IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 2017, No. 1, pp. 259–280. DOI:10.13154/tosc.v2017.i1.259-280

Cube-like Attack on Round-Reduced Initialization of Ketje Sr

Xiaoyang Dong[1,2], Zheng Li[1], Xiaoyun Wang[1,2*] and Ling Qin[3]

[1] Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, Shandong, China
{dongxiaoyang,lizhengcn}@mail.sdu.edu.cn
[2] Institute for Advanced Study, Tsinghua University, Beijing, China
xiaoyunwang@tsinghua.edu.cn
[3] National Digital Switching System Engineering & Technological Research Center, P.O. Box 407, 62 Kexue Road, Zhengzhou, 450001, China

Abstract. This paper studies the Keccak-based authenticated encryption (AE) scheme Ketje Sr against cube-like attacks. Ketje is one of the remaining 16 candidates of third round CAESAR competition, whose primary recommendation is Ketje Sr. Although the cube-like method has been successfully applied to Ketje's sister ciphers, including Keccak-MAC and Keyak – another Keccak-based AE scheme, similar attacks are missing for Ketje. For Ketje Sr, the state (400-bit) is much smaller than Keccak-MAC and Keyak (1600-bit), thus the 128-bit key and cubes with the same dimension would occupy more lanes in Ketje Sr. Hence, the number of key bits independent of the cube sum is very small, which makes the divide-and-conquer method (it has been applied to 7-round attack on Keccak-MAC by Dinur *et al.*) can not be translated to Ketje Sr trivially. This property seems to be the barrier for the translation of the previous cube-like attacks to Ketje Sr.
In this paper, we evaluate Ketje Sr against the divide-and-conquer method. Firstly, by applying the linear structure technique, we find some 32/64-dimension cubes of Ketje Sr that do not multiply with each other as well as some bits of the key in the first round. In addition, we introduce the new dynamic variable instead of the auxiliary variable (it was used in Dinur *et al.*'s divide-and-conquer attack to reduce the diffusion of the key) to reduce the diffusion of the key as well as the cube variables. Finally, we successfully launch a 6/7-round[1] key recovery attack on Ketje Sr v1 and v2 (v2 is presented for the 3rd round CAESAR competition.). In 7-round attack, the complexity of online phase for Ketje Sr v1 is 2^{113}, while for Ketje Sr v2, it is 2^{97} (the preprocessing complexity is the same). We claim 7-round reduced Ketje Sr v2 is weaker than v1 against our attacks. In addition, some results on other Ketje instances and Ketje Sr with smaller nonce are given. Those are the first results on Ketje and bridge the gaps of cryptanalysis between its sister ciphers – Keyak and the Keccak keyed modes.

Keywords: Ketje · CAESAR · Cube-like · Linear Structure · Authenticated Encryption

Introduction

Authenticated encryption (AE) algorithms provide message confidentiality and integrity protection with a single cryptographic primitive. The most widely used AE algorithm is

*Corresponding author

[1] The attacks cover 6/7 rounds, including $n_{start} = 5/6$ initialization rounds and $n_{step} = 1$ step round, which are specified in Subsection 1.4.

Received: 2016-11-23, Accepted: 2017-01-23, Published: 2017-03-08

AES-GCM [NISb]. However, GCM is usually seen as a not robust enough standard [NISa] and therefore a new competition (CAESAR) for authenticated encryption with security, applicability and robustness was launched in 2014 [com14]. 57 algorithms were submitted as the first-round candidates. After 2 rounds assessment from world wide cryptographers and engineers, only 16 survivors left in the third-round of CAESAR.

Ketje [BDP+16a] is one of the 16 candidates, which is submitted by the Keccak team. It is based on the MonkeyDuplex construction and the inner function is Keccak-p permutations. In Ketje v1, Ketje Sr (400-bit state size) is the primary recommendation and Ketje Jr (200-bit state size) is the secondary recommendation. They are both lightweight and oriented towards the resource constrained environments. In Ketje v2, the authors add Ketje Minor (800-bit state size) and Ketje Major (1600-bit state size) into the family to meet the need of high-performance applications. So there are four instances in Ketje family and Ketje Sr is the primary one.

Cube attack [DS09] is a chosen IV key-recovery attack, which was introduced by Dinur and Shamir. Since then, cube attack was applied to many different cryptographic primitives [ADMS09, DS11, FV13]. At Eurocrypt 2015, Dinur *et al.* [DMP+15] used a cube-like attack to recover the key on round-reduced Keccak keyed modes (Keccak-MAC and Keyak). It is a divide-and-conquer method, where the 'borderline' cube variables and some auxiliary variables are carefully selected so that the cube sums depend only on a small number of key bits. They finally achieved the first 7-round key-recovery attack on Keccak-MAC. Huang *et al.* [HWX+] proposed a new *conditional cube attack* on the Keccak keyed modes. By restraining some bit conditions of the key, they obtain a new set of cube variables which do not multiply with each other in the first round, and also meet the condition that one cube variable does not multiply with other cube variables in the second round, then the output degree over cube variables is reduced. The two methods were applied to Ascon [DEMS16] by Dobraunig *et al.* [DEMS15] and Li *et al.* [LDW17].

However, those attacks can not be trivially translated to Ketje Sr. As a matter of fact, there are no attacks on round-reduced Ketje Sr till now. For Ketje Sr, the state is only 400-bit, with 25 16-bit lanes. The 128-bit key occupies 9 lanes alone with some padding bits. For Huang *et al.*'s method, it is hard to find enough cube variables which have similar properties as Keccak-MAC[2]. When applying Dinur *et al.*'s method to Ketje Sr, we need more lanes to be cube variables to achieve the same round attack (6/7-round). This reduces the number of key bits, which are independent of the cube sum after 6/7-round, significantly. Hence, more new techniques are needed to translate Dinur *et al.*'s attacks on Keccak-MAC to Ketje Sr.

Our Contributions

In this paper, we comprehensively study the divide-and-conquer method introduced by Dinur *et al.* [DMP+15] as well as the features of Ketje Sr (the primary recommendation). First, by applying the linear structure technique [GLS16], we find some 32/64-dimension cubes of Ketje Sr that do not multiply with some bits of the key in the first round. Second, we introduce the new dynamic variable instead of the auxiliary variable (it was used in Dinur *et al.*'s divide-and-conquer attack to reduce the diffusion of the key) to reduce the diffusion of the key as well as the cube variables. Therefore, the number of key bits, which are independent of the cube sum after 6/7-round Ketje Sr, increases significantly and only a small fraction of the key bits will appear in the cube sum. Finally, we successfully apply the divide-and-conquer method to the 6/7-round key-recovery attacks on both Ketje Sr v1 and v2. These attacks are the first key-recovery attacks on round-reduced Ketje Sr.

In the 7-round attacks, the time complexity of online phase is 2^{113} for Ketje Sr v1, and 2^{97} for Ketje Sr v2, with the same preprocessing complexity. Hence, we could

[2]More details about this attack are given in Subsection 2.3.

Table 1: Summary of Key-recovery Attacks on Ketje, Keyak and Keccak-MAC

Mode	Attacked Rounds	Time Online	Time offline	Momery	Source
Keccak-MAC	7/24	2^{96}	2^{96}	2^{32}	[DMP+15]
	7/24	2^{72}	-	-	[HWX+]
Lake Keyak	7/12	2^{75}	2^{76}	2^{43}	[DMP+15]
	8/12	2^{74}	-	-	[HWX+]
Ketje Sr v1	6/13	$2^{65.6}$	2^{73}	2^{40}	Section 5
	7/13	2^{113}	2^{115}	2^{50}	
Ketje Sr v2	6/13	$2^{65.6}$	2^{65}	2^{32}	Section 6
	7/13	2^{97}	2^{113}	2^{48}	
Ketje Jr v1	5/13	2^{42}	2^{56}	2^{38}	Section 7
Ketje Jr v2	5/13	2^{48}	2^{50}	2^{32}	Section 7
Ketje Minor/Major v1/2	6/13	2^{64}	2^{64}	2^{32}	Section 7
	7/13	2^{96}	2^{96}	2^{32}	
Ketje Sr v1 128-bit nonce	6/13	2^{80}	2^{72}	2^{40}	Section 7
Ketje Sr v2 128-bit nonce	6/13	2^{64}	2^{96}	2^{64}	Section 7

claim that the 7-round reduced Ketje Sr v2 is weaker than Ketje Sr v1 against our attack. In addition, we also give some results on other Ketje instances and round-reduced Ketje Sr with length-reduced nonce (128-bit). All the attacks do not threaten the full 13 rounds of Ketje. Table 1 compares our attacks with the previous attacks on Keyak and Kecck-MAC. Our contributions are summarised as three folds:

1. By using the linear structure technique [GLS16], we find many 1-round linear structures of Ketje Sr with 32/64 degrees of freedom. We use them as 32/64-dimension cubes. By carefully studying the features of Ketje Sr (properties of the permutation and padding rules), we select 32/64-dimension cubes, which do not multiply with as many key bits as possible after the first round. Hence, the cube sums after 6/7-round are independent from those key bits.

2. In Dinur *et al.*'s work, auxiliary variables are introduced to control the diffusion of some key bits. In our work, we introduce the dynamic variables, which are not only used to reduce the diffusion of the key, but also the cube variables. The advantage of the dynamic variables over auxiliary variables is that it makes the variables more sparse and at meanwhile maintains the the dimension of cube.

3. We launch the 6/7-round key recovery attacks on both versions of Ketje Sr, and some results on other Ketje instances and round-reduced Ketje Sr with length-reduced nonce (128-bit) are given in addition. They are the first results on Ketje and bridge the gaps of cryptanalysis between its sister ciphers – Keyak [BDP+16b] and the Keccak keyed modes.

Organization of the Paper

Section 1 gives some notations, the Keccak-p permutations and Ketje. In Section 2, some related works are discussed. The properties of Ketje Sr are presented in Section 3. In Section 4, we introduce the dynamic variables into divide-and-conquer attack. In Section 5, the attacks on 6/7-round reduced Ketje Sr v1 are presented. In Section 6, the attacks on 6/7-round reduced Ketje Sr v2 are presented. Section 7 gives some discussions on other Ketje instances and length-reduced nonce. In Section 8, the practical evaluation of our attack is presented. Section 9 concludes this paper.

1 Preliminaries

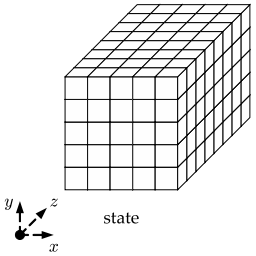
1.1 Notations

$\oplus, \neg$ *and* $\wedge$	bit-wise XOR, negation and logic AND
$A[x, y]$	a lane in x-th column and y-th row of state A
$A[x, y, z]$	z-th bit of $A[x, y]$. $A[x, y, \{z_i, z_j, ...\}]$ means $z_i, z_j, ...$-th bit of $A[x, y]$

1.2 The Keccak-p permutations

The KECCAK-p permutations are derived from the KECCAK-f permutations [BDPA] and have a tunable number of rounds. The KECCAK-p permutation is defined by its width $b = 25 \times 2^l$, with $b \in \{25, 50, 100, 200, 400, 800, 1600\}$, and its number of rounds n_r, denoted as KECCAK-$p[b]$. The round function $\mathtt{R}$ consists of five operations:

$$\mathtt{R} = \iota \circ \chi \circ \pi \circ \rho \circ \theta$$



(a)

0, 0	1, 0	2, 0	3, 0	4, 0
0, 1	1, 1	2, 1	3, 1	4, 1
0, 2	1, 2	2, 2	3, 2	4, 2
0, 3	1, 3	2, 3	3, 3	4, 3
0, 4	1, 4	2, 4	3, 4	4, 4

(b)

Figure 1: (a) The KECCAK State [BDPA], (b) State A in 2-dimension

KECCAK-$p[b]$ works on a state A of size b, which can be represented as 5×5 $\frac{b}{25}$-bit lanes, as depicted in Figure 1, $A[x, y]$ with x for the index of column and y for the index of row. In what follows, indexes of x and y are in set $\{0, 1, 2, 3, 4\}$ and they are working in modulo 5 without other specification.

$$\theta : A[x, y] = A[x, y] \oplus \sum_{j=0}^{4} (A[x-1, j] \oplus (A[x+1, j] \lll 1)).$$
$$\rho : A[x, y] = A[x, y] \lll r[x, y].$$
$$\pi : A[y, 2x + 3y] = A[x, y].$$
$$\chi : A[x, y] = A[x, y] \oplus ((\neg A[x+1, y]) \wedge A[x+2, y].$$
$$\iota : A[0, 0] = A[0, 0] \oplus RC.$$

In KETJE v2, *the twisted permutations*, KECCAK-$p^*[b]$=$\pi\circ$ KECCAK-$p[b] \circ \pi^{-1}$, are introduced to effectively re-order the bits in the state. π^{-1} is the inverse of π, shown in Figure 2.

$$\pi^{-1} : A[x + 3y, x] = A[x, y].$$

1.3 Ketje

KETJE [BDP+16a] is a submission by the KECCAK team. It is a sponge-like construction. In KETJE v1, two instances are proposed, KETJE SR and JR with 400-bit and 200-bit state sizes, respectively. In the latest KETJE v2, another two instances KETJE MINOR and MAJOR are added to the family, with 800-bit and 1600-bit state sizes, respectively. KETJE

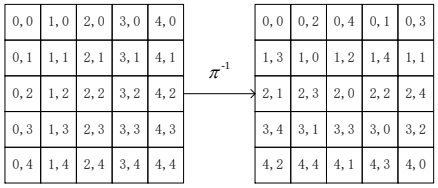


Figure 2: π^{-1}

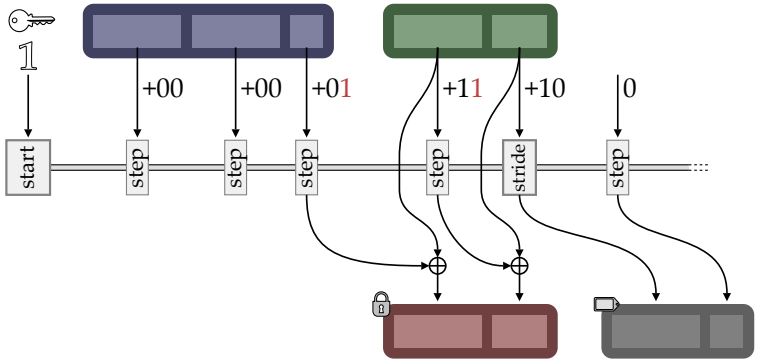


Figure 3: Wrapping a Header and a Body with MonkeyWrap[BDP+16a]

Sr is the primary recommendation. In the following, we give a brief overview about the Ketje v2. For a complete description, we refer to the design document [BDP+16a].

The structure of Ketje is an authenticated encryption mode MonkeyWrap, shown Figure 3, which is based on MonkeyDuplex [BDPA11]. It consists of four parts: the initialization phase, processing associated data, processing the plaintext and finalization. The initialization takes the secret key K, the public nonce N and some paddings as the initial state. Then $n_{start} = 12$ rounds Keccak-p^* is applied. In the subsequent processing of the associated data, ρ-bit blocks are padded to $(\rho + 4)$-bit and absorbed by xoring them to the state, then $n_{step} = 1$ round Keccak-p^* is applied. It should be noted that, if associated data is empty, this procedure is still needed to be applied which means an empty block is padded to $(\rho + 4)$-bit and then processed similarly. Plaintext is processed in ρ-bit blocks in a similar manner, with ciphertext blocks extracted from the state right after adding the plaintext. After all data is processed, the finalization with $n_{stride} = 6$ rounds Keccak-p^* and a series of $n_{step} = 1$ round Keccak-p^*s are performed to get the required length of tag T. In Ketje v1, Keccak-p is used instead of Keccak-p^*.

In Ketje v2, four instances are proposed, shown in Table 2. $n_{start} = 12$, $n_{step} = 1$ and $n_{stride} = 6$. Out of the four instances, Ketje Sr is the primary recommendation. The recommended key length is 128-bit and length of nonce is (400-128-18=254) bits.

Table 2: Four Instances in Ketje v2

Name	f	ρ	Main use case
Ketje Jr	Keccak-p^*[200]	16	lightweight
Ketje Sr	Keccak-p^*[400]	32	lightweight
Ketje Minor	Keccak-p^*[800]	128	lightweight
Ketje Major	Keccak-p^*[1600]	256	high performance

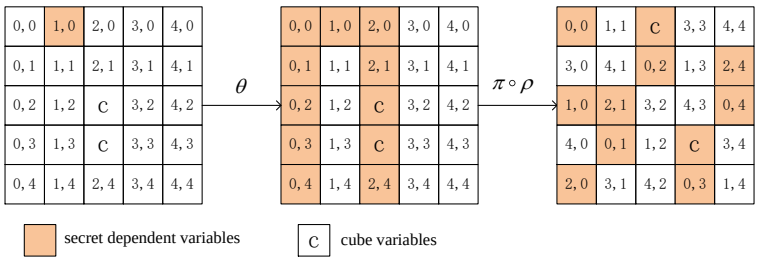


Figure 4: Dinur *et al.*'s Work

1.4 Our Attack Assumptions

In our attacks, we assume the associated data is less than ρ-bit, so a 13-round KECCAK-p^* is applied to the initial state, before the ciphertext outputs. We reduce the full 13 rounds to 6 and 7 rounds to launch the attacks. Hence, the attacks cover 6/7 rounds, including $n_{start} = 5/6$ initialization rounds and $n_{step} = 1$ step round. This paper includes the attacks on round-reduced KETJE SR v1 and v2. In v1, the KECCAK-p^* is replaced by KECCAK-p. The attacks given in Section 5 and Section 6 are on the primary recommendation KETJE SR with the recommended 128-bit key and 254-bit nonce.

2 Related Works

2.1 Cube Attack

The cube attack [DS09] assumes that the output bit of a cipher can be described as a master polynomial $p(k_1, ..., k_n, v_1, ..., v_m)$ over $\mathbb{F}_2$, which can be written as a sum of two polynomials:

$$p(k_1, ..., k_n, v_1, ..., v_m) = t_I \cdot p_{S(I)} + q(k_1, ..., k_n, v_1, ..., v_m),$$

where: $k_1, ..., k_n$ are secret variables (e.g. the key bits), $v_1, ..., v_m$ are public variables (e.g. the nonce or IV bits); t_I is called maxterm and is a product of certain public variables; $p_{S(I)}$ is called superpoly; $q(k_1, ..., k_n, v_1, ..., v_m)$ is the remainder polynomial and none of the terms in it is divisible by t_I. The maxterm t_I is defined through a subset of indices I, called a cube. In order to get the super polynomial $p_{S(I)}$, one assigns all possible values to the variables contained in I, evaluates the master polynomial and sums up the results.

2.2 Dinur *et al.*'s Divide-and-Conquer Method

At EUROCRYPT 2015, Dinur *et al.* launched a cube-attack-like cryptanalysis on KECCAK keyed modes, including the MAC mode, AE scheme KEYAK and the stream cipher mode. The best key recovery attack is based on a divide-and-conquer method. In the attack on 7-round reduced KECCAK-MAC, the 128-bit key is placed in the lane $A[0, 0]$ and $A[1, 0]$. They find if the cube variables are in $A[2, 2]$ and $A[2, 3]$ which are equal in the same column, shown in Figure 4, after θ, ρ and π, the cube variables do not multiply with key in $A[1, 0]$ after the first round. The cube sums after 7-round are independent of the key bits in $A[1, 0]$.

In addition, Dinur *et al.* introduce 32 bits auxiliary variables which are assumed to be equal to key bits in $A[0, 0]$ in the same column, shown in Figure 10. Hence, half of $A[0, 0]$ (32-bit key) and auxiliary variables are diffused to $A[0, 0]$ and $A[1, 3]$ without affecting

many lanes, which makes that cube variables in $A[2,2]$ and $A[2,3]$ do not multiply with those key bits and auxiliary variables in the first round. So only 32 key bits will multiply with the cube variables after the first round, which means only 32 key bits will affect the cube sums of the output after 7-round.

The whole 7-round attack is as follows. In preprocessing phase, the attacker calculates the cube sums for each of 32-bit keys which multiply with cube variables and store them in a list L. In the online phase, for 2^{32} values of 32-bit auxiliary variables, calculate the cube sums for the output bits and search them in L, for each match in L return the corresponding 32-bit key as a candidate. For more details, please refer to [DMP+15].

2.3 Conditional Cube Attack

Conditional cube attack [HWX+] was proposed by Huang *et al.* to attack Keccak keyed modes, and later generalized by Li *et al.* [LDW17]. Inspired by dynamic cube attack [DS09], which reduces the degree of the output polynomials over cube variables by adding some bit conditions on the initial value (IV), they reduce the degree by appending key bit conditions. It is similar to message modification technique [WY05, WYY05] and conditional differential cryptanalysis [KMN10], which used bit conditions to control differential propagation. In conditional cube attack, one firstly choose a so-called conditional cube variable, by adding some conditions on key bits and non-cube IV bits, the diffusion of conditional cube variable is much reduced. Then some so-called ordinary cube variables are chosen to meet that all cube variables do not multiply with each other in the first round and all ordinary cube variables do not multiply with the conditional cube variable in the second round. Hence, under bit conditions, the degree of the output polynomial is reduced by 1. Then one guesses the related key bits and let the bit conditions hold, launch the cube tester to determine if the guessed key bits are right. They gave 7-round cube testers for Keccak-MAC and Lake Keyak. Due to the large output state of Lake Keyak, they could decrypt by one χ to get some output bits of 7.5-round (the cube testers still hold here) and achieve an 8-round key-recovery attack on Lake Keyak. For more details, please refer to [HWX+].

In Ketje Sr, the internal state is much smaller (400-bit), in the second round, the conditional cube variables are much denser, although the bit conditions are added. So the chance that an ordinary cube variable does not multiply with the conditional cube variable in the second round is much smaller than that of the 1600-bit Lake Keyak. In fact, we test Huang *et al.*'s conditional cube attack on a much larger state 800-bit, only 25 such ordinary cube variables are found, which could only give 5-round key-recovery attacks. So we claim that in a much smaller state cipher i.e. 400-bit Ketje Sr, the conditional cube attack will not work when the attacked number of round is reduced to 6 or 7.

2.4 Dynamic Cube Attack

At FSE 2011, Dinur and Shamir [DS11] introduced a variant of cube attack called dynamic cube attack, whose core is to find dynamic variable, which depends on some of the cube public variables and some private variables (the key bits), to nullify the complex function. As an example given in [DS11], a polynomial P is written as $P = P_1 \cdot P_2 + P_3$, $P_1 \cdot P_2$ is the complex function, however, P_1 is a relatively simple function about the cube public variables and some private variables. Choose a dynamic variable to make $P_1 = 0$ and nullify $P_1 \cdot P_2$, and at last P is simplified.

2.5 Linear Structure

Inspired by Dinur *et al.*'s work, Guo, Liu and Song [GLS16] developed a new technique named *linear structure*, that allows linearization of the underlying permutation of Keccak for up to 3 rounds. Then a series of zero-sum distinguishers of Keccak permutations

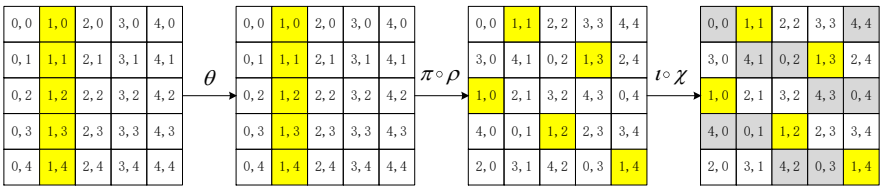


Figure 5: 1-Round Linear Structure

were proposed. Moreover, some preimage attacks against Keccak were launched by this technique.

Let $A[1,i]$, $i=0,1,2,3$, be variables and $A[1,4]=\bigoplus_{i=0}^{3}A[1,i]$, so that variables in this column sum to zero. Figure 5 shows how the variables affect the internal state under the transformation of Keccak-f round function $\mathtt{R}=\iota\circ\chi\circ\pi\circ\rho\circ\theta$. All the bits of the lanes in yellow have algebraic degree 1, those in light grey have algebraic degree at most 1, other lanes are all constants. In fact, only the non-linear operation χ can increase the algebraic degree through two neighbouring bits due to the term $(a_{i+1}\oplus 1)\cdot a_{i+2}$. Hence, the algebraic degree of the state bits remains at most 1 after one round function R. We denote the linear structure as 1-round linear structure. The size of free variables can be at most 4 lanes. If we use the 1-round linear structure as a cube, the cube dimension is 4 lanes, and cube variables will not multiply with each other after 1-round.

3 New Linear Structures of Ketje Sr

As shown in Figure 4, Dinur *et al.* find the 64 cube variables do not multiply with 64-bit key in $A[1,0]$ in the first round. For Ketje Sr v1 and v2, the situations are different. In order to mount a 6 or 7-round attack, one must get enough cube variables (32 and 64 cube variables for 6/7-round attacks, respectively). However, the size of a lane is only 16 for Ketje Sr instead of 64 for Keccak-MAC. So more lanes are needed to be cube variables, which makes the number of key bits that do not multiply with cube variables decrease. Taking advantage of the linear structure technique, we get many 32/64-dimension cubes whose cube variables do not multiply with each other in the first round. Then, we select cubes that do not multiply with as many key bits as possible in the first round. At last, we choose 4 cubes presented in Property 1, 3, 2 and 4 in our attacks.

Property 1. *In Ketje Sr v1, 32 cube variables do not multiply with 32-bit keys in $A[1,0]$ and $A[1,1]$ in the first round, shown in Figure 6, bits of c_i are the cube variables and $c_1\oplus c_2=const_1$, $c_3\oplus c_4=const_2$, $const_1$ and $const_2$ are constants.*

Property 2. *In Ketje Sr v1, without considering the last 2-bit padding in the nonce[3], there are 64 cube variables that do not multiply with 16-bit keys in $A[0,1]$ in the first round, shown in Figure 7, bits of c_i are the cube variables and $c_1\oplus c_2=const_1$,$c_3\oplus c_4\oplus c_5\oplus c_6=const_2$, $const_1$ and $const_2$ are constants.*

Property 3. *In Ketje Sr v2, 32 cube variables do not multiply with 56-bit keys in $A[0,2]$, $A[3,0]$, $A[3,3]$ and half of $A[0,0]$ in the first round, shown in Figure 8, bits of c_i are the cube variables and $c_1\oplus c_2\oplus c_3=const_1$, $const_1$ is constant.*

Property 4. *In Ketje Sr v2, 64 cube variables do not multiply with 32-bit keys in $A[3,0]$ and $A[3,3]$ in the first round, shown in Figure 9, bits of c_i are the cube variables and $c_1\oplus c_2\oplus c_3=const_1$ and $c_4\oplus c_5\oplus c_6=const_2$, $const_1$ and $const_2$ are constants.*

[3]The padding will be considered later in Subsection 5.2.

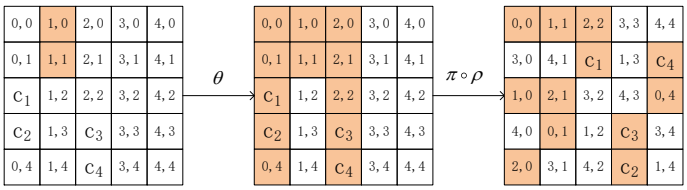


Figure 6: Diffusion of the 32-bit Keys in $A[1,0]$, $A[1,1]$ and 32-dimension Cube

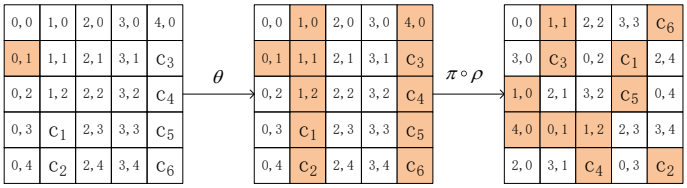


Figure 7: Diffusion of the 16-bit Keys in $A[0,1]$ and 64-dimension Cube

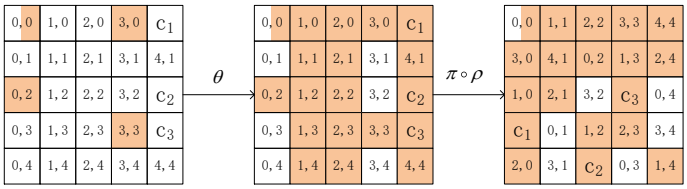


Figure 8: Diffusion of the 56-bit Keys in $A[0,2]$, $A[3,0]$, $A[3,3]$, Half of $A[0,0]$ and 32-dimension Cube

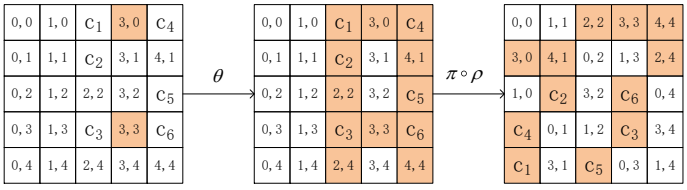


Figure 9: Diffusion of the 32-bit Keys in $A[3,0]$, $A[3,3]$ and 64-dimension Cube

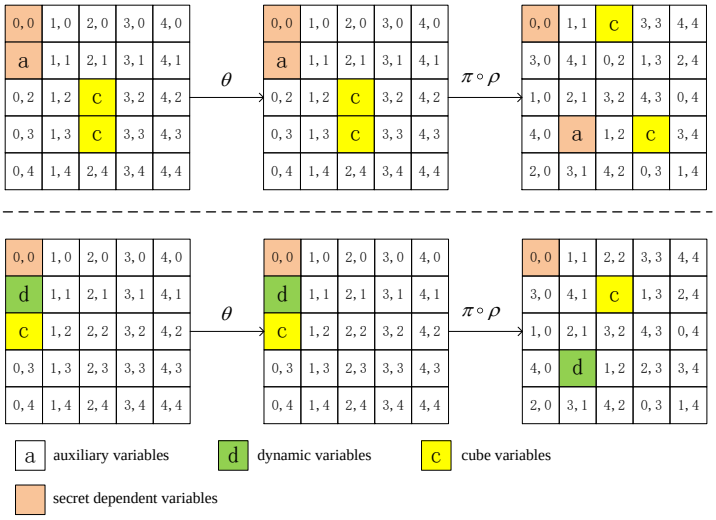


Figure 10: Comparing *Dynamic Variables* with *Auxiliary Variables*

4 Dynamic Variables Used in Our Attacks

In Dinur *et al.*'s work [DMP+15] on Keccak keyed modes, some *auxiliary variables* are introduced to reduce the diffusion of the key bits, shown in Subsection 2.2. In this paper, we firstly introduce the *dynamic variables* to replace *auxiliary variables* in the divide-and-conquer attack, which makes this attack possible to be applied to Ketje Sr.

A *dynamic variable* [DS11] depends on some of the cube variables as well as some private variables (the key bits). An example use of *dynamic variable* is introduced in Figure 10, which also gives a brief comparison on the effect of *auxiliary variables* and *dynamic variables*. The *auxiliary variables* a are equal to the key bits in $A[0,0]$ within the same column; each of *dynamic variables* d in $A[0,1]$ is the sum of a key bit in $A[0,0]$ and a cube variable in c within the same column.

In the upper part of Figure 10, the cube variables in $A[2,2]$ and $A[2,3]$ are set to be equal to reduce the diffusion of cube variables. The *auxiliary variables* a reduce the diffusion of the key bits in $A[0,0]$. At last, the cube variables do not multiply with each other in the first round. In addition, the cube variables do not multiply with *auxiliary variables* a and the key bits in $A[0,0]$ in the first round. The dimension of the cube is 16 and 4 lanes of the state are variables.

In the lower part of Figure 10, a *dynamic variable* in $A[0,1]$ is supposed to be equal to the sum of a key bit in $A[0,0]$ and a cube variable in c within the same column. So one lane of *dynamic variables* can reduce the diffusion of the key bits in $A[0,0]$ as well as cube variables in $A[0,2]$. In the first round, the cube variables do not multiply with each other and additionally do not multiply with *dynamic variables* d and the key bits in $A[0,0]$. The dimension of the cube is the same with upper part, but only 3 lanes of the state are variables. Obviously, the advantage of the *dynamic variable* over *auxiliary variable* is that it provides more sparse variables, meanwhile it maintains the size of cube and reduces the diffusion of the same number of key bits.

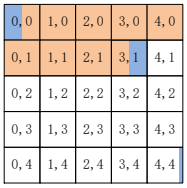


Figure 11: Initialize the State with Key, Padding and Nonce of Ketje Sr v1

Figure 12: Attack on 6-round Ketje Sr v1

5 Attacks on Round-Reduced Ketje Sr v1

In this section, we will introduce the first 6/7-round key-recovery attacks on Ketje Sr v1. As shown in Figure 11, we suppose the key fills the pink lanes. Note that $A[0,0,\{0,1,...,7\}]$, $A[3,1,\{8,9,...,15\}]$ and $A[4,4,\{14,15\}]$ are padding bits highlighted in blue. Other bits are nonce. θ, ρ, π *et al.* will be applied to the state successively.

5.1 Attack on 6-round Ketje Sr v1

We use the 32-dimension cube introduced in Property 1. As shown in Figure 12, the cube variables are placed in $A[0,3]$ and $A[2,4]$, which are denoted as $\{v_0, v_1, ..., v_{15}\}$ and $\{v_{16}, v_{17}, ..., v_{31}\}$. We introduce the dynamic variables in $A[0,2]$ and $A[2,3]$ highlighted in green, denoted as $\{d_0, d_1, ..., d_{15}\}$ and $\{d_{16}, d_{17}, ..., d_{31}\}$ which act as a similar role as auxiliary variables[4] used in Dinur *et al.*'s work [DMP+15]. Note that half of $A[0,0]$ are padding bits (8-bit). We denote the key bits (secret variables) in pink lanes (half of $A[0,0]$, $A[0,1]$, $A[2,0]$ and $A[2,1]$), as $\{k_0, k_1, ..., k_7\}$, $\{k_8, k_9, ..., k_{23}\}$, $\{k_{24}, k_{25}, ..., k_{39}\}$, $\{k_{40}, k_{41}, ..., k_{55}\}$, respectively. Then the dynamic variables meet the following conditions:

$$\begin{cases} d_i = v_i \oplus k_{i+8}, i = 0, 1, ..., 7 \\ d_i = v_i \oplus k_{i-8} \oplus k_{i+8}, i = 8, 9, ..., 15 \\ d_i = v_i \oplus k_{i+8} \oplus k_{i+24}, i = 16, 17, ..., 31 \end{cases} \tag{1}$$

According to Property 1, the cube variables do not multiply with 32-bit keys in $A[1,0]$ and $A[1,1]$ in the first round. If the conditions in Equation 1 hold, 56-bit keys k_i, $i = 0, 1, ..., 55$, will not multiply with the cube variables and dynamic variables either. In addition, the cube variables will not multiply with each other as well in the first round. Then only 40-bit keys placed in $A[3,0]$, $A[4,0]$ and half of $A[3,1]$ will multiply with cube variables in the first round, and then affect the cube sums after 6-round. The divide-and-conquer attack procedures are presented as follows.

[4] Their differences are presented in Section 4.

Preprocessing Phase:

1. Set the $A[0,0,\{0,1,...,7\}] = \{0,1,0,0,1,0,0,0\}$, $A[3,1,\{8,9,...,15\}] = \{1,0,0,0,0,0,0,0\}$ and $A[4,4,\{14,15\}] = \{1,1\}$ to meet the padding rule. Set all other state bits to 0 (except $A[3,0]$, $A[4,0]$, $A[3,1,\{0,1,...,7\}]$, $A[4,1,0]$, 32-bit cube variables and dynamic variables).

2. For the 2^{40} possible values of $(A[3,0], A[4,0], A[3,1,\{0,1,...,7\}])$:

   (a) $A[4,1,0] = 0$, calculate the cube sums after 6 rounds for all the 32 output bits,

   (b) $A[4,1,0] = 1$, calculate the cube sums after 6 rounds for all the 32 output bits,

   (c) Store the two 32-bit cube sums in a sorted list L, next to the value of the corresponding $(A[3,0], A[4,0], A[3,1,\{0,1,...,7\}])$.

Online Phase:

1. For each guess of 2^{32} values: k_{i+8} $(i = 0,1,...,7)$, $k_{i-8} \oplus k_{i+8}$ $(i = 8,9,...,15)$ and $k_{i+8} \oplus k_{i+24}$ $(i = 16,17,...,31)$, which are used to compute dynamic variables according to Equation 1:

   (a) $A[4,1,0] = 0$, request the outputs of the 2^{32} messages that make up the chosen cube (using the same constant as in the preprocessing phase). Note that according to Equation 1, dynamic variables are computed by the values of cube variables and the guessed keys. Calculate the 32-bit cube sums.

   (b) $A[4,1,0] = 1$, request the outputs of the 2^{32} messages that make up the chosen cube (using the same constant as in the preprocessing phase). Calculate the 32-bit cube sums.

   (c) Search cube sums in L.

   (d) For each match in L, retrieve $(A[3,0], A[4,0], A[3,1,\{0,1,...,7\}])$ and store all the candidates combining with 32-bit value of k_{i+8} $(i = 0,1,...,7)$, $k_{i-8} \oplus k_{i+8}$ $(i = 8,9,...,15)$ and $k_{i+8} \oplus k_{i+24}$ $(i = 16,17,...,31)$.

2. For each candidates, guess the remaining unknown $128 - 40 - 32 = 56$ key bits, and use one $(nonce, P, C, T)$ pair to check to get the full 128-bit key.

Complexity Analysis. In the online phase, we can get $2^{-64} \times 2^{40} \times 2^{32} = 2^8$ candidates for $32 + 40 = 72$ bits keys, which are k_{i+8} $(i = 0,1,...,7)$, $k_{i-8} \oplus k_{i+8}$ $(i = 8,9,...,15)$, $k_{i+8} \oplus k_{i+24}$ $(i = 16,17,...,31)$ and $A[3,0], A[4,0], A[3,1,\{0,1,...,7\}]$. In step 2, we need $2^{56+8} = 2^{64}$ encryptions to get the full key.

The time complexity of online phase is $2^{32} \times 2 \times 2^{32} + 2^{64} = 2^{65.6}$ encryptions. The time complexity of the preprocessing phase is $2^{40+1+32} = 2^{73}$ encryptions. The memory complexity is 2^{40} 64-bit words.

5.2 Attack on 7-round Ketje Sr v1

We use the 64-dimension cube introduced in Property 2. However, when considering the padding rule (the last two bits of $A[4,4]$ are padding bits), the cube size will be reduced to 62. So we will at first revise Property 2 a little. As shown in Figure 13, we add two bits as cube variables in $A[1,2,\{14,15\}]$, and meet the condition: $c_0 \oplus c_1 \oplus c_2 = const_1$ in the same column. After θ, ρ and π, the two cube variables will be neighbor to two key bits in $A[0,1]$. Hence, only 14 key bits in $A[0,1]$ do not multiply with all the cube variables in the first round.

We use the revised property to attack 7-round Ketje Sr v1. As shown in Figure 14, the cube variables are placed in $A[1,4]$, $A[4,2]$, $A[4,3]$, $A[4,4,\{0,1,..,13\}]$ and two

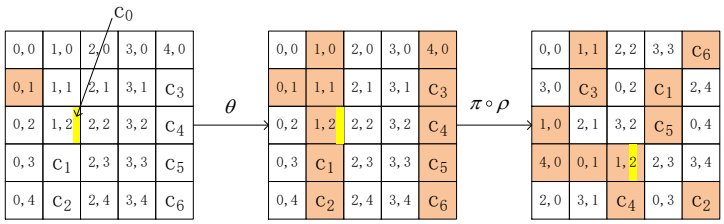


Figure 13: Property 2 Revised

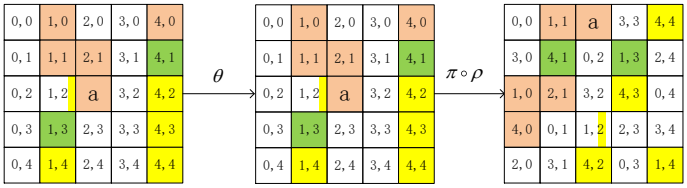


Figure 14: Attack on 7-round Ketje Sr v1

bits of $A[1,2]$, which are denoted as $\{v_0, v_1, ..., v_{15}\}$, $\{v_{16}, v_{17}, ..., v_{31}\}$, $\{v_{32}, v_{33}, ..., v_{47}\}$, $\{v_{48}, v_{49}, ..., v_{60}, v_{61}\}$ and $\{v_{62}, v_{63}\}$, respectively. We introduce the dynamic variables in $A[1,3]$ and $A[4,1]$ highlighted in green, denoted as $\{d_0, d_1, ..., d_{15}\}$ and $\{d_{16}, d_{17}, ..., d_{31}\}$. In addition, 16 auxiliary variables in $A[2,2]$ are introduced as $\{a_0, a_1, ..., a_{15}\}$ We denote the key bits (secret variables) in $A[1,0]$, $A[1,1]$, $A[2,1]$ and $A[4,0]$, as $\{k_0, k_1, ..., k_{15}\}$, $\{k_{16}, k_{17}, ..., k_{31}\}$, $\{k_{32}, k_{34}, ..., k_{47}\}$ and $\{k_{48}, k_{49}, ..., k_{63}\}$, respectively. Then the dynamic variables and auxiliary variables meet the following conditions:

$$\begin{cases} d_i = v_i \oplus k_i \oplus k_{i+16}, i = 0, 1, ..., 13, \\ d_i = v_i \oplus v_{i+48} \oplus k_i \oplus k_{i+16}, i = 14, 15, \\ d_i = v_i \oplus v_{i+16} \oplus v_{i+32} \oplus k_{i+32}, i = 16, 17, ..., 29, \\ d_i = v_i \oplus v_{i+16} \oplus k_{i+32}, i = 30, 31, \\ a_i = k_{i+32}, i = 0, 1, ..., 15. \end{cases} \tag{2}$$

If the conditions are met, the cube variables will do not multiply with 64 bits $k_i, i = 0, 1, ..., 63$ and 14-bit key in $A[0,1]$ in the first round. In addition, the cube variables will do not multiply with each other in the first round. And the cube size is 64. We use this cube to attack 7-round Ketje Sr v1. Only 50-bit keys ($A[0,0,\{8,9,...,15\}]$, $A[2,0]$, $A[3,0]$, $A[3,1,\{0,1,...,7\}]$ and 2-bit in $A[0,1]$) will multiply with cube variables after the first round, and then affect the cube sums after 7-round. The attack procedures are as follows.

Preprocessing Phase:

1. Set the $A[0,0,\{0,1,...,7\}] = \{0,1,0,0,1,0,0,0\}$, $A[3,1,\{8,9,...,15\}] = \{1,0,0,0,0,0,0,0\}$ and $A[4,4,\{14,15\}] = \{1,1\}$ to meet the padding rule. Set all other state bits to 0 (except $A[0,0,\{8,9,...,15\}]$, $A[2,0]$, $A[3,0]$, $A[3,1,\{0,1,...,7\}]$, and 2-bit in $A[0,1]$, $A[3,2,0]$, 64-bit cube variables, 32-bit dynamic variables).

2. For the 2^{50} possible values of ($A[0,0,\{8,9,...,15\}]$, $A[2,0]$, $A[3,0]$, $A[3,1,\{0,1,...,7\}]$ and 2-bit in $A[0,1]$):

   (a) $A[3,2,0] = 0$, calculate the cube sums after 7 rounds for all the 32 output bits,

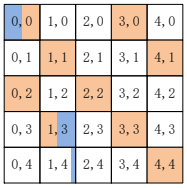


Figure 15: State After π^{-1}

(b) $A[3,2,0]=1$, calculate the cube sums after 7 rounds for all the 32 output bits,

(c) Store the two 32-bit cube sums in a sorted list L, next to the value of the corresponding $(A[0,0,\{8,9,...,15\}], A[2,0], A[3,0], A[3,1,\{0,1,...,7\}]$ and 2-bit in $A[0,1])$.

Online Phase:

1. For each of 2^{48} values: $k_i \oplus k_{i+16}$ $(i=0,1,...,14,15)$, k_{i+32} $(i=0,1,...,30,31)$, which are used to compute dynamic variables and auxiliary variables according to Equation 2:

   (a) $A[3,2,0]=0$, request the outputs of the 2^{64} messages that make up the chosen cube (using the same constant as in the preprocessing phase). Note that according to Equation 2, dynamic variables are computed by the values of cube variables. Calculate the 32-bit cube sums.

   (b) $A[3,2,0]=1$, request the outputs of the 2^{64} messages that make up the chosen cube (using the same constant as in the preprocessing phase). Calculate the 32-bit cube sums.

   (c) Search cube sums in L.

   (d) For each match in L, retrieve $(A[0,0,\{8,9,...,15\}], A[2,0], A[3,0], A[3,1,\{0,1,...,7\}]$ and 2-bit in $A[0,1])$ and store all the candidates combining with 48-bit value of $k_i \oplus k_{i+16}$ $(i=0,1,...,14,15)$, k_{i+32} $(i=0,1,...,30,31)$.

2. For each candidate, guess the remaining unknown $128-50-48=30$ key bits, and use one $(nonce, P, C, T)$ pair to check to get the full 128-bit key.

Complexity Analysis. In the online phase, we can get $2^{-64}\times 2^{50}\times 2^{48}=2^{34}$ candidates for $48+50=98$ bits keys, which are $A[0,0,\{8,9,...,15\}], A[2,0], A[3,0], A[3,1,\{0,1,...,7\}]$ and 2-bit in $A[0,1]$ and $k_i \oplus k_{i+16}$ $(i=0,1,...,14,15)$, k_{i+32} $(i=0,1,...,30,31)$. The last step, we need $2^{34+30}=2^{64}$ encryptions to get the full key.

The time complexity of online phase is $2^{48}\times 2\times 2^{64}+2^{64}=2^{113}$ encryptions. The time complexity of the preprocessing phase is $2^{50+1+64}=2^{115}$ encryptions. The memory complexity is 2^{50} 64-bit words.

6 Attacks on Round-Reduced Ketje Sr v2

In Ketje Sr v2, the state is updated by π^{-1} at first, then θ, ρ, π *et al.* are applied to the state successively. Equivalently, as shown in Figure 15, we suppose the key fills the pink lanes. Note that $A[0,0,\{0,1,...,7\}]$, $A[1,3,\{8,9,...,15\}]$ and $A[1,4,\{14,15\}]$ are padding bits highlighted in blue. Other bits are nonce. Then, θ, ρ, π *et al.* will be applied to the state successively.

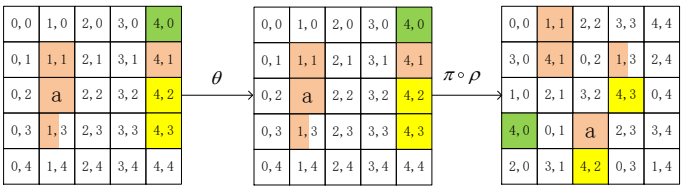


Figure 16: Attack on 6-round KETJE SR v2

6.1 Attack on 6-round Ketje Sr v2

In this attack, we use the 32-dimension cube introduced in Property 3. In Figure 16, the cube variables are placed in $A[4,2]$, $A[4,3]$, which are denoted as $\{v_0, v_1, ..., v_{15}\}$, $\{v_{16}, v_{17}, ..., v_{31}\}$, respectively. We introduce the dynamic variables in $A[4,0]$ highlighted in green, denoted as $\{d_0, d_1, ..., d_{15}\}$. The auxiliary variables in $A[1,2]$ are denoted as $\{a_0, a_1, ..., a_{15}\}$. We denote the key bits (secret variables) in $A[1,1]$, $A[1,3,\{0,1,...,7\}]$, $A[4,1]$ as $\{k_0, k_1, ..., k_{15}\}$, $\{k_{16}, k_{17}, ..., k_{23}\}$, $\{k_{24}, k_{25}, ..., k_{39}\}$, respectively. Then the dynamic variables and auxiliary variables meet the following conditions:

$$\begin{cases} a_i = k_i \oplus k_{i+16}, i = 0, 1, ..., 7, \\ a_i = k_i, i = 8, 9, ..., 15, \\ d_i = v_i \oplus v_{i+16} \oplus k_{i+24}, i = 0, 1, ..., 15. \end{cases} \tag{3}$$

If the conditions are met, the cube variables will do not multiply with 40 bits $k_i, i = 0, 1, ..., 39$ and 56-bit keys in $A[0,2]$, $A[3,0]$, $A[3,3]$ and half of $A[0,0]$ in the first round. In addition, the cube variables will do not multiply with each other in the first round. And the cube size is 32. We use this cube to attack 6-round KETJE SR v2. Only 32 key bits in $A[2,2]$ and $A[4,4]$ will multiply with cube variables in the first round, and then affect the cube sums after 6-round. The attack procedures are as follows.

Preprocessing Phase:

1. Set the $A[0,0,\{0,1,...,7\}] = \{0,1,0,0,1,0,0,0\}$, $A[1,3,\{8,9,...,15\}] = \{1,0,0,0,0,0,0,0\}$ and $A[1,4,\{14,15\}] = \{1,1\}$ to meet the padding rule. Set all other state bits to 0 (except $A[2,2]$, $A[4,4]$, $A[2,4,0]$, 32-bit cube variables, 16-bit dynamic variables).
2. For the 2^{32} possible values of $A[2,2]$ and $A[4,4]$:
   (a) $A[2,4,0] = 0$, calculate the cube sums after 6 rounds for all the 32 output bits,
   (b) $A[2,4,0] = 1$, calculate the cube sums after 6 rounds for all the 32 output bits,
   (c) Store the two 32-bit cube sums in a sorted list L, next to the value of the corresponding $A[2,2]$ and $A[4,4]$.

Online Phase:

1. For each guess of 2^{32} values: $k_i \oplus k_{i+16}$ $(i = 0, 1, ..., 7)$, k_i $(i = 8, 9, ..., 15)$, k_{i+24} $(i = 0, 1, ..., 15)$, which are used to compute dynamic variables and auxiliary variables according to Equation 3:
   (a) $A[2,4,0] = 0$, request the outputs of the 2^{32} nonces that make up the chosen cube (using the same constant as in the preprocessing phase). Note that according to Equation 3, dynamic variables are computed by the values of cube variables. Calculate the 32-bit cube sums.

Figure 17: Attack on 7-round Ketje Sr v2

(b) $A[2,4,0]=1$, request the outputs of the 2^{32} nonces that make up the chosen cube (using the same constant as in the preprocessing phase). Calculate the 32-bit cube sums.

(c) Search cube sums in L.

(d) For each match in L, retrieve $A[2,2]$, $A[4,4]$ and store all the candidates combining with 32-bit value $k_i \oplus k_{i+16}$ $(i=0,1,...,7)$, k_i $(i=8,9,...,15)$, k_{i+24} $(i=0,1,...,15)$.

2. For each candidate, guess the remaining unknown $128-32-32=64$ bits key, and use one $(nonce, P, C, T)$ pair to check to get the full 128-bit key.

Complexity Analysis. In online phase, we can get $2^{-64}\times 2^{32}\times 2^{32}=1$ candidate for $32+32=64$ key bits, which are $A[2,2]$, $A[4,4]$ and $k_i\oplus k_{i+16}$ $(i=0,1,...,7)$, k_i $(i=8,9,...,15)$, k_{i+24} $(i=0,1,...,15)$. In the last step, we need 2^{64} encryptions to get the full key.

The time complexity of online phase is $2^{32}\times 2\times 2^{32}+2^{64}=2^{65.6}$ encryptions. The time complexity of the preprocessing phase is $2^{32+1+32}=2^{65}$ encryptions. The memory complexity is 2^{32} 64-bit words.

6.2 Attack on 7-round Ketje Sr v2

In this attack, we use the 64-dimension cube introduced in Property 4. The cube variables are placed in $A[2,1]$, $A[2,3]$, $A[4,2]$, $A[4,3]$, which are denoted as $\{v_0,v_1,...,v_{15}\}$, $\{v_{16},v_{17},...,v_{31}\}$, $\{v_{32},v_{33},...,v_{47}\}$, $\{v_{48},v_{49},...,v_{63}\}$, respectively.

In Figure 17, we introduce the dynamic variables in $A[2,0]$ and $A[4,0]$ highlighted in green, denoted as $\{d_0,d_1,...,d_{15}\}$ and $\{d_{16},d_{17},...,d_{31}\}$. We denote the key bits (secret variables) in $A[2,2]$, $A[4,1]$, $A[4,4]$, as $\{k_0,k_1,...,k_{15}\}$, $\{k_{16},k_{17},...,k_{31}\}$, $\{k_{32},k_{34},...,k_{47}\}$, respectively. Then the dynamic variables meet the following conditions:

$$\begin{cases} d_i = v_i \oplus v_{i+16} \oplus k_i, i=0,1,...,14,15, \\ d_i = v_{i+16} \oplus v_{i+32} \oplus k_i \oplus k_{i+16}, i=16,17,...,31. \end{cases} \tag{4}$$

If the conditions are met, the cube variables will do not multiply with 48 bits $k_i, i=0,1,...,47$ and 32-bit key in $A[3,0]$ and $A[3,3]$ in the first round. In addition, the cube variables will do not multiply with each other in the first round. And the cube size is 64. We use this cube to attack 7-round Ketje Sr v2. Only 48 key bits $(A[0,0,\{8,9,...,15\}], A[0,2], A[1,1], A[1,3,\{0,1,...,7\}])$ will multiply with cube variables in the first round, and then affect the cube sums after 7-round. The attack procedures are as follows.

Preprocessing Phase:

1. Set the $A[0,0,\{0,1,...,7\}] = \{0,1,0,0,1,0,0,0\}$, $A[1,3,\{8,9,...,15\}] = \{1,0,0,0,0,0,0,0\}$ and $A[1,4,\{14,15\}] = \{1,1\}$ to meet the padding rule. Set all other state bits to 0 (except $A[0,0,\{8,9,...,15\}]$, $A[0,2]$, $A[1,1]$, $A[1,3,\{0,1,...,7\}]$, $A[2,4,0]$, 64-bit cube variables, 32-bit dynamic variables).

2. For the 2^{48} values of $A[0,0,\{8,9,...,15\}]$, $A[0,2]$, $A[1,1]$, $A[1,3,\{0,1,...,7\}]$:

   (a) $A[2,4,0] = 0$, calculate the cube sums after 7 rounds for all the 32 output bits,

   (b) $A[2,4,0] = 1$, calculate the cube sums after 7 rounds for all the 32 output bits,

   (c) Store the two 32-bit cube sums in a sorted list L, next to the value of the corresponding $A[0,0,\{8,9,...,15\}]$, $A[0,2]$, $A[1,1]$, $A[1,3,\{0,1,...,7\}]$.

Online Phase:

1. For each of 2^{32} values: k_i $(i = 0,1,...,14,15)$, $k_i \oplus k_{i+16}$ $(i = 16,17,...,31)$, which are used to compute dynamic variables according to Equation 4:

   (a) $A[2,4,0] = 0$, request the outputs of the 2^{64} messages that make up the chosen cube (using the same constant as in the preprocessing phase). Note that according to Equation 4, dynamic variables are computed by the values of cube variables. Calculate the 32-bit cube sums.

   (b) $A[2,4,0] = 1$, request the outputs of the 2^{64} messages that make up the chosen cube (using the same constant as in the preprocessing phase). Calculate the 32-bit cube sums.

   (c) Search cube sums in L.

   (d) For each match in L, retrieve $A[0,0,\{8,9,...,15\}]$, $A[0,2]$, $A[1,1]$, $A[1,3,\{0,1,...,7\}]$ and store all the candidates combining with 32-bit value of k_i $(i = 0,1,...,14,15)$, $k_i \oplus k_{i+16}$ $(i = 16,17,...,31)$.

2. For each candidate, guess the remaining unknown $128-48-32 = 48$ key bits , and use one $(nonce, P, C, T)$ pair to check to get the full 128-bit key.

Complexity Analysis. In the online phase, we can get $2^{-64} \times 2^{48} \times 2^{32} = 2^{16}$ candidates for $48+32 = 80$ key bits, which are $A[0,0,\{8,9,...,15\}]$, $A[0,2]$, $A[1,1]$, $A[1,3,\{0,1,...,7\}]$ and k_i $(i = 0,1,...,14,15)$, $k_i \oplus k_{i+16}$ $(i = 16,17,...,31)$. In the last step, we need $2^{16+48} = 2^{64}$ encryptions to get the full key.

The time complexity of online phase is $2^{32} \times 2 \times 2^{64} + 2^{64} = 2^{97}$ encryptions. The time complexity of the preprocessing phase is $2^{48+1+64} = 2^{113}$ encryptions. The memory complexity is 2^{48} 64-bit words.

7 Other Ketje Instances and Nonce Lengths

7.1 Attacks on Other Ketje Instances

For Ketje Jr v1 and v2, as shown in Figure 18, the 96-bit key fills the pink lanes and padding bits are highlighted in blue. Other bits are nonce. θ, ρ, π *et al.* will be applied to the state successively. The length of a lane is 8.

As shown in Figure 19, for Ketje Jr v1 with 96-bit key and 86-bit nonce, 5-round key-recovery attack is achieved. The 16-dimension cube is placed in $A[4,2]$, $A[4,3]$ and $A[4,4]$, in which $A[4,2]$ is the sum of $A[4,3]$ and $A[4,4]$ within the same column. Since two bits of $A[4,4]$ are padding, 2 cube variables are set in $A[1,4,\{6,7\}]$, and 2 dynamic

Figure 18: Initialize State with Key, Padding and Nonce in Ketje Jr v1 (a) and v2 (b)

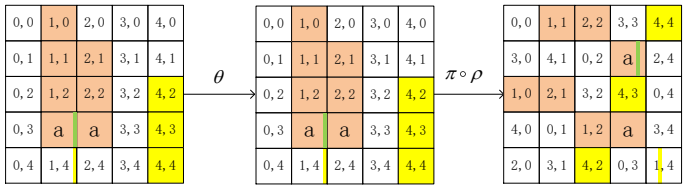


Figure 19: Attack on 5-round Ketje Jr v1

variables are set in $A[1,3,\{6,7\}]$. The (24-6=18)-bit[5] key in $A[0,1]$, $A[0,2]$, $A[3,0]$ and will not multiply with the cube variables without any condition in the first round. By adding auxiliary variables in $A[1,3]$ and $A[2,3]$, which are supposed to be equal to the sum of key bits in the pink lanes of the same column. Hence, the 40-bit key in $A[1,0]$, $A[1,1]$, $A[1,2]$ and $A[2,1]$, $A[2,2]$ will not multiply with the cube variables in the first round. Only 38-bit key ($A[3,1]$, $A[4,0]$, $A[4,1]$, $A[2,0]$, and 2-bit of $A[0,2]$, 4-bit of $A[3,0]$) will multiply with cube variables in the first round, and then affect the cube sums after 5-round. In the preprocessing phase, the time complexity is about 2^{56} encryptions and the memory cost is 2^{38} 64-bit words. In the online phase, 2^{34} encryptions are needed to recover (16+38=54) key bits and the remaining 42-bit key is recovered by exhaustive search. Hence the time complexity of online phase is $2^{34}+2^{42}\approx 2^{42}$ encryptions.

As shown in Figure 20, for Ketje Jr v2 with 96-bit key and 86-bit nonce, 5-round key-recovery attack is achieved. The 16-dimension cube is placed in $A[0,1]$, $A[0,3]$ and $A[0,4]$, in which $A[0,1]$ is the sum of $A[0,3]$ and $A[0,4]$ within the same column. The 32-bit key in $A[1,1]$, $A[1,3]$, $A[4,1]$ and $A[4,4]$ will not multiply with the cube variables without any condition in the first round. By adding auxiliary variables in $A[2,3]$ and $A[3,1]$, which are supposed to be equal to the sum of key bits in $A[2,1]$, $A[2,2]$ and $A[3,0]$, $A[3,2]$, respectively of the same column. Hence, the 32-bit key in $A[2,1]$, $A[2,2]$, $A[3,0]$ and $A[3,2]$ will not multiply with the cube variables in the first round. Only 32-bit key in $A[0,2]$, $A[1,0]$, $A[2,4]$ and $A[3,3]$ will multiply with cube variables in the first round, and then affect the cube sums after 5-round. In the preprocessing phase, the time complexity is about 2^{50} encryptions and the memory cost is 2^{32} 64-bit words. In the online phase, 2^{34} encryptions are needed to recover 48-bit key and the remaining 48-bit key are recovered by exhaustive search. Hence the time complexity of online phase is $2^{34}+2^{48}\approx 2^{48}$ encryptions.

For Ketje Minor and Major v1/2, we can achieve 6-round key-recovery attacks with 2^{64} preprocessing phase time complexity, 2^{64} online phase time complexity, and the memory cost is 2^{32}. The 7-round key-recovery attacks cost 2^{96} offline phase time complexity, 2^{96} online phase time complexity and 2^{32} memory.

[5]Note that 2-bit of $A[0,2]$ and 4-bit $A[3,0]$ will multiply with cube variables in $A[1,4,\{6,7\}]$ and $A[1,3,\{6,7\}]$.

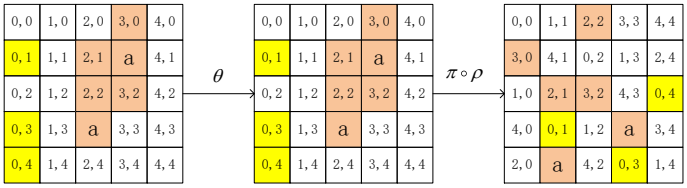


Figure 20: Attack on 5-round KETJE JR v2

Figure 21: Attack on 6-round KETJE SR v1 with 128-bit Nonce

7.2 Attacks on Ketje Sr with length-reduced nonce

When the length of nonce is 128-bit, only 6-round key-recovery attacks on KETJE SR v1 and v2 could be achieved. As shown in Figure 21, the blue lanes are the 128-bit padding of the nonce. For the attack on 6-round KETJE SR v1, the cube variables are placed in $A[1,3]$, $A[4,2]$. The dynamic variables are placed in $A[1,2]$, $A[4,1]$. The auxiliary variables are placed in $A[2,2]$. Note that the key in $A[0,0]$ does not multiply with cube variables without any condition. The attack costs 2^{80} time complexity in online phase and 2^{72} time in the preprocessing phase, and 2^{40} memory complexity.

As shown in Figure 22, the blue lanes are the 128-bit padding of the nonce. For the attack on 6-round KETJE SR v2, the cube variables are placed in $A[2,4]$, $A[4,3]$. The dynamic variables are placed in $A[2,1]$, $A[4,0]$. Note that 16 key bits in $A[3,3]$ do not multiply with cube variables without any condition. The attack costs 2^{64} time complexity in online phase and 2^{96} time in the preprocessing phase, and 2^{64} memory complexity.

8 Practical Evaluation

As the complexities of given attacks are infeasible, we reduce the 400-bit state KETJE SR v1 to a 200-bit state, and give a 5-round experiment using 16-dimension cube to check a limited number of keys. The experiment follows the attack on 6-round KETJE SR v1, using the same linear structure and dynamic cubes shown in Figure 12 of the

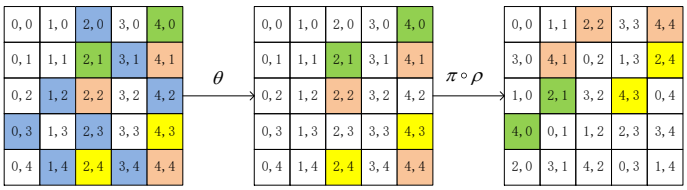


Figure 22: Attack on 6-round KETJE SR v2 with 128-bit Nonce

paper. As the state size is reduced to 200-bit, the full key size is reduced to 64-bit key. To accelerate the test, we set the key in $A[3,0]$ and half of $A[3,1]$ to be zero. So totally (64-8-4=52) key bits are secret. Note that half of $A[0,0]$ and half of $A[3,1]$ are padding bits. Without affecting the test, we simply set all padding bits to be zero. The source code is listed in `https://github.com/dongxiaoyang/Ketje_test`. In a PC using a single core (Intel(R)Core(TM)i7-4790 CPU@3.60GHz 3.60GHz) with Visual studio 2010 Release x64 platform, it cost about 2 hours to recover a 24-bit equivalent key. This experiment confirms the correctness of our attacks.

One experiment result (the key is randomly generated.):

```
24-bit Equivalent Right Key (Key[0]^Key[5],Key[2]^Key[7],Key[4]):[6]
0x21,0xd9,0x73
24-bit Equivalent Candidate Key (Key[0]^Key[5],Key[2]^Key[7],Key[4]):
0x21,0xd9,0x73
```

9 Conclusion

In this paper, we propose the first attacks on 6/7-round reduced Ketje Sr– the primary recommendation of Ketje family. First, by applying the linear structure technique, we find some 32/64-dimension cubes of Ketje Sr that do not multiply with some bits of the key in the first round. Second, we introduce the new dynamic variable instead of the auxiliary variable to reduce the diffusion of the key as well as the cube variables. Therefore, the number of key bits, which are independent of the cube sum after 6/7-round Ketje Sr, increases significantly and only a small fraction of the key bits will appear in the cube sum, which makes the divide-and-conquer attacks successfully translated to the 6/7-round key-recovery attacks on both Ketje Sr v1 and v2. In addition, some results on other Ketje instances and Ketje Sr with smaller nonce are given. They are the first results on Ketje and bridge the gaps of cryptanalysis between its sister cipher Keyak [BDP+16b] and the Keccak keyed modes. For Ketje Sr v1, the time complexity of 6/7-round attack is $2^{65.6}$ and 2^{113}, respectively. For Ketje Sr v2, the 7-round attack is 2^{97}. According to our attacks, we could claim that 7-round reduced Ketje Sr v2 is weaker than that of Ketje Sr v1 against cube-like attack. None of our attacks threaten the full 13-round version of Ketje.

Acknowledgments

We would like to thank Florian Mendel and the anonymous reviewers who helped improve this paper. This work is supported by China's 973 Program (No. 2013CB834205), the Strategic Priority Research Program of the Chinese Academy of Sciences (No. XDB01010600), the National Natural Science Foundation of China (No. 61672019 and 61402256), the Fundamental Research Funds of Shandong University (No. 2016JC029), and the Foundation of Science and Technology on Information Assurance Laboratory (No. KJ-15-002).

References

[ADMS09] Jean-Philippe Aumasson, Itai Dinur, Willi Meier, and Adi Shamir. Cube testers and key recovery attacks on reduced-round MD6 and trivium. In Orr Dunkelman, editor, *FSE 2009*, volume 5665 of *Lecture Notes in Computer Science*, pages 1–22. Springer, 2009.

[6] $Key[i]$s $(0 \le i \le 8)$ are placed in $A[0,0]$,$A[1,0]$, ..., $A[4,0]$, $A[0,1]$, ..., $A[3,1]$, respectively.

[BDP+16a] Guido Berton, Joan Daemen, Michaël Peeters, Gilles Van Assche, and Ronny Van Keer. CAESAR submission: KETJE v2, 2016. http://competitions.cr.yp.to/round3/ketjev2.pdf.

[BDP+16b] Guido Berton, Joan Daemen, Michaël Peeters, Gilles Van Assche, and Ronny Van Keer. CAESAR submission: KEYAK v2, 2016. http://competitions.cr.yp.to/round3/keyakv22.pdf.

[BDPA] Guido Berton, Joan Daemen, Michaël Peeters, and Gilles Van Assche. The KECCAK sponge function family. http://keccak.noekeon.org/.

[BDPA11] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Duplexing the sponge: Single-pass authenticated encryption and other applications. In Ali Miri and Serge Vaudenay, editors, *SAC 2011*, volume 7118 of *Lecture Notes in Computer Science*, pages 320–337. Springer, 2011.

[com14] The CAESAR committee. CAESAR: Competition for authenticated encryption: Security, applicability, and robustness, 2014. http://competitions.cr.yp.to/caesar.html.

[DEMS15] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Cryptanalysis of ascon. In Kaisa Nyberg, editor, *CT-RSA 2015*, volume 9048 of *Lecture Notes in Computer Science*, pages 371–387. Springer, 2015.

[DEMS16] Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Ascon v1.2. 2016.

[DMP+15] Itai Dinur, Pawel Morawiecki, Josef Pieprzyk, Marian Srebrny, and Michal Straus. Cube attacks and cube-attack-like cryptanalysis on the round-reduced keccak sponge function. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015*, volume 9056 of *Lecture Notes in Computer Science*, pages 733–761. Springer, 2015.

[DS09] Itai Dinur and Adi Shamir. Cube attacks on tweakable black box polynomials. In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *Lecture Notes in Computer Science*, pages 278–299. Springer, 2009.

[DS11] Itai Dinur and Adi Shamir. Breaking grain-128 with dynamic cube attacks. In Antoine Joux, editor, *FSE 2011*, volume 6733, pages 167–187. Springer, 2011.

[FV13] Pierre-Alain Fouque and Thomas Vannet. Improving key recovery to 784 and 799 rounds of trivium using optimized cube attacks. In Shiho Moriai, editor, *FSE 2013*, volume 8424 of *Lecture Notes in Computer Science*, pages 502–517. Springer, 2013.

[GLS16] Jian Guo, Meicheng Liu, and Ling Song. Linear structures: Applications to cryptanalysis of round-reduced keccak. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016*, volume 10031 of *Lecture Notes in Computer Science*, pages 249–274, 2016.

[HWX+] Senyang Huang, Xiaoyun Wang, Guangwu Xu, Meiqin Wang, and Jingyuan Zhao. Conditional cube attack on reduced-round KECCAK sponge function. *EUROCRYPT 2017* (*to appear*). http://eprint.iacr.org/2016/790.

[KMN10] Simon Knellwolf, Willi Meier, and María Naya-Plasencia. Conditional differential cryptanalysis of nlfsr-based cryptosystems. In Masayuki Abe, editor, *ASIACRYPT 2010*, volume 6477 of *Lecture Notes in Computer Science*, pages 130–145. Springer, 2010.

[LDW17] Zheng Li, Xiaoyang Dong, and Xiaoyun Wang. Conditional cube attack on round-reduced ascon. *IACR Trans. Symmetric Cryptol.*, 2017(1), 2017. http://eprint.iacr.org/2017/160.

[NISa] NIST VCAT: NIST cryptographic standards and guidelines development process: report and recommendations of the visiting committee on advanced technology of the national institute of standards and technology (2014).

[NISb] NIST. Advanced encryption standard (AES) (november 2001), federal information processing standards publication fips 197.

[WY05] Xiaoyun Wang and Hongbo Yu. How to break MD5 and other hash functions. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 19–35. Springer, 2005.

[WYY05] Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu. Finding collisions in the full SHA-1. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 17–36. Springer, 2005.