УДК 004.056.53

**С.А. ГУБКА, А.С. ГУБКА, Н.Ю. НОСОВА**

***Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Украина***

# МОДЕРНИЗАЦИЯ СИММЕТРИЧНОГО АЛГОРИТМА ШИФРОВАНИЯ

*В работе рассмотрены основные проблемы криптографической стойкости симметричного алгоритма шифрования по ГОСТ 28147-89. На основе анализа этих проблем разработаны методы решения, связанные с двумя основными направлениями: увеличением размера ключа шифрования и усовершенствованием доступа к получению таблицы замен. Модернизированный алгоритм защищен с помощью двойного подхода к шифрованию на основе простой и расширенной сетей Фейстеля. При этом алгоритм сохранил достаточную скорость обработки данных и хорошие показатели по автоматизации с помощью современных средств программирования, а защищенность алгоритма улучшилась по сравнению с оригиналом.*

***Ключевые слова:*** *симметричный алгоритм шифрования, криптоанализ, сеть Фейстеля, криптография, компьютерные системы, защита информации.*

## Введение

В настоящее время в мире наблюдается стремительное внедрение компьютерных систем (КС) во все сферы человеческой деятельности. В связи с этим остро встает задача защиты информации, передаваемой в рамках КС и открытых каналов связи (например, сеть Internet).

Наиболее эффективным средством защиты информации, по мнению большинства специалистов, является криптографические методы защиты информации [1 – 3]. А из них наиболее надежными являются симметрические системы шифрования информации [4]. Симметричные системы шифрования, в свою очередь, могут быть платными (запатентованными) и бесплатными (OpenSourse).

Платные системы существенно увеличивают материальную составляющую процесса защиты, и к тому же не гарантируют полного отсутствия программных закладок и технических входов в сам алгоритм.

Бесплатные системы располагают открытыми алгоритмами и кодами, которые к тому же можно достаточно просто подвергнуть модернизации. Пожалуй, одним из немногих алгоритмов, который стоит особняком, является симметричный алгоритм шифрования по ГОСТ 28147-89 (далее ГОСТ) [5]. Он сочетает в себе достаточно стойкий алгоритм (на уровне запатентованных) и полностью открытую структуру для использования. Это позволяет достаточно просто его использовать и модернизировать.

До 2009 года в Украине этот ГОСТ являлся межгосударственным стандартом и именовался ГОСТ 28147-89. Государственным комитетом Украины по вопросам технического регулирования и потребительской политики [6] ГОСТу присвоен новый индекс ДСТУ ГОСТ 28147:2009. Фактически кроме соответствующих приставок к названию, никаких других изменений в ГОСТ внесено не было. Он даже не был переведен на украинский язык, а так и остался на языке оригинала (т.е. русском). Кроме того, в приказе допускается возможность использования как нового, так и старого названия этого ГОСТа. Исходя из вышесказанного, будем ссылаться на первоисточник, т.е. ГОСТ 28147-89.

## Анализ проблемы

Исходя из вышеизложенного в качестве основы для построения современного симметричного алгоритма шифрования целесообразно использовать алгоритм ГОСТ 28147-89. При этом необходимо выявить «слабые» места этого алгоритма с точки зрения современных подходов криптоанализа и уровня развития КС.

Один из подходов криптоанализа предложен Е.А. Ищуковой [7]. Он основан на возможности алгоритма ГОСТ работать с 32-битными блоками данных и ключей и применении дробления шифртекста на блоки и выявлении элементов ключа, объединение которых и даст полный 256 битный ключ алгоритма ГОСТ.

В источнике [8] предлагается для решения проблемы криптоанализа увеличить размер блока с четырехбитных групп до байтовых групп. Это позволяет существенно увеличить время просчета комбинаций ключа, однако этот подход применим только для «слабых» ключей.

Максимально полный криптоанализ алгоритма ГОСТ выполнен в работе С.П. Панасенко [9], где рассмотрены сразу несколько вариантов, основанных как на «слабых» местах самого алгоритма ГОСТ, так и на современных методах криптоанализа. По результатам сделан вывод о том, что в случае применении не полнораундового алгоритма ГОСТ существует потенциальная возможность вскрытия алгоритма. В случае же полнораундового алгоритма вскрытие этими методами на данный момент не представляется возможным.

Суммируя вышеизложенное, можно сформулировать основные недостатки алгоритма ГОСТ, связанные с неполнотой стандарта в части генерации ключей и таблиц замен. Тривиально доказывается, что у алгоритма существуют «слабые» ключи и таблицы замен, но в стандарте не описываются критерии выбора и отсева «слабых». Также стандарт не специфицирует алгоритм генерации таблицы замен (S-блоков).

Однако, несмотря на перечисленные проблемы алгоритма ГОСТ, на данный момент нет достоверных данных о том, что данный алгоритм может быть вскрыт (при полнораундовой реализации).

Кроме того, процесс развития нанотехнологий и КС не стоит на месте и возможно в ближайшем будущем нас ждет серьезный скачек в производительности КС.

Исходя из проведенного анализа и перспектив развития КС, прослеживается необходимость устранения «слабых» мест алгоритма ГОСТ или усложнения доступа к ним.

## 2. Решение поставленных задач

### 2.1. Процедура деления исходного ключа на составляющие

Для усложнения процесса выбора подключей из общего ключа можно использовать следующую процедуру.

***Шаг 1.*** Генерируем ключ длиной 512 бит и разбиваем его последовательно на 16 блоков по 32 бита.

***Шаг 2.*** Выделяем из общего ключа 512 бит 256 бит ключа для алгоритма шифрования по ГОСТ. Для достижения этой цели можно действовать разными способами. Например, будем использовать простой алгоритм:

**если n=1...5, то $x=2^{n-1}$,**

**если n=6...8, то $x=|2^{n-1}/y|$,**

где n – номер выбираемого блока для алгоритма шифрования по ГОСТу, x – номер выбранного блока из общего числа блоков, y – целое число в диапазоне от 8 до 128.

***Шаг 3.*** По полученным на предыдущем шаге значениям x выбираем 8 блоков из общего 512-го ключа, оставшиеся блоки будут использованы во втором алгоритме.

### 2.2. Первый этап шифрования данных

Шифрование данных выполняется в соответствии с алгоритмом ГОСТ (режим простой замены). В качестве ключа используются 8 выбранных с помощью процедуры деления ключа блоков (256 бит) (см. п.2.1.).

### 2.3. Второй этап шифрования данных

#### 2.3.1. Основной шаг алгоритма

***Шаг 1.*** Полученный шифртекст первого этапа шифрования, используем в качестве исходного текста для второго этапа шифрования (рис. 1).

***Шаг 2.*** Оставшиеся 8 блоков ключа (256 бит) нумеруем $X=\{X_0 \ldots X_7\}$ (аналогично алгоритму ГОСТ).

***Шаг 3.*** Выбираем блок шифртекста (128 бит) и делим его на 4 равные части по 32 бита $N=(N_1, N_2, N_3, N_4)$.

***Шаг 4.*** В цикле 3 раза производим операцию сложения по модулю $2^{32}$ с первыми тремя блоками данных ($N_1$, $N_2$, $N_3$) и элементом ключа $X_0$ ($S_2 = (N_i+X_0) \bmod 2^{32}$).

***Шаг 5.*** Полученный результат сдвигаем на 11 бит влево ($S_1=R_{11L}(S)$).

***Шаг 6.*** Далее повторяем шаг 4 один раз с блоком данных $N_4$ и элементом ключа $X_0$.

***Шаг 7.*** Блоку $N_1$ присваиваем полученное значение $S_2$, а блоку $N_2$ присваиваем значение $S_1$.

***Шаг 8.*** Блоки $N_1$и $N_2$ складываем и получаем блок шифра N (128 бит).

#### 2.3.2. Алгоритм зашифровывания (128/32-З)

***Шаг 1.*** Выбираем блок данных N (128 бит) (рис. 2).

***Шаг 2.*** В цикле, используя блок данных N и элементы ключа в последовательности ($X_5$, $X_6$, $X_7$, $X_8$), реализуем алгоритм основного шага.

***Шаг 3.*** В цикле, используя блок данных N и элементы ключа в последовательности ($X_4$, $X_3$, $X_2$, $X_1$), реализуем алгоритм основного шага.

***Шаг 4.*** Полученные блоки меняем местами.

***Шаг 5.*** В результате получаем блок N (128 бит).

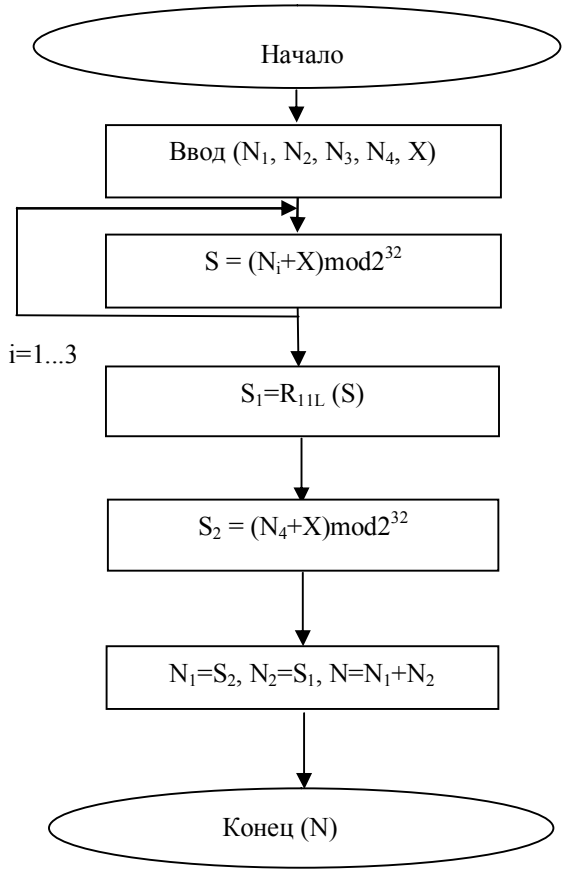


Рис. 1. Основной шаг второго этапа шифрования

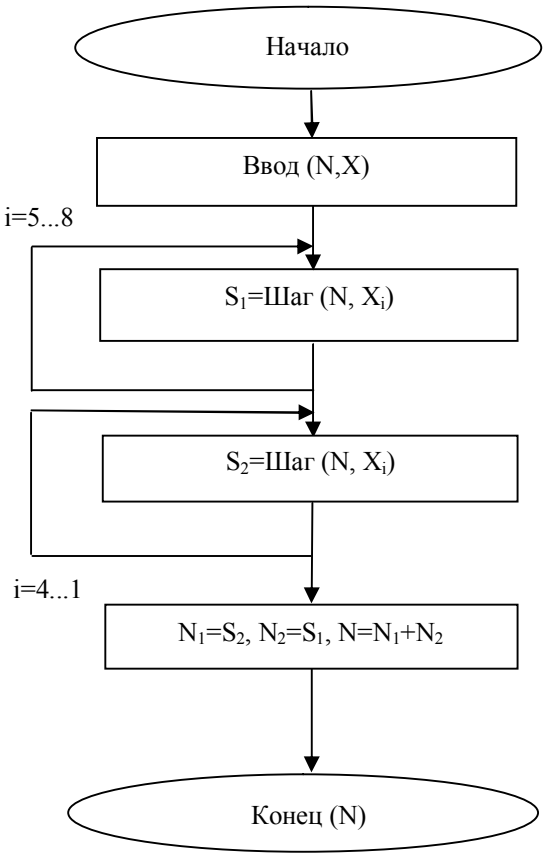


Рис. 2. Алгоритм зашифровывания 128/32-З

#### 2.3.3. Алгоритм расшифровывания (128/32-Р)

Выполняется в строго обратной последовательности относительно алгоритма зашифровывания (рис.3).

***Шаг 1.*** Выбираем блок данных N (128 бит).

***Шаг 2.*** В цикле, используя блок данных N и элементы ключа в последовательности ($X_1$, $X_2$, $X_3$, $X_4$), реализуем алгоритм основного шага.

***Шаг 3.*** В цикле, используя блок данных N и элементы ключа в последовательности ($X_8$, $X_7$, $X_6$, $X_5$), реализуем алгоритм основного шага.

***Шаг 4.*** Полученные блоки меняем местами.

***Шаг 5.*** В результате получаем блок N (128 бит).

## 3. Особенности разработанной схемы усовершенствования алгоритма ГОСТ

1. Применение ключа в 512 бит, возможно, потребует несколько более мощных генераторов псевдослучайных чисел (ГПСЧ). Однако остается возможность оставить исходные ГПСЧ, просто сгенерировав два ключа по 256 бит и объединив их в один большой ключ 512 бит.

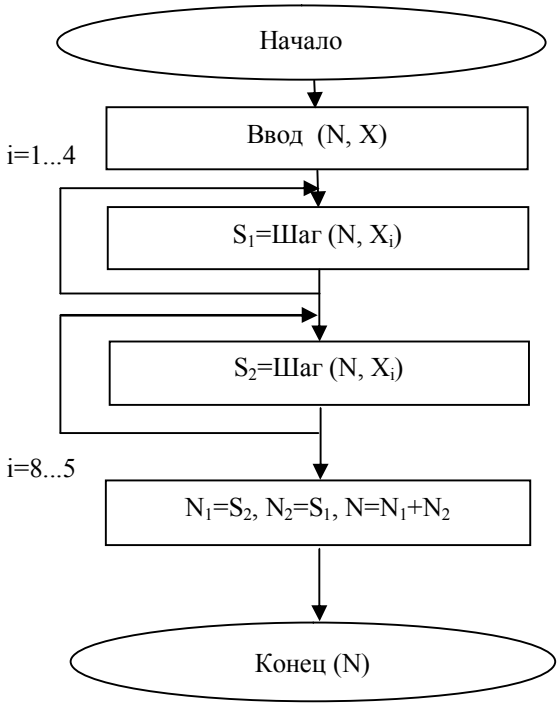


Рис. 3. Алгоритм расшифровывания 128/32-Р

2. Элементы ключа 512 бит не используются одновременно в разных частях алгоритма, что существенно повышает стойкость к вскрытию. Кроме того, процедура выбора элементов ключа существенно затрудняет идентификацию тех блоков ключа, которые используются в том или ином алгоритме, в отличие от аналогичных алгоритмов шифрования, где ключ делиться на равные промежутки и последовательно применяется. Даже зная ключ или некоторые его элементы, злоумышленник затратит дополнительное время на выделение из общего ключа двух подключей.

3. Чтобы усилить алгоритм шифрования обычно используют либо увеличение длины ключа (если это позволяет сам алгоритм), или применяют элементы защиты типа гаммирования, имитовставки и т.п. Фактически эти элементы представляют собой дополнительные шаги основного алгоритма, построенные по его образу и подобию, только включающие меньшее количество шагов (итераций). Кроме того, в этих дополнительных элементах используется тот же ключ, что и в основном алгоритме шифрования.

## Заключение

В предложенном алгоритме произведена попытка устранить вышеописанные недостатки, а именно:

1. Предложенная вторая часть алгоритма реализована на базе расширенной сети Фейстеля в отличие от алгоритма ГОСТ, который основан на простой сети Фейстеля. Таким образом, удалось добиться неравенства исходных блоков для частей алгоритма, что позволяет избавиться от повторяющихся частей шифртекста. Эти части обычно являются одним из тех слабых мест, с помощью которых злоумышленник может вычислить элементы ключа.

2. Использование для первой и второй частей алгоритма неповторяющихся и непересекающихся элементов ключа существенно затрудняет криптоанализ шифртекста.

3. Благодаря использованию принципов, заложенных в расширенную сеть Фейстеля, увеличилась скорость обработки блоков при операциях зашифровывания и расшифровывания во второй части алгоритма (скорость обработки увеличилась в несколько раз, т. к. за один шаг обрабатывается 4, а не 1 блок данных (32 бита)). Таким образом, применение дополнительного алгоритма шифрования существенно не снижает скорость работы алгоритма в целом. Время операций шифрования и расшифровывания увеличится максимум на 20 – 30% [10] от алгоритма ГОСТ, что при нынешнем уровне производительности персональных компьютеров (ПК) составляет от доли секунды до нескольких секунд (табл. 1).

4. В цикле зашифровывания и расшифровывания, в отличие от алгоритма ГОСТ, применяются элементы ключа смешанным образом в рамках одного ключа (256 бит). Этот подход является нестандартным и также вносит дополнительные трудности при проведении криптоанализа.

Таблица 1

Сравнительные характеристики вариантов алгоритма ГОСТ

| Алгоритмы / Варианты реализации | Оригинальный ГОСТ (режим простой замены) | Модернизированный вариант ГОСТ |
|---|---|---|
| ПК (32-разрядный, одноядерныйцентральный процессор Intel Pentium 4, 3,0 GHz) | не менее 16 МБ/сек | не менее 12 МБ/сек |
| В аппаратных комплексах | не менее 24 МБ/сек | не испытывался |
| В потоковых аппаратных комплексах | не менее 48 МБ/сек | не испытывался |

## Литература

*1. Казарин О.В. Безопасность программного обеспечения компьютерных систем / О.В. Казарин. – М.: МГУЛ, 2003. – 212 с.*

*2. Основы криптографии / А.П. Алферов, А.Ю. Зубов, А.С. Кузьмин, А.В. Черемушкин. – М.: Гелиос АРВ, 2001. – 479 с.*

*3. Бабаш А.В. Криптография / А.В. Бабаш, Г.П. Шанкин. – М.: Солон-Р, 2002. – 511 с.*

*4. Казарин О.В. Теория и практика защиты программ / О.В. Казарин. – М.: МГУЛ, 2004. – 450 с.*

*5. Винокуров А.А. Алгоритм шифрования ГОСТ 28147-89, его использование и реализация для компьютеров платформы Intel x86 [Электронный ресурс] / А.А. Винокуров. – Режим доступа: http://www.enlight.ru/crypto/articles/vinokurov/gost_i.htm.*

*6. Про прийняття міждержавних стандартів як національних методом підтвердження та скасування відповідних міждержавних стандартів: Наказ Державного комітету України з питань технічного регулювання та споживчої політики від 22.12.2008 г. №495 [Электронный ресурс]. – Режим доступа: http://www.licasoft.com.ua/index.php/component/lica/?base=1&id=x000CC641.*

*7. Ищукова Е.А. Применение рекурсивного алгоритма поиска в B-деревьях для дифференциального криптоанализа алгоритма шифрования ГОСТ*

*28147-89 [Электронный ресурс] / Е.А. Ищукова. – Режим доступа: http://www.contrterror.tsure.ru/www/magazine12/08-19-Ischykova.htm.*

*8. ГОСТ 28147-89. Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования. – Введен впервые; введ. 02.06.1989 [Электронный ресурс]. – Режим доступа: http://protect.gost.ru/document.aspx?control=7&id=139177.*

*9. Панасенко С.П. Алгоритм шифрования ГОСТ 28147-89 [Электронный ресурс] / С.П. Панасенко. – Режим доступа: http://www.inssl.com/standart-of-cipher.html.*

*10. Программное изделие «LS-Crypt» (библиотеки функций криптографических преобразований) [Электронный ресурс] / Материалы компании «Lime Systems». – Режим доступа: http://lime-systems.com/soft/ls-crypt/.*

*Поступила в редакцию 4.03.2011*

**Рецензент:** д-р техн. наук, проф., заведующий кафедрой "Информационные управляющие системы" В.М. Левыкин, Харьковский национальный университет радиоэлектроники, Харьков, Украина.

## МОДЕРНІЗАЦІЯ СИМЕТРИЧНОГО АЛГОРИТМУ ШИФРУВАННЯ

***С.О. Губка, О.С. Губка, Н.Ю. Носова***

У роботі розглянуті основні проблеми криптографічного стійкості симетричного алгоритму шифрування за ГОСТ 28147-89. На основі аналізу цих проблем розроблені методи рішення, пов'язані з двома основними напрямками: збільшенням розміру ключа шифрування і удосконаленням доступу до отримання таблиці замін. Модернізований алгоритм захищений за допомогою подвійного підходу до шифрування на основі простої і розширеної мереж Фейстеля. При цьому алгоритм зберіг достатню швидкість обробки даних і хороші показники по автоматизації за допомогою сучасних засобів програмування, а захищеність алгоритму покращала в порівнянні з оригіналом.

**Ключові слова:** симетричний алгоритм шифрування, криптоаналіз, мережа Фейстеля, криптографія, комп'ютерні системи, захист інформації.

## MODERNIZATION SYMMETRIC ENCRYPTION ALGORITHMS

***S.A. Gubka, A.S. Gubka, N.Yu. Nosova***

In the article the main problems of cryptographic stability of encryption algorithm, which is described in the State Standard 28147-89, are represented. Based on the analysis of these problems the methods for solving them have been given, which are connected with two basic ways: increasing the size of the encryption key and improving access to the replacement table. Updated algorithm is protected by a dual encryption, which is based on the simple and extended Feistel network. The algorithm kept sufficient data processing rate and good reading of automation using modern programming tools. Besides this, the security of the algorithm has been improved comparing with the original one.

**Keywords:** symmetric encryption algorithm, cryptanalysis, Feistel network, cryptography, computer systems, information security.

**Губка Сергей Алексеевич** – канд. техн. наук, доцент, доцент кафедры "Информационные управляющие системы", Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков, Украина, e-mail: sergey_gubka@mail.ru.

**Губка Алексей Сергеевич** – канд. техн. наук, доцент, доцент кафедры "Информационные управляющие системы", Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков, Украина, e-mail: gubka@gala.net.

**Носова Наталия Юрьевна** – аспирант кафедры "Информационные управляющие системы", Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков, Украина, e-mail: kroha2004@mail.ru.