электронный журнал
МОЛОДЕЖНЫЙ НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК
Издатель ФГБОУ ВПО "МГТУ им. Н.Э. Баумана". Эл No. ФС77-51038.

УДК 004.056.55

Разработка алгоритма защиты информации в системах электронного документооборота

Цибин А.Н., *студент*
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Программное обеспечение ЭВМ и информационные технологии»

Научный руководитель: Филиппов М.В., к.т.н, доцент
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана
bauman@bmstu.ru

Введение

Широкое внедрение компьютерной техники во все сферы деятельности человека привело к изменениям в обществе, пожалуй, не меньшим, чем промышленная революция XIX в. За несколько десятилетий ЭВМ превратились из чисто "вычислительных" машин в мощные системы по обработке, хранению, передаче информации.

В результате создания глобальных компьютерных сетей произошел настоящий переворот в области передачи и распространения информации. С использованием средств удаленного доступа стали проводиться торги, участники которых имеют возможность совершать сделки, не покидая своего офиса. Поручения о переводе денег могут передаваться в банк в виде компьютерных данных за считанные секунды. Чтобы заключить договор в письменной форме, уже необязательно писать его на бумаге и заверять подписью и печатью - достаточно заверить компьютерный файл с текстом договора электронной цифровой подписью и направить его по телекоммуникационным каналам партнеру.

Именно компьютеризация информационной деятельности помогла человечеству существенно ускорить процессы обработки, поиска, передачи информации, повысить надежность хранения больших ее массивов и тем самым преодолеть так называемый "второй информационный барьер".

Вместе с тем компьютеризация привела к возникновению целого ряда проблем, в том числе и правовых. Традиционные правовые системы создавались в эпоху, когда главенствующую роль в рыночном обмене занимали материальные товары ("индустриальная цивилизация"), а теперь, когда информация в стоимостном выражении стала существенной частью рынка, когда новые компьютерные технологии существенно

изменили функционирование самого рынка, указанные системы часто уже не дают адекватных ответов на многие вопросы.

К числу таких проблем можно отнести следующие:

• Появление нового типа преступлений - преступлений в сфере компьютерной информации, связанных с несанкционированным доступом к компьютерной информации, созданием и распространением компьютерных вирусов и т.п.;

• Обеспечение правовой охраны программ для ЭВМ; определение правового режима компьютерной информации, баз данных;

• Регламентация доступа к базам данных, содержащим сведения о личной жизни граждан, определение ответственности за несанкционированное разглашение таких сведений;

• Правовое регулирование использования систем электронного документооборота и в связи с этим определение правового статуса электронных документов;

• Правовые проблемы, связанные с функционированием глобальных сетей, трансграничной передачей данных и трансграничными компьютерными преступлениями, и ряд других.

1. Обзор существующих методов

Существует два метода шифрования – симметричное шифрование (использует один ключ для шифрования и расшифрования) и асимметричное шифрование (использует два различных ключа для шифрования и расшифрования).

Методы симметричного шифрования в свою очередь делятся на 3 вида: поточные шифры (синхронные и самосинхронизирующиеся), блочные шифры, комбинированные шифры.

1.1 Data Encryption Standard (DES)

Симметричный алгоритм шифрования, разработанный фирмой Ай-Би-Эм (IBM) и утверждённый правительством США в 1977 году как официальный стандарт (FIPS 46-3). DES имеет блоки по 64 бита и 16-цикловую структуру сети Фейстеля, для шифрования использует ключ с длиной 56 бит. Алгоритм использует комбинацию нелинейных (S-блоки) и линейных (перестановки E, IP, IP-1) преобразований.

Атаки на DES:

• Метод полного перебора требует одну известную пару шифрованного и расшифрованного текста, незначительный объём памяти, и его выполнение требует около 255 шагов.

• Дифференциальный криптоанализ — первую такую атаку на DES заявили Бихам и Шамир. Эта атака требует шифрования 247 открытых текстов, выбранных нападающим, и для её выполнения нужны примерно 247 шагов. Теоретически являясь точкой разрыва, эта атака непрактична из-за чрезмерных требований к подбору данных и сложности организации атаки по выбранному открытому тексту. Сами авторы этой атаки Бихам и Шамир заявили, что считают DES защищенным для такой атаки.

• Линейный криптоанализ разработан Матсуи. Этот метод позволяет восстановить ключ DES с помощью анализа 243 известных открытых текстов, при этом требуется примерно 243 шагов для выполнения. Первый экспериментальный криптоанализ DES, основанный на открытии Матсуи, был успешно выполнен в течение 50 дней на автоматизированных рабочих местах 12 HP 9735.

1.2 Blowfish

Криптографический алгоритм, реализующий блочное симметричное шифрование. Разработан Брюсом Шнайером в 1993 году. Представляет собой сеть Фейстеля. Выполнен на простых и быстрых операциях: XOR, подстановка, сложение. Является незапатентованным и свободно распространяемым.

На сегодняшний день (июнь 2014) не существует атак, выполняемых за разумное время. Успешные атаки возможны только из-за ошибок реализации.

1.3 Advanced Encryption Standard (AES)

Симметричный алгоритм блочного шифрования (размер блока 128 бит, ключ 128/192/256 бит), принятый в качестве стандарта шифрования правительством США по результатам конкурса AES. Этот алгоритм хорошо проанализирован и сейчас широко используется, как это было с его предшественником DES. Национальный институт стандартов и технологий США опубликовал спецификацию AES 26 ноября 2001 года после пятилетнего периода, в ходе которого были созданы и оценены 15 кандидатур. 26 мая 2002 года AES был объявлен стандартом шифрования. По состоянию на 2009 год AES является одним из самых распространённых алгоритмов симметричного шифрования. Поддержка AES введена фирмой Intel в семейство процессоров x86 начиная с Intel Core i7-980X Extreme Edition, а затем на процессорах Sandy Bridge.

Атаки на AES:

• В октябре 2005 года Даг Арне Освик, Ади Шамир и Эран Трумер представили работу с описанием нескольких атак, использующих время выполнения операций для нахождения ключа. Одна из представленных атак получала ключ всего лишь после 800 операций шифрования. Атака требовала от криптоаналитика возможности запускать программы на той же системе, где выполнялось шифрование.

• В декабре 2009 года была опубликована работа, в которой использование дифференциального анализа ошибок, искусственно создаваемых в матрице состояния на 8-ом раунде шифрования, позволило восстановить ключ за 232 операций.

1.4 Rivest Cipher 4

Потоковый шифр, широко применяющийся в различных системах защиты информации в компьютерных сетях (например, в протоколах SSL и TLS, алгоритме безопасности беспроводных сетей WEP).

Алгоритм RC4, как и любой потоковый шифр, строится на основе параметризованного ключом генератора псевдослучайных битов с равномерным распределением. Длина ключа может составлять от 40 до 2048 бит.

Основные преимущества шифра — высокая скорость работы и переменный размер ключа. RC4 довольно уязвим, если используются не случайные или связанные ключи, один ключевой поток используется дважды. Эти факторы, а также способ использования могут сделать криптосистему небезопасной (например, WEP).

2. Описание разработанного алгоритма.

Разработанный алгоритм является самосинхронизирующимся потоковым шифром на регистрах сдвига с линейной обратной связью.

2.1 Потоковые шифры на регистрах сдвига с линейной обратной связью (РСЛОС)

Есть несколько причин использования линейных регистров сдвига в криптографии:

• высокое быстродействие криптографических алгоритмов

• применение только простейших операций сложения и умножения, аппаратно реализованных практически во всех вычислительных устройствах

• хорошие криптографические свойства (генерируемые последовательности имеют большой период и хорошие статистические свойства)

• легкость анализа с использованием алгебраических методов за счет линейной структуры

Регистр сдвига с линейной обратной связью длины L состоит из L ячеек пронумерованных 0,1,2…L-1, каждая из которых способна хранить 1 бит и имеет один вход и один выход; и синхросигнала (clock), который контролирует смещение данных. В течение каждой единицы времени выполняются следующие операции:

содержимое ячейки L-1 формирует часть выходной последовательности;

содержимое i-той ячейки перемещается в ячейку i+1 для любого i, 0<i<L.

новое содержимое ячейки 0 определяется битом обратной связи, который вычисляется сложением по модулю 2 с определёнными коэффициентами битов ячеек 0,1,2,…L-1.

На первом шаге:

$$S_L = c_1 \cdot S_{L-1} \oplus c_2 \cdot S_{L-2} \oplus \ldots \oplus c_L \cdot S_0.$$

На втором шаге:

$$S_{L+1} = c_1 \cdot S_L \oplus c_2 \cdot S_{L-1} \oplus \ldots \oplus c_L \cdot S_1.$$

Следующее соотношение описывает в общем виде работу РСЛОС:

$$S_j = c_1 \cdot S_{j-1} \oplus c_2 \cdot S_{j-2} \oplus \ldots \oplus c_L \cdot S_{j-L}.$$

Если мы запишем во все ячейки биты равны нулю, то система будет генерировать последовательность, состоящую из всех нулей. Если записать ненулевые биты, то получим полубесконечную последовательность. Последовательность определяется коэффициентами $c_1, c_2, \ldots, c_L$

Посмотрим, каким может быть период такой системы:

Число ненулевых заполнений: $2^L - 1.$

Значит: $T \leq 2^L - 1$.

После возникновения одного заполнения, которое было раньше, процесс начнёт повторяться. Процесс заполнения регистра, как показано выше, представим линейным разностным уравнением. Перенесём все члены в одну часть равенства, получим:

$$S_j \oplus c_1 \cdot S_{j-1} \oplus c_2 \cdot S_{j-2} \oplus \ldots \oplus c_L \cdot S_{j-1} = 0$$

Обозначим:

$$S_j = D^{-j}.$$

Тогда:

$$(1 \oplus c_1 \cdot D \oplus c_2 \cdot D^2 \oplus \ldots \oplus c_L \cdot D^L) \cdot D^{-j}.$$

$$C(D) = 1 \oplus c_1 \cdot D \oplus c_2 \cdot D^2 \oplus \ldots \oplus c_L \cdot D^L.$$

Важным свойством этого многочлена является - приводимость.

Многочлен называется приводимым, если он может быть представлен как произведение двух многочленов меньших степеней с коэффициентами из данного поля (в нашем случае с двоичными коэффициентами). Если такое представление не имеет место, то многочлен $C(D)$ называется неприводимым.

Если многочлен является неприводимым и примитивным, то период будет совпадать с максимально возможным периодом, равным $2^L - 1.$

Пример схемы РСЛОС приведен на рисунке 1.

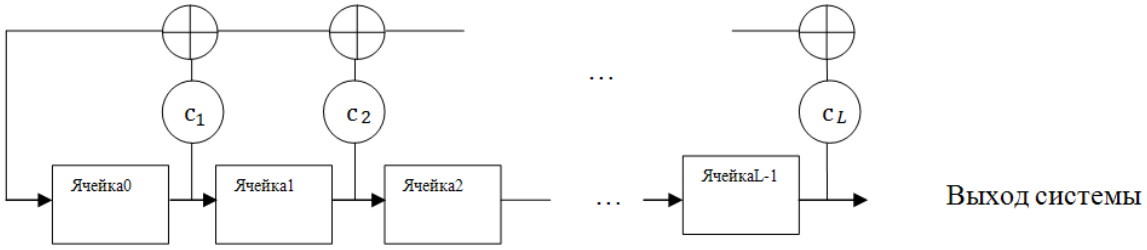

Рис.1. Регистр сдвига с линейной обратной связью

2.2 Линейная сложность

Линейная сложность бинарной последовательности – одна из самых важных характеристик работы РСЛОС. Поэтому остановимся на этой теме поподробнее. Прежде чем дать определение линейной сложности введём некоторые обозначения. S - бесконечная последовательность с членами $S1, S2, S3, \ldots$

S_n – конечная последовательность длины n, членами которой являются $S_1, S_2, S_3, \ldots, S_{n-1}.$

Говорят, что РСЛОС генерирует последовательность S, если существует некоторое исходное состояние, при котором выходная последовательность РСЛОС совпадает с S. Аналогично, говорят, что РСЛОС генерирует конечную последовательность S_n, если существует некоторое начальное состояние, для которого выходная последовательность РСЛОС имеет в качестве первых n членов члены последовательности S_n. Наконец дадим определение линейной сложности бесконечной двоичной последовательности.

Линейной сложностью бесконечной двоичной последовательности называется число $L(S)$, которое определяется следующим образом:

Если S – нулевая последовательность $S = 0, 0, 0, 0, \ldots,$ то $L(S) = 0.$

Если не существует РСЛОС, который генерирует S, то $L(S)$ равно бесконечности.

Иначе, $L(S)$ есть длина самого короткого РСЛОС, который генерирует S.

Линейной сложностью конечной двоичной последовательности S_n называется число $L(S_n)$, которое равно длине самого короткого РСЛОС, который генерирует последовательность, имеющую в качестве первых n членов S_n.

Свойства линейной сложности: Пусть S и K – двоичные последовательности. Тогда:

1. Для любого $n > 0$ линейная сложность подпоследовательности S_n удовлетворяет неравенствам $0 \leqslant L(S_n) \leqslant n$.
2. $L(S_n) = 0$ тогда и только тогда, когда S_n – нулевая последовательность длины n.
3. $L(S_n) = n$ тогда и только тогда, когда $S_n = 0, 0, 0, \ldots, 1$.
4. Если S – периодическая с периодом N, тогда $L(S_n) \leqslant N$.
5. $L(S \oplus K) \leqslant L(S) + L(K)$.

2.3 Нелинейная комбинация генераторов

Известно, что каждая Булева функция $f(x_1, x_2, \ldots, x_n)$ может быть записана как сумма по модулю 2 произведений порядков m независимых переменных, $0 \leqslant m \leqslant n$. Это выражение называется алгебраической нормальной формой функции f. Нелинейным порядком функции f называется максимальный порядок членов в записи её алгебраической нормальной формы. Например, Булева функция $f(x_1, x_2, x_3, x_4) = x_1 \oplus x_2 \oplus x_4 \oplus x_1x_3 \oplus x_2x_3x_4$ имеет нелинейный порядок 3. Максимально возможный нелинейный порядок Булевой функции равен количеству переменных n.

Предположим теперь, что у нас n регистров сдвига с линейной обратной связью, их длины $L_1, L_2, L_3, \ldots, L_n$ попарно различны и больше двух. Все регистры объединены нелинейной функцией $f(x_1, x_2, \ldots, x_n)$, как показано на рисунке. Функция f представлена в алгебраической нормальной форме. Тогда линейная сложность потока ключей равна $f(L_1, L_2, \ldots, L_n)$.

Если $L_1, L_2, \ldots, L_n$ – попарно взаимно-простые числа, то длина периода ключевого потока равна: $(2^{L_1} - 1) \cdot (2^{L_2} - 1) \cdots (2^{L_n} - 1)$. Например, если $L_i = 10$, то $(2^{L_1} - 1) \approx 10^3$. И длина периода ключевого потока равна $(10^3)^n$.

Пример РСЛОС с нелинейной комбинацией генераторов представлен на рисунке 2.

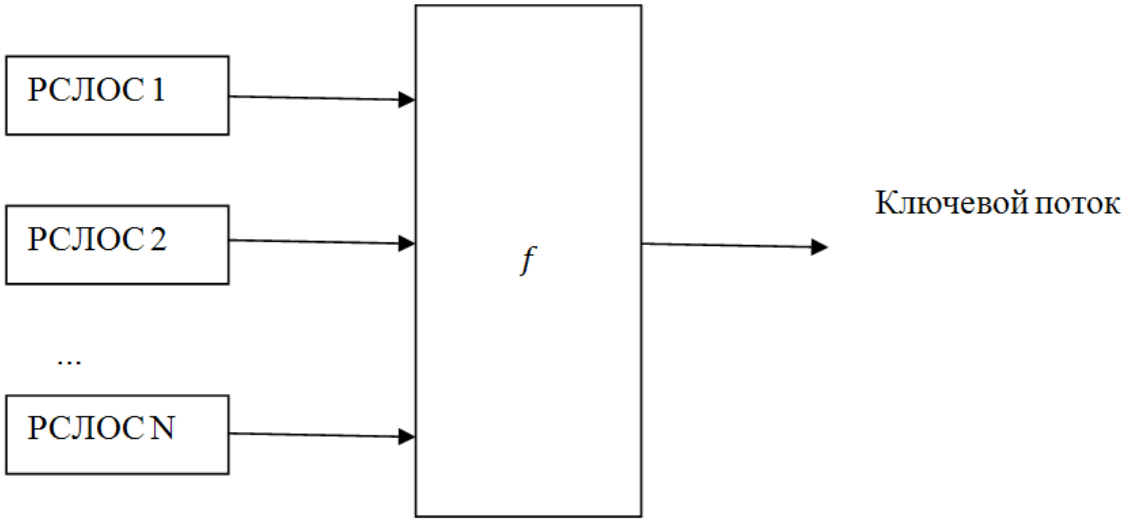

Рис.2. Нелинейная комбинация генераторов

2.4 Описание основных операций, переменных и функций

$+$: x+y означает x+y mod 2^{32}, где $0 \leq x < 2^{32}$ и $0 \leq y < 2^{32}$

$\boxminus$: x $\boxminus$ y означает x - y mod 1024

$\oplus$: побитовое исключающее ИЛИ

$\|$: конкатенация

$\gg$: оператор сдвига вправо на указанное количество бит

$\ll$: оператор сдвига влево на укзанное количество бит

$\ggg$: циклический сдвиг вправо, x $\ggg$ n означает ((x $\gg$ n) $+$ (x $\ll$ (32 - n)), где $0 \leq n < 32$ и $0 \leq x < 2^{32}$

P : таблица из 1024-х 32-битных элементов. Каждый элемент обозначается P[i], где $0 \leq i \leq 1023$.

Q : таблица из 1024-х 32-битных элементов. Каждый элемент обозначается P[i], где $0 \leq i \leq 1023$.

K : 256-битный ключ алгоритма.

V : 256-битный вектор инициализации алгоритма.

S : ключевая последовательность, созданная алгоритмом. 32-битный элемент на выходе i-го шага обозначается S_i . S=$S_0 \| S_1 \| S_2 \| ...$

$$f_1(x) = (x >>> 7) \oplus (x >>> 18) \oplus (x >> 3)$$

$$f_2(x) = (x >>> 17) \oplus (x >>> 19) \oplus (x >> 10)$$

$$g_1(x, y) = ((x >>> 10) \oplus (y >>> 23)) + Q[(x \oplus y) \bmod 1024]$$

$$g_2(x, y) = ((x >>> 10) \oplus (y >>> 23)) + P[(x \oplus y) \bmod 1024]$$

$$h_1(x) = Q[x_0] + Q[256 + x_1] + Q[512 + x_2] + Q[768 + x_3]$$

$$h_2(x) = P[x_0] + P[256 + x_1] + P[512 + x_2] + P[768 + x_3]$$

$$x = x_0||x_1||x_2||x_3 - 32x - \text{битное слово}, \ x_i - 8 \text{ бит}$$

Функции f взяты из SHA-2 и служат для однозначного отображения одного 32-битного числа в другое. Константы соответствуют максимальному периоду.

Функции h используются для обновления значений таблиц Q и P на каждом шаге.

Функции g служат для получения очередного элемента ключевого потока.

2.5 Процесс инициализации

Процесс инициализации состоит из преобразования P и Q с помощью ключа и вектора инициализации и запуска шифрования 4096 раз без генерации выходного результата.

1. Составление $K_0 \| K_1 \| \ldots \| K_7$ и $V_0 \| V_1 \| \ldots \| V_7$, где K_i и V_i состоят из 32-х бит. Ключ и вектор инициализации расширяются в массив W_i ($0 \leq i \leq 2559$).

W_i принимает следующие значения:

$$K_i, \ 0 \leq i \leq 7$$

$$V_{i-8}, \ 8 \leq i \leq 15$$

$$f_2(W_{i-2}) + W_{i-7} + f_1(W_{i-15}) + W_{i-16} + i, \ 16 \leq i \leq 2559$$

2. Обновление таблиц P и Q с помощью W:

$$P[i] = W_{i+512}, \ 0 \leq i \leq 1023$$

$$Q[i] = W_{i+1536}, \ 0 \leq i \leq 1023$$

3. Запуск шифрования (алгоритм генерации ключевой последовательности) 4096 раз без генерации выходного результата.

2.6 Общая схема алгоритма

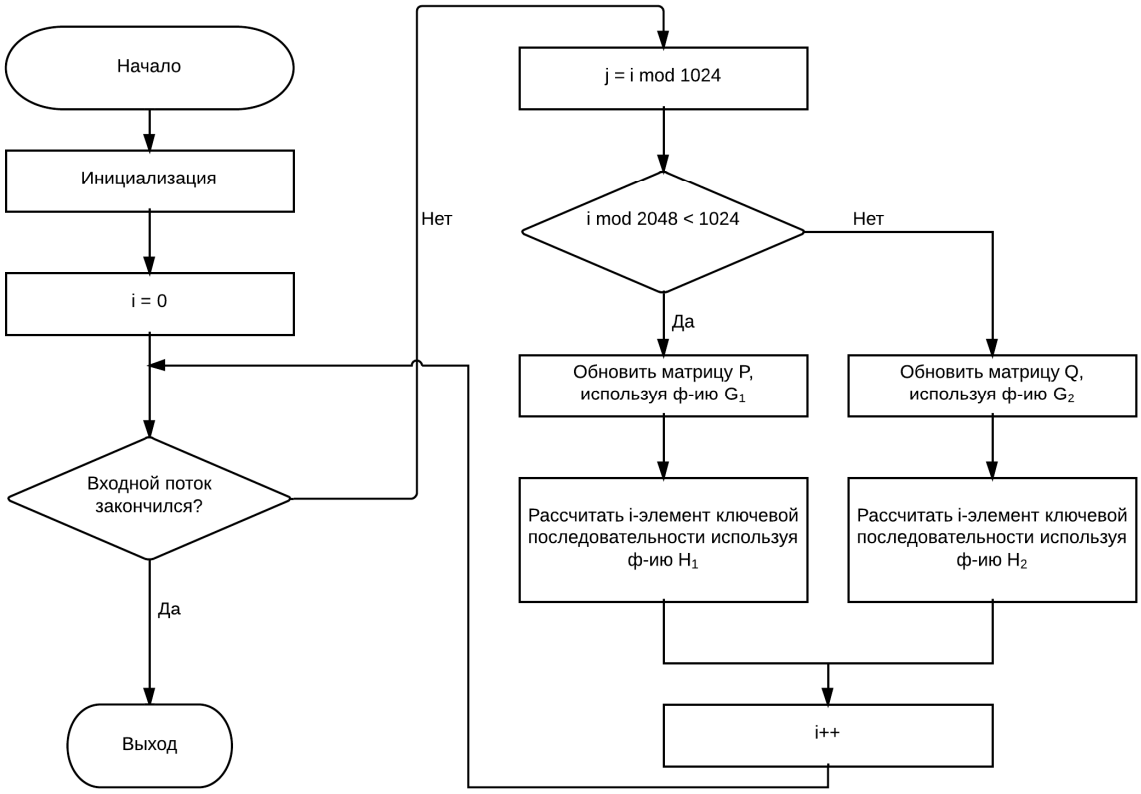

Рис. 3. Схема алгоритма

Заключение

В рамках работы над проектом были выполнены следующие задачи:

• Разработан алгоритм шифрования

• Разработана структура ПО

• Выполнена программная реализация, разработан пользовательский интерфейс

• Выполнено сравнение с другими существующими решениями

Дальнейшим развитием проекта может стать:

• Улучшение быстродействия, распараллеливание вычислений

• Уменьшение потребляемой памяти

Список литературы

1. Ященко В.В. Введение в криптографию. М.: МЦНМО, 2000. 272 с.
2. Коблиц Н. Курс теории чисел и криптографии. М.: ТВП, 2001. 254с.
3. Schneier B. Applied Cryptography // John Wiley & Sons, 2012. 784p.
4. Lowy J. Programming WCF Services, 3rd Edition // O'Reilly Media, 2010. 910p.

5. Encryption. Available at: http://en.wikipedia.org/wiki/Encryption, accssed 26.06.2014.
6. Symmetric-key algorithm. Available at: https://en.wikipedia.org/wiki/Symmetric-key_algorithm, accssed 06.06.2014.