МЕТОДЫ И СИСТЕМЫ ЗАЩИТЫ ИНФОРМАЦИИ, ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ / METHODS AND SYSTEMS OF INFORMATION PROTECTION, INFORMATION SECURITY

DOI: https://doi.org/10.60797/IRJ.2024.147.15

РАЗРАБОТКА ПРОГРАММНОГО КОМПЛЕКСА ПОТОКОВОГО ШИФРА RC4 ДЛЯ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ «КРИПТОГРАФИЯ»

Научная статья

Шарипов Р.Р.[1, *], Макаров С.П.[2], Кассирова А.А.[3]

[1, 2, 3] Казанский национальный исследовательский технический университет им. А. Н. Туполева - КАИ, Казань, Российская Федерация

* Корреспондирующий автор (riphat[at]mail.ru)

Аннотация

В данной статье актуализирована криптография как эффективное средство защиты данных. Для ускорения времени шифрования и расшифрования данных актуальны потоковые шифры, реализованные на РСЛОС, но у этих типов генераторов недостаточная криптостойкость. Рассматривается алгоритм RC4, представлен алгоритм, а также представлена программная реализация алгоритма на языке C++/CLI. Дается подробное описание исходного кода и графического интерфейса, разработанного программного комплекса и проверяется корректность его работы. Проведена демонстрация работы комплекса: задана кодировка символов, заданы массивы и получены S-блоки, представлено шифрование сообщения и получена шифрограмма. Разработанный программный комплекс и представленные алгоритмы могут быть использованы в учебном процессе в рамках дисциплины «Криптография» для обучающихся по направлению «Информационная безопасность».

Ключевые слова: криптография, RC4, программный комплекс, потоковое шифрование, РСЛОС, исходный код, шифрование, расшифрование, обучающиеся, информационная безопасность.

DEVELOPMENT OF RC4 STREAM CIPHER SOFTWARE FOR STUDENTS OF "CRYPTOGRAPHY" DISCIPLINE

Research article

Sharipov R.R.[1, *], Makarov S.P.[2], Kassirova A.A.[3]

[1, 2, 3] Kazan National Research Technical University named after A.N. Tupolev–KAI, Kazan, Russian Federation

* Corresponding author (riphat[at]mail.ru)

Abstract

This article actualizes cryptography as an effective means of data protection. Stream ciphers implemented on LFSR are relevant to speed up the time of encryption and decryption of data, but these types of generators have insufficient cryptographic strength. The RC4 algorithm is discussed, the algorithm is presented, and the software implementation of the algorithm in C++/CLI is presented. A detailed description of the source code and graphical interface of the developed software complex is given and the correctness of its operation is checked. Demonstration of the complex operation is carried out: character encoding is set, arrays are set and S-blocks are obtained, message encryption is presented and cryptogram is received. The developed software complex and the presented algorithms can be used in the educational process in the framework of the discipline "Cryptography" for students in the direction of "Information Security".

Keywords: cryptography, RC4, software complex, stream encryption, LFSR, source code, encryption, decryption, students, information security.

Введение

На сегодняшний день криптография – это одно из надёжных направлений при защите информации. Эффективные криптографические системы делают данные защищёнными от нарушения их конфиденциальности и целостности. Современные алгоритмы очень криптостойкие и шифротексты не поддаются взлому, включающие различные методы криптоанализа. Известно, что алгоритмы бывают симметричные (одноключеные) и ассиметричные (двухключевые) [1]; ассиметричные используются, например, для идентификации отправителя данных или же для защиты от изменений данных при передаче, симметричные для шифрования самих данных, которые передаются по каналу. Кроме того, к системам передачи данных предъявляются требования по скорости обработки данных и передачи их [2]. Однако в реальных системах, учитывая различные сложности современных симметричных алгоритмов шифрования, падает скорость обработки и передачи данных. Например, в известном алгоритме Advanced Encryption Standard (AES) данных разбиваются по 128 битовые блоки и проходят 10 итерационных шагов табличной замены байтов, смещения строк, смешивания столбцов и добавления раундового ключа [3]. Всё это требует затрат вычислительных ресурсов.

Для ускорения процессов криптографического преобразования данных в реальных системах используют поточные шифры. Потоковый шифр – это симметричный шифр, который шифрует каждый символ открытого текста в зависимости от ключа и его расположения в открытом тексте. Для работы таких шифров необходима псевдослучайная последовательность чисел (гамма), которая накладывается на текст передаваемого сообщения [4]. Обычно для генерации гаммы используется регистр сдвига с линейной обратной связи (РСЛОС). Генератор псевдослучайной последовательности, основанный на РСЛОС, является хорошим, так как перед началом работы в регистр заносится случайная последовательность бит, однако не обладает достаточной криптостойкостью из-за линейных связей [5].

Более распространённым является шифр – RC4. Известно, что потоковый шифр RC4 применяется в различных сетевых протоколах безопасности, таких как SSL и TLS. Также RC4 используется для обеспечения безопасности в беспроводных сетях Wi-Fi, в алгоритмах WEP и WPA. Главными преимуществами RC4 являются высокая скорость работы и программной, и аппаратной реализаций, и переменная длинна ключа [6]. Длина ключа может варьироваться от 40 до 2048 бит. Поэтому также в RC4 для генерации гаммы используется алгоритм «key-scheduling algorithm», сокращено KSA [7].

Целью данной статья является разработка учебного комплекса для изучения алгоритма RC4 обучающимся. Для достижения поставленной цели необходимо выполнить следующие задачи:

– исследовать алгоритм RC4 и представить пошаговые операции;

– разработать программную реализацию алгоритма RC4 с графической формой ввода данных и получения результатов;

– провести демонстрацию работы разработанного комплекса, получить результаты и перепроверить их.

Рассмотрим более подробнее алгоритм RC4.

Алгоритм RC4

Введем следующие обозначения:

- *S[i]* – значение i-ого S-блока;
- *Key[i]* – значение i-ого элемента ключевой последовательности;
- *L* – длина ключевой последовательности;
- *a mod b* – остаток от деления a на b.

RC4 состоит из следующих шагов [8]:

1. Массива блоков S. Блок S содержит одно из 2^n, где n это Размерность слова, используемого в алгоритме. Значение n также определяет размер S-блока. Размерность массива S-блоков равна 2^n.
2. Двух внутренних счетчиков, обозначаемых как i и j.
3. Ключевой последовательности Key.
4. Псевдослучайного слова K.

Описание двух этапов работы алгоритма потокового шифра RC4.

Этап первый – алгоритм KSA:

1. Сначала все S-блоки заполняются значениями от 0 до 2^n-1:

$$S[i] = i \text{ , где } \overline{i = 0, 2^n - 1}$$

2. Затем значения S-блоков перемешивается с помощью ключевой последовательности Key:

– Сначала обнулим счетчик j:

$$j = 0$$

– Затем выполним следующие действия для $\overline{i = 0, 2^n - 1}$

$$j = (j + S[i] + Key[i \bmod L]) \bmod 2^n$$

3. Меняем местами S[i] и S[j].

Этап второй – генерация псевдослучайного слова K:

1. Сначала обнулим счетчики i и j:

$$j = 0$$
$$i = 0$$

2. Затем следующие действия выполняются циклически необходимо число раз:

$$i = (i + 1) \bmod 2^n$$
$$j = (j + S[i]) \bmod 2^n$$

3. Меняем местами S[i] и S[j]

$$t = (S[i] + S[j]) \bmod 2^n$$
$$K = S[t]$$

В *K* будет находиться сгенерированное псевдослучайное слово, которое накладывается на символ текста, с помощью XOR, для шифровки или расшифровки.

Программная реализация алгоритма RC4

Для наглядной демонстрации работы алгоритма RC4 было разработано программное обеспечение «RC4», которое эмулирует работу потокового шифра RC4, для n = 6. Программа разработана на языке C++/CLI.

На рис. 1 и рис. 2 представлен исходный код основной функции разработанной программы. Где:

– RC4_Funct – функция эмулирующая работу алгоритма RC4;
– i и j – внутренние счетчики RC4;
– L – длина ключевой последовательности;
– S[64] – массива S-блоков;
– CopyMes – строка, содержащая значение сообщения в кодировке (рис. 5);
– CopyKey – строка, содержащая значение ключа в кодировке с (рис. 5).

```
// Функция RC4_Fucnt эмулирует работу алгоритма RC4 для n = 6
// Функция шифрует значение поля «Сообщение» и выводит результат в поле «Результат»
// Генерируемая псевдослучайная последовательность, выводится в поле «Генерируемая последовательность»
private: System::Void RC4_Funct(System::Object^ sender, System::EventArgs^ e) {

    // Перед началом шифрования, программа проверяет, что поля «Сообщение» и «Ключ» не пустые,
    // и не содержат запрещенных символов.
    if (KeyText->Text->Length && MessageText->Text->Length && ValidationOfInput()) {
        // Объявление двух внутренних счетчиков RC4
        unsigned char i = 0;
        unsigned char j = 0;
        // Объявление переменной, хранящей длину ключевой последовательности
        unsigned char L = KeyText->Text->Length;
        // Объявление пустого массива S-блоков
        unsigned char S[64];
        // Объявление массива, для хранения значение поля «Сообщение», после перевода в
        // алфавит alf.
        unsigned char CopyMes[1000];
        // Объявление массива, для хранения значение поля «Ключ», после перевода в
        // алфавит alf.
        unsigned char CopyKey[1000];

        // Перевод значения поля «Сообщение» в алфавит alf
        for (i = 0; i < MessageText->Text->Length; i++)
            for (j = 0; j < 64; j++)
                if (alf[j] == MessageText->Text[i])
                    CopyMes[i] = j;

        // Перевод значения поля «Ключ» в алфавит alf
        for (i = 0; i < KeyText->Text->Length; i++)
            for (j = 0; j < 64; j++)
                if (alf[j] == KeyText->Text[i])
                    CopyKey[i] = j;

        // Заполнение массива S-блоков значениями от 0 до 63 включительно
        for (i = 0; i < 64; i++)
            S[i] = i;

        // Перемешивание массива S-блоков с помощью ключевой последовательности
        for (i = 0; i < 64; i++) {
            j = (j + S[i] + CopyKey[i % L]) % 64;
            unsigned char copy = S[i];
            S[i] = S[j];
            S[j] = copy;
        }
```

Рисунок 1 - Первая часть кода функции RC4_Funct
DOI: https://doi.org/10.60797/IRJ.2024.147.15.1

```
        // Обнуление счетчиков
        i = 0;
        j = 0;

        // Очищение полей «Результат» и «Генерируемая последовательность»
        // перед выводом новых значений
        if (ResultText->Text->Length != 0) {
            ResultText->Text = ResultText->Text->Remove(0);
            GammaText->Text = GammaText->Text->Remove(0);
        }

        // Цикл, в котором шифруется значение поля «Сообщение»
        for (int k = 0; k < MessageText->Text->Length; k++) {

            i = (i + 1) % 64;
            j = (j + S[i]) % 64;

            unsigned char copy = S[i];
            S[i] = S[j];
            S[j] = copy;

            unsigned char t = (S[i] + S[j]) % 64;

            // Вывод очередного зашифрованного символа на экран в поле «Результат»
            ResultText->Text += alf[CopyMes[k] ^ S[t]];

            // Вывод очередного сгенерированного числа псевдослучайной последовательности
            // в поле «Генерируемая последовательность»
            GammaText->Text += S[t];
            GammaText->Text += " ";
        }
    }
}
```

Рисунок 2 - Вторая часть кода функции RC4_Funct
DOI: https://doi.org/10.60797/IRJ.2024.147.15.2

Кодировка на рис. 5 хранится в программе в виде переменной alf (рис. 3). Работа данной функции ничем не отличается от работы алгоритма, представленного выше, кроме двух циклов в начале, которые переводят значения ключа и сообщения из кодировки UTF-16, используемой внутри программы, в кодировку (рис. 5).

```
// Переменная alf содержит алфавит, который используется в программе для шифрования сообщения.
private: String^ alf = gcnew String(L" .0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ");
```

Рисунок 3 - Алфавит в программе
DOI: https://doi.org/10.60797/IRJ.2024.147.15.3

На рис. 4 представлено рабочее окно «RC4», где:

1. «Сообщение» – это поле предназначено для ввода текста, которое шифруется или расшифровывается.
2. «Ключ» – это поле предназначено для ввода ключа, который используется при шифровании или расшифровании введенного в поле «Сообщение» текста.
3. «Результат» – это поле предназначено для вывода результата шифрования или расшифрования текста, введенного в поле «Сообщение».
4. «Генерируемая последовательность» – это поле предназначено для вывода псевдослучайной последовательности, которая генерируется при шифровании или расшифровании.
5. «Зашифровать/Расшифровать» – это кнопка, при нажатии на которую происходит шифрование или расшифровка текста, введенного в поле «Сообщение».
6. «Выход» – это кнопка, при нажатии на которую приложение завершает свою работу.

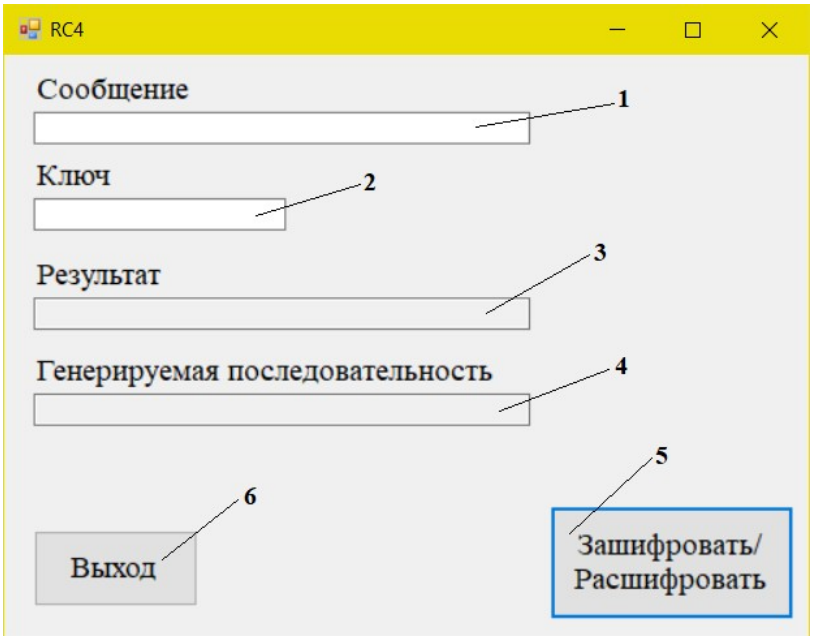

Рисунок 4 - Окно программы «RC4»:
1 - поле для ввода текста; *2* - поле для ввода ключа шифрования; *3* - поле для вывода результата; *4* - поле для вывода псевдослучайной последовательности; *5* - кнопка действия; *6* - кнопка завершения работы
DOI: https://doi.org/10.60797/IRJ.2024.147.15.4

Демонстрация работы комплекса

Для демонстрации работы алгоритма и проверки корректности работы программы проведем эксперимент и зашифруем сообщение «*MSP*», при помощи ключа «*Key*». В качестве кодировки символов будем использовать алфавит, представленный на рис. 5.

Символ	Код	Символ	Код	Символ	Код	Символ	Код
«Пробел»	0	'e'	16	'u'	32	'K'	48
'.'	1	'f'	17	'v'	33	'L'	49
'0'	2	'g'	18	'w'	34	'M'	50
'1'	3	'h'	19	'x'	35	'N'	51
'2'	4	'I'	20	'y'	36	'O'	52
'3'	5	'j'	21	'z'	37	'P'	53
'4'	6	'k'	22	'A'	38	'Q'	54
'5'	7	'l'	23	'B'	39	'R'	55
'6'	8	'm'	24	'C'	40	'S'	56
'7'	9	'n'	25	'D'	41	'T'	57
'8'	10	'o'	26	'E'	42	'U'	58
'9'	11	'p'	27	'F'	43	'V'	59
'a'	12	'q'	28	'G'	44	'W'	60
'b'	13	'r'	29	'H'	45	'X'	61
'c'	14	's'	30	'I'	46	'Y'	62
'd'	15	't'	31	'J'	47	'Z'	63

Рисунок 5 - Алфавит

DOI: https://doi.org/10.60797/IRJ.2024.147.15.5

Этап первый:

Массив S-блоков заполняется значение от 0 до 63. Далее счетчики j и i обнулятся и значения массива перемешиваются с помощью ключа:

$$S[i] = i \text{ , где } i = \overline{0,63}$$
$$i = 0$$
$$j = 0$$

Далее для всех значений $i = \overline{0,63}$ выполняется следующее:

$$j = (j + S[i] + Key[i \ mod \ 3]) \ mod \ 64$$

S[i] и S[j] меняются местами.

Рассмотрим несколько итераций:

При i=0:

$$j = (0 + S[0] + Key[0 \ mod \ 3]) \ mod \ 64 = (0 + 0 + 48) \ mod \ 64 = 48$$

S[0] и S[48] меняются местами.

При i=1:

$$j = (48 + S[1] + Key[1 \ mod \ 3]) \ mod \ 64 = (48 + 1 + 16) \ mod \ 64 = 1$$

S[1] и S[1] меняются местами.

При i=2:

$$j = (1 + S[2] + Key[2 \ mod \ 3]) \ mod \ 64 = (1 + 2 + 36) \ mod \ 64 = 39$$

S[2] и S[39] меняются местами.

И так далее. В конце перемешивания в массиве S-блоков будет находиться последовательность, представленная на рис. 6.

S[0] = 40	S[16] = 43	S[32] = 17	S[48] = 0
S[1] = 33	S[17] = 11	S[33] = 60	S[49] = 61
S[2] = 39	S[18] = 27	S[34] = 4	S[50] = 15
S[3] = 26	S[19] = 57	S[35] = 1	S[51] = 46
S[4] = 34	S[20] = 42	S[36] = 62	S[52] = 2
S[5] = 23	S[21] = 22	S[37] = 30	S[53] = 31
S[6] = 24	S[22] = 38	S[38] = 3	S[54] = 36
S[7] = 54	S[23] = 6	S[39] = 16	S[55] = 45
S[8] = 52	S[24] = 13	S[40] = 48	S[56] = 9
S[9] = 12	S[25] = 21	S[41] = 7	S[57] = 58
S[10] = 35	S[26] = 47	S[42] = 56	S[58] = 50
S[11] = 18	S[27] = 32	S[43] = 10	S[59] = 8
S[12] = 37	S[28] = 44	S[44] = 49	S[60] = 53
S[13] = 28	S[29] = 5	S[45] = 20	S[61] = 53
S[14] = 29	S[30] = 14	S[46] = 51	S[62] = 55
S[15] = 19	S[31] = 25	S[47] = 59	S[63] = 41

Рисунок 6 - Значения S-блоков после перемешивания

DOI: https://doi.org/10.60797/IRJ.2024.147.15.6

Этап второй:

Обнуляем счетчики i и j.

$$j = 0$$
$$i = 0$$

И вычисляем значения K_1.

$$i = (0 + 1) \ mod \ 64 = 1$$

$$j = (0 + S[1])\ mod\ 64 = (0 + 33)\ mod\ 64 = 33$$

Меняем местами S[1] и S[33]

$$t = (S[1] + S[33])\ mod\ 64 = (60 + 33)\ mod\ 64 = 29$$
$$K_1 = S[29] = 5$$

Вычисляем значения K_2.

$$i = (1 + 1)\ mod\ 64 = 2$$
$$j = (33 + S[2])\ mod\ 64 = (33 + 39)\ mod\ 64 = 8$$

Меняем местами S[2] и S[8]

$$t = (S[2] + S[8])\ mod\ 64 = (39 + 52)\ mod\ 64 = 27$$
$$K_2 = S[27] = 32$$

Вычисляем значения K_3.

$$i = (2 + 1)\ mod\ 64 = 3$$
$$j = (8 + S[3])\ mod\ 64 = (8 + 26)\ mod\ 64 = 34$$

Меняем местами S[3] и S[34]

$$t = (S[3] + S[34])\ mod\ 64 = (26 + 4)\ mod\ 64 = 30$$
$$K_3 = S[30] = 14$$

Шифруем сообщение с помощью полученных значений:

$$«M» \ XOR\ K_1 = 50\ XOR\ 5 = 55 = «R»$$
$$«S» \ XOR\ K_2 = 56\ XOR\ 32 = 24 = «m»$$
$$«P» \ XOR\ K_3 = 53\ XOR\ 14 = 59 = «V»$$

Для проверки результата зашифруем сообщение «MSP» ключом «Key» используя разработанное программное обеспечение (рис. 7).

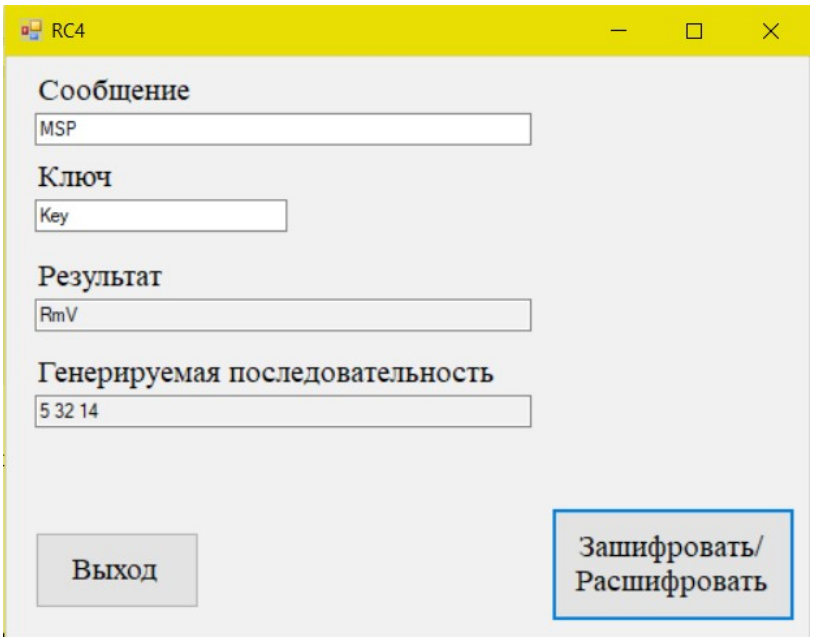

Рисунок 7 - Результат работы приложения «RC4»
DOI: https://doi.org/10.60797/IRJ.2024.147.15.7

В итоге, результаты работы приложения «RC4» совпадает с ручными расчетами:
– K_1=5;
– K_2=32;
– K_3=14;
– зашифрованное сообщение – «RmV».

Кроме этого, мы можем и легко расшифровать сообщение с исходным ключом.

Результатом демонстрации является верность ручного шифрования данных, поскольку они идентичны с результатом программного комплекса «RC4» при идентичных входных значениях.

Обсуждение

Новизной данной работы является программная реализация алгоритма RC4. Программный комплекс «RC4» является оригинальным решением с удобным интерфейсом для ввода-вывода данных, а представленный алгоритм позволяет шифровать данные вручную с целью получения практических навыков обучающимися.

Существуют и подобные работы [9], [10], [12], [13], но в других научных направлениях и во время разработки нашего программного комплекса мы опирались на некоторые рекомендации, предложенные в них, и актуализировали их. Разработанный программный комплекс «RC4» имеет удобный интерфейс (рис.4) для ввода входных данных и получения результата работы программы (рис.7).

Несмотря на то что данный алгоритм уже имеет несколько модификаций, разработанный комплекс может быть использован в учебном процессе в рамках учебной дисциплины – «Криптография» по направлению «Информационная

безопасность». Обучающие сначала изучают алгоритм RC4, задают входные параметры, формируют таблицы данных и, выбрав необходимый по длине ключ шифрования, формируют ключевую последовательность. После этого выбирают открытый текст и выполняют шифрование выработанной на предыдущем этапе ключевой последовательностью. После получения шифрограммы обучающие, используя программный комплекс «RC4», проверяет результаты своей работы и если результат расчёта и «RC4» совпадут, то обучающиеся верно выполнили расчеты. Проведённые тесты показали корректность работы разработанного приложения.

Заключение

Проведенные исследования показали, что разработанное приложение «RC4» может корректно эмулировать работу потокового шифра RC4, выдавая результат, совпадающий с рассчитанными вручную данными. Это доказывает, что данное приложение может быть использовано в учебном процессе для формирования у обучающихся компетенций в области потокового шифрования. Обучение с помощью разработанного программного комплекса и методических рекомендаций позволит обучающимся лучше понять принципы и методы, лежащие в основе алгоритма RC4. Таким образом, представленный в статье программный комплекс может быть использован в учебном процессе в рамках дисциплины «Криптография» для обучающихся по направлению «Информационная безопасность».

Конфликт интересов

Не указан.

Рецензия

Все статьи проходят рецензирование. Но рецензент или автор статьи предпочли не публиковать рецензию к этой статье в открытом доступе. Рецензия может быть предоставлена компетентным органам по запросу.

Conflict of Interest

None declared.

Review

All articles are peer-reviewed. But the reviewer or the author of the article chose not to publish a review of this article in the public domain. The review can be provided to the competent authorities upon request.

Список литературы / References

1. Гибадуллин Р.Ф. Потокобезопасные вызовы элементов управления в обогащенных клиентских приложениях / Р.Ф. Гибадуллин // Программные системы и вычислительные методы. — 2022. — № 4. — С. 1-19. — DOI: 10.7256/2454-0714.2022.4.39029.

2. Шарипов Р.Р. Методы анализа клавиатурного почерка пользователей с использованием эталонных гауссовских сигналов / Р.Р. Шарипов, А.С. Катасев, А.П. Кирпичников // Вестник Технологического университета. — 2016. — Т. 19. — № 13. — С. 157-160.

3. Buluş A. Cipher with AES / A. Buluş, E. Buluş // 2018 3rd International Conference on Computer Science and Engineering (UBMK). — IEEE, 2018. — P. 27-30.

4. Гибадуллин Р.Ф. Разработка и тестирование программных модулей для оценки производительности CUDA и OPENCL технологий / Р.Ф. Гибадуллин, Д.Д. Фирсова, Н.В. Кормильцев [и др.] // Вестник Технологического университета. —2018. — Т. 21. — № 9. — С. 171-175.

5. Макаров С.П. Разработка программного комплекса регистра сдвига с линейной обратной связью / С.П. Макаров, А.А. Кассирова // Актуальные проблемы науки и образования в условиях современных вызовов (шифр – МКАП 25) : Сборник материалов XXV Международной научно-практической конференции, Москва, 17 ноября 2023 года. — Москва: Печатный цех, 2023. — С. 27-34.

6. Гибадуллин Р.Ф. Анализ и модернизация защищенности стандарта IEEE 802.11i / Р.Ф. Гибадуллин, А.Р. Галимов, Н.В. Кормильцев [и др.] // Вестник Технологического университета. — 2018. — Т. 21. — № 8. — С. 100-108.

7. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы и исходный код на С / Б. Шнайер. — 2022.

8. Кассирова А.А. Исследование алгоритма "Берлекэмпа-Месси" на простых регистрах сдвига с линейной обратной связью / А.А. Кассирова, С.П. Макаров // Актуальные проблемы науки и образования в условиях современных вызовов (шифр – МКАП 25) : Сборник материалов XXV Международной научно-практической конференции, Москва, 17 ноября 2023 года. — Москва: Печатный цех, 2023. — С. 218-228.

9. Яганова А.А. Опыт проведения практических занятий с использованием систем электронного документооборота при подготовке студентов / А.А. Яганова // Вестник РГГУ. Серия: Документоведение и архивоведение. Информатика. Защита информации и информационная безопасность. — 2014. — № 2(124). — С. 258-262.

10. Грызлова М.С. Методика проведения интерактивного занятия на примере лекции-визуализации «Информационная безопасность в сети Интернет» / М.С. Грызлова // Информационная безопасность и вопросы профилактики киберэкстремизма среди молодежи : Материалы внутривузовской конференции, Магнитогорск, 09-12 октября 2015 года / Под ред. Г.Н. Чусавитиной, Е.В. Черновой, О.Л. Колобовой. — Магнитогорск: Магнитогорский государственный технический университет им. Г.И. Носова, 2015. — С. 175-183.

11. Бакулин В.М. Методика преподавания темы «Организация парольной защиты в файловых системах» для обучающихся нетехнических специальностей / В.М. Бакулин, Д.Л. Еськин // Современные наукоемкие технологии. — 2016. — № 8-2. — С. 290-293.

12. Гибадуллин Р.Ф. Реконструкция томографических снимков с применением многопроцессорных систем / Р.Ф. Гибадуллин, А.А. Максимов, А.А. Новиков [и др.] // Вестник Технологического университета. — 2017. — Т. 20. — № 12. — С. 87-89.

13. Шарипов Р.Р. Исследование электрических параметров пороговых извещателей / Р.Р. Шарипов, Б.З. Юсупов // Программные системы и вычислительные методы. — 2023. — № 3. — С. 29-47. — DOI: 10.7256/2454-0714.2023.3.43682.

Список литературы на английском языке / References in English

1. Gibadullin R.F. Potokobezopasnye vyzovy jelementov upravlenija v obogashhennyh klientskih prilozhenijah [Thread-safe calls of control elements in enriched client applications] / R.F. Gibadullin // Programmnye sistemy i vychislitel'nye metody [Software Systems and Computational Methods]. — 2022. — № 4. — P. 1-19. — DOI: 10.7256/2454-0714.2022.4.39029. [in Russian]

2. Sharipov R.R. Metody analiza klaviaturnogo pocherka pol'zovatelej s ispol'zovaniem jetalonnyh gaussovskih signalov [Methods of analysing the keyboard handwriting of users using reference Gaussian signals] / R.R. Sharipov, A.S. Katasev, A.P. Kirpichnikov // Vestnik Tehnologicheskogo universiteta [Bulletin of the University of Technology]. — 2016. — Vol. 19. — № 13. — P. 157-160. [in Russian]

3. Buluş A. Cipher with AES / A. Buluş, E. Buluş // 2018 3rd International Conference on Computer Science and Engineering (UBMK). — IEEE, 2018. — P. 27-30.

4. Gibadullin R.F. Razrabotka i testirovanie programmnyh modulej dlja ocenki proizvoditel'nosti CUDA i OPENCL tehnologij [Development and testing of software modules for performance evaluation of CUDA and OPENCL technologies] / R.F. Gibadullin, D.D. Firsova, N.V. Kormil'cev [et al.] // Vestnik Tehnologicheskogo universiteta [Bulletin of Technological University]. —2018. — Vol. 21. — № 9. — P. 171-175. [in Russian]

5. Makarov S.P. Razrabotka programmnogo kompleksa registra sdviga s linejnoj obratnoj svjaz'ju [Development of the software complex of the shift register with linear feedback] / S.P. Makarov, A.A. Kassirova // Aktual'nye problemy nauki i obrazovanija v uslovijah sovremennyh vyzovov (shifr – MKAP 25) : Sbornik materialov XXV Mezhdunarodnoj nauchno-prakticheskoj konferencii, Moskva, 17 nojabrja 2023 goda [Current problems of science and education in the conditions of modern challenges (cipher - ICAP 25) : Collection of materials of XXV International Scientific and Practical Conference, Moscow, 17 November 2023]. — Moscow: Print Shop, 2023. — P. 27-34. [in Russian]

6. Gibadullin R.F. Analiz i modernizacija zashhishhennosti standarta IEEE 802.11i [Analysis and modernization of IEEE 802.11i standard security] / R.F. Gibadullin, A.R. Galimov, N.V. Kormil'cev [et al.] // Vestnik Tehnologicheskogo universiteta [Bulletin of Technological University]. — 2018. — Vol. 21. — № 8. — P. 100-108. [in Russian]

7. Schneier B. Prikladnaja kriptografija. Protokoly, algoritmy i ishodnyj kod na C [Applied cryptography. Protocols, algorithms and source code in C] / B. Schneier. — 2022. [in Russian]

8. Kassirova A.A. Issledovanie algoritma "Berlekjempa-Messi" na prostyh registrah sdviga s linejnoj obratnoj svjaz'ju [Study of the algorithm "Berlecamp-Messy" on simple shift registers with linear feedback] / A.A. Kassirova, S.P. Makarov // Aktual'nye problemy nauki i obrazovanija v uslovijah sovremennyh vyzovov (shifr – MKAP 25) : Sbornik materialov XXV Mezhdunarodnoj nauchno-prakticheskoj konferencii, Moskva, 17 nojabrja 2023 goda [Topical problems of science and education in the conditions of modern challenges (cipher – MKAP 25) : Collection of materials of XXV International Scientific and Practical Conference, Moscow, 17 November 2023]. — Moscow: Print Shop, 2023. — P. 218-228. [in Russian]

9. Jaganova A.A. Opyt provedenija prakticheskih zanjatij s ispol'zovaniem sistem jelektronnogo dokumentooborota pri podgotovke studentov [Experience in conducting practical classes with the use of electronic document management systems in training students] / A.A. Jaganova // Vestnik RGGU. Serija: Dokumentovedenie i arhivovedenie. Informatika. Zashhita informacii i informacionnaja bezopasnost' [Bulletin of RSGU. Series: Documentology and Archival Science. Computer science. Information protection and information security]. — 2014. — № 2(124). — P. 258-262. [in Russian]

10. Gryzlova M.S. Metodika provedenija interaktivnogo zanjatija na primere lekcii-vizualizacii «Informacionnaja bezopasnost' v seti Internet» [Methodology of the interactive lesson on the example of lecture-visualization "Information security on the Internet"] / M.S. Gryzlova // Informacionnaja bezopasnost' i voprosy profilaktiki kiberjekstremizma sredi molodezhi : Materialy vnutrivuzovskoj konferencii, Magnitogorsk, 09-12 oktjabrja 2015 goda [Information security and issues of prevention of cyberextremism among young people : Proceedings of the Intrauniversity Conference, Magnitogorsk, 09-12 October 2015] / Ed. by G.N. Chusavitinva, E.V. Chernova, O.L. Kolobova. — Magnitogorsk: Magnitogorsk State Technical University named after G.I. Nosov, 2015. — P. 175-183. [in Russian]

11. Bakulin V.M. Metodika prepodavanija temy «Organizacija parol'noj zashhity v fajlovyh sistemah» dlja obuchajushhihsja netehnicheskih special'nostej [Methods of teaching the topic "Organization of password protection in file systems" for students of non-technical specialities] / V.M. Bakulin, D.L. Es'kin // Sovremennye naukoemkie tehnologii [Modern science-intensive technologies]. — 2016. — № 8-2. — P. 290-293. [in Russian]

12. Gibadullin R.F. Rekonstrukcija tomograficheskih snimkov s primeneniem mnogoprocessornyh sistem [Reconstruction of tomographic images using multiprocessor systems] / R.F. Gibadullin, A.A. Maksimov, A.A. Novikov [et al.] // Vestnik Tehnologicheskogo universiteta [Bulletin of Technological University]. — 2017. — Vol. 20. — № 12. — P. 87-89. [in Russian]

13. Sharipov R.R. Issledovanie jelektricheskih parametrov porogovyh izveshhatelej [Study of electrical parameters of threshold detectors] / R.R. Sharipov, B.Z. Jusupov // Programmnye sistemy i vychislitel'nye metody [Software systems and computational methods]. — 2023. — № 3. — P. 29-47. — DOI: 10.7256/2454-0714.2023.3.43682. [in Russian]