Рандомизированный алгоритм для псевдо-вероятностного шифрования

И. К. Абросимов
Кафедра «Информационная безопасность»
СПбГЭТУ «ЛЭТИ»
ivnabr@yandex.ru

Д. Н. Молдовян[1], Н. А. Молдовян[2]
Кафедра «Информационные системы»
СПбГЭТУ «ЛЭТИ»
[1]nmold@mail.ru, [2]mdn.spectr@mail.ru

***Аннотация*. Представлен метод рандомизации псевдослучайного поточного шифрования и алгоритм, реализующий этот метод. Разработанный алгоритм направлен на обеспечение устойчивости к атакам с принуждением, при которых у атакующего есть возможность перехватывать зашифрованные тексты, соответствующие повторно зашифрованному исходному сообщению. Предлагаемый алгоритм выполняет рандомизированное поточное шифрование двух разных сообщений, фиктивного и секретного. Сообщения шифруются одновременно с использованием двух разделяемых ключей, фиктивного и секретного. Раскрытие каждого из исходных сообщений выполняется независимо с использованием одного из ключей и одного и того же алгоритма дешифрования. Ключи используются для генерации двух или трех потоков ключевых последовательностей, которые зависят от значения вектора инициализации и отправленного отправителем получателю зашифрованного текста.**

Ключевые слова: рандомизированный поточный шифр; отрицаемое шифрование; принуждающая атака; псевдо-вероятностное шифрование; вероятностное шифрование

I. Введение

Псевдовероятностное шифрование – это особый метод отрицаемого шифрования с разделяемым ключом. Понятие отрицаемого шифрования было введено в работе [1] в связи с проблемой обеспечения устойчивости криптографических протоколов к атакам с принуждением. В модели таких атак предполагается, что противник (принуждающий) может заставить отправителя или получателя или обе стороны какого-либо протокола связи раскрыть ключ шифрования и исходное сообщение после сеанса связи.

Схемы отрицаемого шифрования с открытым ключом могут применяться для предотвращения покупки голосов в системах интернет-голосования [2, 3] и обеспечения безопасности протоколов многосторонних вычислений [4]. Криптосхемы отрицаемого шифрования с разделяемым ключом представляют значительный интерес как новый механизм защиты информации в компьютерных и телекоммуникационных системах в случае принуждающих атак [5].

Псевдовероятностное шифрование (ПШ) было предложено в качестве конкретного метода отрицаемого шифрования с открытым ключом [5,6]. ПШ представляет собой процесс детерминированного шифрования, при котором два независимых сообщения, фиктивное и секретное, преобразуются одновременно с использованием двух ключей, фиктивного и секретного. ПШ характеризуется созданием одного зашифрованного текста, который может быть получен с помощью некоторого алгоритма вероятностного шифрования (ВШ), применяемого к фиктивному сообщению. Известны поточные [6] и блочные [7] алгоритмы ПШ, обладающие достаточно высокой производительностью. Чтобы обеспечить более высокое сопротивление блочных алгоритмов ПШ принуждающим атакам, при которых у принуждающего есть возможность повторить зашифрование одних и тех же исходных сообщений в работе [7] было предложено ввести рандомизацию в процедуру блочного ПШ.

Поскольку поточный алгоритм ПШ хорошо подходит к различным приложениям, связанным с компьютерной и телекоммуникационной безопасностью [8, 9], он представляет значительный интерес для изучения механизмов внедрения рандомизации в поточные алгоритмы ПШ. В настоящей статье предлагается метод рандомизации процедуры поточного ПШ, и предлагается алгоритм поточного ПШ на основе этого метода.

II. Метод рандомизированного поточного псевдо-вероятностного шифрования

A. Критерии проектирования

Для построения рандомизированного поточного алгоритма ПШ использовались следующие критерии:

- алгоритм должен обеспечивать двухстороннюю отрицаемость;
- алгоритм должен реализовывать процедуру шифрования типа потока;
- процесс ПШ должен реализовываться как одновременное шифрование секретного T и фиктивного M сообщений с использованием секретного Q и фиктивного K ключей, которые разделяются двумя сторонами протокола связи;
- зашифрованный текст, созданный поточным алгоритмом ПШ, должен быть вычислительно неотличим от зашифрованного текста, созданного некоторым алгоритмом ВШ, используемым для шифрования фиктивного сообщения с фиктивным ключом;

Работа выполнена при финансовой поддержке РФФИ, проект №18-57-54002-Вьет_а

- алгоритм должен обладать достаточно высокой производительностью (скоростью шифрования);
- соответствующий алгоритм дешифрования должен обеспечивать независимое восстановление секретного и фиктивного сообщений в зависимости от используемого ключа, секретного или фиктивного, соответственно.

B. Метод шифрования

Пусть $M=(m_1,m_2,...,m_z)$ и $T=(t_1,t_2,...,t_z)$ представлены в виде u-битовых символов m_i и t_i ($i=1,2,...,z$) и для генерации двух ключевых потоков Γ и Γ', зависящих от используемого ключа K и Q соответственно используется функция блочного шифрования E:

$$\Gamma=\{(\alpha_1,\beta_1),(\alpha_2,\beta_2),...,(\alpha_z,\beta_z)\}$$
$$\Gamma'=\{(\alpha_1',\beta_1'),(\alpha_2',\beta_2'),...,(\alpha_z',\beta_z')\}$$

Элементы Γ и Γ' представляют собой пары u-битовых символов, которые используются для одновременного преобразования символов m_i и t_i. Последние значения пар преобразуются в процессе решения системы сравнений по модулю взаимно простых двоичных многочленов, представленных в виде следующих битовых строк $1\|\beta_i$, $1\|\beta_i'$, и ρ, где $\|$ - операция сцепления строк и ρ – случайный полином длины $u+1$. Рандомизация процесса преобразования пары m_i и t_i определяется использованием случайного значения ρ. Преобразование выполняется как решением следующей системы сравнений

$$\begin{cases} c_i \equiv m_i \oplus \alpha_i \bmod \eta \\ c_i \equiv t_i \oplus \alpha_i' \bmod \lambda \\ c_i \equiv t_i \oplus \mu \bmod \rho \end{cases}, \quad (1)$$

где c_i – i-й символ выходного шифротекста; $\oplus$ – побитовое сложение по модулю 2; $\eta=1\|\beta_i$; $\lambda=1\|\beta_i'$; μ – случайное u-битное значение. Элементы ключевых последовательностей Γ и Γ' зависят от ключа K и Q соответственно, номера символа сообщения i и от значения вектора инициализации V, который отправляется отправителем получателю посредством открытого канала. Таким образом, из-за использования случайного значения V различные сообщения будут зашифрованы различными парами ключевых последовательностей Γ и Γ'.

C. Алгоритм генерации элементов ключевых последовательностей

Предположим, что блок входных данных блочной функции E шифрования имеет размер $2n$ бит, а значения V и i записываются как n битовых строк, где $n>u$. Следующий алгоритм обеспечивает генерацию требуемых пар ключевых элементов (α_i,β_i) и (α_i',β_i').

Вход: значения K, Q, V и i.

1. Вычислить значение $E_K(V,i) \bmod 2^{2u}=(\alpha_i,\beta_i)$.
2. Вычислить значение $E_Q(V,i) \bmod 2^{2u}=(\alpha_i',\beta_i')$.
3. Составить двоичный многочлен $\lambda=1\|\beta_i'$.
4. Составить двоичный многочлен $\eta=1\|\beta_i$.
5. Вычислить значение $\gcd(\eta,\lambda)$, где gcd обозначает наибольший общий делитель.
6. Если $\gcd(\eta,\lambda)\neq 1$, то преобразовать битовую строку β_i в соответствии с формулой $\beta_i \leftarrow (\beta_i+1) \bmod 2^u$, где битовая строка β_i рассматривается как число, представленное в двоичном виде и перейти к шагу 3.
7. В качестве i-го элемента ключевой последовательности Γ взять пару u-битовых символов (α_i,β_i) и пару (α_i',β_i') в качестве i-го элемента ключевой последовательности Γ'.

Выход: ключевые элементы (α_i,β_i) и (α_i',β_i').

III. Рандомизированный поточный псевдовероятностный алгоритм шифрования

Шифрование потока двух сообщений M и T выполняется как последовательное преобразование пар u-битовых символов m_i и t_i в $3u$-битовые символы c_i выходного зашифрованного текста следующим образом:

Вход: значения K, Q, V, i, m_i и t_i.

1. Используя алгоритм из подраздела С второго раздела вычислить ключевые элементы (α_i,β_i) и (α_i',β_i').
2. Сгенерировать u-битовое случайное μ.
3. Сгенерировать $u+1$-битовое случайное ρ.
4. Считая $\alpha+\beta$ двоичным (1) полиномом, вычислить $\gcd(\eta,\rho)$ и $\gcd(\lambda,\rho)$, где $\eta=1\|\beta_i$ и $\lambda=1\|\beta_i'$.
5. Если $\gcd(\eta,\rho)\neq 1$ или $\gcd(\lambda,\rho)\neq 1$, то перейти к шагу 3.
6. Вычислить $3u$-bit символ c_i выходного шифротекста, используя следующую формулу:

$$c_i=\left[m_i\lambda\rho\left(\lambda^{-1}\rho^{-1} \bmod \eta\right)\oplus t_i\eta\rho\left(\eta^{-1}\rho^{-1} \bmod \lambda\right)\oplus \mu\eta\lambda\left(\eta^{-1}\lambda^{-1} \bmod \rho\right)\right] \bmod \eta\lambda\rho \quad (2)$$

Выход: i-ый символ c_i шифротекста.

Формула (2) описывает решение системы линейных сравнений (1), поэтому обратное преобразование символа сі выполняется как вычисление значения и значения. Можно отметить, что каждый из символов t_i и m_i можно восстановить независимо другой. Поэтому во время атаки с принуждением каждая сторона протокола связи может открывать только фиктивное сообщение и фиктивный ключ, утверждая, что они использовали алгоритм поточного ВШ, описанный ниже.

Вход: значения K, V и сообщение M.

1. Для каждого $i = 1,2,...,z$ выполнить шаги:

1.1. Вычислить $E_K(V,i) \bmod 2^{2u} = (\alpha_i, \beta_i)$

1.2.Сформировать двоичный полином $\eta = 1 \| \beta_i$

1.3. Сгенерировать u-битовое случайное μ.

1.4. Сгенерировать $u+1$-битовое случайное ρ.

1.5. Считая ρ двоичным полиномом, вычислить $\gcd(\eta, \rho)$

1.6. Если $\gcd(\eta, \rho) \neq 1$, то перейти к шагу 1.4

1.7. Вычислить $3u$-bit символ c_i выходного шифротекста, используя следующую формулу:

$$c_i = [m_i \rho(\rho^{-1} \bmod \eta) \oplus \mu\eta(\eta^{-1} \bmod \rho)] \bmod \eta\rho \quad (3)$$

2. Соединить символы c_i шифротекста в криптограмму $C = (c_1, c_2, ..., c_z)$

Выход: шифротекст C.

Описанный алгоритм ВШ называется ассоциированным поточным алгоритмом ВШ. Формула (3) описывает решение следующей системы двух сравнений:

$$\begin{cases} c_i \equiv m_i \oplus \alpha_i \bmod \eta \\ c_i \equiv t_i \oplus \mu \bmod \rho \end{cases}, \quad (4)$$

Поэтому легко видеть, что восстановление символа сообщения m_i из символа c_i зашифрованного текста криптограммы C, созданной с помощью ассоциированного поточного алгоритма ВШ, может быть выполнено с помощью формулы (4). Таким образом, дешифрование криптограммы C выполняется с помощью следующего алгоритма дешифрования.

Вход: значения K, V и шифротекст C.

1. Для каждого $i = 1,2,...,z$ выполнить шаги:

1.1. Вычислить $E_K(V,i) \bmod 2^{2u} = (\alpha_i, \beta_i)$

1.2.Сформировать двоичный полином $\eta = 1 \| \beta_i$

1.3. Вычислить u-битовый символ m_i раскрываемого сообщения по формуле $m_i \equiv c_i \oplus \alpha_i \bmod \eta.$

2. Соединить символы m_i сообщения $M = (m_1, m_2, ..., m_z)$

Выход: сообщение M.

Для раскрытия секретного сообщения следует использовать как секретный ключ, так и фиктивный ключ. Соответственно, алгоритм восстановления секретного сообщения из зашифрованного текста отличается от алгоритма восстановления фиктивного сообщения и описывается следующим образом.

Вход: значения Q, K, V и шифротекст C.

1. Для каждого $i = 1,2,...,z$ выполнить шаги:

1.1. Вычислить значения $E_K(V,i) \bmod 2^{2u} = (\alpha_i, \beta_i)$ и $E_Q(V,i) \bmod 2^{2u} = (\alpha_i', \beta_i')$.

1.2.Сформировать двоичные полиномы $\eta = 1 \| \beta_i$ и $\lambda = 1 \| \beta_i'$

1.3. Вычислить $\gcd(\eta, \lambda)$

1.4. Если $\gcd(\eta, \lambda) \neq 1$, то преобразовать битовую строку β_i в соответствии с формулой $\beta_i \leftarrow (\beta_i + 1) \bmod 2^u$, где битовая строка β_i рассматривается как число, представленное в двоичном виде и перейти к шагу 1.2.

1.5. Вычислить u-битовый символ t_i раскрываемого сообщения по формуле $t_i \equiv c_i \oplus \alpha_i' \bmod \lambda.$

2. Соединить символы t_i сообщения $T = (t_1, t_2, ..., t_z)$

Выход: секретное сообщение T.

Последние два алгоритма используют разные шаги для генерации ключевых элементов, используемых для расшифровки символов шифротекста c_i. Это различие может потенциально использоваться принуждающим, чтобы продемонстрировать, что стороны протокола связи используют другой ключ, отличный от открытого ключа K. Например, у него есть потенциальная возможность исследовать машинный код программы, выполняющей дешифрование на мобильном устройстве получателя сообщения. Чтобы обеспечить устойчивость в момент принудительных атак такого типа, следует использовать метод ПШ, в котором определены полностью схожие процедуры восстановления фиктивного и секретного сообщений. В следующем разделе описывается конструкция рандомизированного ПШ, удовлетворяющего этому требованию.

IV. Способ обеспечения схожести процедур расшифрования фальшивого и сектретного сообщений

Чтобы обеспечить сходство процедур расшифрования, предполагается использовать фальшивый ключ (K, S) и секретный ключ (Q, S), где S является дополнительным подключом, определяющим генерацию третьего ключевого потока $\Gamma^* = \{\omega_1, \omega_2, ..., \omega_z\}$, где $\omega_i = E_S(V,i) \bmod 2^u$. Кроме того, указывается неприводимый двоичный многочлен ψ, имеющий степень, равную u. Ключи (K, S) и (Q, S) генерируются так, что зависящие от ключа значения $\varphi = k \oplus s$ и $\varphi' = q \oplus s$ удовлетворяющих соотношению $\varphi \neq \varphi'$, где k, q и s – битовые строки, совпадающие с u правыми битами подключей K, Q и S соответственно.

Шифрование пары символов m_i и t_i входных сообщений M и T осуществляется как вычисление символа промежуточного шифротекста $b_i = (b_i', b_i'')$, после чего выполняется рандомизация значения b_i. Значение b_i вычисляется как решение системы уравнений в конечном поле $GF(2^u)$:

$$\alpha + \beta = \gamma. \quad (1) \qquad (1)$$

$$\begin{cases} b_i' \oplus (\varphi \oplus \omega_i) b_i'' \equiv m_i \oplus \alpha_i \bmod \psi \\ b_i' \oplus (\varphi' \oplus \omega_i) b_i'' \equiv t_i \oplus \alpha_i' \bmod \psi \end{cases} \quad (5)$$

Для рандомизации значения b_i имеется сгенерированный двоичный полином θ степени $2u$, зависящий от ключа, такой, что $\gcd(\theta, \psi) = 1$ и случайная u-битовая строка μ, после чего решается два сравнения

$$\begin{cases} c_i \equiv b_i \bmod \theta \\ c_i \equiv \mu \bmod \psi \end{cases} \quad (6)$$

Описанный метод преобразования пар символов определяет следующий рандомизированный поточный алгоритм ПШ

Вход: значения K, Q, S, V, сообщение M и сообщение T.

1. Вычислить значения $\varphi = k \oplus s$ и $\varphi' = q \oplus s$.

2. Для каждого $i = 1, 2, \ldots, z$ выполнить.

2.1. Вычислить значения $(\alpha_i, \beta_i) = E_K(V, i) \bmod 2^{2u}$, $(\alpha'_i, \beta'_i) = E_Q(V, i) \bmod 2^{2u}$ и $\omega_i = E_S(V, i) \bmod 2^u$.

2.2. Вычислить значения $b_i = (b'_i, b''_i)$ как решение системы сравнений (5).

2.3. Сформировать двоичный полином $\theta' = \omega_i \| \omega_i$ степени $2u - 1$.

2.4. Сформировать двоичный полином $\theta = 1 \| \theta'$ степени $2u$.

2.5. Вычислить $\gcd(\theta, \psi)$. Если $\gcd(\theta, \psi) \neq 1$, то $\theta' \leftarrow \theta' + 1 \bmod 2^{2u}$, где $2u$-битовая строка интерпретируется как число, записанное в двоичной форме, и переход к шагу 2.4.

2.6. Сгенерировать u-битовую строку μ.

2.7. Вычислить символ шифртекста c_i как решение системы двух сравнений (6):

$$c_i = \left[\mu\theta(\theta^{-1} \bmod \psi) \oplus b_i \psi(\psi^{-1} \bmod \theta)\right] \bmod \psi\theta .$$

3. Соединив символы c_i составить выходную криптограмму $C = (c_1, c_2, \ldots, c_i, \ldots, c_z)$.

Выход: шифротекст C.

Расшифрование шифротекста осуществляется с помощью поточного алгоритма расшифрования.

Вход: ключ (W, S), шифротекст C, и значение инициализирующего вектора V.

1. Вычислить значения $\varphi = w \oplus s$, где w представляет собой u правых бит подключа W.

2. Для каждого $i = 1, 2, \ldots, z$ выполнить следующие шаги.

2.1. Вычислить значения $(\alpha_i, \beta_i) = E_W(V, i) \bmod 2^{2u}$ и $\omega_i = E_S(V, i) \bmod 2^u$.

2.2. Сформировать двоичный полином $\theta' = \omega_i \| \omega_i$ степени $2u - 1$.

2.3. Сформировать двоичный полином $\theta = 1 \| \theta'$ степени $2u$.

2.4. Вычислить $\gcd(\theta, \psi)$. Если $\gcd(\theta, \psi) \neq 1$, то $\theta' \leftarrow \theta' + 1 \bmod 2^{2u}$, где $2u$-битовая строка интерпретируется как число, записанное в двоичной форме, и переход к шагу 2.3.

2.5. Вычислить значение $b_i = (b'_i, b''_i) = c_i \bmod \theta$.

2.6. Вычислить значение d_i:

$$d_i = \left[b_i' \oplus (\varphi \oplus \omega_i) b_i''\right] \oplus \alpha_i \bmod \psi .$$

3. Соединив символы d_i составить раскрываемое сообщение $D = (d_1, d_2, \ldots, d_i, \ldots, d_z)$.

Выход: исходное сообщение D.

Если $W = K$, то $D = M$. Если $W = Q$, то $D = T$.

$\alpha + \beta = V$. Заключение (1)

Известные поточные алгоритмы ПШ определяют детерминированный процесс шифрования, состоящий в одновременном преобразовании двух независимых сообщений, фальшивого и секретного. В статье впервые предложен метод рандомизации поточных алгоритмов ПШ. На основе этого метода были разработаны рандомизированные алгоритмы ПШ, обеспечивающие достаточно высокую производительность. Описанные алгоритмы удовлетворяют критерию вычислительной неразличимости от связанных с ними поточных алгоритмов ВШ. Дополнительное внимание было уделено разработке поточных схем ПШ, в которых точно такой же алгоритм дешифрования используется для раскрытия фальшивого и секретного сообщений.

В дальнейших исследованиях мы планируем разработать метод комбинированного поточного шифрования, который будет чередовать вероятностное шифрование исходных текстовых символов с псевдо-вероятностным.

Список литературы

[1] Canetti R., Dwork C., Naor M., Ostrovsky R. Deniable Encryption // Proceedings Advances in Cryptology – CRYPTO 1997. Lectute Notes in Computer Science. Springer – Verlag. Berlin, 1997. Вып. 1294. С. 90–104.

[2] Barakat T.M. A New Sender-Side Public-Key Deniable Encryption Scheme with Fast Decryption // KSII Transactions on Internet and Information Systems. 2014. Вып. 8, № 9, С. 3231–3249.

[3] Meng B. A secure internet voting protocol based on non-interactive deniable authentication protocol and proof protocol that two ciphertexts are encryption of the same plaintext // Journal of Networks. 2009. Вып. 4, № 5, С. 370–377.

[4] Ishai Y., Kushilevitz E., Ostrovsky R., Prabhakaran M., Sahai A. Efficient Non-interactive Secure Computation // Advances in Cryptology – EUROCRYPT 2011: 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer – Berlin, Heidelberg, 2011. С. 406–425.

[5] Al-Majmar N.A., Nguyen Duc Tam, Nguyen Nam Hai, Nguyen Hieu Minh Deniability of Symmetric Encryption Based on Computational Indistinguishability from Probabilistic Ciphering // Information Systems Design and Intelligent Applications: Proceedings of the Fourth International Conference INDIA 2017. Advances in Intelligent Systems and Computing. Springer Nature - Singapore, 2018. Вып. 672, С. 209-218.

[6] Moldovyan N.A., Moldovyan A.A., Moldovyan D.N., Shcherbacov V.A. Stream Deniable-Encryption Algorithms // Computer Science Journal of Moldova. 2016. Вып. 24, № 1, С. 68–82.

[7] Moldovyan A.A., Moldovyan N.A., Berezin A.N., Shapovalov P.I. Randomized pseudo-probabilistic encryption algorithms // Proceedings of 2017 20th IEEE International Conference on Soft Computing and Measurements. 2017, С. 14–17.

[8] Zou M.H, Ma K, Wu K.J. Scan-based attack on stream ciphers: A case study on eSTREAM finalists // Computer science and technology. 2014. Вып. 29 С. 646–655.

[9] Hwang T., Gope P. Robust stream-cipher mode of authenticated encryption for secure communication in wireless sensor network // Security and communication networks. 2016. Вып. 9, № 7, С. 667-679.