 gives the comparison between our results and previous results.

ORGANIZATION OF THE PAPER. The rest of the paper is organised in the following way: In Section 2 we present some necessary notations that will be used throughout the paper. Khudra cipher is briefly described in Section 3. MITM attack on full round Khudra is described in Section 4. In Section 5 a general setup of biclique cryptanalysis for block ciphers is described. Section 6 provides biclique attack on full round Khudra. Finally Section 7 concludes the paper.

2. Notations

Before describing the Khudra cipher, we shall now fix some notations that will be used throughout the paper.

- **word** – a 4-bit sequence i.e., a nibble will be identified as a *word*. Thus Khudra operates on 16-word plaintext using 20-word secret key.
- A 16-bit block X will be identified by the ordered sequence $X = (X[0], X[1], X[2], X[3])$ of 4-words. In this case $X = X[0]||X[1]||X[2]||X[3]$, where || is the concatenation operator.
- A 64-bit block X will be identified by the ordered sequence $X = (X[0], X[1], X[2], X[3])$ of 16-bit blocks. In this case $X = X[0]||X[1]||X[2]||X[3]$.
- A 80-bit block X will be identified by the ordered sequence $X = (X[0], X[1], X[2], X[3], X[4])$ of 16-bit blocks. In this case $X = X[0]||X[1]||X[2]||X[3]||X[4]$.
- $\oplus$ – bitwise XOR.
- With the above notations, if X denotes a 64 or 80 bit block then $X[i][j]$ represents the j-th word of the 16-bit block $X[i]$ and $(4i + j)$-th word of X.

3. Brief description of Khudra

Khudra [1] is a lightweight block cipher. It operates on 64-bit plaintext blocks using an 80-bit key. By iterating over 18 rounds, it outputs an 64-bit ciphertext block. Khudra uses the S-Box, as shown in Table 2, of the block cipher PRESENT [7]. The S-Box operates on 4-bit blocks.

Let P be a 64-bit plaintext and k be an 80-bit key. The resulting 64-bit ciphertext is denoted by C. The key scheduling algorithm takes the 80-bit master key $k = (k[0], k[1], k[2], k[3], k[4])$ and produces 16-bit round keys RK_i $(0 \le i < 36)$ and 16-bit whitening keys WK_i $(0 \le i < 4)$. Whitening keys and round keys are generated according to Algorithm 1, as described below. RC_i denotes the 16-bit round constant and $i_{(6)}$ is the 6-bit representation of the round counter i.

Algorithm 1: Khudra key scheduling algorithm.

Input: $k = (k[0], k[1], k[2], k[3], k[4])$
Output: Whitening keys and round keys
$\mathrm{WK}_0 = k[0]$, $\mathrm{WK}_1 = k[1]$, $\mathrm{WK}_2 = k[3]$, $\mathrm{WK}_3 = k[4]$
for $i = 0$ *to* 35 **do**
  $\mathrm{RC}_i = \{0|i_{(6)}|00|i_{(6)}|0\}$
  $\mathrm{RK}_i = k[i \bmod 5] \oplus \mathrm{RC}_i$
end

The structure of the cipher is depicted in Fig. 1. The left structure in Fig. 1 is considered as Outer Structure, while the right structure, that is the structure for the F-function is considered as Inner Structure. In the inner structure, 4×4 S-boxes are used to provide non-linearity.

4. MITM Attack on Khudra

In this section we consider meet-in-the-middle attack on the full round Khudra. Note that the 18 khudra rounds are denoted by $i = 0, 1, \ldots, 17$. Let us denote the 64-bit state of Khudra after i-th round by S^i. For example, S^0 is the state after pre-whitening followed by one Khudra round, whereas S^{17} is the state just before post-whitening. We now consider Khudra as a composition of 3 subciphers: Khudra $= g \circ \epsilon \circ f$, as shown below

$$P \xrightarrow[f]{k} S^5 \xrightarrow[\epsilon]{k} S^{15} \xrightarrow[g]{k} C.$$

4.1. Key space partitioning

Let k be an 80-bit i.e., 20-word Khudra key. Then $k = (k[0], k[1], k[2], k[3], k[4])$ where $k[i] = (k[i][0], k[i][1], k[i][2], k[i][3])$. We call the key k to be a *base key* if $k[4][0] = 0$. The number of all possible base keys is 2^{76}.

We now consider the following 20-word difference δ_i $(i \in \{0, \ldots, 2^4 - 1\})$ defined by $\delta_i[4][0] = i$, all other words being 0. Then 2^4 keys $\{k_i = k \oplus \delta_i\}$ are in a group with the base key k where $k_0 = k$. This yields a partition of the master key space into 2^{76} groups of 2^4 keys each.