УДК 681.391

В. А. Пласковицкий, аспирант (БГТУ);
П. П. Урбанович, доктор технических наук, профессор,
заведующий кафедрой информационных систем и технологий (БГТУ)

ШИФРОВАНИЕ КОДОВ ПРОГРАММ НА ОСНОВЕ КЛЮЧА, ЗАДАВАЕМОГО РЕКУРРЕНТНЫМИ МАТЕМАТИЧЕСКИМИ СООТНОШЕНИЯМИ

Рассмотрен новый подход к решению проблемы защиты программных кодов на основе использования рекуррентных математических соотношений в качестве ключа зашифрования данных. Это позволяет избежать коллизий с однозначностью перевода шифруемых символов, а также снижает уязвимость метода, обусловленную длиной ключа. Основным моментом является использование нестандартных математических операций, рекуррентно зависящих от предыдущих и составных действий, с динамически изменяемыми параметрами.

A new approach to software protection codes through the use of recursive mathematical relations as a key to encrypt data is consided. This eliminates conflicts with the unambiguous transfer of encrypted characters, as well as reducethe vulnerability of the method due to key length. The key is to use a non-standard mathematical operations recursively depend on their previous actions and composite action with dynamic parameters.

Введение. Одной из актуальных проблем в области информационных технологий является защита электронных документов от несанкционированного использования. Известные решения этой проблемы основываются чаще всего на использовании криптографического шифрования либо на применении специальных алгоритмов, затрудняющих анализ документов, если таким документом является код программы. Последние методы получили название обфусцирующих [1].

В данной работе рассмотрен новый подход к решению проблемы, предполагающий использование рекуррентного математического соотношения в качестве ключа при шифровании данных. Это позволяет избежать коллизий с однозначностью перевода шифруемых символов, а также уязвимости, обусловленной длиной ключа [2]. Эта уязвимость приводит к периодическим повторениям шифруемых участков, что особенно актуально при шифровании программного кода, обладающего большим объемом данных и часто повторяющимися последовательностями символов [3].

Основная часть. Сущность предлагаемого здесь подхода состоит в следующем. К некоторому однозначному числовому определению шифруемого символа A (например, кодовому значению из используемой таблицы) добавляется некоторое число $F(X)$, полученное в результате вычисления неизвестного (третьей стороне) математического выражения. В качестве переменных X при этом могут выступать известные данные, например, порядковый номер символа (в шифруемом сообщении), кодовое значение предыдущего символа в этом сообщении:

$$A + F(X) = B, \qquad (1)$$

$$B - F(X) = A. \qquad (2)$$

Первое из приведенных соотношений применяется в режиме зашифрования, второе – расшифрования.

При использовании в качестве аргумента кодового значения текущего символа упомянутые выше преобразования можно представить соответственно так:

$$F(A + X) = B,\ F^{-1}(B) = A, \qquad (3)$$

где $F^{-1}(B)$ – обратное значение функции, которое нужно знать для расшифрования текста. Это требование ограничивает сложность ключа, но уменьшает количество известных для расшифрования данных.

Спецификой математической формулы является возможность получения одинаковых значений для ряда передаваемых переменных при различных наборах математических операций. Благодаря этому нельзя с уверенностью полагаться на то, что ключ найден, если с его помощью удалось расшифровать отдельный набор данных.

Приведем пример. Есть ключ в виде выражения $1 + X\%123$, где X – порядковый номер шифруемого символа. Второе слагаемое для первых 123 символов роли не играет, так как результатом вычисления будет число 0, как если бы ключ был просто единицей. Но, начиная с 123 символа, появляется дополнительный сдвиг, который нельзя предсказать, не зная настоящую формулу. Разумеется, в качестве такого дополнительного воздействия может выступать выражение любой сложности, дающее в результате 0 при определенных условиях, или наоборот.

На рис. 1 приведен пример изменения входных данных в зависимости от использования различных математических выражений.

Формула	1	2	3	4	5	6	7	8	9	10
1	1	1	1	1	1	1	1	1	1	1
x/5	0	0	0	0	1	1	1	1	1	2
sumDel(x)	1	2	3	6	5	11	7	14	12	17
sumDelM(x)	0	0	0	2	0	5	0	6	3	7
sin(x)*10	8	9	1	-7	-9	-2	6	9	4	-5
delModuleD (x;1.75)	1	0	1	1	1	0	2	1	0	0
max(x;3)% min(x;3)	0	1	0	1	2	0	1	2	0	1
x++ + x	3	5	7	9	11	13	15	17	19	21

Рис. 1. Сдвиг данных при различных математических соотношениях

Использование простых математических операций не представляет значительной сложности для операции криптоанализа (слома шифра). Поэтому важнейшей особенностью данного подхода являются динамические и рекуррентные преобразования.

Такие преобразования основаны на использовании в одних и тех же операциях значений, изменяющихся на основе предыдущих действий (операций).

Например, деление по модулю, где в качестве модуля выступает число, увеличиваемое каждый раз по определенному правилу. Или сама переменная, передаваемая в формулу, будет изменяться при каждом повторном использовании в заданном выражении.

Авторами разработано программное средство, позволяющее производить зашифрование и расшифрование сообщения на основе описанной методики (рис. 2).

Приложение состоит из текстовых полей для ввода шифруемых данных, ключа, отображения результата зашифрования и расшифрования, настроек процесса зашифрования, а также таблицы с выводом информации о численных значениях, поступающих на вход шифратора, ключевых вычислений в процессе зашифрования и результирующих значений.

На текущий момент реализована обработка около 40 различных операторов, среди которых есть как односоставные операторы (тригонометрические функции, округление, возведение в степень), так и сложные (вычисление суммы делителей числа, динамическое деление по модулю, изменение переменных).

Просмотреть доступные операторы можно по нажатию на кнопку «Список возможных операций», расположенную слева от поля для ввода формулы.

Процесс зашифрования является нетривиальным, состоящим из перевода текстового вида формулы в пригодный для использования компьютером вид. С учетом ряда особенностей в программе была специально разработана структура, хранящая отдельные блоки математической формулы, их взаимоотношения и другие характеристики, необходимые для реализации алгоритмов, которые основаны на рекуррентных и других динамических отношениях.

Значительная оптимизация процесса обработки данных (по критерию времени) происходит за счет поиска одинаковых участков математического выражения и вычисления их только один раз на протяжении зашифрования одного символа. Например, в качестве ключа используется формула вида: $(X + 1) + X^{X+1}$. Очевидно, что результат вычисления первого слагаемого можно использовать при нахождении второго слагаемого. Программа осуществляет поиск таких повторяющихся фрагментов в ключевой информации.

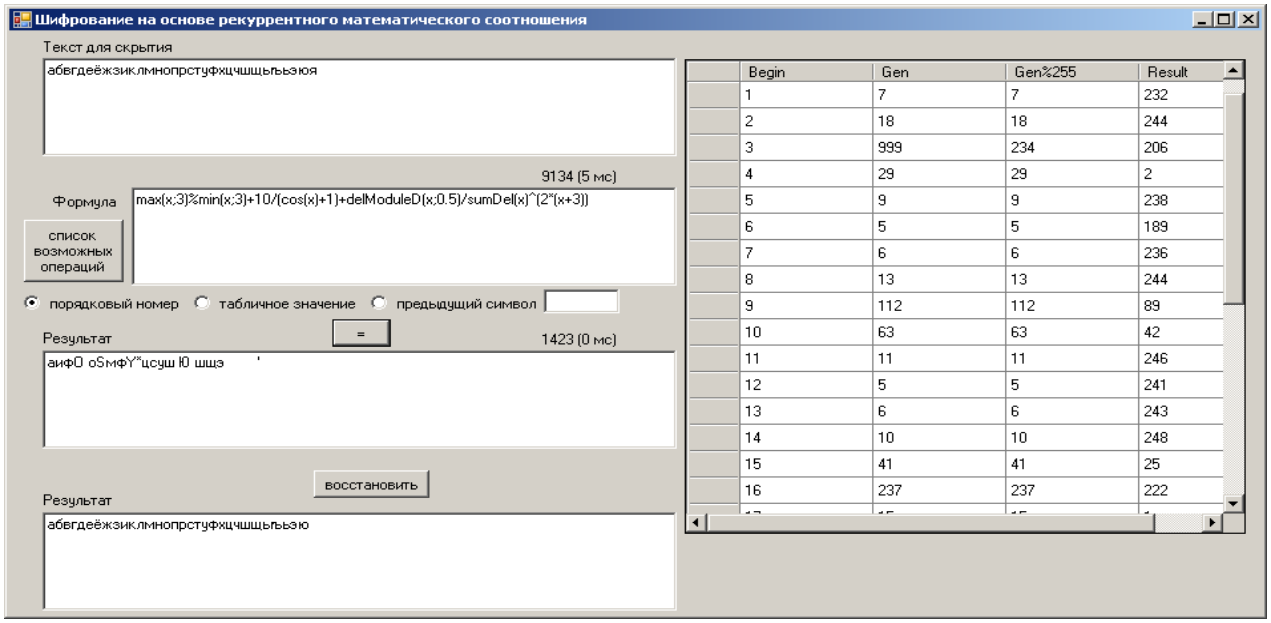

Рис. 2. Интерфейс приложения для зашифрования данных

Тестовые результаты данного эффекта показаны на рис. 3.

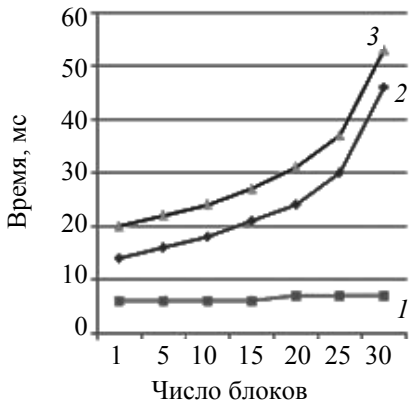

Рис. 3. Зависимость времени вычисления (зашифрования) от сложности формулы:
1 – операции на основе формулы;
2 – время анализа формулы;
3 – результирующее время

Из приведенного рисунка следует, что суммарное время (кривая *3*) зависит, в основном, от времени анализа формулы. В качестве повторяющегося блока использовалось выражение: *round(x^20+x/PI+delModuleD(x;0.5)+sumDel(x)+abs(x-10)*cos(x)/(sin(x)+1))*.

В качестве данных использовалось 33 символа русского алфавита.

Численные значения данных замеров приведены в нижеследующей таблице.

Численные значения замеров времени зашифрования 33 символов при различном количестве повторяющихся блоков математических операций

Количество блоков	Обработка, мс	Вычисление, мс	Общее время, мс
1	14	6	20
5	16	6	22
10	18	6	24
15	21	6	27
20	24	7	31
25	30	7	37
30	46	7	53

В дальнейшем планируется сократить это время за счет оптимизации используемых алгоритмов обработки [4]. В то же время это не является критичным моментом, поскольку такая обработка производится лишь один раз и используется в дальнейшем для зашифрования/расшифрования неограниченного объема поступающих данных.

Заключение. К преимуществам предложенного метода зашифрования/расшифрования сообщений на основе ключа в виде рекуррентных соотношений по сравнению с подобными методами, основанными на использовании статического ключа, относятся:

– устойчивость к большим объемам данных;

– устойчивость к часто повторяющимся данным;

– специфика зашифрования, позволяющая создавать трудно анализируемые выражения.

Недостатками являются:

1) зависимость криптостойкости ключа от составленной формулы, что ограничивает круг пользователей, способных самостоятельно создать сложные ключи;

2) более медленная обработка по сравнению с прямым гаммированием.

Особенности данного метода особенно подходят для шифрования программного кода.

Литература

1. Пласковицкий, В. А. Защита программного обеспечения от несанкционированного использования и модификации методами обфускации / В. А. Пласковицкий, П. П. Урбанович // Труды БГТУ. – 2011. – № 6: Физ.-мат. науки и информатика. – С. 173–176.

2. Урбанович, П. П. Информационная безопасность и надежность систем / П. П. Урбанович, Д. М. Романенко, Е. В. Романцевич. – Минск: БГТУ, 2007. – 86 с.

3. Ярмолик, В. М. Криптография, стеганография и охрана авторского права: монография / В. Н. Ярмолик, С. С. Портянко, С. В. Ярмолик. – Минск: Издат. центр БГУ, 2007. – 240 с.

4. Пласковицкий, В. А. Внедрение регулярных выражений в программные коды, реализующие алгоритмы обфускации / В. А. Пласковицкий, П. П. Урбанович // XIV Республиканская конференция студентов и аспирантов: сб. науч. работ. – Гомель: ГГУ им. Ф. Скорины, 2011. – С. 261–262.

Поступила 01.03.2012