A chosen key attack against the secret S-boxes of GOST

Unpublished manuscript from 1998.

Markku-Juhani O. Saarinen <mjos@iki.fi>

Historical notes

I am making this work from August 1998 available for historical reasons. It has been cited as an "unpublished manuscript" more than two dozen times over the years – even though it has not been publicly available for almost 20 years.

The memo describes a simple technique for extracting the S-Boxes from a GOST 29147-89 chip with 2^{32} effort – typically just few hours, depending on the chip (key set-up is fast with GOST). The original Soviet standard left the S-Boxes undefined and different S-Boxes were used by different branches of the government. In 1990s it was widely assumed that the S-Boxes were essentially a static part of the secret keying material – Bruce Schneier included speculation to this effect in "Applied Cryptography" (2nd Ed., Wiley 1996).

The same algorithm is now part of the GOST 34.12-2015 standard – which *does* define a standard set of substitution values (значения подстановок π_i). The algorithm has also acquired a name, "Magma", to distinguish it from a newer GOST block cipher, 128-bit "Kuznyechik". Both are in very active use, especially in Russia as they are the only approved ciphers for many use cases.

"Magma" («Магма») was apparently also a codename used by the KGB and the affiliated cryptographers that originally designed the algorithm. I have also been told that my chosen key attack is rather well known in Russia.

The S-Box extraction technique uses a **slide attack**, but that term is not used. In the memo I am simply referring to the "lifting property" of Feistel networks. This is because the term "slide attack" was not yet established at the time the text was written. Our current terminology comes from the paper "Slide Attacks", published by Alex Biryukov and David Wagner in CRYPTO '99.

I understand that the first published cryptanalytic use of Feistel sliding was by E. K. Grossman and B. Tuckerman in "Analysis of a Feistel-like Cipher Weakened by Having No Rotating Key", IBM Thomas J. Watson Research Division, 1977. Unfortunately I have not been able to find that report myself.

Please ignore the part about Finnish Air Forces crypto modems – this is probably not accurate. There was a company in Finland that manufactured cryptosystems based on GOST chips, and I was under the impression that Finnish Defence Forces was one of their customers. I was probably mistaken.

The paper is included in its original form, with all of its errors. I was just a 3rd-year mathematics undergraduate at the time, studying at University of Jyväskylä (Finland). I recall wasting most of my time on activities quite unrelated to pure mathematics – such as hacking cryptographic chips!

Dr. Markku-Juhani O. Saarinen
May 21, 2019
Oxford, UK

A chosen key attack against the secret S-boxes of GOST

Markku-Juhani Saarinen
mjos@math.jyu.fi

August 12, 1998

Abstract

We show that a simple "black box" chosen-key attack against GOST can recover secret S-boxes with approximately 2^{32} encryptions.

1 Introduction

The specification for the "Russian DES" GOST 28147-89 [1] does not specify its S-boxes. Different applications are known to use different S-boxes. For example, the crypto modems used by the Finnish Air Force are rumored to use one set of S-boxes which is hidden in an encryption chip. The idea is to prevent encryption and decryption without such a chip, even when the keys are known.

[2] and [3] consider the S-boxes as a fixed part of the secret key material, thus bringing the total amount of secret key material to $256 + 8 \lg(16!) = 256 + 354 = 610$ bits. If this would be the case, a chosen-key attack against the S-boxes would require 2^{354} effort.

We will describe a simple chosen key attack that allows the S-boxes (those fixed 354 bits) to be recovered with 2^{32} encryptions under a single chosen key. This means that the S-boxes of a typical GOST chip (such as the one described in [4]) can be found in a couple of hours without physical reverse engineering.

2 The lifting property of a Feistel network

Only one half of the block gets changed in one round of a Feistel network. If the rounds are identical and we can correctly guess the other half of the block after the first round, putting this block through the entire Feistel network results in only one half of the output block being changed. The entire Feistel network gets "lifted" by one round, hence the name.

More formally, consider an r - round Feistel network with the same round key for every round:

1. set $(L_0, R_0) = P$ (plaintext)
2. for $i = 1$ to r do
   $L_i = R_{i-1}$
   $R_i = L \oplus f(R_{i-1})$

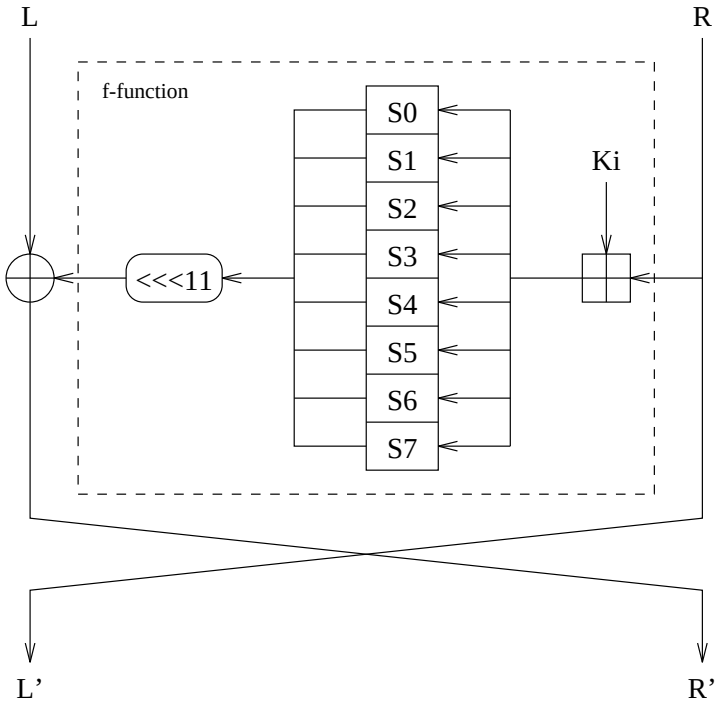

Figure 1: One round of GOST

3. set (ciphertext) $C = (R_i, L_i)$

Encrypting $P_1 = (y, x)$ produces a ciphertext block $C_1 = (b, a)$. If $f(x) = y$, encrypting $P_2 = (x, 0)$ will produce $C_2 = (a, c)$. This means that the right half of C_1 is the same as the left half of C_2.

3 The attack

The attack proceeds in two steps. The first step searches for a 32-bit "zero vector" $z = f(0)$. This step requires no more than 2^{32} encryptions.

The second step examines one S-box at the time and extracts the contents of that S-box. The second step requires approximately 2^{11} encryptions, not raising the total attack complexity significantly over 2^{32}.

3.1 Finding the zero vector

We will set the key to 0. As a result all of the round keys are 0.

Encrypt a zero block $(0, 0)$. Let a be the right half the ciphertext. Loop over the plaintexts $(z, 0)$ until a ciphertext is found that has a as the left half. There is a high probability that this is the zero vector $z = f(0)$. If the second step fails for this z, we will continue searching from $z + 1$; the zero vector is always found with 2^{32} encryptions.

3.2 Extracting the S-boxes

Encrypt $(a, 0)$. Let x be right half of the ciphertext. Since the lifting property holds, encrypting (b, a) will result the left half of ciphertext to be x if $f(a) = b$.

To test whether $S_i[u] = v$, one can set

$$\begin{aligned} a &= u \lll 4i \\ b &= (z \wedge \neg(1111_2 \lll 4i + 11)) \vee (v \lll 4i + 11) \end{aligned}$$

and perform the check described above. Since there are eight $4 * 4$ S-boxes, a naive algorithm will discover the contents of all S-boxes in no more than $8 * 2^4 * 2^4 = 2^{11}$ encryptions if z is known.

If the resulting S-boxes are not permutations of the numbers $0 \ldots 15$ or are otherwise broken, we will continue searching at step 1.

4 Conclusion

We have described a chosen-key attack against the secret S-boxes of GOST. The complexity of this attack is 2^{32} and the attack has been carried out in practice. We can conclude that keeping the GOST S-boxes secret does not increase the security of the cryptosystem as is commonly thought.

References

[1] I. A. Zabotin, G. P. Glazkov, V. B. Isaeva. *Cryptographic Protection for Data Processing Systems.* Government Standard of the U.S.S.R. 28147 - 89. (In Russian.)

[2] B. Schneier *Applied Cryptography, second edition.* John Wiley and Sons, New York, 1996.

[3] C. Charnes, L. O'Connor, J. Pieprzyk, R. Safavi-Naini, and Y. Zheng *Comments on Soviet encryption algorithm.* Advances in Cryptology – EUROCRYPT '94 (LNCS 950), 433 – 438, Springer-Verlag, New York, 1995.

[4] A. V. Pichuev, A. G. Ryabchenko, D. G. Titov, and S. A. Frolov. On designing a high-speed VLSI Data Ciphering Processor. *Optoelectronics, Instrumentation and Data Processing (Avtometriya).* Allerton Press, New York, 1994.