ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ ЗАЩИТЫ ИНФОРМАЦИИ МОДИФИКАЦИЕЙ ШИФРА ВИЖИНЕРА

Н. И. КОРСУНОВ[1)]
А. И. ТИТОВ[2)]

[1)] Белгородский государственный университет

e-mail: korsunov@intbel.ru

[2)] Белгородский государственный технологический университет им. В.Г. Шухова

e-mail: titov@programist.ru

В статье рассмотрена проблема защиты информации. Предложены методы предотвращающие утечку стратегической информации с серверов, даже при получении злоумышленником доступа к файлам. Отображен разработанный криптографический алгоритм в основу которого заложен шифр Виженера. Предложенная модификация позволяет проводить блочные итерации со смещением ключа, обеспечивая надежную защиту информации и дает возможность применять его к любому типу файлов.

Ключевые слова: криптография, стойкость криптографическая, итеративный алгоритм шифрования, метод протяжки вероятного слова, шифр Виженера, блочное шифрование.

Защита информации является существенной проблемой в информационных Web технологиях

Для шифрования данных расположенных на сервере используются различные криптоалгоритмы: генератора псевдослучайных чисел [1], алгоритм DES [2] [3], шифр Вижинера [1], алгоритм RSA [1].

Эффективные методы защиты основаны на классической криптографии, для которой характерно использование одной секретной единицы — ключа. Используемый ключ позволяет отправителю зашифровать сообщение, а получателю расшифровать его. В случае шифрования данных, хранимых на магнитных или иных носителях информации, ключ позволяет зашифровать информацию при записи на носитель и расшифровать при чтении с него.

Наиболее известными и широко используемыми методами симметричного шифрования являются алгоритм DES [2] [3] и шифр Вижинера [1].

Шифр Виженера — метод полиалфавитного шифрования буквенного текста с использованием ключевого слова.

Этот метод является простой формой многоалфавитной замены. Метод прост для понимания и реализации, он является недоступным для простых методов криптоанализа.

Алгоритм DES осуществляет шифрование 32,64 или 128-битовых блоков данных с помощью ключа размерностью от 0 до 2040 бит.

Дешифрование в DES является операцией обратной шифрованию и выполняется путем повторения операций шифрования в обратной последовательности. Процесс шифрования заключается в начальной перестановке битов 64-битового блока, шестнадцати циклах шифрования и, наконец, обратной перестановки битов.

Необходимо отметить, что при используемые при шифровании таблицы, являются стандартными, и следовательно, должны включаться в реализацию алгоритма в неизменном виде.

В асимметричных криптосистемах [1] ключ, используемый для шифрования, отличен от ключа дешифрования. При этом ключ шифрования не секретен и может быть известен всем пользователям системы. Однако дешифрирование с помощью известного ключа шифрования невозможно. Для дешифрования используется специальный, секретный ключ. Знание открытого ключа не позволяет определить секретный ключ. Таким образом, дешифровать сообщение может только его получатель, владеющий секретным ключом.

При защите информации на Web – сервере данная система имеет недостаток связанный, с тем, что необходимо для каждого пользователя ресурса генерировать

свой секретный ключ. Считается, что системы с открытым ключом больше подходят для шифрования передаваемых данных, чем для защиты данных, хранимых на носителях информации [4].

Приведенные недостатки систем с открытым ключом делают предпочтительным использование алгоритма Вижинера при защите информации на Web – серверах.

Шифр Вижинера требует хранения одного ключа, задаваемого набором из b букв. Такие наборы подписываются с повторением под сообщением, а, затем, полученную последовательность складывают с открытым текстом по модулю **n** (мощность алфавита).

Шифрование осуществляется согласно выражению:

Vigd(mi)=(mi+ki mod d)(mod n),

а дешифрование:Vigd(mi)=(mi-ki mod d)(mod n)

Известны модификации алгоритма: шифр Вижинера с автоключом,[5] шифр Вижинера с перемешанным один раз алфавитом,[5] шифр Бофорта.[5]

Алгоритм Вижинера и известные его модификации[1][5], не дают надежной защиты информации так как при однократном шифровании, зная алфавит и шифрованный текст, криптоаналитик, используя метод протяжки вероятного слова[6], может подобрать ключ шифрования**.**

Целью статьи является разработка эффективного способа защиты исходных кодов программного продукта и файлов, расположенных на стороннем сервере

В основу предлагаемого метода защиты положен шифр Вижинера [1] как наиболее простой алгоритм симметричной защиты.

Для достижения поставленной цели предлагается использовать многократную итерацию, при которой соответствующие алгоритмы шифрования и дешифрования состоят из последовательных однотипных циклов шифрования.

Воспользуемся методами защиты по алгоритму Вижинера и его модификациям использующим средства защиты без обратной связи, так как использование метода с обратной связью не возможно при возникновении шума в канале связи, вследствие того, что изменение одного бита в шифрованном сообщении приводит к ошибке дешифрования всего сообщения. Это приводит к тому, что в заданной ситуации необходимо запрашивать всё сообщение повторно, что ведёт к временным затратам и занятости канала связи.

Использование блочных кодов позволяет проводить неполное дешифрование для получения информации о файле. Это позволяет сократить загруженность сервера при большом количестве подключенных пользователей. Для сохранения этого достоинства, предлагается при модификации кода Вижинера, воспользоваться принципом блочного кодирования. При величине блока восемь бит, следует соответствие его шифру Вижинера [1] при использовании алфавита размерностью n=256. Алфавитом предложенным в таблице 1 можно шифровать любые файлы независимо от типа и размерности. Первая строка таблицы является прямым алфавитом, все последующие строки сдвинуты на один элемент.

Таблица 1

Код Виженера для шифрования файлов

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24.....253 254 255
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24.....253 254 255 0
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24.....253 254 255 0 1
3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24.....253 254 255 0 1 2
4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24.....253 254 255 0 1 2 3
5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24.....253 254 255 0 1 2 3 4
6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24.....253 254 255 0 1 2 3 4 5
7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24.....253 254 255 0 1 2 3 4 5 6
.....
.....
254 255 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24.....253
255 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24.....253 254

Аналитическое представление алгоритмов шифрования и дешифрования, полученные из таблицы 1, имеют следующий вид:

Для шифрования задают два параметра i – один байт шифруемого файла, j – один байт ключа шифрования. Результатом шифрования является один байт шифрованного файла.

```
crypt(i,j:byte):byte;
begin
if j<(256-i) then
crypt:=i+j
else
crypt:=j-(256-i)
end;
```

Дешифрование работает аналогично шифрованию – вычисляется байт открытого файла.

```
decrypt(i,x:byte):byte;
begin
if x>(i-1) then
decrypt:=x-i
else
decrypt:=x+(256-i)
end;
```

Входные данные: байт ключа, байт шифрованного файла.

Выходные данные: Байт открытого сообщения.

По таблице 1 легко проверить, что эти выражения справедливы для всех наборов значении байта.

Однако, этот подход не решает проблемы соответствия блоков открытого и шифрованного сообщения. Так, например, при шифровании представленной табл. 2:

Таблица 2

Пример шифрования файла

Ключ	1	21	31	41	51	61	31	1	21	31	41	51	61
Открытый файл	121	145	0	18	35	43	0	0	9	15	5	6	3
Шифрованный файл	122	166	31	59	86	104	31	1	30	46	46	56	64

Видна проблема соответствия шифрованного и открытого файла, так как использование одинаковых бит в открытом файле позволяет восстановить открытый ключ.

Для устранения этого предлагается использовать многократный итерационный метод при шифровании и дешифровании. Для увеличения криптостойкости ключ шифрования смещается на втором и последующих шагах итерации. Смещение вычисляется по остатку ключа с предыдущей итерации.

Прямолинейный процесс шифрования-дешифрования представляется следующей последовательностью шагов. Первый шаг соответствует таблице 2, и вводятся новые шаги приведенные в таблицах 3-5. В введенном шаге шифрования в качестве открытого файла используется шыфрованный файл с предыдущего шага, и происходит смещение ключа шифрования на длину остатка на предыдущем шаге.

Таблица 3

Второй шаг шифрования

Ключ	31	1	21	31	41	51	61	31	1	21	31	41	51
Шифрованный файл 1 шага	122	166	31	59	86	104	31	1	30	46	46	56	64
Шифрованный файл	153	167	52	90	127	155	92	32	31	67	77	97	115

Таблица 4

Третий шаг шифрования

Ключ	61	31	1	21	31	41	51	61	31	1	21	31	41
Шифрованный файл 2 шага	153	167	52	90	127	155	92	32	31	67	77	97	115
Шифрованный файл	214	198	53	111	158	196	143	93	62	68	98	128	156

Таблица 5

Четвертый шаг шифрования

Ключ	51	61	31	1	21	31	41	51	61	31	1	21	31
Шифрованный файл 3 шага	214	198	53	111	158	196	143	93	62	68	98	128	156
Шифрованный файл	9	3	82	112	179	227	184	144	123	99	99	149	187

Сравнения данных приведенных в таблицах 3,4,5 с данными приведенными в табл. 2 показывает, что начиная со второго шага, устраняется проблема соответствия байт в открытом и шифрованном файлах, это приводит к невозможности выявлении закрытого ключа методом протяжки вероятностного слова. При этом, начиная с четвертого шага разные байты открытого файла могут давать одинаковые байты шифрованного файла, что усложняет определение закрытого ключа методами частотного анализа.

Для N – итерационного шифрования необходимо проходить исходный файл N раз. При прямолинейном подходе это действительно так, но при этом теряется возможность использования данного алгоритма для блочного шифрования. Обобщим метод на блочное шифрование. Блочные криптосистемы разбивают текст сообщения на отдельные блоки и затем осуществляют преобразование этих блоков с использованием ключа.

Для организации блочного шифра, заменим многократный проход исходного файла на шифрование блоками по одному байту, независимо от остальных байт в файле.

При этом каждый блок будем шифровать в несколько проходов с использованием различных байт ключа, позиции которых вычисляются.

Для получения выражений, используемых при вычислениях, принимаем первоначальное смещение равное нолю, что соответствует первому шагу шифрования отображенному на рис. 1. Используя остаток ключа вычисляем начальное смещения для второго шага шифрования.

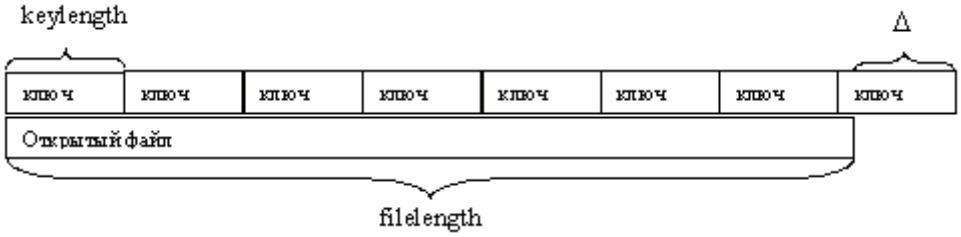

Рис. 1. Вычисление смещения для второго шага шифрования,
где filelength – количество байт в открытом файле,
keylengt – количество байт в ключе шифрования,
Δ – величина смещения ключа на втором шаге итерации.

Из приведенных обозначений следует, что:

$$\Delta = filelength \bmod keylength$$

И показывает что остаток ключа, переходящий на следующую итерацию, отождествляет смещение ключа шифрования.

Вычисления смещения на любом шаге приведено на рисунке 2 , из которого следует что вычисления смещения на любом шаге шифрования представляется в виде.

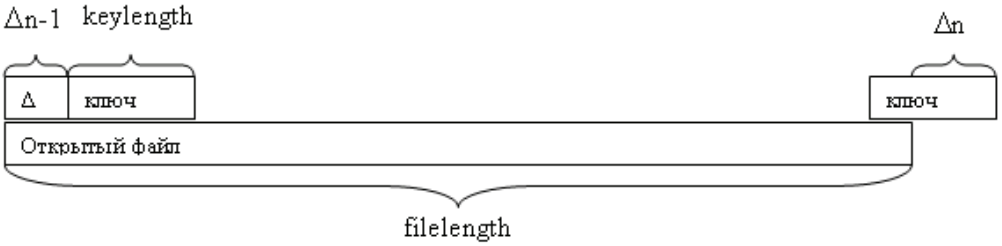

Рис. 2. Общий вид алгоритма вычисления смещения:

$$\Delta_n = (filelength - (keylength - \Delta_{n-1})) \bmod keylength,$$

где Δ_n – вычисляемая величина смещения ключа на шаге n,

Δ_{n-1} – величина смещения ключа на предыдущем шаге(n-1)

Из вышесказанного следует, что длинна ключа есть величина неопределённая, и в случаях, когда

filelength mod keylengt = 0.

Чтобы вторая и последующие итерации не проходили в пустую, необходимо смещать ключ на T байт. Число T – величина задаваемая в момент настройки криптосистемы.

Таким образом, предложенный модифицированный алгоритм Вижинера с применением блочного шифрования, основанный на варьировании количества итерации со смещением ключа, позволяет, в отличие от известных алгоритмов, более надежно защищать данные на Web – сервере.

Литература

1. Альферов А.П. “Основы криптографии учебное пособие” – Зубов А.Ю., Кузьмин А.С., Черемушкин А.В. [Текст] // 2-е изд., испр. и доп. – М.: Гелиос АРВ, 2002. – 480 с., ил.

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

3. Thomas W. Cusick, Pantelimon Stanica. “Cryptographic Boolean Functions and Applications” [Text] // Academic Press is an imprint of Elsevier 525 B Street, Suite 1900, San Diego, CA 92101-4495, USA Linacre House, Jordan Hill, Oxford OX2 8DP, UK. First edition 2009.

4. Зубов А.Ю., “Совершенные шифры”: [Текст]// М.: Гелиос АРВ 2003 160с., ил.

5. Криптография и алгоритмы шифрования – [электронный ресурс]// [http://vse-shifri.ru/].

6. Bruce Schneier. “Applied Cryptography”[Text]// Second Edition: Protocols, Algorthms, and Source Code in C (cloth), Publication Date: 01/01/96.

IMPROVING THE EFFECTIVENESS OF INFORMATION SECURITY MODIFICATION CIPHER VIZHINERA

N. I. KORSUNOV[1]
A. I. TITOV[2]

1) Belgorod State University

e-mail: korsunov@intbel.ru

2) Belgorod State Technological University them. VG Shukhov

e-mail: titov@programist.ru

In article the problem of protection of the information is considered. Methods preventing leak of the strategic information from servers, even are offered at reception by the malefactor of access to files. The developed cryptographic algorithm in which basis is displayed code number already checked up in the years is put by Vigenere. Modernisation of algorithm Vigenere offered by the author of article, allows to spend block iterations with key displacement, and gives the chance to apply it to any type of files.

Key words: Cryptography, cryptographic security, iterative encryption algorithm,moving probable word cryptanalysis, the code number of Vizhenera, block enciphering.