جامـــــعة المـــــستـقـبل
AL MUSTAQBAL UNIVERSITY

قســـم الامــــــن الـــــــــسيبرانــــــــي

DEPARTMENT OF CYBER SECURITY

SUBJECT:

GOST CIPHER

CLASS:

SECOND

LECTURER:

ASST. LECT. MUSTAFA AMEER AWADH

LECTURE: (4)

GOST

GOST 28147-89

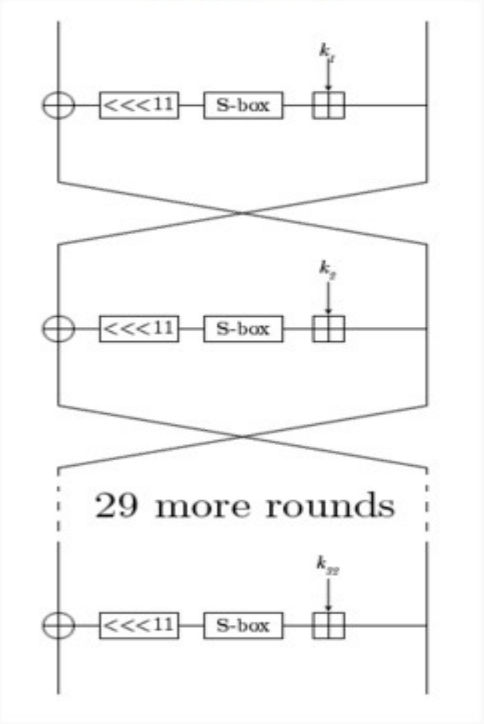


Diagram of GOST

General

GOST is a block algorithm from the former Soviet Union. “GOST” is an acronym for “Gosudarstvennyi Standard,” or Government Standard, sort of similar to a FIPS, except that it can (and does) refer to just about any kind of standard. (Actually, the full name is Gosudarstvennyi Standard Soyuza SSR, or Government Standard of the Union of Soviet Socialist Republics.) This standard is number 28147-89. The Government Committee for Standards of the USSR authorized the standard, whoever they were.

I don’t know whether GOST 28147-89 was used for classified traffic or just for civilian encryption. A remark at its beginning states that the algorithm “satisfies all cryptographic requirements and not limits the grade of information to be protected.” I have heard claims that it was initially used for very high-grade communications, including classified military communications.

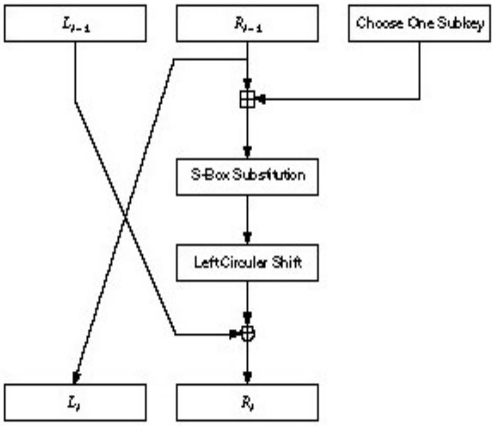
Description of GOST

GOST is a 64-bit block algorithm with a 256-bit key. GOST also has some additional key material that will be discussed later. The algorithm iterates a simple encryption algorithm for 32 rounds.

To encrypt, first break the text up into a left half, *L*. and a right half, *R*. The subkey for round *i* is K_i. A round, *i*, of GOST is:

$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$$

Figure is a single round of GOST. Function f is straightforward. First, the right half and the *i*th subkey are added modulo 2^{32}. The result is broken into eight 4-bit chunks, and each chunk becomes the input to a different S-box. There are eight different S-boxes in GOST; the first 4 bits go into the first S-box, the second 4 bits go into the second S-box, and so on. Each S-box is a permutation of the numbers 0 through 15. For example, an S-box might be:



One round of GOST.

7, 10, 2, 4, 15, 9, 0, 3, 6, 12, 5, 13, 1, 8, 11

In this case, if the input to the S-box is 0, the output is 7. If the input is 1, the output is 10, and so on. All eight S-boxes are different; these are considered additional key material. The S-boxes are to be kept secret.

The outputs of the eight S-boxes are recombined into a 32-bit word, then the entire word undergoes an **11-bit left circular shift**. Finally, the result XORed to the left half to become the new right half, and the right half becomes the new left half. Do this 32 times and you're done.

The subkeys are generated simply. The 256-bit key is divided into eight 32-bit blocks: k_1, k_2,..., k_8. Each round uses a different subkey, as shown in following Table. Decryption is the same as encryption with the order of the k_is reversed.

The GOST standard does not discuss how to generate the S-boxes, only that they are somehow supplied. This has led to speculation that some Soviet organization would supply good S-boxes to those organizations it liked and bad S-boxes to those organizations it wished to eavesdrop on. This may very well be true, but further conversations with a GOST chip manufacturer within Russia offered

Use of GOST Subkeys in Different Rounds

Round:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Subkey:	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
Round:	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Subkey:	1	2	3	4	5	6	7	8	8	7	6	5	4	3	2	1

More recently, a set of S-boxes used in an application for the Central Bank of the Russian Federation surfaced. These S-boxes are also used in the GOSTone-way hash function (see section 18.11) [657].They are listed in Table 14.2.

Cryptanalysis of GOST

These are the major differences between DES and GOST.

- DES has a complicated procedure for generating the subkeys from the keys. GOST has a very simple procedure.
- DES has a 56-bit key; GOST has a 256-bit key. If you add in the secret S-box permutations,.

Table
GOST S-Boxes

S-box 1:	4	10	9	2	13	8	0	14	6	11	1	12	7	15	5	3
S-box 2:	14	11	4	12	6	13	15	10	2	3	8	1	0	7	5	9
S-box 3:	5	8	1	13	10	3	4	2	14	15	12	7	6	0	9	11
S-box 4:	7	13	10	1	0	8	9	15	14	4	6	12	11	2	5	3
S-box 5:	6	12	7	1	5	15	13	8	4	10	9	14	0	3	11	2
S-box 6:	4	11	10	0	7	2	1	13	3	6	8	5	9	12	15	14
S-box 7:	13	11	4	1	3	15	5	9	0	10	14	7	6	8	2	12
S-box 8:	1	15	13	0	5	7	10	4	9	2	3	14	6	11	8	12

- The S-boxes in DES have 6-bit inputs and 4-bit outputs; the S-boxes in GOST have 4bit inputs and outputs. Both algorithms have eight S-boxes, but an S-box in GOST is one-fourth the size of an S-box in DES.
- DES has an irregular permutation, called a P-box; GOST uses an 11-bit left circular shift.
- DES has 16 rounds; GOST has 32 rounds.

If there is no better way to break GOST other than brute force, it is a very secure algorithm. GOST has a 256-bit key—longer if you count the secret S-boxes. Against differential and linear cryptanalysis, GOST is probably stronger than DES. Although the random S-boxes in GOST are probably weaker than the fixed S-boxes in DES, their secrecy adds to GOST's resistance against differential and linear attacks. Also, both of these attacks depend on the number of rounds: the more rounds, the more difficult the attack. GOST has twice as many rounds as DES; this alone probably makes both differential and linear cryptanalysis infeasible.

The other parts of GOST are either on par or worse than DES. GOST doesn't have the same expansion permutation that DES has. Deleting this permutation from DES weakens it by reducing the avalanche effect; it is reasonable to believe that GOST is weaker for not having it. GOST's use of addition instead is no less secure than DES's XOR.

The greatest difference between them seems to be GOST's cyclic shift instead of a permutation. The
DES permutation increases the avalanche effect. In GOST a change in one input bit affects one S-box in one round, which then affects two S-boxes in the next round, three the round after that, and so on. GOST requires 8 rounds before a single change in an input affects every output bit; DES only requires 5 rounds. This is certainly a weakness. But remember: GOST has 32 rounds to DES's 16.

GOST's designers tried to achieve a balance between efficiency and security. They modified DES's basic design to create an algorithm that is better suited for software implementation. They seem to have been less sure of their algorithm's security, and have tried to compensate by making the key length very large, keeping the S-boxes secret, and doubling the number of iterations. Whether their efforts have resulted in an algorithm more secure than DES remains to be seen.