Improved meet-in-the-middle attacks on reduced-round Joltik-BC

Manman Li | Shaozhen Chen

State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou, China

Abstract

Joltik-BC is an internal tweakable block cipher of the authenticated encryption algorithm Joltik, which was a second-round finalist in the CAESAR competition. The authors study the key-recovery attacks on Joltik-BC under meet-in-the-middle attack. Utilising the subweakkey difference cancellation, the freedom of the tweak and the differential enumeration, they attack on nine-round Joltik-BC-64-64 by constructing a precise six-round meet-in-the-middle distinguisher with $2^{53}$ plaintext–tweak combinations, $2^{52.91}$ Joltik-BC blocks and $2^{54.1}$ nine-round Joltik-BC-64-64 encryptions. Moreover, they attempt to attack on 11-round Joltik-BC-128-64 for the first time by constructing a seven-round meet-in-the-middle distinguisher with $2^{53}$ plaintext–tweak combinations, $2^{114}$ Joltik-BC blocks and $2^{123}$ 11-round Joltik-BC-128-64 encryptions.

1 | INTRODUCTION

As an important part of cryptography, block cipher can be used for data encryption, message authentication code and pseudorandom number generation. Recently, the design and analysis of tweakable block cipher has attracted increasing attention. Lisov et al. proposed tweakable block ciphers [1] in CRYPTO 2002. They have a public input tweak that can increase the diversity of block ciphers, which is different to traditional block ciphers. It is relatively more convenient and less costly to change the tweak than the secret key. Therefore, tweakable block ciphers are widely used in cryptographic schemes, for example, format-preserving encryption, disk encryption and authenticated encryption algorithms [2, 3].

In ASIACRYPT 2014, a common framework was put forward by Jean et al. called TWEAKEY [4] used to construct tweakable block ciphers. The authors presented three particular instances of the framework, including Deoxys-BC, Joltik-BC and Kiasu-BC, all of which were AES-based tweakable block ciphers [5]. In TWEAKEY framework, instead of distinguishing the key and tweak, the key and the tweak cascading are treated as a whole, called TWEAKEY, and the tweak schedule algorithm and the key schedule algorithm as the tweakkey schedule algorithm. Because the structure is very simple and the resistance is reasonably strong to related-key attacks, the TWEAKEY framework has received most attention from researchers. The analysis of specific algorithm instances helps to assess its security and will also assist designers in building more valuable tweakable block ciphers in the future.

Joltik-BC is an internal tweakable block cipher of Joltik as a second-round finalist in CAESAR competition, which is an authenticated encryption algorithm. The results of security analysis on Joltik-BC were initially carried out against meet-in-the-middle attacks and differential attacks [6]. A related-key impossible differential attack on 10-round Joltik-BC-64-64 was proposed by Zong et al. [7] in 2018. Then, a meet-in-the-middle attack on nine-round Joltik-BC-64-64 was presented by Liu et al. [8] combined with the subweakkey difference cancellation in 2019. In the same year, Li et al. [9] constructed a meet-in-the-middle attack on 10-round Joltik-BC-128-64 taking advantage of the tweak differential property. In Table 1, the valid attack results of Joltik-BC are summarised.

The authors analyse the security of Joltik-BC-64-64 and Joltik-BC-128-64 against the meet-in-the-middle attack. Firstly, they clearly distinguish the tweak and the key to propose a precise six-round meet-in-the-middle distinguisher using the subweakkey difference cancellation property, the tweak difference and the differential enumeration technique, based on the results...
Table 1 Summary of attacks on Joltik-BC

| Cipher       | Technique | Rounds | Time  | Memory | Data (CP) | Reference |
|--------------|-----------|--------|-------|--------|-----------|-----------|
| Joltik-BC-64-64 | MITM      | 8      | \(2^{34}\) | –      | –         | [6]       |
| Joltik-BC-64-64 | MITM      | 8      | \(2^{53.6}\) | \(2^{53}\) | \(2^{53.5}\) | [9]       |
| Joltik-BC-64-64 | ID        | 9      | \(2^{61.7}\) | \(2^{50}\) | \(2^{60}\) | [7]       |
| Joltik-BC-64-64 | MITM      | 9      | \(2^{56.6}\) | \(2^{5291}\) | \(2^{53}\) | [8]       |
| Joltik-BC-64-64 | MITM      | 9      | \(2^{54.1}\) | \(2^{5291}\) | \(2^{53}\) | Section 3.2 |
| Joltik-BC-64-64 | MITM      | 8      | \(2^{528}\) | –      | –         | [6]       |
| Joltik-BC-64-64 | MITM      | 10     | \(2^{145.5}\) | \(2^{123.5}\) | \(2^{16.1}\) | [9]       |
| Joltik-BC-64-4 | MITM      | 11     | \(2^{123}\) | \(2^{114}\) | \(2^{53}\) | Section 4.2 |

Abbreviations: CP, chosen plaintext; ID, impossible differential; MITM, meet-in-the-middle.

of Joltik-BC-64-64 given in [8]. Then, by adding one round to the top and two to the bottom, they present a meet-in-the-middle attack with \(2^{53}\) plaintext–tweak combinations, \(2^{52.91}\) Joltik-BC blocks and \(2^{54.1}\) nine-round Joltik-BC-64-64 encryptions. Similarly by constructing a seven-round meet-in-the-middle distinguisher in the offline, they present a meet-in-the-middle attack on 11-round Joltik-BC-128-64 for the first time with \(2^{53}\) plaintext–tweak combinations, \(2^{113}\) Joltik-BC blocks and \(2^{123}\) 11-round Joltik-BC-64-64 encryptions. Section 2 provides some preliminaries and briefly introduces some points including the specification of Joltik-BC, the property of the tweakkey schedule algorithm and the development of meet-in-the-middle attack. Section 3 gives the specific process and complexity on nine-round Joltic-BC-64-64, followed by the proposition of the new result of the meet-in-the-middle attack on Joltie-BC-128-64 in Section 4 and a summary in Section 5.

2 | PRELIMINARIES

2.1 | Notation

\(STK, \text{ tweak}, P, C\) the subtw eakey, tweak, plaintext, ciphertext;
\(\Delta\text{tweak}, \Delta P, \Delta C\) the differences of tweak, plaintext and ciphertext;
\(x_0, y_i, z_i, w_i\) the internal states before \(\text{SubNibbles}, \text{ShiftRows}, \text{MixNibbles}\) and \(\text{AddRoundTweakey}\) operations in the \(i\)-th round, respectively;
\(x[i]\) byte in position \(i\) of state \(x\);
\(x[i, \ldots, j]\) bytes in position \(i, \ldots, j\) of state \(x\).

2.2 | Description of Joltik-BC

Figure 1 shows the overall structure of Joltik-BC.

Adopting an AES-like structure, Joltik-BC includes the two versions of Joltik-BC-128 and Joltik-BC-192. The designers used a \(4 \times 4\) matrix to express the internal state of Joltik-BC. Table 2 shows the Joltik-BC parameters for each version.

Joltik-BC is an iterative substitution permutation network, with one round consisting of four operations:

**SubNibbles(SN)**: Apply a 4-bit Sbox \(S\) adopted by Piccolo [10] to each nibble of the internal state, as shown in Table 3.

**ShiftRows(SR)**: The 4-nibble \(i\)-th row is rotated left by \(i\) nibbles in the state matrix, where \(i = 0, 1, 2, 3\), obviously;

**MixNibbles(MN)**: Multiply each internal state by an MDS matrix as follows:

\[
M = \begin{bmatrix}
1 & 4 & 9 & 13 \\
4 & 1 & 13 & 9 \\
8 & 13 & 1 & 4 \\
13 & 9 & 4 & 1
\end{bmatrix} = M^{-1}
\]

**AddRoundTweakey(ART)**: The 64-bit internal state is xored a 64-bit round subtw eakey.

Before the 64-bit plaintext accesses the round function, an additional **AddRoundTweakey** is performed. To visually describe the round tweakkey, the two versions as Joltik-BC-64-64 (i.e. Joltik-BC-128 with 64-bit tweak and 64-bit key) and Joltik-BC-128-64 (i.e. Joltik-BC-192 with 128-bit tweak and 64-bit key) are referred to.

The authors define the \(i\)-th 64-bit round tweakkey as \(STK_i\) and represent \(KT\) for the concatenation of the key \(K\) and the tweak \(T\), which is expressed as \(KT = K \| T\). In the specific operation, the tweakkey \(KT\) is then divided into several 64-bit words. Therefore, in detail, the size of \(KT\) is 128 bits for the Joltik-BC-64-64 as the first 64 bits of \(KT\) is represented by \(W_1\) and the second word is represented by \(W_2\). Then, the size of \(KT\) is 192 bits for Joltik-BC-128-64 and the authors denoted the first, second and third 64-bit words of \(KT\) by \(W_1, W_2, W_3\), respectively. The subtw eakey is defined as \(STK_i = TK_1 \oplus TK_2 \oplus RC_i\) for Joltik-BC-64-64 and defined as \(STK_i = TK_1 \oplus TK_2 \oplus TK_3 \oplus RC_i\) for Joltik-BC-128-64. Here \(RC_i\) are the key schedule round constants. The 64-bit words \(TK_i\) are outputs produced by a special tweakkey schedule algorithm: Let \(TK_0^1 = W_1\) and \(TK_0^2 = W_2\) for Joltik-BC-64-64; let \(TK_0^1 = W_1, TK_0^2 = W_2\) and \(TK_0^3 = W_3\) for Joltik-BC-128-64. Thus, the tweakkey schedule algorithm is represented as

\[
TK_{i+1} = g_1(b(TK_i)),
\]

\[
TK_{i+1} = g_2(b(TK_i)),
\]

\[
TK_{i+1} = g_3(b(TK_i)),
\]
TABLE 2 The Joltik-BC parameters

| Version    | Block length | Key length | Tweak length | Round |
|------------|--------------|------------|--------------|-------|
| Joltik-BC128 | 64           | 64         | 64           | 24    |
| Joltik-BC192 | 64           | 128        | 64           | 32    |

TABLE 3 The Shox of Joltik-BC

| $x$ | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| $S(x)$ | 14 | 4  | 11 | 2  | 3  | 8  | 0  | 9  | 1  | 10 | 7  | 15 | 6  | 12 | 5  | 13 |

where the function $g_a$ is a finite field multiplication of each nibble by the element $a$ and the nibble permutation $h$ is defined as

$$(0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \ 15 \ 1 \ 6 \ 11 \ 12 \ 5 \ 10 \ 15 \ 0 \ 9 \ 14 \ 3 \ 4 \ 13 \ 2 \ 7 \ 8)$$

For Joltik-BC-64-64, $TK_1$ is denoted as $k_0^i$ and $TK_2^i$ as $t_i$ or $k_i^j$ for convenience, then the tweak key schedule algorithm is represented as

$$k_0^{i+1} = b(k_0^i),$$
$$t_{i+1} = g_2(b(t_i)),$$

where $k_0^i = K$ and $t_0 = T$. Therefore the result $STK_i = k_i^j \oplus t_i \oplus RC_i$ is achieved.

For Joltik-BC-128-64, $TK_1^i$ is denoted as $k_0^i$, $TK_2^i$ is denoted as $k_1^i$ and $TK_3^i$ as $t_i$ or $k_i^j$, then the tweak key schedule algorithm is represented as

$$k_0^{i+1} = b(k_0^i),$$
$$k_1^{i+1} = g_2(b(k_1^i)),$$
$$t_{i+1} = g_4(b(t_i)),$$

where $t_0 = T$, $k_0^0$ is the $K$ most significant 64-bit words and $k_0^i$ is the $K$ least significant 64-bit words. Then similarly, $STK_i = k_1^i \oplus k_0^i \oplus t_i \oplus RC_i$ can be the result.

2.3 | Description of the meet-in-the-middle attack

As an effective chosen plaintext attack proposed by Diffie and Helman when they analysed DES in 1977 [11], meet-in-the-middle attack utilises the cipher-slice idea and the space–time tradeoff technique in security analysis of hash function and block cipher. Recently, many techniques and methods have been proposed to improve the results of meet-in-the-middle analysis.

Demirci and Selcuk proposed a new meet-in-the-middle attack model [12] in 2008, called Demirci-Selcuk meet-in-the-middle attack, which combined the idea of collision analysis [13] with meet-in-the-middle attack in the process of analysing AES. Its general idea is that a block cipher $E$ is divided into three consecutive parts, $E = E_2E_0E_1$, as shown in Figure 2.

The attack process consists of the offline phase and the online phase. In the former, that is the precomputation phase $E_0$, a precomputation table needs to be established to store specific input and output by constructing a meet-in-the-middle distinguisher; in the latter, that is the key recovery phase, the adversary expects to encrypt some chosen plaintexts and decrypt the corresponding ciphertexts by guessing some related subkeys $k_1$ and $k_2$ used in $E_1$ and $E_2$. Then, the adversary looks forward to check whether the internal values match with the precomputation table. In general, the guessed subkeys may be true if the values computed in the latter satisfy the precomputation table; if not, it will be filtered out.

In 2010, Dunkelman et al. [14] presented a differential enumeration technique to solve the large storage complexity of Demirci-Selcuk meet-in-the-middle attack: when the input difference and the output difference satisfy the specific differential structure, the range of internal state can be limited. Meanwhile, the key-bridging technique was proposed to find the algebraic relationship between subkeys and reduce the time complexity of online analysis on Demirci-Selcuk attack. In 2014, Wang et al. [15] put forward the key-dependent sieve technique to further narrow the value range of internal state to reduce the storage complexity of Demirci-Selcuk analysis by combining with the differential enumeration technique. Subsequently, the Demirci-Selcuk meet-in-the-middle analysis technique was used to analyse various types of block ciphers [16–19]. In 2019, Ahmadi et al. [20] proposed the generalised
meet-in-the-middle attack and achieved an automated attack along with some certain ideas which were applied on different block ciphers.

3 | MEET-IN-THE-MIDDLE ATTACK ON NINE-ROUND JOLTIK-BC-64-64

In theory, for a tweakable block cipher, the distinction between the tweak input and the key input is clear: the former is public and can be fully controlled by the attacker, while the latter is secret [4]. When a block cipher is analysed it is better to distinguish the key and the tweak to achieve accurate attack results.

In [8], the authors proposed a six-round meet-in-the-middle distinguisher to attack nine-round Joltik-BC-64-64, but they did not distinguish the concepts of key and tweak. This idea resulted in the complexity beyond exhaustive range when they attacked 10-round Joltik-BC-64-64. Therefore the authors think that the analysis results do not affect the security of the Joltik-BC-64-64 encryption authentication algorithm.

Referring to the ideas in [9], the authors precisely distinguish the concepts of the key and the tweak for Joltik-BC-64-64 and accurately construct a new six-round meet-in-the-middle distinguisher in this section. It is useful for to understand the seven-round meet-in-the-middle distinguisher in Section 4. The definition and property as follow are used.

**Definition 1** [21] A $b$-δ-set is a set including the $2^b$ state values, which are entirely different in $b$ state bits (active bits) and are entirely equal in the remaining state bits (inactive bits).

**Property 1** [22] For a given bijective S-box $S$, let $\Delta_0$ and $\Delta_1$ be two non-zero differences, then the number of solutions satisfying the equation $S(x) \oplus S(x \oplus \Delta_0) = \Delta_1$ is 1 on average.

3.1 | A six-round distinguisher on Joltik-BC-64-64

In this part, it is considered that the nibble $\omega_0[1]$ is active and that the corresponding ordered sequence $\Delta x^t[5,7]$ can be calculated after six-round Joltik-BC-64-64 encryption. In order to optimise the analysis results, some special tweaks in the phase are selected, such as $\Delta k^t_0[1] = \Delta w_0[1]$, that is $\Delta x^t_1[1] = 0$.

Firstly, a property of the Joltik-BC-64-64 called Subtweak Difference Cancellation used in the distinguisher is presented.

**Property 2** [6,7] The designers affirmed that it occurs once for each 15-round Joltik-BC-64-64 with respect to a subtweak difference cancellation. Assume that both $TK^1$ and $TK^2$ have one active nibble and name the differences of these two active nibbles as $a_1$ and $a_2$, respectively. Then, in the first round, the subtweak difference is $a_1 \oplus a_2$ at the active cell and $g^t_2(a_2) \oplus a_1$ in the $i$-th round if ignoring the position permutation $b$. Due to both $a_1$ and $a_2$ being non-zero, the equation $g^t_2(a_2) \oplus a_1 = 0$ cannot occur more than one time for every 15 rounds.

According to Property 2, in the second round of the distinguisher, some related keys and tweaks can be selected to make the subtweak difference cancellation occur. The specific content of the distinguisher is as follows.

**Theorem 1** [8] Let $\{w_0^0, \omega_0^1, ..., \omega_0^{15}\}$ and $\{t_0^0, t_0^1, ..., t_0^{15}\}$ be two $b$-δ-sets where $b = 4$ and satisfy $\omega_0^t[1] = t_0^t[1] \oplus t_0^{t[1]}[1] = a(a = 0, 1, 2, ..., 15)$. Meanwhile, the truncated differential characteristic needs to be generated by a pair of message-tweak combinations $(\omega_0^t, t_0^t)$, $(\omega_0^t, t_0^t)$, as outlined in Figure 3. Among the characteristics, the keys and the tweaks must have an exact selection such that $\Delta STK_2 = 0$ according to Property 2. Considering the encryption of $\omega_t^t$ under the tweak $t_0^t$ through six-round Joltik-BC-64-64 encryption, the ordered sequence $(x_i^t[5,7] \oplus x_t^t[5,7], x_i^t[5,7] \oplus x_t^t[5,7], ..., x_i^{t[5,7]} \oplus x_t^{5[5,7]} \oplus x_t^{5[5,7]})$ only takes $2^{12}$ values (out of the $2^{120}$ theoretical values).

**Proof.** The process to propagate the ordered sequence $(x_i^t[5,7] \oplus x_t^t[5,7], x_i^t[5,7] \oplus x_t^t[5,7], ..., x_i^{t[5,7]} \oplus x_t^{5[5,7]} \oplus x_t^{5[5,7]})$ by using the differential enumeration technique from two $b$-δ-sets is shown.

First, it can be proved that the ordered sequence $(x_i^t[5,7] \oplus x_t^t[5,7], x_i^t[5,7] \oplus x_t^t[5,7], ..., x_i^{t[5,7]} \oplus x_t^{5[5,7]} \oplus x_t^{5[5,7]})$ is determined by 27-nibble parameters, namely:

$w_0^t[1], x_i^t[15], x_i^t[0, 1, 2, 3, 8], x_t^t[3, 4, 9, 14]$.

Denote the difference between $x^t$ and $x^t$ as $\Delta x^t$ in the following proof. It can be deduced that $t_0^t[1] \oplus t_0^{t[1]}[1]$ and $\Delta t_0^t, \Delta t_0^t, \Delta t_0^t, \Delta t_0^t, \Delta t_0^t, \Delta t_0^t, \Delta t_0^t, \Delta t_0^t, \Delta t_0^t$ with $w_0^t[1]$. Since $\Delta w_0^t[1] = \Delta t_0^t[1] = a \oplus w_0^t[1]$, then $\Delta x^t = 0$. Some special tweaks and keys can be selected such that $\Delta STK_2 = 0$ and $\Delta x^t = 0$ deduced by Property 2. Hence, $\Delta x^t[15] = \Delta x^t[15]$. Then we can infer $\Delta x^t[15]$ linearly from $\Delta x^t[15]$ and $x_i^t[15]$. $\Delta x^t[0, 1, 2, 3, 8]$ is deduced by ShiftRow, MixNibbles and AddRoundTweak operations. Similarly, $\Delta x^t$ is deduced with $x_i^t[0, 1, 2, 3, 8], x_i^t[3, 4, 9, 14]$ with $x_i^t$ and $\Delta x^t[5,7]$, with $x_i^t[3, 4, 9, 14]$. Therefore the sequence $(x_i^t[5,7] \oplus x_t^t[5,7], x_i^t[0, 1, 2, 3, 8], x_t^t[3, 4, 9, 14])$. Therefore the sequence $(x_i^t[5,7] \oplus x_t^t[5,7],...$
In the section, a meet-in-the-middle attack on nine-round Joltik-BC-64-64 is presented with $2^{53}$ plaintext–tweak combinations, $2^{52.01}$ Joltik-BC blocks and $2^{54.1}$ nine-round Joltik-BC-64-64 encryptions based on a six-round distinguisher by adding one round to the top and two to the bottom. As MixNibbles and AddRoundTweakey transformations are both linear, the order of the MixNibbles and the AddRoundTweakey operations are exchanged in the eighth and the ninth rounds, respectively, to obtain an equivalent relation, which can reduce nibbles guessed, as shown in Figure 4.

The attack process consists of two parts: the pre-computation phase and the online phase.

Precomputation phase: all of the $2^{52}$ possible values of the ordered sequence $(x_{i}^{k}[5,7] \oplus x_{i}^{k}[5,7], x_{i}^{k}[5,7] \oplus x_{i}^{k}[5,7] \ldots, x_{i}^{k}[5,7] \oplus x_{i}^{k}[5,7])$ need to be computed as described in Theorem 1. Then a hash table $H$ is built to store them.

The online phase includes the following attack procedures.

Step 1: Firstly, a set $S$ of $2^{16}$ plaintexts is defined satisfying that the state of $P[0,5,10,15]$ takes all the possible $2^{16}$ values and at the same time the remaining 12 nibbles are going to be some constants. Then, similarly a set $T$ of $2^{4}$ tweaks is defined satisfying that the values of $T[0]$ take all the possible and the remaining 15 nibbles are going to be some constants. Considering the sets and structures of plaintexts, $2^{16} \times 2^{4} = 2^{20}$ plaintext–tweak combinations can be obtained and for one structure generate $2^{20} \times (2^{20} - 1)/2 \approx 2^{39}$ pairs of plaintext–tweak combinations.

Step 2: The probability of the truncated differential characteristic is $2^{-(3+4+8+6)} \times 4 = 2^{-12}$, so $2^{72-39} = 2^{33}$ structures need to be chosen, which can yield about $2^{33} \approx 2^{35}$ pairs.

Step 3: To the plaintext–tweak combinations for each structure, the corresponding ciphertexts are queried and the value $s_{9}$ is obtained by partially decrypting the ciphertexts. It is expected to have $2^{2^{2}-(8 \times 4)} = 2^{40}$ pairs satisfying that the difference is 0 in nibbles $s_{9} \in \{1, 3, 4, 6, 9, 11, 12, 14\}$. For each of the $2^{40}$ remaining pairs, the following steps are performed:

i. Compute $\Delta x_{k}^{0}[1], \Delta x_{k}^{0}[7]$ and $\Delta x_{k}^{0}[0]$ with $\Delta x_{k}^{0}[0]$. Since $\Delta x_{k}^{0}[1] = \Delta x_{k}^{0}[1]$, it can be deduced that $\Delta x_{k}^{0}[5, 10, 15]$. Meanwhile $\Delta x_{k}^{0}[5, 10, 15]$ can be obtained with the plaintexts. Then, $x_{k}[5, 10, 15]$ can be obtained by Property 1 and $k_{0}[5, 10, 15]$ deduced.

ii. Guess $\Delta y_{k}[5, 7]$, compute $\Delta x_{k}[0, 1, 2, 3, 8, 9, 10, 11]$ and deduce $\Delta y_{k}[0, 1, 2, 3, 8, 9, 10, 11]$ with the ciphertexts. According to Property 1, we know the value of $x_{k}[0, 1, 2, 3, 8, 9, 10, 11]$ and $y_{k}[0, 1, 2, 3, 8, 9, 10, 11]$. We get $z_{k}[0, 2, 5, 7, 8, 10, 13, 15]$ with $y_{k}[0, 1, 2, 3, 8, 9, 10, 11]$. Deduce $s_{9}[0, 2, 5, 7, 8, 10, 13, 15]$ with the ciphertexts. Then $u_{k}[0, 2, 5, 7, 8, 10, 13, 15]$ is obtained.

iii. Take one of these pairs and partially encrypt it to construct two $\delta$-sets as defined by Theorem 1. Through partial decryption, the 16 plaintext–tweak combinations $(P^{0}, T^{0}), (P^{1}, T^{1}), \ldots, (P^{15}, T^{15})$ can be deduced.

iv. Guess $u_{k}[1, 11]$. For the 16 plaintext–tweak combinations, query corresponding ciphertexts and decrypt them by the value of $u_{k}[1, 11]$ and...
4 | MEET-IN-THE-MIDDLE ATTACK ON 11-ROUND JOLTIK-BC-128-64

The authors construct a seven-round meet-in-the-middle distinguisher in the offline and present a meet-in-the-middle attack on 11-round Joltik-BC-128-64 for the first time by adding one round to its top and three rounds to its bottom in this section.

4.1 | A seven-round distinguisher on Joltik-BC-128-64

In [6], it can be seen that though it is slightly different in key schedule between Joltik-BC-128-64 with Joltik-BC-64-64, the version still satisfies the subtwakey difference cancellation property. According to the property, the specific content of the distinguisher constructed in the same way is as follows.

Theorem 2. Let \( w_0, w_1, \ldots, w_{35} \) and \( t_0, t_1, \ldots, t_{35} \) be two \( b\)-\( \delta \)-sets where \( b = 8 \) and satisfy \( w_0[1] = t_0[1] \oplus t_0'[1] = \alpha(a = 0, 1, 2, \ldots, 255) \). Meanwhile, the truncated differential characteristic needs to be generated by a pair of message-tweak combinations \((w_0', t_0'), (w_0', t_0')\), as outlined in Figure 5. Among the characteristics, the keys and tweaks must have an exact selection such that \( \Delta SK_2 = 0 \) according to Property 2. Consider the encryption of \( w_0(0 \leq a \leq 63) \) under the tweak \( t_0' \) through seven-round Joltik-128-64 encryptions, then the ordered sequence \((x_0'[11] \oplus x_0'[11], x_0'[11] \oplus x_0'[11], \ldots, x_0'[11] \oplus x_0'[11])\) only takes \( 2^{12} \) values (out of the \( 2^{352} \) theoretical values).

Proof. The process to propagate the ordered sequence \((x_0'[11] \oplus x_0'[11], x_0'[11] \oplus x_0'[11], \ldots, x_0'[11] \oplus x_0'[11])\) using the differential characteristic technique from two \( b\)-\( \delta \)-sets is shown.

First, it can be proved that the ordered sequence is determined by 43-nibble parameters, namely:
The attack process consists of two parts: the pre-computation phase and the online phase.

Precomputation phase: all of the $2^{112}$ possible values of the ordered sequence $(x^i_8[11] \oplus x^i_8[11], x^i_8[11] \oplus x^i_8[11], \ldots, x^i_8[11] \oplus x^i_8[11])$ need to be computed, as described in Theorem 2. Then a hash table $H$ is built to store them.

The online phase includes the following attack procedures.

Step 1: It is expected to choose $2^{33}$ structures which are the same as those in Section 3.2 and get $2^{33+20} = 2^{53}$ plaintext–tweak combinations. There are $2^{72}$ pairs of plaintext–tweak combinations satisfying the truncated differential characteristic. Perform the following steps for each of these $2^{72}$ pairs:

i. To the plaintext–tweak combinations for each structure, the corresponding ciphertexts are queried. Compute $\Delta x^i_0[1], \Delta x^i_0[1]$ and $\Delta x^i_0[6]$ with $\Delta x^i_0[0]$. Since $\Delta x^i_0[1] = \Delta x^i_0[1], \Delta y^i_0[0, 5, 10, 15]$ can be deduced and $\Delta x^i_0[0, 5, 10, 15]$ obtained with the plaintexts, the $x^i_0[0, 5, 10, 15]$ can be obtained by Property 1 and $(k^0 \oplus k^0)[0, 5, 10, 15]$ deduced.

ii. Guess the value of $\Delta y^i_0[0, 1, 2, 3, 12, 13, 14, 15]$, compute $\Delta x^i_0$ and $\Delta x^i_0$ with the ciphertexts. According to Property 1, the values of $x^i_0$ and $y^i_0$ are known. Then deduce $u^i_0 \oplus u^i_1$. Guess $\Delta y^i_0[0, 11]$ and compute $\Delta x^i_0[0, 1, 2, 3, 12, 13, 14, 15]$. According to Property 1, $y^i_0[0, 1, 2, 3, 12, 13, 14, 15]$ is obtained. Then $(u^i_0 \oplus u^i_1)[0, 3, 6, 7, 9, 10, 12, 13]$ is obtained.

iii. Take one of these pairs and partially encrypt it to construct two $\delta$-sets as defined by Theorem 2. Through partial decryption, the 64 plaintext–tweak combinations $(P^0, T^0), (P^1, T^1), \ldots, (P^{25}, T^{25})$ can be deduced.

iv. Guess $(u^i_0 \oplus u^i_1)[0, 15]$. For the 64 plaintext–tweak combinations, query corresponding ciphertexts and decrypt them by the value of $u^i_1 \oplus u^i_1$ and $(u^i_0 \oplus u^i_0)[0, 3, 6, 7, 9, 10, 12, 13]$. The ordered sequence $(x^i_1[11] \oplus x^i_8[11], x^i_8[11] \oplus x^i_8[11], \ldots, x^i_8[11] \oplus x^i_8[11])$ can then be calculated.

v. It needs to be verified that the sequence matches the hash table $H$. If not, the subkey $u^i_0 \oplus u^i_1$, $(u^i_0 \oplus u^i_0)[0, 3, 6, 7, 9, 12, 13]$, $(u^i_0 \oplus u^i_0)[0, 15]$ and $(k^0 \oplus k^0)[0, 5, 10, 15]$ is filtered. For an incorrect subkey, the probability to pass the test is about $2^{112-252} \approx 2^{-140}$. Hence, there is about $1 + 2^{24+12+4-140} \approx 1$ remaining subkey candidate.

Step 2: By exhaustive key search of the remaining 16 nibbles of $u^i_1$, the master key can be retrieved.
Complexity analysis. The data complexity is $2^{33+20} = 2^{53}$ plaintext–tweak combinations. The memory complexity is $2^{112} \times 252 + 64 \approx 2^{114}$ 64-bit blocks determined by building the precomputation table $H$ and the time complexity is $2^{112} \times 2^5 \times 41 \div (11 \times 16) \approx 2^{116}$ 11-round Joltik-BC-128-64 encryptions in the precomputation phase. The time complexity is $2^{32} \times 2^{12} \times 2^5 \times 30 \div (11 \times 16) \approx 2^{123}$ 11-round Joltik-BC-128-64 encryptions in the online phase. Finally, the whole complexity of the attack is $2^{53}$ plaintext–tweak combinations, $2^{114}$ Joltik-BC blocks and $2^{116} + 2^{123} \approx 2^{123}$ 11-round Joltik-BC-128-64 encryptions.

5 | CONCLUSION

A favourable meet-in-the-middle attack against Joltik-BC has been presented herein. By utilising the subtwakey difference cancellation, the freedom of the tweak and the differential enumeration, the authors clearly distinguish the tweak and the key to precisely construct a six-round meet-in-the-middle distinguisher. Based on the six-round distinguisher, they present an attack on nine-round Joltik-BC-64-64 with $2^{33}$ plaintext–tweak combinations, $2^{52.91}$ Joltik-BC blocks and $2^{54.1}$ 9-round Joltik-BC-64-64 encryptions. Moreover, they give a seven-round meet-in-the-middle distinguisher for the first time to achieve the result in 11-round Joltik-BC-128-64 with $2^{53}$ plaintext–tweak combinations, $2^{112}$ Joltik-BC blocks and $2^{123}$ 11-round Joltik-BC-128-64 encryptions.

ACKNOWLEDGEMENTS

Foundation items: National Cipher Development Foundation (MMJ20180203); State Key Laboratory of Mathematical Engineering and Advanced Computation Open Foundation (2019A08); Henan Key Laboratory of Network Cryptography Technology (LNCT-2019-S03).

REFERENCES

1. Lisakov, M., Rivest, R., Wagner, D.: Tweakable Block Ciphers. CRYPTO 2002, CA, USA, LNCS 2442, 31–46. Springer (2002)
2. Krovetz, T., Rogaway, P.: The software performance of authenticated-encryption modes. In: Proceedings of Fast Software Encryption - FSE 2014, 306–327. Springer, Berlin (February 13C16)
3. Peyrin, T., Seurin, Y.: Counter-in-tweak authenticated encryption modes for tweakable block ciphers. In: Proceedings of Advances in Cryptology - CRYPTO, 33C63 (2016)
4. Jean, J., Nikolic, I., Peyrin, T.: Twkeys and keys for block ciphers: the TWEAKEY framework. In: ASIACRYPT 2014, PART II, LNCS 8874, 274–288. Springer (2014)
5. Minematsu, K.: Building blockcipher from small-block tweakable blockcipher. Des. Codes Cryptogr. 74, 645–663 (2015)
6. Jean, J., Nikolic, I., Peyrin, T.: Joltik v1.3’. Submission to the CAESAR competition. (2015). http://www1.spms.ntu.edu.sg/syllab/Joltik
7. Zong, R., Dong, X., Wang, X.Y.: MILP-aided related-tweak/key impossible differential attack and its applications to QARMA, Joltik-BC. IEEE Access. 7, 1–1 (2019)
8. Liu, Y., Shi, Y., Gu, D.W.: Improved meet-in-the-middle attacks on reduced-round Kiasu-BC and joltik-BC. Comput. J. 62(Issue 12), 1761–1776 (2019). https://doi.org/10.1093/comput/hzx059
9. Li, R., Jin, C., Pan, H.: Key recovery attacks on reduced-round joltik-BC in the single-key setting. Inf. Process. Lett.151 (2019). https://doi.org/10.1016/j.ipl.2019.105834
10. Shibutani, K., Isobe, T., Hiwatari, H.: Piccolo: an ultra-lightweight blockcipher. In: International Conference on Cryptographic Hardware & Embedded Systems. Springer, Berlin (2011)
11. Gilbert, H., Minier, M.: A collision attack on 7 rounds of Rijndael. In: The Third AES Candidate Conference (2000)
12. Diffie, W., Hellman, M.: Exhaustive cryptanalysis of the NBS data encryption standard. Computer. 10, 74–84 (1977)
13. Demirci, H., Selçuk, A.: A meet-in-the-middle attack on 8-round AES'. In: Fast Software encryption-FSE 08, LNCS 5086, 116–126. Springer Verlag (2008)
14. Dunkelman, O., Keller, N., Shamir, A.: Improved single-key attacks on 8-round AES-192 and AES-256. In: ASIACRYPT10, LNCS 6477, 158–176. Springer-Verlag (2010)
15. Li, L., Jia, K., Wang, X.Y.: Improved Single-key attacks on 9-round AES-192/256. Fast Software encryption-FSE 14, LNCS 8540, 127–146. Springer-Verlag (2015)
16. Dong, X., Li, L., Jia, K.: Improved attacks on reduced-round Camellia-128/192/256. In: CT-RSA 15, LNCS 9048, pp. 59–83. Springer-Verlag (2015)
17. Lin, L., Wu, W.L., Zhang, Y.: Improved meet-in-the-middle distinguisher on Feistel schemes. In: Selected areas in cryptography-SAC 15, LNCS 9566, 122–142. Springer-Verlag (2016)
18. Biryukov, A., Derbez, P., Perrin, L.: Differential analysis and meet-in-the-middle attack against round-reduced TWINE. In: Fast Software encryption-FSE 15, LNCS 9054, 3–27. Springer-Verlag (2016)
19. Derbez, P., Perrin, L.: Meet-in-the-Middle attacks and structural analysis of round-reduced PRINCE'. In: Fast Software encryption-FSE 15, LNCS 9054, pp. 190–216. Springer-Verlag (2015)
20. Ahmadi, S., Aref, M.: Generalized meet in the middle cryptanalysis of block ciphers with an automated search algorithm. IEEE Access, 2284–2301 (2019)
21. Biham, E., Shamir, A.: Differential cryptanalysis of DES–like cryptosystems. CRYPTO 90, LNCS 537, 2–21. Springer (1990)
22. Kanda, M.: Practical security evaluation against differential and linear cryptanalyses for Feistel ciphers with SPN round function. In: Proceedings of Selected Areas in cryptography, SAC 00, LNCS 7707, pp. 324–338. Springer-Verlag (2012)

How to cite this article: Li M, Chen S. Improved meet-in-the-middle attacks on reduced-round Joltik-BC. IET Inf. Secur. 2021;15:247–255. https://doi.org/10.1049/ise2.12019