УДК 004.056.55

ОБЕСПЕЧЕНИЕ СКРЫТНОСТИ КОДИРОВАНИЯ ДАННЫХ ПОМЕХОУСТОЙЧИВЫМИ КОДАМИ

А. И. ТИТОВ
Н. И. КОРСУНОВ

Белгородский государственный национальный исследовательский университет

e-mail: titov@programist.ru

В статье рассматривается эволюционный метод как обобщающий существующих симметричных методов кодирования данных. Парадигма эволюционного кодирования данных вводит не используемую раннее мутацию. Предложен пример использование мутации на основе помехоустойчивого кодирования Хемминга.

Ключевые слова: скрещивание, ген, поколение, мутация, родитель, кодирование, декодирование, слово, ключ.

Возможности метода кодирования данных можно выявить сравнением его с другими методами. Будем считать, что эффективность метода кодирования данных определяется расширением его возможностей по сравнению с другим, что можно представить в виде:

$$M \in Mi, \quad (1)$$

где M – метод эффективность которого оценивается,
Mi – известный метод, i – 1,2,3,....,n.

Для рассмотрения эволюционного метода кодирования данных[1], его положительных сторон и недостатков, необходимо разобрать метод на составляющие.

1 этап – выбор родителей;
2 этап – скрещивание;
3 этап – формирование потомка;
4 этап – мутация;
5 этап – селекция потомства.

Первые три этапа не являются новыми и присутствуют в существующих методах криптографии уже давно. Соответственно в той или иной форме изменяя последовательность этапов и критерии отбора на каждом этапе, мы можем получить любой из существующих симметричных шифров. И проанализировав наличие или отсутствие приведенных этапов в методе Mi, также определим положительные и отрицательные стороны присутствия этапов Mi в M.

В качестве Mi выберем: M1 – алгоритм Виженера [2]; M2 – алгоритм ГОСТ 28147-89 [3]; M3 – алгоритм Rijndael [4].

Выделим в каждом алгоритме предложенные выше этапы.

M1 формирование мультиязычных систем шифрования, используем 1-3 этап с четко заданными параметрами.

1 этап – выбор родителей.

Параметры:

1 родитель – i-тая, буква открытого текста. (1 байт сообщения.)
2 родитель –i-тая буква ключа. (1 байт ключа)
Если i>lengkey
то i mod lengkey
2 этап – скрещивание родителей по таблице Вижнера или по алгоритму[2] .
3 этап – результат скрещивания напрямую записывается в потомство.

M2 можно также представить как частный случай эволюционного кодирования. В работе алгоритма используются 1, 2, 5 этапы.

1 этап – выбор 1 родителя L =32 бита
выбор 2 родителя R=32 бита.

Родители в этом случае левая и правая части 64 битного блока.

2 этап – скрещивание можно описать следующими действиями:

$$Li = Ri - 1, \quad (2)$$

$$Ri = Li - 1\ f(Ri - 1, ki). \quad (3)$$

5 этап – Алгоритм ГОСТ селекция для отбора потомка используется итеративное 32 кратное выполнение второго этапа, после чего потомок записывается в закрытый текст.

М3 является более изощренным[5], но поддающимся описанию поэтапной моделью эволюционного метода.

Рассматривая принятый за стандарт алгоритм с размерностью блока 128 бит выделяем следующее:

1 этап – выбор родителя осуществляется в зависимости от этапа либо блок сообщения + ключ, либо блок сообщения + блок сообщения, выборка из сообщения осуществляется в прямом порядке.

2 этап – скрещивание производится с использование S-box(соответствует замене из мультиязычных систем) и сложением по модулю с ключом.

3 этап – для большего распределения в процесс формирование потомка заложен сдвиг блоков и перемешивание.

5 этап – селекция потомка производится на 10, 12 или 14 раунде шифрования в зависимости от длинны ключа. Здесь уже нет четкой фиксации, но для всего сообщения этап селекции будет одинаковым.

Эволюционный метод кодирования данных можно считать обобщенным методом для всех симметричных алгоритмов. Из пятиэтапного описанного выше метода, в существующих системах встречаются различные этапы в разном порядке следования, за исключением этапа мутации.

Термин «мутация» был введен одним из ученых, переоткрывших законы Менделя, – Г. де Фризом в 1901 г. (от лат. мутатио – изменение, перемена). Этот термин означал вновь возникшие, без участия скрещиваний, наследственные изменения. Мутации делят на генные мутации , хромосомные мутации и геномные мутации. В эволюционном кодирование особое место занимает генные мутации. При переносе определения генной мутации на кодирование данных получим следующее определение: генные мутации – это мутация при которой изменяется количество и значение хромосом или хромосомных наборов, без участия скрещиваний [6].

Основываясь на предложенной модели эволюционного кодирования, разработаем алгоритм и ПО для обеспечения безопасности при несанкционированном доступе.

История шифрования показывает нам несложные технологии повышения криптостойкости.

Усиление Алгоритма DES при помощи введения в него тройной итерации, привело к повышению более стойкого TREE DES. Здесь введение итераций можно сравнить с этапом №5 эволюционного метода.

Усовершенствовав введением селекции Алгоритм Вижнера мы получим новый шифр[7].

Недостатком данного метода является последовательность шифрования и дешифрования. Для устранения этого недостатка и приведения Алгоритма к блочной структуры внесет изменения [3].

В представленном Алгоритме большую роль играет период ключа шифрования. Даже после формирования маски шифрования могут встречаться ситуации, показывающие соответствие.

При наличии у злоумышленника большого объема закрытых и открытых сообщений, есть вероятность подбора маски шифрования для сообщения конкретной длины. Определение ключа шифрования, при наличии точной маски шифрования и знания длины файла, осуществляется грубым перебором.

Для формирования более стойкого Алгоритма возьмем за прототип сегодняшний стандарт шифрования ГОСТ. Его основа скрещивание левой и правой части блока и выбор потока на 32-ой селекции. Здесь этап селекции явно прописан в Алгоритме и не может быть изменен без вмешательства разработчиков в код программы.

Устраним этот недостаток введением в ключ шифрования шага селекции. В ГОСТ 24 шага идёт взаимодействие с прямым ключом, а последние 8 с обратным. Это возможно из-за его четкого определения количества итераций. В нашем Алгоритме шаг следующий задан не явно, поэтому этап скрещивания будет иметь следующее представление:

Кодирование данных:

1) Читаем в закрытый ключ первый и второй биты. В зависимости от значений закрепляем блоки данных «0» – закрепляем первый блок сова, «1»- закрепляем второй блок слова. Здесь первый бит информации отвечает за первое слово, второй бит за второе слово.

Оставшиеся блоки записываем из первого слова в третью позицию, из второго слова в четвёртую.

2) Делаем скрещивание C1 XOR C2 записываем результат в D1.

3) Читаем закрытый ключ третий бит, если:

«0» – берём C1 последний, C2 первый(записываем в D2);

«1» – берём C1 первый, C2 последний;

4) Если количество поколений не удовлетворяет критерию то повторить шаг 1-3, при этом ключ шифрования двигается дальше.

Наглядный пример работы алгоритма кодирования:

Ключ шифрования: 011100

Входные слова: первое слово(10001101)

второе слово(10110110)

Количество поколений для достижения стойкости: 2

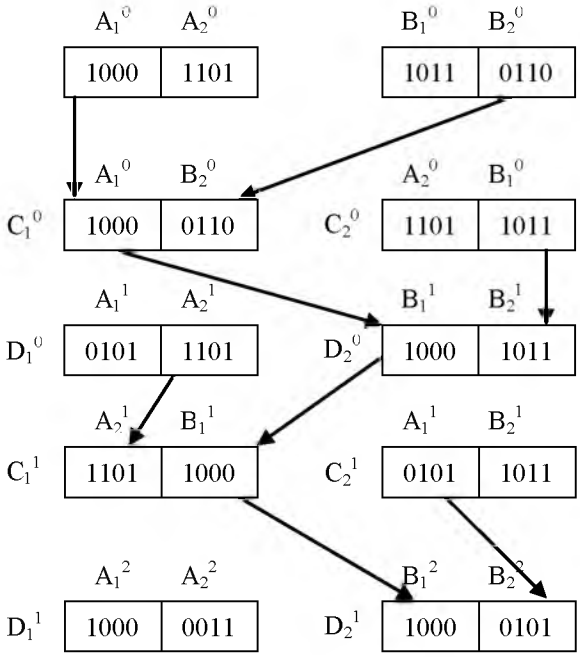

Рис. 1. Работа алгоритма кодирования данных эволюционными методами

Декодирование данных предполагает наличие секретного ключа использованного при кодировании. Обмен ключом происходит по уже существующим закрытым каналам связи, или же во время «рукопожатия»[8].

Декодирование:

1) Читаем ключ с конца.

Если бит имеет значение «0» то B_1^0 во второй блок слова C1(Y_2), а B_2^0 в первый блок слова C2(X_1);

Если бит имеет значение «1» то B_1^0 в первый блок слова C1(Y_1), а B_2^0 во второй блок слова C2(X_2);

2) В зависимости от значения считанного бита ключа на предыдущем шаге проводим восстановление предка по генотипу потомков.
Если «0» то $Y_1=(A_1^0 \text{ XOR } X_1)$; $X_2=(A_2^0 \text{ XOR } Y_2)$
Если «1» то $Y_2=(A_2^0 \text{ XOR } X_2)$; $X_1=(A_1^0 \text{ XOR } Y_1)$
3) Читаем следующие с хвоста два бита ключа, в зависимости от их значений меняем блоки местами:

Таблица

Скрещивание блоков слова

Значение бит ключа	Запись блоков информации			
00	$A_1=Y_1$	$A_2=X_1$	$B_1=Y_2$	$B_2=X_2$
01	$A_1=Y_1$	$A_2=X_1$	$B_1=X_2$	$B_2=Y_2$
10	$A_1=X_1$	$A_2=Y_1$	$B_1=Y_2$	$B_2=X_2$
11	$A_1=X_1$	$A_2=Y_1$	$B_1=X_2$	$B_2=Y_2$

4) Если количество поколений не удовлетворяет критерию то повторить шаг 1-3, при этом ключ шифрования двигается дальше.

Наглядный пример работы алгоритма декодирования данных:

Ключ шифрования: 011100

Входные скрытые слова: первое слово(10000011)
второе слово(10000101)

Количество поколений для достижения стойкости: 2

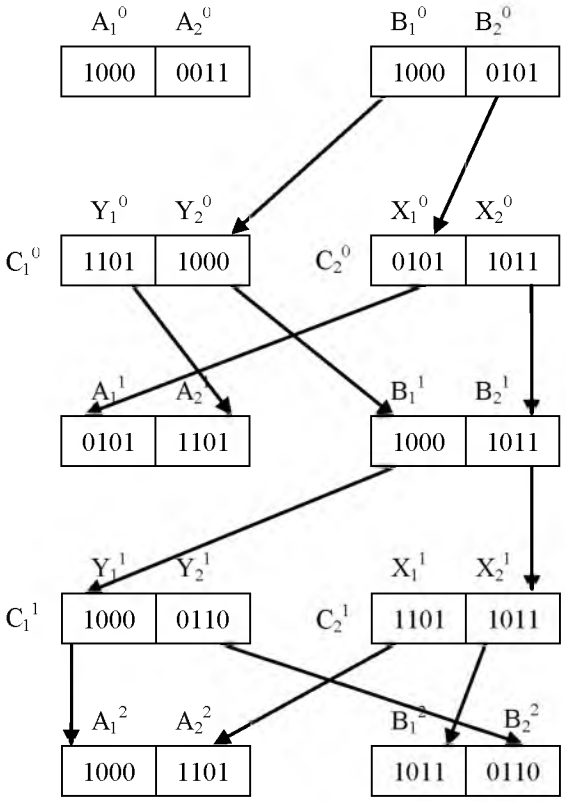

Рис. 2. Работа алгоритма декодирования данных эволюционными методами

Наилучший результат и полное соответствие с моделью эволюционного кодирования даст введение мутации.

Мутацию будем базировать на известном методе помехоустойчивого кодирования Хемминга. Так как предложенный Алгоритм скрещивания дает нам возмож-

ность без труда изменять величину входных блоков данных, то соответственно эта величина может быть секретной. Идея предложенного Хеммингом метод помехоустойчивости кодирования в следующем: все биты, номера которых есть степень 2, — контрольные, остальные — биты сообщения. Каждый контрольный бит отвечает за чётность суммы некоторой группы бит. Один и тот же бит может относиться к разным группам. Чтобы определить какие контрольные биты контролируют бит в позиции k надо разложить k по степеням двойки: если k=11=8+2+1, то этот бит относится к трём группам — к группе, чья чётность подсчитывается в 1-ом бите, к группе 2-ого и к группе 8-ого бита.

Этап мутации представляется как добавление контрольных битов в потомка и введение однократной случайной ошибки. Без знания величины блока шифрования определить местоположение контрольных битов можно только в 1 блоке. Для введения ошибки в мутации будем использовать генератор случайной последовательности (ГСП) ANSI X9.17

Ранее ГСП применялся только для формирования ключа шифрования[9] или используемых в алгоритме блоков и никогда не являлись непосредственно частью процесса модификации данных.

Литература

1. Титов А.И. Эволюционные методы кодирования данных, пример работы алгоритмов кодирования и декодирования/Корсунов Н.И.// «Научные ведомости» БелГУ №1(120) 2012.

2. Панасенко С.П. «Алгоритмы шифрования». Специальный справочник. [Текст]// СПб.: БХВ-Петербург. – 2009. – 576с.: ил.

3. Титов А.И. Модифицированный алгоритм шифрования данных /Корсунов Н.И. // Научно-технический журнал «Информационные системы и технологии» №2(64) 2011

4. ГОСТ 28147-89. Системы обработки информации. Защита крипто-графическая. Алгоритм криптографического преобразования. // М.:Госстандарт СССР. – 1989.

5. Joan Daemen The Design of RijndaeL: AES – The Advanced Encryption Standard (Information Security and Cryptography) /Vincent Rijmen//QA 76.9.A25 D 32 2001 ISBN 3540-42580-2.

6. Рутковская Д. Нейронные сети, генетические алгоритмы и нечеткие системы = Sieci neuronowe, algorytmy genetyczne i systemy rozmyte/ Пилиньский М., Рутковский Л. // М.:Горячая линия-телеком. – 2-е изд. -2008. — С. 452.

7. Корсунов Н.И. Повышение эффективности зашиты информации модификацией шифра Вижинера /Титов А.И. // Научные ведомости БелГУ. –2010. № 7 (78) вып. 14. – С. 171-175.

8. Баричев С.Г. Основы современной криптографии./Баричев С.Г., Гончаров В.В., Серов О.Е. // М.:Горячая линия-телеком.- 2001.

9. Proakis J.G. Digital communications/ перевод на русский язык – Кловский Д.Д. Николаев Б.И.//М-Радио и связь. 2000 – 800с.: ил.

ENSURING DATA SECRECY CODING NOISE-RESISTANT CODES

A. I. TITOV
N. I. KORSUNOV

Belgorod National Research University

e-mail: titov@programist.ru

The article discusses the evolutionary synthesis method as the existing symmetric data encryption methods. The paradigm of evolutionary coding data input is not used early mutation. An example of the use of mutation on the basis of the Hamming error-correcting coding.

Keywords: crossing, the gene, generation, mutation, parent, encoding, decoding, word key.