Math-Net.Ru

Общероссийский математический портал

V. A. Kiryukhin, Атака со связанными ключами на 5 раундов шифра Кузнечик, *Матем. вопр. криптогр.*, 2020, том 11, выпуск 2, 53–67

DOI: <https://doi.org/10.4213/mvk321>

Использование Общероссийского математического портала Math-Net.Ru подразумевает, что вы прочитали и согласны с пользовательским соглашением
<http://www.mathnet.ru/rus/agreement>

Параметры загрузки:

IP: 18.234.197.8

26 апреля 2023 г., 14:37:14

Related-key attack on 5-round Kuznyechik

V. A. Kiryukhin

JSC «InfoTeCS», LLC «SFB Lab», Moscow

Получено 05.XI.2019

Abstract. The first related-key attack on 3-round (of 9) Kuznyechik with 2-round (of 8) key schedule was presented in CTSrypt'18. This article describes a related-key attack on 5-round cipher with the same key schedule. The presented attack also has a practical complexity (2^{32} operations, 2^{30} memory, 2^{16} related keys, one chosen plaintext-ciphertext pair for each key) and was verified in practice. Our results was obtained due to the simultaneous use of the integral properties of the cipher transformations and the key schedule.

Keywords: Kuznyechik, related-key attack, integral cryptanalysis

Атака со связанными ключами на 5 раундов шифра Кузнечик

В. А. Кирюхин

ОАО «ИнфоТеКС», ООО «СФБ Лаб», Москва

Аннотация. Первая атака на связанных ключах на шифр Кузнечик была представлена на конференции CTSrypt'18. Атаковались 3 раунда шифра (из 9) и 2 раунда развертки ключа (из 8). В настоящей работе представлена атака на 5-раундовый шифр с аналогичной процедурой развертки ключа. Атака имеет практическую сложность (2^{32} операций, 2^{30} памяти, 2^{16} связанных ключей, одна пара выбранный текст-шифртекст для каждого ключа), была реализована на ЭВМ. Основной результат получен благодаря одновременному использованию интегральных свойств самого шифра и процедуры развертки ключа.

Ключевые слова: шифр Кузнечик, атака со связанными ключами, интегральный криптоанализ

1. Introduction

The setting of a related-key attack on ciphers was introduced in [6]. Informally this model assumes that adversary has access to several encryptors with different unknown keys, but he knows a certain simple relationship (for example, bitwise XOR) between these keys.

In some cases the related-key model is quite consistent with reality. A good example is an iterative hash function with block cipher as a part of compression function. In this case the adversary has a possibility to manipulate the encryption keys. Some cryptographic protocols may use related keys by design. One such related-key protocol CTRR was proposed at CTRCrypt'18 [2].

In the same publication [2], the first related-key attack on a reduced variant of block cipher Kuznyechik [1] was proposed. This approach exploits the ability of attacker to manipulate keys and the similarity of the functions in the encryption and the key schedule procedures.

In this paper we present a related-key attack on 5-round (of 9) Kuznyechik with 2-round (of 8) key schedule. Main result is obtained by means of the integral properties [4, 5] of the encryption and the key schedule. We also had used some approaches from [3]. The simplified versions of Kuznyechik are described in the next section (equations (2) and (3)).

The presented attack was verified in practice with C++ implementation. Source codes may be found at <https://gitlab.com/v.kir/rk-5R-kuznyechik>.

Comparative characteristics of attacks are presented in Table.

Table: Related-key attacks on Kuznyechik

Cipher rounds	Key schedule rounds	Operations	Keys	Memory	Source
3	2	2^{12}	2^{12}	$\sim$	[2]
5	2	2^{32}	2^{16}	2^{30}	Section 4

2. Definitions

Let $\mathbb{F}_{2^8}$ be a finite field as defined in [1]. Each element of $\mathbb{F}_{2^8}$ may be interpreted as an integer or as a binary vector. Field elements are indicated by lowercase letters: a, b . Denote vector space of dimension $n \in \mathbb{N}$ over $\mathbb{F}_{2^8}$ by $\mathbb{F}_{2^8}^n$. Elements from $\mathbb{F}_{2^8}^n$ will be denoted by capital letters: A, B . Blocks of plaintext and ciphertext also belong to $\mathbb{F}_{2^8}^n$.

Denote bitwise XOR operation by symbol $\oplus$. For a sequence of blocks

$$B_0, \dots, B_d \in \mathbb{F}_{2^8}^n, \quad d \in \mathbb{N},$$

we will call the sequence

$$\Delta \mathbf{B} = (B_0 \oplus B_1, B_0 \oplus B_2, \dots, B_0 \oplus B_d) \in (\mathbb{F}_{2^8}^n)^d \quad (1)$$

by a difference. Throughout the paper we always use the notation $d = 2^8 - 1$. Differences are indicated by bold: κ , $\Delta \mathbf{K}$.

The transformations over $\mathbb{F}_{2^8}^n$ (or sets of elements from $\mathbb{F}_{2^8}^n$) are denoted by Sans Serif font: $\mathbf{f}$, $\mathbf{S}$, $\mathbf{L}$. Such characters may mean a bijective transformation of blocks ($\mathbf{f}(A)$, $A \in \mathbb{F}_{2^8}^n$) or non-bijective transformation of differences to the set of differences (for example, $\mathbf{S}(\kappa)$ is a set of differences, $\kappa \in (\mathbb{F}_{2^8}^n)^d$). The notation $\mathbf{LS}$ indicates a composition of transformations, where $\mathbf{S}$ is applied first.

The difference $\Delta \in (\mathbb{F}_{2^8}^n)^d$ may also be interpreted as n «columns» of d bytes each: $\Delta \in (\mathbb{F}_{2^8}^d)^n$. If i -th «column» ($i = 1, 2, \dots, n$) $\alpha \in \mathbb{F}_{2^8}^d$ contains all different non-zero bytes, we say that i -th position has an integral property *All* (denoted by $\mathbf{A}$). Similarly, if XOR of all bytes is equal to zero, then i -th position of the difference has an integral property *Zero* (denoted by $\mathbf{0}$). If all positions of the block have the property $\mathbf{0}$, then the entire block has the property $\mathbf{0}$. Obviously, the property $\mathbf{A}$ implies the property $\mathbf{0}$. If at least one byte in such «column» is non-zero, we say that i -th position is active, otherwise inactive.

Kuznyechik

Kuznyechik [1] realizes a sequence of 9 rounds and a post-whitening key addition. Each round contains three operations:

$\mathbf{X}$ – modulo 2 addition of an input block with an iterative key, $\mathbf{X}[C](A) = C \oplus A$, $C, A \in \mathbb{F}_{2^8}^n$,

$\mathbf{S}$ – parallel application of a fixed bijective substitution to each byte of the block,

$\mathbf{L}$ – linear transformation defined as an LFSR over $\mathbb{F}_{2^8}$.

The block size is 128 bits ($n = 16$ bytes), the size of key K is equal to 256 bits.

Key schedule uses round constants $C_i \in \mathbb{F}_{2^8}^n$, $i = 1, 2, \dots, 32$.

Round keys $K_i \in \mathbb{F}_{2^8}^n$, $i = 1, 2, \dots, 10$, are derived from a master key K

as follows:

$$\begin{aligned} K &= K_1 || K_2, \\ (K_{2i+1}, K_{2i+2}) &= F[C_{8(i-1)+8}] \dots F[C_{8(i-1)+1}](K_{2i-1}, K_{2i}), \quad i = 1, 2, 3, 4, \\ F[C](A_1, A_2) &= (\text{LSX}[C](A_1) \oplus A_2, A_1), \quad C, A_1, A_2 \in \mathbb{F}_{2^8}^n. \end{aligned}$$

We define 3-round Kuznyechik as in [2]. Each round of the key schedule has only 2 rounds of basic cipher's Feistel rounds:

$$\begin{aligned} E_{K_1, K_2}(A) &= X[K_4] \text{LSX}[K_3] \text{LSX}[K_2] \text{LSX}[K_1](A), \\ (K_3, K_4) &= F[C_2] F[C_1](K_1, K_2), \\ K_3 &= K_1 \oplus \text{LSX}[C_2](K_2 \oplus \text{LSX}[C_1](K_1)), \\ K_4 &= K_2 \oplus \text{LSX}[C_1](K_1). \end{aligned} \tag{2}$$

5-round Kuznyechik is defined in a similar way:

$$\begin{aligned} E_{K_1, K_2}(A) &= X[K_6] \text{LSX}[K_5] \text{LSX}[K_4] \text{LSX}[K_3] \text{LSX}[K_2] \text{LSX}[K_1](A), \\ (K_3, K_4) &= F[C_2] F[C_1](K_1, K_2), \\ (K_5, K_6) &= F[C_4] F[C_3](K_3, K_4). \end{aligned} \tag{3}$$

Denote also the block before addition of the key K_i by P_i (for example, $P_2 = \text{LSX}[K_1](A)$).

3. Technical lemmas and concepts

The polytopic cryptanalysis was first introduced in [3]. We will use some techniques from this concept along with integral cryptanalysis [4].

In particular, we use the difference (1) as « d -difference» in [3]. Let's consider how cipher transformations change this difference.

It's easy to see that addition of a *same* round key does not change the difference. The attack presented in section 4 uses non-equal keys. In this case, the difference between the round keys is added to the difference between the intermediate states. Note that if both such differences $\Delta, \kappa \in (\mathbb{F}_{2^8}^n)^d$, have the integral property $\mathbf{0}$, then $\Delta \oplus \kappa$ has the same property.

Suppose that the difference $\Delta \in (\mathbb{F}_{2^8}^n)^d$ has only one active position, then after the S-transformation we have no more than 2^8 possible differences. Indeed, all inactive positions remain inactive. We have one non-zero «column» $\alpha = (c_1, c_2, \dots, c_d) \in \mathbb{F}_{2^8}^d$ and after substitution layer:

$$s(\alpha) = \{(s(x \oplus c_1) \oplus s(x), s(x \oplus c_2) \oplus s(x), \dots, s(x \oplus c_d) \oplus s(x)), x \in \mathbb{F}_{2^8}\},$$

where $\mathbf{s} : \mathbb{F}_{2^8} \rightarrow \mathbb{F}_{2^8}$ is the cipher S-box. Obviously, the number of differences $\mathbf{s}(\alpha)$ does not exceed the number of different x . In most cases, these numbers are equal. If all bytes in α are different, all bytes in $\mathbf{s}(\alpha)$ are also different (the bijective S-box preserve the integral property **A**). If we know α and $\alpha' \in \mathbf{s}(\alpha)$, we can easily find the corresponding x .

The L-transformation bijectively maps one difference to another:

$$\Delta = (\Delta_1, \Delta_2, \dots, \Delta_d), \quad \mathbf{L}(\Delta) = (\mathbf{L}(\Delta_1), \mathbf{L}(\Delta_2), \dots, \mathbf{L}(\Delta_d)).$$

If only one position in input difference is active, then all positions in output difference are active (this is true if $\mathbf{L}$ is MDS matrix). Under the same conditions, if one position has the property **A**, then all output positions will have this property. The integral property **0** is preserved by L-transformation:

$$\bigoplus_{i=1}^d \Delta_i = 0, \quad \bigoplus_{i=1}^d \mathbf{L}(\Delta_i) = \mathbf{L} \left(\bigoplus_{i=1}^d \Delta_i \right) = 0.$$

We will use the so-called integral property [4, 5] of LSXLSX transformation.

Lemma. *Let one position in the difference $\Delta \in (\mathbb{F}_{2^8}^n)^d$ has integral property **A** and all other positions are inactive (so-called δ -set). Then any difference from LSXLSX(Δ) has the integral property **0**.*

Proof. Adding a round key does not change the difference. Thus, we have LSLS(Δ). After the first substitution layer, one position will have the property **A** and all others will remain inactive. The first linear transformation will make all bytes active. Each of them will have the property **A**. The second S transformation will preserve **A** and consequently the property **0**. Hence, after the last linear transformation we have the property **0** in each position of the difference. $\square$

Equivalent representation of the last two rounds

The presented attack uses an equivalent representation of the last two rounds.

Let $A, B \in \mathbb{F}_{2^8}^n$ be a plaintext and ciphertext correspondingly. $K_1, \dots, K_r, K_{r+1}$ be round keys, $K_i \in \mathbb{F}_{2^8}^n$, $i = 1, 2, \dots, r+1$.

The original cipher has the form

$$B = \mathbf{X}[K_{r+1}]\mathbf{LSX}[K_r] \dots \mathbf{X}[K_1](A) = \mathbf{E}_{r+1}(A).$$

Apply the inverse linear transformation to the known ciphertext

$$\begin{aligned}\mathbf{L}^{-1}(B) &= \mathbf{L}^{-1}(\mathbf{X}[K_{r+1}]\mathbf{LSX}[K_r] \dots \mathbf{X}[K_1](A)), \\ \mathbf{L}^{-1}(B) &= \mathbf{L}^{-1}(K_{r+1}) \oplus \mathbf{SX}[K_r] \dots \mathbf{X}[K_1](A).\end{aligned}$$

We denote $B' = \mathbf{L}^{-1}(B)$, $K'_i = \mathbf{L}^{-1}(K_i)$, then the cipher takes the form

$$B' = \mathbf{X}[K'_{r+1}]\mathbf{SX}[K_r]\mathbf{LSX}[K_{r-1}] \dots \mathbf{X}[K_1](A).$$

Similarly, for the penultimate round. Let's consider the transformation

$$\mathbf{X}[K_r]\mathbf{L}(A) = K_r \oplus \mathbf{L}(A) = \mathbf{L}(A \oplus \mathbf{L}^{-1}(K_r)) = \mathbf{LX}[K'_r](A).$$

Therefore, the cipher transformation may be represented by the formula

$$B' = \mathbf{X}[K'_{r+1}]\mathbf{SLX}[K'_r]\mathbf{SX}[K_{r-1}] \dots \mathbf{X}[K_1](A).$$

4. Related-key attack

Let's represent the 5-round Kuznyechik (3) in an equivalent form:

$$\begin{aligned}\mathbf{E}_{K_1, K_2}(A) &= \mathbf{X}[K'_6]\mathbf{SLX}[K'_5]\mathbf{SX}[K_4]\mathbf{LSX}[K_3]\mathbf{LSX}[K_2]\mathbf{LSX}[K_1](A), \\ (K_3, K_4) &= \mathbf{F}[C_2]\mathbf{F}[C_1](K_1, K_2), \\ K_4 &= K_2 \oplus \mathbf{LSX}[C_1](K_1), \\ K_3 &= K_1 \oplus \mathbf{LSX}[C_2](K_4), \\ (K_5, K_6) &= \mathbf{F}[C_4]\mathbf{F}[C_3](K_3, K_4), \\ K_6 &= K_4 \oplus \mathbf{LSX}[C_3](K_3), \quad K'_6 = \mathbf{L}^{-1}(K_6), \\ K_5 &= K_3 \oplus \mathbf{LSX}[C_4](K_6), \quad K'_5 = \mathbf{L}^{-1}(K_5).\end{aligned}$$

The attack consists of the following steps:

- 1) Adversary chooses 2^8 collections of related keys, 2^8 keys in each collection. One plaintext C_1 (first constant in the key schedule) will be used.
- 2) For one of these collections, the special easy verifiable property (integral distinguisher) is true.
- 3) The round keys K_6 , K_5 are recovered by using integral and polytopic properties.

Let's describe these steps in more detail. We denote by

$$\kappa = \begin{pmatrix} 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & \dots & 0 & 2 \\ \dots & \dots & \ddots & \dots & \dots \\ 0 & 0 & \dots & 0 & 255 \end{pmatrix}$$

$\underbrace{\hspace{10em}}_{n=16}$

the difference between keys K_1 . The set $\text{LS}(\kappa)$ contains 2^8 differences. The collection of the related keys looks like

$$(K_1, K_2) \text{ and the set } (K_1 \oplus \kappa, K_2 \oplus \kappa''), \text{ where } \kappa'' \in \text{LS}(\kappa).$$

It is easy to see that each collection contains the «main» key and a set of 255 related keys. Adversary does not know the keys, but he know all relations (κ and $\kappa'' \in \text{LS}(\kappa)$) between them. Adversary encrypts only one plaintext C_1 and gets 2^8 ciphertexts for each collection of keys. In total we have $1 + 2^8 \cdot (2^8 - 1)$ different keys and different ciphertexts correspondingly. In the same collection we refer to the difference between i -th round keys K_i as ΔK_i , for example $\kappa = \Delta K_1$ and $\kappa'' = \Delta K_2$.

4.1. Integral property

Figure 1 shows the propagation of differences, which is true for only one collection of keys (namely, for such that $\kappa'' \in \text{LS}(\kappa)$). Active S-boxes have a gray background. Integral properties are indicated in bold (**A** denotes that all bytes are different, **0** – that bitwise XOR of all bytes is zero). More detailed pictures are presented in the Appendix B.

Let's see the key schedule. After the first $\text{LS}(C_1 \oplus K_1)$ transformation we have the difference $\kappa' \in \text{LS}(\kappa)$. This difference is the same for all collections of keys, but only for one collections the equality $\kappa'' = \kappa'$ is true. If so, we have $\Delta K_4 = \mathbf{0}$ and $\Delta K_3 = \kappa$. The difference κ has one active byte. In the difference $\kappa'' \in \text{LS}(\kappa)$ all bytes are active and have an integral property **A**.

Figure 1: Related-key attack on 5-round Kuznyechik

Similarly,

$$\Delta K_6 = \kappa''' \in \text{LS}(\kappa)$$

and

$$\Delta K_5 \in \text{LS}(\kappa''') \oplus \kappa = \{\delta \oplus \kappa, \delta \in \text{LS}(\kappa''')\}.$$

We use the equivalent representation of the last two rounds, therefore we consider difference between the keys $K'_5 = L^{-1}(K_5)$ and $K'_6 = L^{-1}(K_6)$ instead of K_5 and K_6 . Thus we find that $\Delta K'_6$ belongs to $S(\kappa)$ (correspondingly, $\Delta K'_5 \in \text{SLS}(\kappa) \oplus L^{-1}(\kappa)$).

All bytes of $\Delta K'_5$ are active and have an integral property **0** (see lemma). The difference $\Delta K'_6$ has one active byte with the property **A**.

Now let's consider the encryption functions. We use only one plain-text C_1 , therefore the difference ΔP_1 is equal to zero. Note that the first round of encryption also has the form $\text{LS}(C_1 \oplus K_1)$. Because of this, the difference between the blocks is also equal to κ' .

If in the key schedule $\kappa' = \kappa'' = \Delta K_2$ then the difference between the blocks becomes zero and also $\Delta P_3 = 0$.

The addition of the third round key K_3 adds the non-zero difference κ . We do not know the exact value of the difference ΔP_4 , but we know that ΔP_4 belongs to the set $\text{LS}(\kappa)$. Similarly, after the following substitution

layer, we have $\Delta P'_5 \in \text{SLS}(\kappa)$. All bytes of $\Delta P'_5$ have an integral property **A** and consequently property **0**. The second one is also true for $\Delta K'_5$. Therefore, their sum $\Delta P'_5 \oplus \Delta K'_5$ has an integral property **0**. The linear transformation preserves this one.

Obviously, we know the corresponding ciphertexts and the difference ΔB between them. The difference $\Delta K'_6 \in \text{S}(\kappa)$ has one active byte.

Let's propagate the difference through S^{-1} . For each of 15 S-boxes we have 2^8 possible differences and for 16'th S-box we get 2^{16} differences due to $\Delta K'_6 \in \text{S}(\kappa)$.

Let's check the integral property **0** for each obtained difference. If we correctly guessed κ'' , then there must be at least one such difference for each S-box. Otherwise, if we do not guess it correctly, then there is at least one S-box for which there is no such difference. Generally speaking, it is possible that a «false» collection of the related keys will satisfy this property. The probability of the existence of such «false» collection is approximately 0.23 (for more details see Appendix A). It does not lead to the failure of the attack. We will be able to distinguish this case through the next step.

We also expect that for each of 15 S-boxes about 2 differences have integral property **0**. For the last S-box about 2^8 differences have such property. Thus, the set $S^{-1}(\Delta B \oplus \text{S}(\kappa))$ will contain about $2^{15} \cdot 2^8 = 2^{23}$ possible differences each having the property **0**.

4.2. Recovery of the round keys

Let's consider the last linear transformation. We know that $\Delta P'_5 \in \text{SLS}(\kappa)$ and $\Delta K'_5 \in \text{SLS}(\kappa) \oplus L^{-1}(\kappa)$. The difference before the linear transformation is the sum

$$\begin{aligned} \Delta P'_5 \oplus \Delta K'_5 &\in (\text{SLS}(\kappa) \oplus \text{SLS}(\kappa) \oplus L^{-1}(\kappa)) \\ &= \{ \delta_1 \oplus \delta_2 \oplus L^{-1}(\kappa), \delta_1 \in \text{SLS}(\kappa), \delta_2 \in \text{SLS}(\kappa) \}. \end{aligned}$$

On the other hand we have the set of possible differences $S^{-1}(\Delta B \oplus \text{S}(\kappa))$ after the linear transformation.

The intersection of sets

$$(\text{SLS}(\kappa) \oplus \text{SLS}(\kappa) \oplus L^{-1}(\kappa)) \cap L^{-1}S^{-1}(\Delta B \oplus \text{S}(\kappa))$$

must contain at least one element. We use only one byte position to determine the inequality of elements from these two sets.

After checking the integral property in the set $L^{-1}S^{-1}(\Delta B \oplus \text{S}(\kappa))$ there will be about 2^{23} possible differences.

Recall that the set $S(\kappa)$ contains 2^8 elements. The linear transformation does not change the number of differences ($LS(\kappa)$ contains 2^8 elements). After another substitution layer we have 2^{16} possible differences at each S-box. The difference κ is known, therefore $L^{-1}(\kappa)$ is also known. Consequently, the set $SLS(\kappa) \oplus SLS(\kappa) \oplus L^{-1}(\kappa)$ contains

$$\frac{2^{16} \cdot (2^{16} - 1)}{2} + 1 < 2^{31}$$

possible differences at each S-box.

Select the position of one of the block bytes. Recall also that each difference contains $2^8 - 1$ vectors and consequently difference in one position contains $2^8 - 1$ bytes. We store in memory all possible differences from $SLS(\kappa) \oplus SLS(\kappa) \oplus L^{-1}(\kappa)$ for selected position. Let's iterate through all differences γ in $L^{-1}S^{-1}(\Delta B \oplus S(\kappa))$. If γ matches one of the stored differences, then we assume that $\gamma = \Delta P'_5 \oplus \Delta K'_5$. We can expect that γ is the only such element even if we compare on eight bytes of a difference rather than $2^8 - 1$. Note that if the collection of the related keys is «false» ($\kappa'' \neq \kappa'$), the match will probably not be found.

At this step we know $\Delta P'_5 \oplus \Delta K'_5$, $L(\Delta P'_5 \oplus \Delta K'_5)$, $\Delta P'_6$, $\Delta K'_6$, ΔB . Block

$$Y: S^{-1}(\Delta P'_6) = L(\Delta P'_5 \oplus \Delta K'_5)$$

may be easily found. Let B_0 be first ciphertext, then $K'_6 = B_0 \oplus Y$. The entire set of related keys K'_6 can also be obtained by summing with $\Delta K'_6$. Therefore, it is possible to decipher all 2^8 ciphertexts through the last round.

We know that $\Delta P_4 \in LS(\kappa)$, $\Delta K'_5 \in S(\Delta K'_6) \oplus L^{-1}(\kappa)$ and also the value $\Delta P'_5 \oplus \Delta K'_5$. Let's iterate through possible $\tau \in S(\Delta K'_6) \oplus L^{-1}(\kappa)$ and propagate $\Delta P'_5 \oplus \Delta K'_5 \oplus \tau$ through S^{-1} . If we guess $\tau = \Delta K'_5$, then $S^{-1}(\Delta P'_5 \oplus \Delta K'_5 \oplus \tau) = S^{-1}(\Delta P'_5) \in LS(\kappa)$. Otherwise, we expect that $S^{-1}(\Delta P'_5 \oplus \Delta K'_5 \oplus \tau) \notin LS(\kappa)$. In the matching process, each S-box may be considered independently of the others. After that we will know the differences ΔP_4 , $\Delta P'_5$, $\Delta K'_5$. The ciphertexts after 5-th round are also known. Therefore, the keys K'_5 may be found in the same way as K'_6 . Due to the reverse key schedule, the master key $K = K_1 || K_2$ may be obtained easily.

4.3. Complexity

As mentioned before, the attack requires $1 + 2^8 \cdot (2^8 - 1) < 2^{16}$ related keys and one chosen ciphertext.

The integral property for all 2^8 related key collections may be checked in about $2^8 \cdot (15 \cdot 2^8 + 2^{16}) \approx 2^{24}$ operations.

The most time-consuming stage is the construction of the set $\text{SLS}(\kappa) \oplus \text{SLS}(\kappa) \oplus L^{-1}(\kappa)$. We construct this set for only one S-box, and store only eight bytes for each difference. It requires about 2^{31} operations and $2^{31} \cdot 8 = 2^{34}$ bytes of memory. These constructed differences are stored in a hash table. The set $L^{-1}S^{-1}(\Delta B \oplus S(\kappa))$ contains much fewer elements. Checking for a single element in a hash table requires constant time. Therefore, the complexity of constructing the hash table will be the most important. The difficulty of recovery the keys K'_5 is also small: $16 \cdot 2^8 \cdot 2^8 + 2^8 \approx 2^{20}$ operations.

The total complexity does not exceed 2^{32} memory access operations and 2^{30} memory (in sixteen-byte blocks). We also note that the attack is deterministic.

We had modeled the attack with a non-optimized C++ implementation. The average attack time was about 5 minutes on a common PC. The amount of used memory did not exceed 17 GB.

5. Conclusion

In this paper we present the related-key attack on 5-round Kuznyechik with 2-round key schedule. The attack has a practical complexity (2^{32} operations, 2^{30} memory, 2^{16} related keys) and has been verified by means of C++ implementation. The experiments had confirmed the correctness of the attack.

Source codes may be found at <https://gitlab.com/v.kir/rk-5R-kuznyechik>.

The main result was achieved by means of the well-known integral property of LSX-transformations. We were able to use this property both in the cipher itself and in the key schedule.

We did not use any specific properties of the linear transformation and the S-box. The exception is to use the maximum branch number of the linear layer. We think that through the use of such properties it is possible to obtain new results. Another possible way is the use of integral distinguishers for a larger number of rounds.

The presented attack also shows a significant security margin of the Kuznyechik's key schedule.

Acknowledgement: The author is grateful to Sergey Svetlov for help with experiments and verification, to Anton Naumenko and Igor Arbekov for support and valuable comments.

References

- [1] *GOST R 34.12-2015. National standard of the Russian Federation. Information technology Cryptographic data security Block ciphers*, 2015 (in Russian).
- [2] Alekseev E., Goncharenko K., Marshalko G., “Provably secure counter mode with related-key-based internal re-keying”, *J. Comput. Virol. Hack. Tech.*, 2020, <https://doi.org/10.1007/s11416-020-00357-9>.
- [3] Tiessen T., “Polytopic cryptanalysis”, *EUROCRYPT 2016, Part I, Lect. Notes Comput. Sci.*, **9665**, 2016, 214–239.
- [4] Daemen J., Knudsen L., Rijmen V., “The block cipher Square”, *FSE 1997, Lect. Notes Comput. Sci.*, **1267**, 1997, 149–165.
- [5] Barreto P., Rijmen V., “The Khazad legacy-level block cipher”, *First open NESSIE Workshop*, 2000, Submission to NESSIE.
- [6] Biham E., “New types of cryptanalytic attacks using related keys”, *EUROCRYPT 93, Lect. Notes Comput. Sci.*, **765**, 1993, 398–409.

A. Probability aspects and experimental verification

«True» and «false» collections of related keys

We know that there is at least one «true» collection. What is the probability that the integral property (section 4.1) will be correct for the «false» collection?

Assume that all ciphertexts are equally probable and independent of each other. We propagate the difference of each S-box through the nonlinear layer. For each of 15 S-boxes we'll have 2^8 possible differences and for 16-th S-box we get 2^{16} differences. We also assume that the sum of the elements of any difference is uniformly distributed. Hence, the probability of the property **0** is equal to $p = \frac{1}{256}$ for each difference of any S-box. Denote the probability of the opposite event by $q = 1 - p = \frac{255}{256}$.

Thus, we have:

$q^{2^8} = 0.367\dots$ is the probability that no difference has the property **0** for one S-box,

$1 - q^{2^8} = 0.632\dots$ is the probability that there is at least one such difference,

$(1 - q^{2^8})^{15} = 0.001\dots$ is the probability that there is at least one such difference for each of the 15 S-boxes.

The probability that one collection of the related keys has the integral property is

$$r = \left(1 - q^{2^8}\right)^{15} \cdot \left(1 - q^{2^{16}}\right) = 0.001\dots$$

We have 2^8 collections of keys and only one «true» collection. The probability that «false» collections do not exist is

$$(1 - r)^{255} = 0.765\dots$$

The probability of the opposite event is

$$1 - (1 - r)^{255} = 0.234 \dots$$

We have performed $N = 5000$ experiments. The number of cases where «false» collections were obtained was equal to 1179. The corresponding frequency $\frac{1179}{5000} = 0.236$ is close to the theoretical estimate.

Number of possible differences

Let we have «true» collection of the related keys. Now we estimate the number of possible differences in the set $L^{-1}S^{-1}(\Delta B \oplus S(\kappa))$.

Each S-box gives at least one possible difference. The probability of the property **0** is equal to $p = \frac{1}{256}$ for each difference of any S-box. We also have 2^8 possible differences for each of 15 S-boxes and 2^{16} for 16-th S-box.

Thus, the average number of elements in the set is equal to

$$\left(1 + \frac{1}{256} \cdot (2^8 - 1)\right)^{15} \cdot \left(1 + \frac{1}{256} \cdot (2^{16} - 1)\right) \approx 2^{23} \ll 2^{31}.$$

The average experimental value is $2^{22.7}$. The maximum value among all N experiments was 2^{29} .

Matching differences

The intersection of sets

$$(SLS(\kappa) \oplus SLS(\kappa) \oplus L^{-1}(\kappa)) \cap L^{-1}S^{-1}(\Delta B \oplus S(\kappa))$$

must contain at least one element. We use only one position to determine the inequality of elements from these two sets.

One position of the first set contains no more than 2^{31} differences. The number of elements in the second set is approximately 2^{23} . We also assume that the elements of these sets are random and equiprobable.

Only the first 8 bytes (64 bits) of the difference are stored in memory. Then the average number of «false» matches may be estimated as

$$\frac{2^{31} \cdot 2^{23}}{2^{64}} = 2^{-10}.$$

A «false» match may be easily detected by an additional check. In $N = 5000$ experiments, we got only 7 cases of it.

Eight-byte numbers were chosen for ease of implementation.

B. Detailed pictures

Figure 2: The difference propagation through the key schedule

Figure 3: The difference propagation through the cipher