VERSITA

TATRA MOUNTAINS
Mathematical Publications

DOI: 10.2478/tmmp-2013-0032
Tatra Mt. Math. Publ. **57** (2013), 1–19

ROTATIONAL CRYPTANALYSIS OF GOST WITH IDENTICAL S-BOXES

Pavol Zajac — Michal Ondroš

ABSTRACT. Rotational cryptanalysis was introduced by Khovratovich and Nikolić as a tool to analyse ARX-type cipher designs. GOST 28147-89 is a former Soviet Union cipher standard based on a Feistel construction with 32 rounds. Each round function adds the round key modulo 2^{32}, transforms the result with 4-to-4 bit S-boxes, and rotates the output. We apply the rotational cryptanalysis to a version of GOST using eight identical S-boxes, such as GOST-PS. We show the existence of (practical) rotational distinguisher in related key model for full GOST. Furthermore, there is a set of weak keys (rotationally symmetric keys) that enables rotational attacks in single-key model as well. Finally, we show a simple attack on the last round that uses the rotational distinguisher to reduce the complexity of the full GOST to 208 bits.

1. Introduction

GOST 28147-89 is a former Soviet Union cipher standard [12]. It has a 64-bit block, and 256-bit key. Due to its simple structure it is very suitable for lightweight implementations. Until GOST was submitted to ISO for standardization in 2010, the published cryptanalytic results about GOST were very sparse. However, after getting a stronger attention of professional cryptographers, many new results that show various weaknesses of the algorithm have been published [1–4, 6]. The best attacks are still impractical, but they significantly reduce the expected security level of the cipher.

In our recent analysis [13], we have focused on the resistance of GOST against generic algebraic attacks. In this article we focus on the rotational cryptanalysis of GOST. Rotational cryptanalysis is a tool introduced by Khovratovich and Nicolić in 2010 [7] to study ARX designs submitted to SHA-3 competition.

2010 Mathematics Subject Classification: 94A60, 62P99.
Keywords: rotational cryptanalysis, GOST.
This research was supported by grant VEGA 1/0173/13, APVV-0586-11, and by NATO's Public Diplomacy Division in the framework of "Science for Peace", SPS Project 984520.

Although GOST is not an ARX design, we can transform it to ARX construction by removing S-boxes. Moreover, if all eight S-boxes in GOST realize the same permutation, GOST's round function becomes almost rotationally invariant (except the key addition). The results obtained from analyzing GOST without S-boxes then apply also to the case with a single type of S-box for rotational amounts that are multiples of 4.

We analyse the security of the selected variants of GOST in the related key model. Our model requires that each of eight 32-bit subkeys of the key in two instances form rotational pairs. The probability of obtaining a valid pair of related keys in practice is very small, but the results can be exploited by attacks on hash modes of GOST, when the attacker controls the key as well. Moreover, the results also hold for a class of keys that are rotationally self-similar, i.e., these keys that remain the same after rotating its 32-bit parts (using the same rotational amount). If a key from this class is used, it can be detected by the attacker using the rotational distinguisher.

In the first part of the article we summarize the preliminaries: we present in more details GOST, and rotational cryptanalysis, respectively. Our analysis is covered in Section 3, which is split into three principal parts. In Subsection 3.1 we start with the analysis of a generic model without S-boxes, and with independent subkeys. We focus on properties of Feistel networks which have rotationally distinguishable round functions. In the following Subsection 3.2, we present our main result. We derive the formula for rotational probability of the scheme with reused subkeys. This probability is significantly higher than in the case when encryptions are completely independent. Finally, in Subsection 3.3 we apply the results to GOST with eight S-boxes realizing the same permutation. In the final section, we summarize the results, and present some recommendations for GOST implementers.

2. Preliminaries

In this section we summarize the basic notions about rotational cryptanalysis, and about the block cipher GOST, along with our notation used in the rest of the article.

2.1. Rotational cryptanalysis

Let us consider an algorithm that processes bit vectors of fixed length $n > 0$. Let $r > 0$ be an integer, and let a, b denote dually two n-bit unsigned integers, as well as their n-bit vector representation (in base-2, big endian).

ARX encryption scheme consists of only three types of operations:

1. **A**ddition, ADD — addition of two arguments as integers without carry (modulo 2^n), denoted by $a + b$;
2. **R**otation, ROT — rotation of the (n-bit vector) argument by a specified amount r, denoted by $a \lll r$;
3. **X**OR — bitwise modulo 2 addition of the two bit-vector arguments, denoted by $a \oplus b$;

Let $hi_r(a) = \lfloor a/2^{n-r} \rfloor$ denote the first (most significant) r bits of a, and let $lo_r(a) = a \mod 2^{n-r}$ denote the remaining bits of a. The rotation can be expressed as

$$a \lll r = lo_r(a)2^r + hi_r(a)\,,$$

i.e., the upper and lower part of a are swapped. It is also possible to define a complementary operation $\ggg$, that is a rotation in the opposite direction (towards the least significant bit). This operation is however unnecessary as $a \ggg r = a \lll (n - r)$. If r is a fixed constant, we will call it a rotational amount. In this case we will also simplify the notation by writing $\overleftarrow{a} = a \lll r$. A pair of vectors $(a, \overleftarrow{a})$ is called a rotational pair.

Rotational cryptanalysis was introduced by K h o v r a t o v i c h and N i c o l i ć in 2010 [7], as a tool to analyse ARX schemes. Rotational cryptanalysis studies the (rotational) response of the (ARX) encryption scheme to the rotation of inputs. We suppose that the attacker encrypts rotational pairs of inputs (all input n-bit vectors are rotated by the rotational amount), and then observes the (statistical) properties of the corresponding pairs of outputs (considered as sets of n-bit vectors).

The encryption scheme is secure only if the attacker cannot distinguish, whether the outputs were produced by the encryption scheme, or by a random permutation (or a random function, depending on the model). In the ideal case, the outputs of two encryptions (with rotated inputs) must be independent, and taken uniformly from a set of all possible vectors. For each output vector there is a single rotated output vector (we remark that rotational amount r is fixed). Therefore a pair of outputs, in ideal case, is a rotational pair with the probability 2^{-n}.

The rotational cryptanalysis is particularly useful against ARX schemes. The main observation is that operations ROT and XOR preserve rotational pairs, i.e., if all inputs of these operations are rotational pairs (input-wise), then also the outputs form a rotational pair. The operation ADD preserves a rotational pair with a relatively high probability (depending on the rotational amount) [7]

$$p_r = Pr(\overleftarrow{x + y} = \overleftarrow{x} + \overleftarrow{y}) = 1/4(1 + 2^{r-n} + 2^{-r} + 2^{-n})\,. \tag{1}$$

Therefore, if we analyze ARX schemes with a low number of additions, we may find that the expected probability of rotational outputs is higher than in the ideal case.

Another observation used in [7], and even more in [8], is that even if the addition does not preserve the rotational pair, the rotational error (or difference)

$$(\overleftarrow{x+y}) \oplus (\overleftarrow{x} + \overleftarrow{y})$$

has a relatively low Hamming weight (with high probability). Rotational errors can also be caused by the addition (or xor-ing) of constants in the encryption scheme. The full rotational analysis of the ARX scheme with constants tries to compensate these rotational errors against each other. We remark that in our analysis we do not study rotational errors. The actual attacks that take rotational errors into account might be stronger than the attacks presented in this paper.

2.2. GOST

Block cipher GOST [5,12] is a Feistel cipher with 32 rounds. Its block size is 64 bits, and key-size is 256 bits. Let L^i, R^i denote two 32-bit halves of the input block of the ith round, $i \geq 1$, and let K^i denote the 32-bit expanded key for the ith round. The input of the next round is computed as

$$\begin{aligned} L^{i+1} &= R^i, \\ R^{i+1} &= L^i \oplus \left(S(R^i \boxplus K^i) \lll 11\right), \end{aligned}$$

where $\oplus$ denotes XOR (modulo 2 addition of individual bits), $\boxplus$[1] denotes addition modulo 2^{32}, $\lll$ denotes the (left) bit rotation, and S denotes the application of 8 parallel 4-to-4-bit S-boxes. The scheme of the encryption is depicted in Figure 1.

The key expansion of GOST is very simple: 256-bit key is split into eight 32-bit words $K^0, \ldots, K^7$. Subkeys for rounds $1-24$ are simply $K^r = K^{(r-1 \mod 8)}$, where r denotes the round number. For the last 8 rounds, the parts of the key are used in the reverse direction, i.e., $K^r = K^{(32-r \mod 8)}$.

The GOST standard [12] does not prescribe a fixed set of S-boxes (although some default sets were published in [10]). If we replace all S-boxes by identity, we get an ARX scheme without constants, and with a relatively low number of additions. We can thus directly apply the techniques of rotational cryptanalysis to study the properties of this encryption scheme. We show, that the probability to obtain rotational pairs of outputs cannot be directly computed from the number of addition, but it is higher due to the effects of the GOST's weak key schedule.

[1] We use the $\boxplus$ notation in this section only to maintain correspondence with Figure 1, otherwise we will use just $+$.

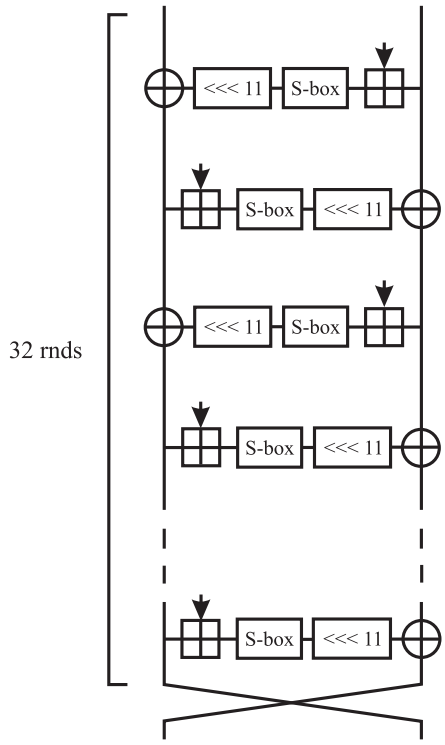

FIGURE 1. The scheme of the block cipher GOST.

In Section 3.3 we also study the case when all eight S-boxes realize the same permutation. Although this is not (directly) an ARX scheme, the rotational pairs on the input of the S-box layer are preserved, if the rotational amount is a multiple of 4.

3. Rotational cryptanalysis of GOST with and without S-boxes

Let us first consider a modification of GOST with omitted S-boxes (each S-box is replaced by identity mapping). This is an ARX design, with the only source of non-linearity provided by key-additions in round functions. The key addition is also the only source of rotational errors.

We work in a model with related keys. Rotational pairs are defined over sets of 32-bit vectors that consist of two halves of the input block, two halves of the output block, and 32-bit parts of the key, respectively. The attacker encrypts plaintext $P = (P_L|P_R)$, with an unknown key $K = (K^0|K^1|\cdots|K^7)$, [2] and obtains ciphertext $C = (C_L|C_R)$. The attacker can then encrypt the rotated plaintext $\overleftarrow{P} = (\overleftarrow{P_L}|\overleftarrow{P_R})$, using a related (but still unknown) key $\overleftarrow{K} = (\overleftarrow{K^0}|\overleftarrow{K^1}|\cdots|\overleftarrow{K^7})$, obtaining ciphertext $C' = (C'_L|C'_R)$. The cipher preserves the input rotational pair if $C' = \overleftarrow{C}$, that is if $C'_L = \overleftarrow{C_L}$, and $C'_R = \overleftarrow{C_R}$.

For an ideal cipher with encryption function E, the cipher outputs are essentially independent random values. Thus $p_r(E) = Pr(\overleftarrow{E_K(X)} = E_{\overleftarrow{K}}(\overleftarrow{X}) = 2^{-n}$, as there is a single n-bit value which is a rotation of $Y = E_K(X)$. Similarly, if the output block is divided into m n-bit vectors, then $Pr(\overleftarrow{E_K(X)} = E_{\overleftarrow{K}}(\overleftarrow{X}) = 2^{-mn}$.

In this section we study the probability, that GOST preserves rotational pairs (for a fixed rotational amount r), that is

$$p_r(GOST) = Pr\big(\overleftarrow{GOST_K(P)} = GOST_{\overleftarrow{K}}(\overleftarrow{P})\big),$$

under a uniformly random choice of K, and P. We show that this probability is significantly higher than 2^{-64} (the ideal case), if all S-boxes of GOST realize the same permutation.

3.1. Random subkeys

Let us consider encryption of a rotational pair $(L|R)$, $(\overleftarrow{L}\,|\,\overleftarrow{R})$ by a Feistel scheme with independent round functions F. Let us denote the probability that the round function preserves a rotational pair constant for each round by $p_r(F)$ (if r is understood, we can omit it). We first consider that these probabilities are independent of each other. This is a reasonable approximation, if we consider that round function depends in a complex way on subkeys, and if these subkeys are independently chosen from the set of all possible subkeys. This condition holds if we consider the first eight rounds of GOST, and randomly chosen key, or if we imagine a GOST-like cipher with a more complex key schedule. However, in the real GOST this condition is violated due to a subkey reuse. We study the real situation in more details in the next section (Section 3.2).

In the Feistel structure of GOST, all operations outside the round function (XOR of the output of the round function, swap) preserve rotational pairs. This leads to a generic property of a Feistel cipher, with round function F, that the rotational probabilities depend only on the probability $p_r(F)$. We do not provide an exact theorem, but we can summarize this result in a form of the following Proposition.

[2] We abuse the notation slightly by defining rotated version of a set of vectors, e.g., if $X = (X_1|X_2)$, than $\overleftarrow{X} = (\overleftarrow{X_1}, \overleftarrow{X_2})$.

Proposition 1. *Let $F : (\mathbb{Z}_2^n)^2 \to \mathbb{Z}_2^n$. Let $p_r(F)$ denote a conditional probability that output of F is rotational, if the input is rotational. That is,*

$$p_r(F) = Pr(\overleftarrow{F(X)} = F(\overleftarrow{X})).$$

Let us construct a Feistel cipher E_K with N rounds and round function F with input (L_0, R_0), and set of round keys $K = (K^0, K^1, \ldots, K^{N-1})$ as follows:

$$L^{i+1} = R^i,$$
$$R^{i+1} = L^i \oplus F(R^i, K^i).$$

Let round keys K^i be considered stochastically independent random values. Let

$$(L, R) = E_K(L^0, R^0), \quad \overleftarrow{K} = \big(\overleftarrow{K^0}, \overleftarrow{K^1}, \ldots, \overleftarrow{K^{(N-1)}}\big), \quad (L^*, R^*) = E_{\overleftarrow{K}}(\overleftarrow{L^0}, \overleftarrow{R^0}).$$

Let $p_r(F) \gg 2^{-n}$. Then output of the Feistel cipher is rotational with probability approximately $(p_r(F))^N$, i.e.,

$$p_r(E_K) = Prob\big((L^*, R^*) = (\overleftarrow{L}, \overleftarrow{R})\big) \approx \big(p_r(F)\big)^N.$$

This also means, that if the output of the Feistel cipher is rotational, then each output of F is rotational as well with very high probability.

As the Proposition 1 is not exact, we do not provide a proof, only a sketch. Let us start with the first two rounds. The input of the cipher consists of two rotational pairs in each half. In the first round, one half of the input is unchanged (but swapped), the second half remains rotational with probability $p = p_r(F)$. XOR operation preserves the rotational pair, if it was preserved by F. Now the inputs to the round function in the second round (i.e., $R^1 = L \oplus F(R)$) form a rotational pair only with the probability p. We suppose that the second execution of F is independent of the first one (due to independently chosen random subkey). Thus the output of the round function, as well as the new right half of the state (R^2), form rotational pairs with probability p^2.

After the third application of a round function, $F(R^2)$'s form a rotational pair with the probability p^3. We XOR it with $L^2 = R^1$, which form a rotational pair only with the probability p. If $F(R^2)$, and L^2 were independent, the rotational probability for R^3 would be p^4. However, if R^1's do not form a rotational pair, the probability that R^2's, and consequently $F(R^2)$'s form rotational pairs, is negligible (we suppose that $p \gg 2^{-n}$). On the other hand, if R^1 is rotational, the conditional probability that $F(R^2)$ is rotational becomes p^2. Thus the probability that R^3 is rotational is p^3.

Similarly, for any pair R^i, $\bar{R}^i$, where R^i is a right-hand part of the state after i round of encrypting $(L|R)$, and $\bar{R}^i$ is a right-hand part of the state after i round of encrypting $(\overleftarrow{L}\,|\,\overleftarrow{R})$, we can show that $Pr(\bar{R}^i = \overleftarrow{R^i}) \approx p^i$. We stress, that this

is not an exact value, as we ignore the situation, when rotational pair can arise from a non-rotational pair. We can do this if $p_r(F) \gg 2^{-n}$, which is a typical situation in practice. On the other hand, the situation when $p_r(F)$ is near 2^{-n}, is not suitable for rotational cryptanalysis.

Let us apply Proposition 1 to simplified GOST without S-boxes, and with a random selection of all subkeys. In this case $p_r(F)$ is exactly p_r from equation (1), as F consists only of a single addition, and a single rotation (which preserves rotational pairs with probability 1). If $r = n/2$, we get $p_r(F) \approx 1/4$, and with 32 applications of F the reduced GOST cannot be reliably distinguished from the ideal cipher, using the rotational characteristics. On the other hand, for any smaller rotational amount r, we get a rotational distinguisher. For example, for $r = 1$, we get $p_r(F) \doteq 0.375$, which means that for 32 rounds we get $\big(p_r(F)\big)^{32} \doteq 2^{-45} \gg 2^{-64}$. In this case at least 46 rounds are required to hide rotational properties of the simplified GOST.

3.2. Influence of the key schedule

Key schedule of the GOST is very simple: The 256-bit key is split into eight 32-bit subkeys $(K^0, K^1, \ldots, K^7)$, which are applied in the same order three times in the first 24 rounds, and then in the reverse order in the last 8 rounds. That is, the whole expanded key is a sequence of 32-bit words

$$\big(K^0, K^1, \ldots, K^7, K^0, \ldots, K^7, K^0, \ldots, K^7, K^7, K^6, \ldots, K^0\big)\,.$$

Unlike the case discussed in Section 3.1, the subkeys are not independent. If we suppose that key is chosen randomly, we can consider each of $K^0, K^1, \ldots, K^7$ independent random 32-bit vectors, but each of them is repeated four times (up to four times, if the number of rounds is reduced). In this section we focus on the probabilities of preserving (randomly, and independently chosen) rotational inputs, under repeated addition of the same subkey.

More formally: Let K, and $X_1, X_2, \ldots, X_t$, be random n bit vectors. Let $\overleftarrow{K} = K \lll r$, and $\overleftarrow{X}_i = X_i \lll r$, be their rotational counterparts. We want to compute probability π_t, that $\overleftarrow{K + X_i} = \overleftarrow{K} + \overleftarrow{X}_i$, for each $i = 1, 2, \ldots, t$.

Let us denote the first r bits of the n-bit vector X by $hi(X)$ and the remaining $n - r$ bits by $lo(X)$. Equality $\overleftarrow{K + X_i} = \overleftarrow{K} + \overleftarrow{X}_i$ holds iff $hi(K) + hi(X_i) < 2^r$, and $lo(K) + lo(X_i) < 2^{n-r}$. Otherwise, carry bits from the addition of $hi(X)$, and $lo(X)$, respectively, are not preserved by the rotation. If K is fixed, the probability that this equality holds for a random choice of X_i is

$$\big(2^r - hi(K)\big) \cdot \big(2^{n-r} - lo(K)\big) \cdot 2^{-n}.$$

We suppose that X_i's are chosen independently[3]. Thus the probability π_t given fixed K, and t values X_i can be computed as

$$\pi_t(K) = \left(2^r - hi(K)\right)^t \cdot \left(2^{n-r} - lo(K)\right)^t \cdot 2^{-nt}. \tag{2}$$

Now we can also compute the probability for a random selection of K as

$$\pi_t = 2^{-n} \sum_{K=0}^{2^n-1} \pi_t(K),$$

or equivalently as

$$\pi_t = 2^{-n(t+1)} \sum_{K=0}^{2^n-1} \left(2^r - hi(K)\right)^t \cdot \left(2^{n-r} - lo(K)\right)^t. \tag{3}$$

Using $K = hi(K) \cdot 2^{n-r} + lo(K)$, we can rewrite this as

$$\begin{aligned}
\pi_t &= 2^{-n(t+1)} \sum_{hi(K)=0}^{2^r-1} \sum_{lo(K)=0}^{2^{n-r}-1} \left[\left(2^r - hi(K)\right)^t \cdot \left(2^{n-r} - lo(K)\right)^t\right] \\
&= 2^{-n(t+1)} \sum_{hi(K)=0}^{2^r-1} \left[\left(2^r - hi(K)\right)^t \cdot \sum_{lo(K)=0}^{2^{n-r}-1} \left(2^{n-r} - lo(K)\right)^t\right] \\
&= 2^{-n(t+1)} \left[\sum_{hi(K)=0}^{2^r-1} \left(2^r - hi(K)\right)^t\right] \cdot \left[\sum_{lo(K)=0}^{2^{n-r}-1} \left(2^{n-r} - lo(K)\right)^t\right] \\
&= 2^{-n(t+1)} \left(\sum_{x=1}^{2^r} x^t\right) \cdot \left(\sum_{y=1}^{2^{n-r}} y^t\right).
\end{aligned}$$

After adapting the notation $S_t(n) = \sum_{x=1}^{n} x^t$, we finally get

$$\pi_t = 2^{-n(t+1)} S_t(2^r) \cdot S_t(2^{n-r}). \tag{4}$$

The exact formulas for individual sums $S_t(n)$ with $t = 1, 2, 3, 4$ are summarized in Table 1. Although the fully evaluated formulas are quite complicated, for the purposes of the cryptanalysis we can simplify them by omitting all exponential terms lower than 2^{-r}, or 2^{r-n}. The approximate probabilities are summarized in Table 1. In the case, when $r < n/2$, we can omit also the term 2^{r-n}. Similarly, if $r > n/2$, we can omit terms with 2^{-r}. We remark that both the simplified, and the fully expanded formulas, do not depend on the direction of the rotation (for any t), i.e., they are symmetric in r, and $n - r$, respectively.

[3]In GOST, the X_i's which are added to the same subkey value, are not really independent. On the other hand, they are separated by 8 rounds of Feistel network (expect for the specific last round), and thus it is reasonable to treat them as if they were independent.

TABLE 1. Sums $S_t(n)$, and approximate values of π_t, for $t = 1, 2, 3, 4$.

t	$S_t(n)$	π_t (approx.)
1	$(n^2 + n)/2$	$1/4 + 1/4 \cdot 2^{-r} + 1/4 \cdot 2^{r-n}$
2	$(2n^3 + 3n^2 + n)/6$	$1/9 + 1/6 \cdot 2^{-r} + 1/6 \cdot 2^{r-n}$
3	$(n^4 + 2n^3 + n^2)/4$	$1/16 + 1/8 \cdot 2^{-r} + 1/8 \cdot 2^{r-n}$
4	$(6n^5 + 15n^4 + 10n^3 - n)/30$	$1/25 + 1/10 \cdot 2^{-r} + 1/10 \cdot 2^{r-n}$

Let us extend the results to (simplified) GOST. According to Section 3.1, the probability of GOST output being rotational is given as a product of rotational probabilities for the round function F under the condition of the independence of their execution. In the first 8 rounds, we consider the subkeys to be independent 32-bit random values, and we can approximate the rotational probabilities for the 8-round GOST by multiplying 8 rotational probabilities for the round function.

Let us consider the addition of the 9th round. Subkey K^0 is repeated. If the output of the 8-round GOST is rotational, with probability near to one also the output of the first round function (using K^0) is rotational. In our model, the only obstacle to output F being rotational is the key addition, thus the we know that $\overleftarrow{R^0 + K^0} = \overleftarrow{R^0} + \overleftarrow{K^0}$. We now ask whether again $\overleftarrow{R^7 + K^0} = \overleftarrow{R^7} + \overleftarrow{K^0}$. Due to 7 rounds of encryption with independent subkeys, we can consider R^7 to be random value independent from R^0. Equivalently, we can ask what is the probability, that both equations $\overleftarrow{R^0 + K^0} = \overleftarrow{R^0} + \overleftarrow{K^0}$, $\overleftarrow{R^7 + K^0} = \overleftarrow{R^7} + \overleftarrow{K^0}$, hold. This is exactly π_2 derived in the previous part of this section.

Let us have a closer look at 9-round GOST in our model. The rotational probability for full GOST is the same as the probability that each $\overleftarrow{R^i + K^i} = \overleftarrow{R^i} + \overleftarrow{K^i}$, for $i = 0, 1, \ldots, 7$, as well as $\overleftarrow{R^7 + K^0} = \overleftarrow{R^7} + \overleftarrow{K^0}$, hold. Under the condition that K^i's are independent random values we can estimate this probability as $\pi_1^7 \cdot \pi_2$. That is, seven subkeys are added once, and one subkey is added twice, and each time the addition of rotated inputs must produce rotated outputs.

We can analyse the other cases in a similar way. The results are summarized in Table 2. We remark that these results are approximations based on the specific assumptions used (independence of inputs, ignoring the probability that a rotational pair can be produced from a non-rotational input, approximation of $S_t(n)$ formulas). However, as shown in later sections, the experimental data provide a supporting evidence that the approximate results can be good enough in practice.

TABLE 2. Rotational probabilities for N-round simplified GOST (without S-boxes).

N	Formula for π	$-\log_2 \pi$		
		$r = 1$	$r = 4$	$r = 16$
1	π_1	1.4	1.9	2.0
2	π_1^2	2.8	3.8	4.0
⋮				
8	π_1^8	11.3	15.3	16.0
9	$\pi_1^7\pi_2$	12.3	16.4	17.2
10	$\pi_1^6\pi_2^2$	13.2	17.6	18.3
⋮				
16	π_2^8	18.9	24.3	25.4
17	$\pi_2^7\pi_3$	19.5	25.1	26.2
18	$\pi_2^6\pi_3^2$	20.2	25.9	27.0
⋮				
24	π_3^8	24	30.6	32.0
25	$\pi_3^7\pi_4$	24.5	31.2	32.6
26	$\pi_3^6\pi_4^2$	24.9	31.8	33.3
⋮				
32	$\pi_3\pi_4^7$	27.3	34.9	36.5
32	π_4^8	27.8	35.5	37.2

Table 2, and Figure 2, summarize the results for selected rotational amounts. Considering the estimated results, we can see that GOST's key schedule have the most adverse effect on the rotational characteristics of (a simplified) GOST, as even for the worst rotational amount $r = n/2$, we get a strong distinguisher of a simplified GOST.

Figure 3 compares experimental results with theoretical estimates, both in the model with independent subkeys, and with GOST's key schedule. The experimental data were computed from a dataset obtained using 10^9 random plaintexts and keys. The differences in the tail part of the chart are on the order of magnitude of the statistical error (for $r = 16$ there is not enough data to get any rotational pair for more rounds than 22).

3.3. Repeated S-box

The specification of GOST does not prescribe any particular set of S-boxes [5]. There are eight 4×4 S-boxes that need to be specified as an additional parameter (e.g., [10]). To simplify the implementation, we might consider to use only a

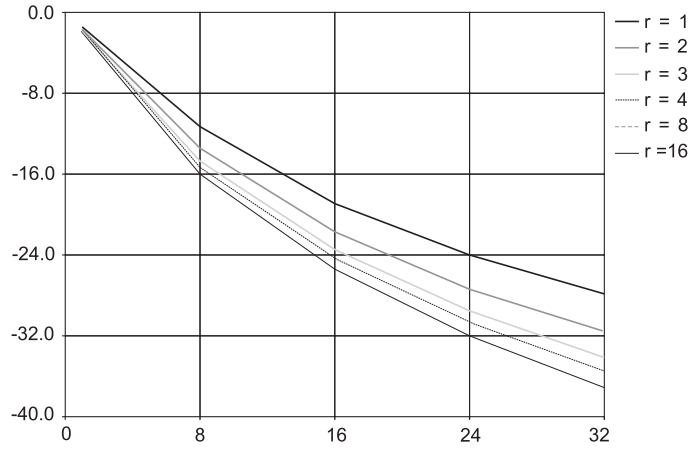

FIGURE 2.
Base-2 logarithm of theoretical rotational probability
for $r = 1, 2, 3, 4, 8$, and 16, as a function of the number of rounds.

single S-box eight times, instead of eight different S-boxes. This was already exploited in the design of lightweight version GOST-PS [11], where all S-boxes are taken from cipher PRESENT. In this section, we show that this selection unfortunately leads to rotational attacks.

Let us consider rotational pairs of inputs to GOST's round function F with $r = 4$, i.e., let $\overleftarrow{x} = x \lll 4$, as well as $\overleftarrow{k} = k \lll 4$. If the rotational pair of inputs is preserved by the key addition, the same S-boxes are applied to each 4-bit substring of either $x + k$, and $\overleftarrow{x} + \overleftarrow{k}$. Thus

$$Pr\big(\overleftarrow{F(x)} = F(\overleftarrow{x})\big) = Pr(\overleftarrow{x + k} = \overleftarrow{x} + \overleftarrow{k}).$$

Similar observation can be made for any rotation size that is a multiple of 4. Note however, that this does not hold when S-boxes are different, as individual S-boxes transform different parts of $(x + k)$, and $(\overleftarrow{x} + \overleftarrow{k})$, respectively.

If we only consider probability of rotational pairs, we get the same results as in the case where no S-boxes are used. However, the S-boxes increase the diffusion, so if the key addition breaks the rotational pair, their difference spreads faster, and we expect that the distribution of rotational errors, i.e., $w_H(y \oplus \overleftarrow{y})$, is nearer the expected distribution from ideal cipher.

Figure 4 compares experimental results with, and without S-boxes, respectively, with theoretical estimates. The experimental data were computed from a dataset obtained using 10^9 random plaintexts and keys. The differences in the tail part of the chart are on the order of magnitude of the statistical error.

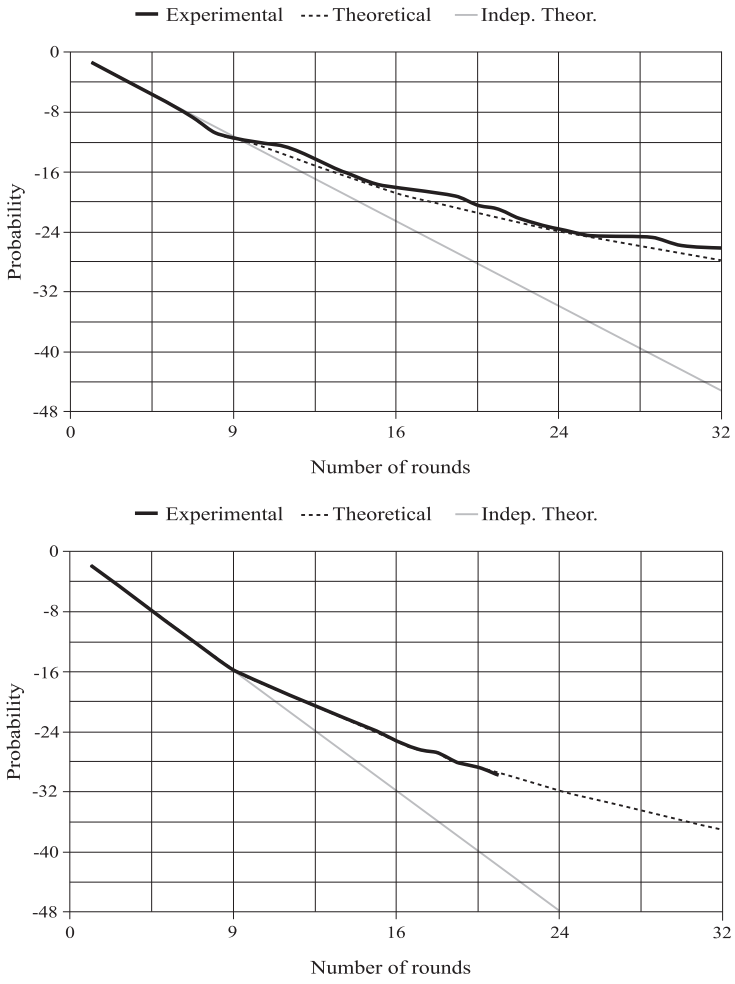

FIGURE 3.

Base-2 logarithm of rotational probability for $r = 1$ (on the top), and $r = 16$ (at the bottom), as a function of the number of rounds.

3.4. A note on weak keys

Instead of a related key model, we can consider a situation, where $K = \overleftarrow{K}$, i.e., when the key is chosen in such a way that each subkey is rotationally symmetric. All results from the related key model hold also for this special class of keys. Rotationally symmetric keys can thus be considered as weak keys for the GOST with a single S-box.

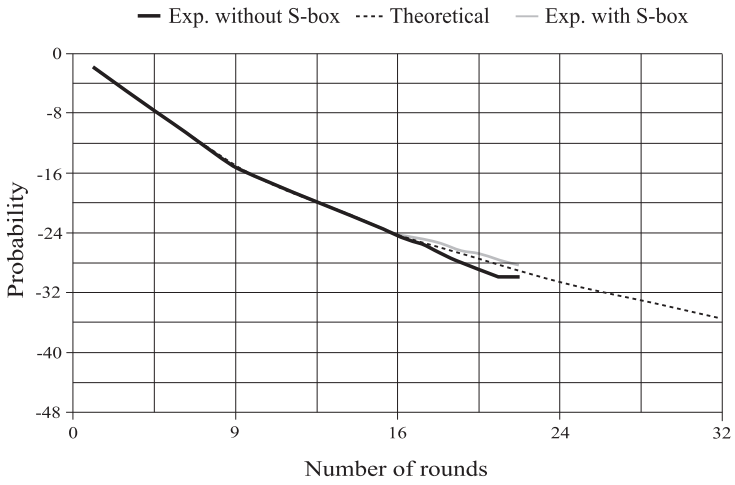

FIGURE 4. Base-2 logarithm of rotational probability (for $r = 4$, theoretical, and experimental with and without S-boxes) as a function of the number of rounds.

The size of the set of weak keys depends on the rotational amount. The number of rotationally symmetric n-bit vectors is

$$N_r = \#\{X = x \cdot 2^{n-r} + y; x(2^{n-r} - 1) = y(2^r - 1), 0 \leq x < 2^r, 0 \leq y < 2^{n-r}\}.$$

If $r \leq n - r$, for any choice of x (out of 2^r possible), we get a rotationally symmetric X only if $x\frac{2^{n-r}-1}{2^r-1}$ is a whole number. Let $d = \gcd(2^{n-r} - 1, 2^r - 1)$. Then by a simple algebra we get that $N_r = d + 1$. If $r|(n - r)$, we can show by using cyclotomic polynomials that $(2^r - 1)|(2^{n-r} - 1)$, and in this case $N_r = 2^r$. In the case of GOST, the largest possible set of weak keys is obtained for $r = 16$ with the cardinality $N_r^8 = 2^{128}$. A probability that we randomly select a weak key is thus negligible (lower than 2^{-128}).

If the keys are not selected randomly, a care should be also be taken to avoid weak keys. In this article we have assumed that rotational probability is computed over a random selection of keys over the set of all possible keys. However, if we restrict the selection of keys, the rotational probabilities may change. For example, if $K = 0$, then there does not occur any carry during the key additions. In this case all rotational pairs are preserved, i.e.,

$$Pr\big(\overleftarrow{GOST_0(L|R)} = GOST_0(\overleftarrow{L}\,|\,\overleftarrow{R})\big) = 1.$$

If we know only some bits of the key, the precise rotational probabilities can be recomputed to reflect this knowledge. To demonstrate, let us consider a situation, when rotational amount is $r = 1$ (and no S-boxes are used).

Let us suppose that we know the most significant bit of K^i. The rotational probability over randomly selected key with $MSB(K^i) = 1$, is only half of the rotational probability for keys with $MSB(K^i) = 0$. This can be computed using similar techniques as in Section 3.2, but it is easy to see that $MSB(K^i) = 1$ means more additions with carry from the most significant bit.

3.5. Key recovery attacks

Although our article focuses on distinguishing attacks, they can be easily expanded to key-recovery scenarios.

Let us suppose that either the cipher is keyed by the weak rotationally symmetric key, or the attacker can operate in the related key scenario. Furthermore suppose that attacker finds the first suitable pair of rotated plaintext that encrypts to rotated ciphertext (the key is not fixed in the first phase). The probability to obtain such a pair (under a random selection of keys) is given in Table 2, so the expected number of tests is reciprocal of this probability.

Let us consider that $(N-1)$-rounds of GOST preserve rotational inputs. Now, due to the nature of the Feistel network, also the left part of the ciphertext (for N-round GOST) stays rotated. In fact, we do not need the final right part of the ciphertext to be rotated, so the probability to obtain valid P-C pairs for our attack comes from $(N-1)$-round distinguishing probability.

Our experiments [9] show that once the attacker has obtained the first suitable rotated pair, he can find more suitable pairs for the same weak key/rotated key-pair more easily. This is due to the fact, that if the key is fixed, the scenario for repeated subkeys discussed in Section 3.2 applies.

Let us denote the left part of ciphertext L, and the right part R, respectively. We also denote the rotated left part of the paired ciphertext $\overleftarrow{L}$, and the corresponding right part of the paired ciphertext R' (in general, it is not a rotated version of R). Attacker guesses the last subkey $K_i^{(N-1)} = i$ (32-bit expansion of $i = 0, 1, \ldots, 2^{32} - 1$), and computes the outputs of round function

$$g_i = F\left(L, K_i^{(N-1)}\right) \quad \text{and} \quad g_i' = F\left(\overleftarrow{L}, \overleftarrow{K}_i^{(N-1)}\right).$$

Values (g_i, g_i') in general do not form a rotational pair. However, due to the nature of Feistel scheme (as analysed in Section 3.1, values $(g_i \oplus R, g_i' \oplus R')$ form a rotational pair for a correct key guess. So if $g_i' \oplus R' \neq (g_i \oplus R) \lll r$, we know that guess $K_i^{(N-1)}$ is incorrect, and we can remove the key from a set of potential keys.

Due to low diffusion in GOST, there are still many guesses that lead to a valid rotated $g_i' \oplus R'$. Our practical experiments (see Figure 5) show that attacker can learn approximately 6 bits of the key with data complexity 2^6. But if we increase the number of P-C pairs above 2^6 (in the experiments we used up to 2^{11} pairs), the number of recovered key-bits does not increase.

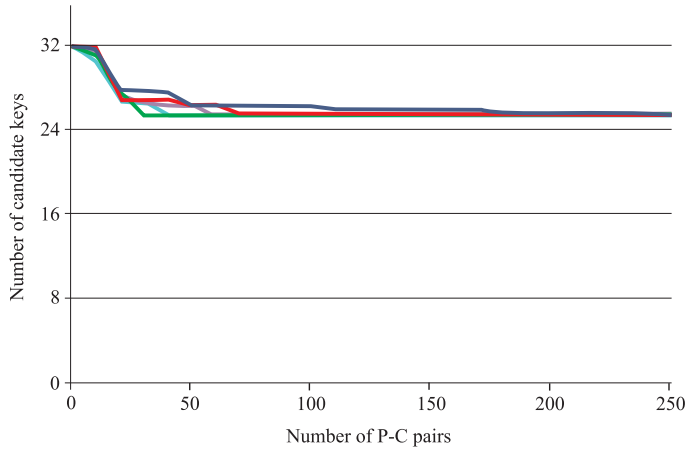

FIGURE 5. Base-2 logarithm of the number of remaining keys for key recovery attack as a function of the number of P-C pairs.

For each remaining key guesses, we can remove the last round of the encryption, and continue the attack recursively with fewer rounds (we do not have to remember all candidates if we use depth-first search). If we suppose that the subkeys are independent random values, we can expect to recover a similar amount of key-bits when guessing values for each of the eight subkeys. Recovering 6-bits out of each 32 amounts to testing $2^{(32-6)\cdot 8} = 2^{208}$ key hypotheses overall.

Let us summarize the attack. The attack is CPA with related keys. It requires that the set of identical S-boxes is used. Attacker encrypts D known pairs of plaintexts P with key K, and corresponding $\overleftarrow{P}$ with the related key $\overleftarrow{K}$.

The number of plaintext pairs depends on the number of rounds, and r, see Table 2. For a full GOST, $D = k \cdot 2^{34.9}$ for $r = 4$, and $D = k \cdot 2^{36.5}$ for $r = 16$. Constant k depends on the number of required P-C pairs to be used as a filter for key candidates. Out of D P-C pairs, we expect (more than) k P-C pairs to have rotationally symmetric right part usable in the attack. The ciphertexts for these P-C pairs are stored, and can be considered a part of memory complexity ($2 \cdot k$ 64-bit vectors). Experimentally, having $k = 2^6$ correct P-C pairs, we can filter out 6 bits out of every 32-bit subkey. The time to evaluate each candidate is approximately $c = 2 \cdot k \cdot 2^{-5}$ GOST encryptions (two one-round GOST encryptions for each of the k correct ciphertext pairs). Thus, using parameters $r = 16$, $k = 2^6$, we get the data complexity $2^{42.5}$ (chosen P-C pairs), memory complexity 2^7 (ciphertexts), and the time complexity 2^{210} (GOST encryptions).

The situation is better for the attacker in the scenario with rotationally symmetric keys. The prerequisite of the single-key attack is again the set of identical S-boxes. Moreover, we suppose that a weak key with rotationally symmetric subkeys is used. The weak key density is at most 2^{-128} (see Section 3.4) for rotational amount $r = 16$, and lower for other rotational amounts.

If the key is rotationally symmetric, the key has a reduced entropy. For example, if $r = 16$, the rotationally symmetric key can only consists of 32-bit subkeys in the form $(k_0 k_1 \dots k_{15} k_0 k_1 \dots k_{15})$. Thus, the attacker only needs to find 128-bits, instead of 256. For other r's the number of unknown bits is even lower, but so is the weak key density. Similarly to the related-key attack, for $r = 16$, $k = 2^6$, we get the same data complexity $2^{42.5}$ (chosen P-C pairs), and memory complexity 2^7 (ciphertexts). If we can filter out full 6 bits out of each subkey, the time complexity is reduced to checking $2^{(16-6)\cdot 8} = 2^{80}$ key hypotheses, which is equivalent to 2^{82} GOST encryptions.

Although these attacks are not practical, they significantly reduce the expected strength of the cipher. Furthermore, we have only explored the basic attacks that do not take into account the distribution of rotational errors. The model with rotational errors can possibly lead to new classes of weak keys (where rotational errors cancel each other due to slow diffusion of GOST), and possibly faster key recovery attacks.

4. Conclusions

There are many published attacks on GOST [1–4, 6], and many of these attacks exploit the weak key scheduling of GOST. In this paper we show that the repeated subkeys have also a significant adverse effect on the GOST's rotational properties. Our main results concern GOST without S-boxes. Although it is not realistic to expect that the user does not choose a set of strong S-boxes, some implementations may allow the adversary to manipulate the S-box selection, or to trick the user into using identity mapping instead of a set of valid non-linear S-boxes. We also note that a similar approach (removing GOST's S-boxes) was used to analyse of GOST's fixpoints in [14].

The cryptographic attacks described in this paper (restricted to rotational amounts that are multiples of 4) apply also to the case with S-boxes, specifically when all eight S-boxes in GOST are defined by a single permutation. We remark that this is not explicitly prohibited in the RFC's concerning GOST [5, 10]. Using a single table for all S-boxes is a tempting choice for lightweight implementations to conserve resources, as demonstrated by GOST-PS design [11]. However, in this case GOST is distinguishable from a random cipher using rotational distinguishers with related keys with the expected probability of an output

rotational pair as high as $2^{-35.5}$ (for $r = 4$, under a random key selection). Moreover, the dependence of rotational properties on the key selection produces a set of weak keys. Rotationally symmetric keys can be identified by the attacker even in the single key attack model (CPA).

Our analysis does not cover the distribution of rotational errors. The preliminary experimental results [9] show that the distribution of rotational errors might be exploited as a distinguisher as well (but we do not provide a mathematical model in this case). This may lead to stronger attacks of rotational type on GOST in the future. It is thus strongly recommended for each implementation of GOST either to use a fixed asymmetric set of S-boxes, or to check its S-box parameters explicitly for exploitable rotational symmetries.

REFERENCES

[1] COURTOIS, N. T: *Security evaluation of GOST 28147-89 in view of international standardisation*, Cryptologia **36** (2012) 2–13.

[2] COURTOIS, N. T.—MISZTAL, M.: *Differential cryptanalysis of GOST*, Cryptology ePrint Archive, Report 2011/312, 2011, `http://eprint.iacr.org/`.

[3] COURTOIS, N. T.: *Algebraic Complexity reduction and cryptanalysis of GOST*, Cryptology ePrint Archive, Report 2011/626, 2011, `http://eprint.iacr.org/2011/626`.

[4] DINUR, I.—DUNKELMAN, O.—SHAMIR, A.: *Improved attacks on full GOST*, in: Fast Software Encryption (A. Canteaut, ed.), LNCS Vol. 7549, Springer, Heidelberg, 2012, pp. 9–28.

[5] DOLMATOV, V.: *GOST 28147-89: Encryption, decryption, and message authentication code (MAC) algorithms*, RFC 5830 (Informational), March 2010.

[6] ISOBE, T.: *A single-key attack on the full GOST block cipher*, in: Fast Software Encryption (A. Joux, ed.), LNCS Vol. 6733, Springer, Heidelberg, 2011, pp. 290–305.

[7] KHOVRATOVICH, D.—NIKOLIĆ, I.: *Rotational cryptanalysis of ARX*, in: Fast Software Encryption 2010 (S. Hong, T. Iwata, eds.) LNCS Vol. 6147, Springer, Heidelberg, 2010, pp. 333–346.

[8] KHOVRATOVICH, D.—NIKOLIĆ, I.—RECHBERGER, C.: *Rotational rebound attacks on reduced skein*, in: Adv. in Cryptology-ASIACRYPT 2010 (M. Abe, ed.), LNCS 6477 Springer, Heidelberg 2010, pp. 1–19.

[9] ONDROŠ, M.: *ARX Ciphers*, in: Master's Thesis, Slovak University of Technology in Bratislava, 2013. (In Slovak)

[10] POPOV, V.—KUREPKIN, I. —LEONTIEV, S.: *Additional Cryptographic Algorithms for Use with GOST 28147-89, GOST R 34.10-94, GOST R 34.10-2001, and GOST R 34.11-94 Algorithms*, RFC 4357 (Informational), January 2006.

[11] POSCHMANN, A.—LING, S.—WANG, H.: *256 bit standardized crypto for 650 ge–gost revisited*, Cryptographic Hardware and Embedded Systems, CHES 2010 (S. Mangard, F.-X. Standaert, eds.) LNCS 6225, Springer, Heidelberg, 2010, pp. 219–233.

[12] GOSUDARSTVENNYI STANDART SOJUZA SSR: *Sistemy obrabotki informacii. Zashchita kryptograficheskaya, Algoritm kriptograficheskogo preobrazovaniya. Gosudarstvennyi Standart Soyuza SSR, GOST: 28147-89*, IPK Izdatelstvo standartov, Moskva, 1989.

[13] ZAJAC, P.—ČAGALA, R.: *Local reduction and the algebraic cryptanalysis of the block cipher GOST*, Periodica Mathematica Hungarica **65** (2012), 239–255.

[14] ZANECHAL, M.: *An algebraic approach to fix points of GOST-algorithm*, Mathematica Slovaca **51** (2001), 583–591.

Received October 18, 2013

Institute of Computer Science and Mathematics
Faculty of Electrical Engineering and
Information Technology
Slovak University of Technology in Bratislava
Ilkovičova 3
SK–812-19 Bratislava
SLOVAKIA

E-mail: pavol.zajac@stuba.sk
xondros@is.stuba.sk