УДК 512.624.95

# ИСТОРИЧЕСКИЕ МЕТОДЫ ШИФРОВАНИЯ: КРИПТОСИСТЕМЫ ХИЛЛА И КАРДАНО

*Державская Е. В., Линник М. В*

*Белорусский государственный университет информатики и радиоэлектроники*
*г. Минск, Республика Беларусь*

*Малышева О. Н. – канд. физ.-мат. наук*

**Аннотация.** В статье рассматриваются криптосистемы Хилла и Кардано, их принципы работы и особенности. Представлен способ их комбинирования для создания более надёжного метода шифрования и построения криптографической хеш-функции. Описана реализация разработанной на основе этого способа хеш-функции на языке C++, включающая поддержку расширенного алфавита.

**Ключевые слова.** Криптосистемы Хилла и Кардано, хеш-функция.

Первые методы шифрования появились около четырёх тысяч лет назад, и с этого времени они, меняясь и развиваясь, помогали людям безопасно передавать важные сведения. Статья посвящена одним из самых популярных таких методов: криптосистемам Хилла и Кардано.

Криптосистема Кардано [1], представляющая собой метод перестановки символов в сообщении с помощью квадратной (иногда прямоугольной) решётки размером $2m * 2k$ , была разработана итальянским ученым XVI века Джероламо Кардано и получила в его честь своё название. Четверть ячеек в решётке должна быть вырезана таким образом, чтобы при наложении её на чистый лист бумаги четырьмя возможными способами, определёнными заранее, вырезы полностью покрывали всю площадь листа. Передаваемое таким способом сообщение должно содержать столько же букв, сколько в решётке ячеек: $4m * k$. Буквы сообщения последовательно вписываются в вырезы решётки, заполняя буквенную матрицу, каждый столбец которой записывается в строку. Иногда пустые поля заполняются случайными символами, и такой способ шифровки текста называют «решётка Кардано с мусором», при этом дешифровать текст, записанный таким образом, не имея решётки-ключа, становится гораздо сложнее. К тому же этот способ позволяет не подбирать фразы, содержащие определённое количество букв, что также значительно упрощает процесс зашифровки.

Гораздо позже, в 1929 году, американским математиком Лестером Хиллом был изобретён первый шифр, построенный на синтезе модульной арифметики и линейной алгебры [1]. Все вычисления производятся на множестве целых $Z/m\,Z$, являющемся кольцом, на котором определены операции сложения, вычитания и умножения по модулю $m$.

Опишем процесс шифрования сообщения методом Хилла. Сначала сообщение разбивается на блоки по $n$ символов, то есть формируются векторы $x$. Зашифрованный текст получается умножением матрицы-ключа $T$ на вектор $x$, при этом ключ должен быть обратимым, то есть для него существует $\text{матрица}T^{-1}$, такая что $T * T^{-1} = I$ (единичная матрица), по модулю $m$. Зашифровка текста является, по сути, линейным отображением в кольце $Z/m\,Z$ по модулю $m$. Расшифровывать текст нужно тоже блоками по $n$ символов, умножая их на обратную по модулю $m$ к ключу матрицу $T^{-1}$.

Предложенный нами метод шифрования состоит в комбинировании описанных криптосистем. Были созданы две его разновидности: одна для шифрования вручную, а другая для создания криптографической хеш-функции.

Комбинирование методов использует последовательное шифрование исходного сообщения методом Хилла, а затем – перемешивание полученного криптотекста методом Кардано с добавлением мусора. Поскольку этот способ использует больше одной криптосистемы, то даже в случае попадания одного из ключей «не в те руки» сообщение остаётся надёжно зашифрованным.

Криптографическая хеш-функция – это алгоритм, который принимает на вход сообщение и превращает его в уникальный массив фиксированного размера. Такой массив называется хешем, а сам процесс – хешированием. Для каждого сообщения алгоритм создаёт свой уникальный хеш. Если заменить в исходных данных хотя бы одну букву, хеш изменится до неузнаваемости. Ключевое свойство хеш-функции – необратимость. Тогда, используя метод Хилла, можно написать хеш-функцию, подобрав такую матрицу-ключ, обратной к которой не существует, что сделает процесс дешифровки невозможным.

Основная суть создания хеш-функции заключается в применении метода Хилла для преобразования входной строки в числовые представления, которые, в свою очередь, подвергаются дополнительному перемешиванию и суммированию для получения фиксированного хеш-значения.

Последующее применение решётки Кардано добавляет элемент перестановки и пространственного распределения в хеш-функцию, увеличивая итоговую эффективность шифрования. Полученный таким образом хеш невозможно дешифровать, если выполняется условие:

$$\text{НОД}(detT, m) \neq 1,$$

где $detT$ - определитель матрицы-ключа, m - данный модуль.

На языке программирования C++ мы разработали алгоритм создания хеш-функции, совмещающий оба криптографических метода. Разберём его ключевые моменты.

Мы добавили поддержку двух алфавитов (рисунок 1), а также специальных символов, так что у каждого символа имеется свой порядковый номер.

```
const std::string RALPHABET = "АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ_.,!";
const std::string EALPHABET = "ABCDEFGHIJKLMNOPQRSTUVWXYZ_.,!";
```

Рисунок 1 – Расширенные алфавиты для русского и английского языков

Также мы задали ключевые матрицы для методов Хилла и Кардано (рисунок 2). Матрица для метода Хилла, как было сказано ранее, не будет обратимой. Решётка Кардано представлена в виде матрицы нулей и единиц, где единицы — это прорези. Для ее использования были написаны функции отражения решётки по вертикали и горизонтали (рисунок 3).

```
// Ключевая матрица 8×8 для хеширования (используется в методе Хилла)
const int KEYHill[8][8] = {
    {2, 3, 1, 0, 8, 5, 7, 2},
    {1, 2, 5, 3, 4, 6, 2, 7},
    {4, 1, 2, 5, 3, 7, 6, 2},
    {3, 4, 6, 2, 7, 1, 5, 3},
    {5, 6, 3, 7, 2, 4, 1, 6},
    {2, 5, 4, 1, 1, 2, 3, 4},
    {7, 1, 2, 1, 3, 5, 4, 1},
    {10, 7, 6, 3, 1, 2, 5, 2}
};

// Решётка Кардано 4x4
const int KEYCardano[4][4] = {
    {0, 1, 0, 0},
    {0, 0, 0, 1},
    {0, 1, 0, 0},
    {1, 0, 0, 0},
};
```

Рисунок 2 – Ключевые матрицы для методов Хилла и Кардано

```
void mirrorVertical(int g[4][4]) {
    for (int i = 0; i < 4; ++i)
        for (int j = 0; j < 2; ++j)
            std::swap(g[i][j], g[i][3 - j]);
}

void mirrorHorizontal(int g[4][4]) {
    for (int i = 0; i < 2; ++i)
        for (int j = 0; j < 4; ++j)
            std::swap(g[i][j], g[3 - i][j]);
}
```

Рисунок 3 – Функции отражения решётки Кардано по вертикали и горизонтали

Функция хеширования методом Хилла (рисунок 4) шифрует переведённую в числовой вектор и дополненную до кратного восьми строку, обрабатывая её описанным нами методом. Следом за ней идёт функция перемешивания методом Кардано (рисунок 5), которая переставляет числа в векторе, дополнительно усложняя шифрование.

```cpp
// Хеширование методом Хилла
std::vector<int> hillHash(const std::string& text, const std::string& lang) {
    int MOD = (lang == "1") ? 36 : 30;
    std::vector<int> nums = textToNums(text, lang);
    if (nums.empty()) return {};

    std::vector<int> hashhill(HASH_SIZE, 0);

    // Обрабатываем вход по блокам по 8 чисел
    for (size_t i = 0; i < nums.size(); i += 8) {
        int x[8];
        for (int j = 0; j < 8; ++j)
            x[j] = nums[i + j];

        int result[8] = { 0 };
        for (int row = 0; row < 8; ++row) {
            for (int col = 0; col < 8; ++col) {
                result[row] += KEYHill[row][col] * x[col];
            }
            result[row] = abs(result[row] % MOD);
        }

        for (int j = 0; j < 8; ++j) {
            hashhill[j] = (hashhill[j] + result[j]) % MOD;
            hashhill[j + 8] = (hashhill[j + 8] * 2 + result[j]) % MOD;
        }
    }
    return hashhill;
}
```

Рисунок 4 – Функция шифрования методом Хилла

```cpp
// Функция Кардано возвращает вектор из 16 чисел
std::vector<int> cardano(const std::vector<int>& hillHash) {
    int matrix[4][4] = { 0 };  // Здесь будем хранить итоговые числа
    int g[4][4];
    memcpy(g, KEYCardano, sizeof(KEYCardano));

    int index = 0;
    for (int step = 0; step < 4; step++) {
        for (int i = 0; i < 4; ++i)
            for (int j = 0; j < 4; ++j)
                if (g[i][j] == 1)
                    matrix[i][j] = hillHash[index++];

        if (step % 2 == 0)
            mirrorVertical(g);
        else
            mirrorHorizontal(g);
    }


    std::vector<int> result;
    // Читаем матрицу столбцами
    for (int j = 0; j < 4; ++j)
        for (int i = 0; i < 4; ++i)
            result.push_back(matrix[i][j]);

    return result;
}
```

Рисунок 5 – Функция шифрования методом Кардано

На рисунке 6 продемонстрированы результаты работы полной программы. Покажем, что хеш-функция работает корректно: если дважды ввести одни и те же данные, получим одинаковый хеш, что говорит об его однозначности. Если даже незначительно изменить входные данные, например, ввести "ПРИВЕТ,_МИР" без восклицательного знака, то результат хеширования кардинально изменится.

```
Введите язык:
 1 - русский
 2 - английский
 Или нажмите 0, чтобы завершить работу программы:
1
Введите фразу (в верхнем регистре, без Ё, если ввод на русском, с _ вместо пробела): ПРИВЕТ,_МИР!
Hash8x8 (128 бит): [ 28 8 7 18 12 27 0 17 15 30 28 30 26 21 8 2 ]
Введите язык:
 1 - русский
 2 - английский
 Или нажмите 0, чтобы завершить работу программы:
1
Введите фразу (в верхнем регистре, без Ё, если ввод на русском, с _ вместо пробела): ПРИВЕТ,_МИР!
Hash8x8 (128 бит): [ 28 8 7 18 12 27 0 17 15 30 28 30 26 21 8 2 ]
Введите язык:
 1 - русский
 2 - английский
 Или нажмите 0, чтобы завершить работу программы:
1
Введите фразу (в верхнем регистре, без Ё, если ввод на русском, с _ вместо пробела): ПРИВЕТ,_МИР
Hash8x8 (128 бит): [ 14 6 33 14 12 25 26 11 1 24 26 26 26 15 6 32 ]
```

Рисунок 6 – Результаты работы программы

Криптосистемы Хилла и Кардано представляют собой два фундаментально различных, но исторически значимых подхода к обработке и защите данных. Объединение этих методов в рамках экспериментальной хеш-функции, представленной выше, выявляет потенциал таких систем в современных прикладных задачах, особенно в образовательной сфере, позволяя сформировать понимание того, как простые математические конструкции могут быть объединены в более сложные криптографические схемы, а также использованы в хешировании.

***Список использованных источников:***

*1. Алферов А.П. Основы криптографии: Учеб. пособ. / А.П. Алферов, А.Ю. Зубов, А.С. Кузьмин [и др.]. – 2-е изд., испр. и доп. – М.: Гелиос АРВ, 2002.*

*2. Обзор шифра Хилла [Электронный ресурс]. Режим доступа: https://habr.com/ru/articles/710890/. Дата доступа: 03.03.2025*

UDC 512.624.95

# HISTORICAL ENCRYPTION METHODS: HILL AND CARDANO CRYPTOSYSTEMS

*Derzhavskaya E. V., Linnik M. V.*

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

*Malysheva O. N. – PhD in Phisics and Mathematics, Associate Professor*

**Annotation.** The article describes the Hill's and Cardano's cryptosystems, their principles of operation and features. A method of combining them is presented to create a more reliable encryption method and construct a cryptographic hash function. The implementation of the hash function developed on the basis of this method in the C++ language, including support for the extended alphabet, is described.

**Keywords.** Hill and Cardano cryptosystems, hash function.