UDC 004.021:004.056.55-083.74

THE DEVELOPMENT OF A CRYPTOALGORITHM BASED ON DOMESTIC STANDARDS

Vasilevsky I. P., Dobygin.M .D.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Goncharova I.V. – Lecturer at the Department of Foreign Languages

Annotation. A symmetric cryptoalgorithm based on the GOST 28147-89 standard has been developed and implemented. The algorithm provides a high degree of plaintext dispersion and processing speed. The possibility of independent decryption of blocks has been implemented, which makes it possible to extract information in case of damage to individual parts of the ciphertext. The results are applicable in government agencies and backup systems.

Keywords: information security, cryptographic algorithm, symmetric encryption, domestic standards

Introduction. Cybersecurity is a major concern for businesses, governments, and individuals around the world. As cybercrime continues to grow, it is more important than ever to protect sensitive data. Encryption is an important tool to achieve this goal by converting data into encrypted text using a set of mathematical procedures known as an encryption algorithm. In this work, only the symmetric type of encryption will be considered, the most famous examples of which are DES, AES and the domestic GOST 28147-89.

Main part. GOST 28147-89 [1] [2] is a block cipher adopted as a standard for block encryption algorithms with a private key in 1989. The cipher proposed by GOST 28147-89 is built according to the same principles as the American DES, however, in comparison with DES, the domestic encryption standard is more convenient for software implementation. The main parameters of the cryptographic data conversion algorithm GOST 28147-89 are as follows: the block size is 64 bits, the key size is 256 bits, and the number of rounds is 32.

One of the algorithms for encrypting (decrypting) data of the GOST 28147-89 standard is the simple replacement mode. This is the simplest mode in which 64-bit data blocks are processed independently of each other.

Encryption by the GOST 28147-89 [1] standard in the simple replacement mode has both positive and negative sides. Positive ones:

1 Simplicity. For simple substitution encryption using the GOST 28147-89 algorithm, it is only necessary to split the data into blocks of 8 bytes each.

2 The possibility of parallelization. Since the blocks are not interconnected, nothing prevents the task from being distributed across multiple processor cores and processing multiple blocks at once.

3 Damage to a single block or a change in the sequence of blocks does not prevent decryption of the remaining blocks in the chain.

The main disadvantage of this mode is the preservation of the statistical features of the plaintext, since the same ciphertext blocks correspond to the same plaintext blocks.

Figure 1– Plaintext in the form of an image

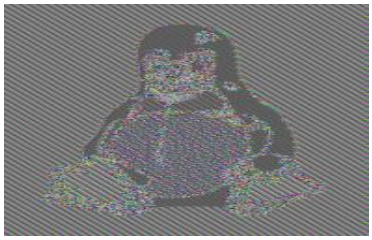

Figure 2 – Cryptogram obtained using the simple substitution mode

Figure 3 – Cryptogram obtained in a mode other than simple substitution

Let's look at the encryption process using the examples shown in Figures 1, 2, and 3. The first figure shows plaintext in the form of an image. The second figure shows a cryptogram obtained using the simple substitution mode, which shows the preservation of the statistical features of the plaintext, which makes encryption secure. To solve this problem, various modes have been invented, such as the ciphertext block coupling mode, ciphertext feedback mode, output feedback mode, and others [5]. While these alternative modes effectively address the issue of statistical pattern preservation, they introduce a new set of operational drawbacks. Their design inherently links ciphertext blocks to their predecessors, creating a dependency chain. The result of encryption using such modes is shown in Figure 3, where the original image structure is completely obfuscated. However, this security comes at a cost, as modern encryption (decryption) modes have the following problems, namely:

1 The inability to parallelize encryption/decryption. This significantly slows down the process of encrypting a large amount of data.

2 Changing the sequence of blocks or damaging a single block prevents decryption of the remaining blocks in the chain.

3 The complexity of the implementation. Such modes of operation are more difficult to create and require more calculations.

The conflict between the need for statistical dispersion and the desire to retain the operational benefits of parallelization and fault tolerance presents a significant challenge. The simple replacement mode offers speed and resilience but is cryptographically weak. Conversely, existing chaining modes provide robust security but sacrifice performance and data recovery capabilities. This situation motivated the search for a compromise solution that combines the strengths of both approaches.

The following cryptoalgorithm has been developed to solve these problems.

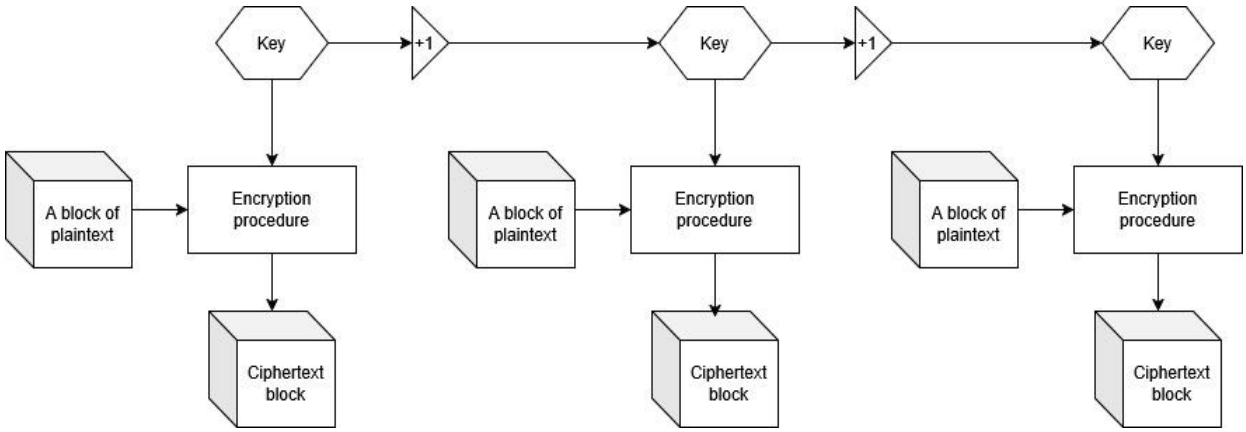

Figure 4 – Cryptoalgorithm

A key aspect of the proposed solution is the dynamic generation of the encryption key for each individual block. Instead of complicating the connections between blocks, this approach complicates the key material itself, making it unique for every segment of data. This ensures that even if the same plaintext block appears multiple times within a message, the resulting ciphertext blocks will be completely different, thereby eliminating statistical pattern leakage. Furthermore, by deriving the per-block key from a base secret using a simple deterministic function, the scheme retains the independent block processing characteristic of the simple replacement mode, preserving its advantages in parallel computing and fault tolerance.

Let P_i be the ith block of plaintext, C_i be the ith block of ciphertext, E – be the encryption procedure, k be the key, then

$$k_i = k_0 + i$$
$$C_i = E_k(P_i)$$

That is, one of the 2^{256} keys is generated for each block, which eliminates repetition on any practically achievable amounts of data. The decryption block diagram is analogous to the encryption scheme. This cryptoalgorithm has a number of advantages over those listed above:

1 The algorithm is simple to implement and does not require complex constructions.

2 Since the plaintext and ciphertext blocks are not connected to each other in any way, this allows you to distribute calculations between processor cores and increase the speed of both encryption and decryption.

3 Damage or modification of the sequence of individual blocks does not prevent the decryption of other blocks in any way.

To demonstrate the operation of the proposed cryptoalgorithm, a software implementation capable of encrypting and decrypting text data was developed. Thanks to it, it is possible to visually see the differences in processing speed using multithreaded mode and decryption of previously corrupted ciphertext. The program successfully decrypted the remaining part of the ciphertext.

Table 1 – Testing of the developed cryptoalgorithm

Mode	**Single-threaded**		**Multithreaded (4 threads)**	
Iteration	1	2	3	4
Encryption	19747ms	19786ms	5542ms	5501ms
Decryption	19986ms	19845ms	5453ms	5433ms

The practical significance of the developed cryptoalgorithm extends beyond mere performance metrics. Its architecture is particularly well-suited for modern distributed storage systems and cloud environments, where data is often fragmented across multiple physical locations. In such systems, the ability to decrypt surviving fragments independently– without relying on the integrity of the entire dataset– provides a critical layer of resilience against partial data corruption or loss. This fault-tolerant property, combined with the observed acceleration on multi-core processors, positions the proposed algorithm as a viable solution for high-load information systems where both security and availability are paramount.

Thus, the proposed cryptographic algorithm makes it possible to effectively disperse the statistical features of the text due to the changing key, and provides significant performance gains with an increase in the number of cores, which is confirmed by the results of experimental testing (Table 1). The algorithm's fault tolerance to damage to individual ciphertext segments was also confirmed, which is a critical factor for high-capacity data storage systems.

Conclusion. In the course of the research work, a symmetric cryptoalgorithm was developed and implemented, based on the domestic standard GOST 28147-89. It is important to note that the proposed cryptographic algorithm is compatible with any block ciphers, for example, AES, DES, «Grasshopper» or the BelT algorithm [4], which, according to STB 34.101.31-2020 [3], is mandatory for all organizations and enterprises of the Republic of Belarus in accordance with the requirements of the Operational Analytical Center under the President of the Republic of Belarus.

References

1. GOST 28147-89. Information processing systems. Cryptographic protection. Cryptographic transformation algorithm. Introduced 1990-07-01. Moscow: IPK Izdatelstvo Standartov, 1989. – 28 p.

2. Vinokurov, A. GOST 28147-89 encryption algorithm, its use and software implementation for Intel x86 platform computers. – Mode of access: https://kaf401.rloc.ru/Criptfiles/gost28147/GOST28147.htm – Date of access: 05.03. 2026.

3. STB 34.101.31-2020. Information technology and security. Cryptographic algorithms for encryption and integrity control. Introduced 2020-11-01. Minsk: Gosstandart, 2020. – 46 p.

4. Agievich, S. V., et al. BelT block cipher algorithm [Electronic resource]. National Research Center for Applied Problems of Mathematics and Informatics, BSU. – Mode of access: https://apmi.bsu.by/assets/files/agievich/BelT.pdf – Date of access: 05.03. 2026..

5. Babenko, L. K., & Ishchukova, E. A. Cryptographic information protection: symmetric encryption: Textbook. Rostov-on-Don: Southern Federal University Publishing House, 2015. – 220 p.