УДК 003.26

ГИБРИДНЫЙ ПРОТОКОЛ ПСЕВДОВЕРОЯТНОСТНОГО БЕСКЛЮЧЕВОГО ШИФРОВАНИЯ[1]

Костина А. А. – н.с. НИЛ КБ и ПКС СПИИРАН,
anna-kostina1805@mail.ru

Мирин А. Ю. – к.т.н., с.н.с. НИЛ КБ и ПКС СПИИРАН,

mirin@cobra.ru

Молдовян Н.А. – д.т.н., г.н.с. НИЛ КБ и ПКС СПИИРАН,

nmold@mail.ru

Фахрутдинов Р.Ш. – к.т.н., зав. НИЛ КБ и ПКС СПИИРАН,

fahr@cobra.ru

Аннотация

Предложен общий способ построения протоколов отрицаемого шифрования, ориентированного на обеспечение защиты информации в случае атак с принуждением к раскрытию ключа шифрования. Представлен гибридный протокол псевдовероятностного коммутативного шифрования, в котором для выполнения аутентификации передаваемых сообщений используются открытые ключи участников защищенного коммуникационного протокола, а шифрование передаваемого сообщения реализуется по схеме бесключевого шифрования.

[1] Санкт-Петербургский институт информатики и автоматизации Российской академии наук. Работа выполнена при частичной финансовой поддержке РФФИ (проект № 18-57-54002-Вьет_а)

Введение

На практике возможны потенциальные атаки на протоколы и алгоритмы шифрования, в рамках которых атакующий получает ключ шифрования. Для обеспечения стойкости к таким атакам предложен способ отрицаемого шифрования [1], реализуемый в виде процесса псевдовероятностного (ПВ) шифрования, который является детерминированной процедурой совместного шифрования двух независимых сообщений и формирует шифртекст, вычислительно неразличимый от шифртекста, полученного в ходе вероятностного шифрования одного из сообщений. Концепция ПВ шифрования представлена в [2,3]. В статье [4] сделана попытка построения протоколов отрицаемого бесключевого шифрования, однако задача обеспечения защиты от атак с одновременным принуждением отправителя и получателя сообщений к раскрытию ключей шифрования осталась нерешенной.

В настоящей статье предложен протокол бесключевого ПВ шифрования, обладающей стойкостью к принуждающим атакам всех типов.

Протокол ПВ бесключевого шифрования.

Рассмотрим реализацию протокола отрицаемого коммутативного шифрования, который не требует наличия у отправителя и получателя секретного сообщения общих секретных значений. Стойкость к потенциальным атакам с принуждением со стороны активного нарушителя реализуется использованием в протоколе открытых ключей отправителя и получателя.

Воспользуемся схемами цифровой подписи Шнорра и открытого согласования ключа и разовыми открытыми ключами в качестве случайных значений. Предполагается, что пользователи А (отправитель) и В (получатель секретного сообщения) являются владельцами зарегистрированных в удостоверяющем центре открытых ключей y_A и y_B соответственно. Открытые ключи вычислялись по случайно выбранным значениям личных секретных ключей x_A и x_B в соответствии с формулами $y_A = g^{x_A} \bmod p$ и $y_B = g^{x_B} \bmod p$, где p – достаточно большое простое число; g – число, порядок которого по модулю p равен достаточно большому простому числу r (разрядности 160 – 512 бит).

В качестве специфицированного параметра протокола используется также число α, являющееся примитивным элементом по модулю p.

Отправитель передает по открытому каналу связи секретное сообщение T:

1. Пользователь A генерирует случайное число k_A, удовлетворяющее условию $0 < k_A < \mathrm{p} - 1$, и вычисляет значение $R_A = \alpha^{k_A} \bmod p$, получая разовый открытый ключ. Используя свой личный секретный ключ x_A, по схеме Шнорра формирует свою подпись $\mathrm{Sign_A}(R_A)$ к значению R_A и направляет значения R_A и $\mathrm{Sign_A}(R_A)$ пользователю B.
2. Пользователь B, используя открытый ключ y_A, проверяет подлинность подписи $\mathrm{Sign_A}(\mathrm{R_A})$. Если подпись подлинная, он генерирует случайное число k_B, удовлетворяющее условию $0 < k_B < p - 1$, и вычисляет значение $R_B = \alpha^{k_B} \bmod p$ – разовый открытый ключ. Используя свой личный секретный ключ x_B, формирует свою подпись $\mathrm{Sign_B}(R_A)$ к значению R_A и свою подпись $\mathrm{Sign_B}(R_B)$ к значению R_B и направляет значения R_B, $\mathrm{Sign_B}(R_A)$ и $\mathrm{Sign_B}(R_B)$ пользователю A.
3. Пользователь A, используя открытый ключ y_B, проверяет подлинность подписи $\mathrm{Sign_B}(R_A)$ к случайному значению, которое он направлял пользователю B, и подлинность подписи $\mathrm{Sign_B}(R_B)$ к значению R_B. Если обе подписи подлинные, то он зашифровывает и передает секретное сообщение T $(T < p)$ пользователю B, участвуя в выполнении следующих интерактивных шагов:

   3.1. Вычисляет значение $U_{AB} \equiv R_B^{k_A} \equiv \alpha^{k_B k_A} \bmod p$, используемое как разовый общий секретный ключ.

   3.2. Вычисляет общий секретный ключ $Z_{AB} \equiv y_B^{x_A} \equiv g^{x_B x_A} \bmod p$.

   3.3. Вычисляет сеансовый ключ $K = Z_{AB} R_A R_B \bmod p$.

   3.4. Генерирует фиктивное сообщение $M < p$ и разовый личный секретный ключ (e_1, d_1, e_2, d_2), элементы которого удовлетворяют условиям $e_1 d_1 = 1 \bmod p - 1$ и $e_2 d_2 = 1 \bmod p - 1$.

   3.5. Вычисляет шифртекст $C^{(1)}$ как решение $C^{(1)} = \left(C_1^{(1)}, C_2^{(1)}\right)$ следующей системы уравнений в конечном поле $GF(p)$ относительно неизвестных $C_1^{(1)}$ и $C_2^{(1)}$:

$$\begin{cases} U_{AB} C_1^{(1)} + U_{AB}^{2} C_2^{(1)} = T^{e_1} \bmod p, \\ K C_1^{(1)} + K^2 C_2^{(1)} = M^{e_2} \bmod p. \end{cases}$$

4. Пользователь B вычисляет значения $U_{BA} \equiv R_A^{k_B} \equiv \alpha^{k_B k_A} \bmod p$ (разовый общий секретный ключ), $Z_{BA} \equiv y_A^{x_B} \equiv g^{x_A x_B} \bmod p$ (общий секретный ключ) и $K = Z_{BA} R_A \; R_B \bmod p$ (сеансовый ключ). Затем вычисляет шифртекст $C^{(2)} = \left(C_1^{(2)}, C_2^{(2)}\right)$:

   4.1. Вычисляет значения C′ и C″ по формулам

$$C' = U_{BA} C_1^{(1)} + U_{BA}^2 C_2^{(1)} = T^{e_1} \bmod p,$$
$$C'' = K C_1^{(1)} + K^2 C_2^{(1)} = M^{e_2} \bmod p.$$

   4.2. Генерирует свой личный секретный ключ $(\varepsilon_1, \delta_1, \varepsilon_2, \delta_2)$ в соответствии с требованием выполнимости аналогичных соотношений $\varepsilon_1\delta_1 = 1 \bmod p - 1$ и $\varepsilon_2\delta_2 = 1 \bmod p - 1$. Затем формирует криптограмму $C^{(2)}$ как решение $C^{(2)} = \left(C_1^{(2)}, C_2^{(2)}\right)$ следующей системы сравнений относительно неизвестных $C_1^{(2)}$ и $C_2^{(2)}$:

$$\begin{cases} U_{AB} C_1^{(2)} + U_{AB}^2 C_2^{(2)} = C'^{\varepsilon_1} = T^{e_1\varepsilon_1} \bmod p \\ K C_1^{(2)} + K^2 C_2^{(2)} = C''^{\varepsilon_2} = M^{e_2\varepsilon_2} \bmod p \end{cases}.$$

   4.3. Направляет шифртекст $C^{(2)} = \left(C_1^{(2)}, C_2^{(2)}\right)$ пользователю A.

5. Пользователь A формирует шифртекст $C^{(3)} = \left(C_1^{(3)}, C_2^{(3)}\right)$:

   5.1. Вычисляет значения C' и C'' по формулам

$$C' = U_{BA} C_1^{(2)} + U_{BA}^2 C_2^{(2)} = T^{e_1\varepsilon_1} \bmod p,$$
$$C'' = K C_1^{(2)} + K^2 C_2^{(2)} = M^{e_2\varepsilon_2} \bmod p.$$

   5.2. Формирует криптограмму $C^{(3)}$ как решение $C^{(3)} = \left(C_1^{(3)}, C_2^{(3)}\right)$ системы сравнений относительно неизвестных $C_1^{(3)}$ и $C_2^{(3)}$:

$$\begin{cases} U_{AB} C_1^{(3)} + U_{AB}^2 C_2^{(3)} = C'^{d_1} = T^{e_1 d_1 \varepsilon_1} = T^{\varepsilon_1} \bmod p, \\ K C_1^{(3)} + K^2 C_2^{(3)} = C''^{d_2} = M^{e_2 d_2 \varepsilon_2} = M^{\varepsilon_2} \bmod p. \end{cases}$$

   5.3. Вычисляет свою подпись $\mathrm{Sign}_A(C^{(3)})$ к шифртексту

$C^{(3)} = \left(C_1^{(3)}, C_2^{(3)}\right)$ и направляет $C^{(3)}$ пользователю B.

6. Пользователь B проверяет подлинность подписи $\text{Sign}_A(C^{(3)})$. Если подпись подлинная, расшифровывает секретное сообщение T по формуле

$$T = \left(U_{AB} C_1^{(3)} + U_{AB}^2 C_2^{(3)}\right)^{\delta_1} = \left(T^{\varepsilon_1}\right)^{\delta_1} = T^{\varepsilon_1 \delta_1} \bmod p,$$

а фиктивное – по формуле

$$M = \left(K C_1^{(3)} + K^2 C_2^{(3)}\right)^{\delta_2} = \left(M^{\varepsilon_2}\right)^{\delta_2} = M^{\varepsilon_2 \delta_2} \bmod p.$$

В случае принуждающей атаки пользователям A и B следует раскрыть атакующему секретные ключи x_A, x_B, (e_2, d_2) и $(\varepsilon_2, \delta_2)$ соответственно и фиктивное сообщение M. Предполагается, что атакующий может, используя ключи, вычислить из перехваченных шифртекстов сообщение M и проверить связь шифртекстов с раскрытым сообщением.

При этом перехваченные шифртексты $C^{(1)}$, $C^{(2)}$ и $C^{(3)}$ потенциально могли быть сгенерированы при определенных значениях случайных параметров в вероятностном протоколе, описанном далее.

Ассоциируемый протокол вероятностного коммутативного шифрования

В данном протоколе пользователи используют открытые ключи друг друга y_A и y_B для проверки цифровых подписей и формирования общего секретного ключа. Этим обеспечивается защита от активного нарушителя, пытающегося выдать себя за отправителя или получателя сообщения M, включая случай подмены подлинного пользователя после одного или нескольких выполненных шагов. Протокол включает следующие шаги:

1. Пользователь A генерирует случайное R_A и, используя свой личный секретный ключ x_A, формирует свою подпись $\text{Sign}_A(R_A)$ к значению R_A и направляет значения R_A и $\text{Sign}_A(R_A)$ пользователю B.
2. Пользователь B, используя открытый ключ y_A, проверяет подлинность подписи $\text{Sign}_A(R_A)$. Если подпись подлинная, то он генерирует случайное число R_B и, используя свой личный секретный ключ x_B,

формирует подписи $\mathrm{Sign}_B(R_A)$ и $\mathrm{Sign}_B(R_B)$. Затем он направляет значения R_B, $\mathrm{Sign}_B(R_A)$ и $\mathrm{Sign}_B(R_B)$ пользователю A.

3. Пользователь A, используя открытый ключ y_B, проверяет подлинность подписей $\mathrm{Sign}_B(R_A)$ и $\mathrm{Sign}_B(R_B)$. Если обе подписи подлинные, зашифровывает и передает секретное сообщение T $(T < p)$ пользователю B, участвуя в выполнении следующих шагов:
   3.1. Вычисляет значение $Z_{AB} = y_B{}^{x_A} \bmod p$ (общий секретный ключ) и сеансовый ключ $K = Z_{AB} R_A R_B \bmod p$.
   3.2. Генерирует разовый личный секретный ключ (e_2, d_2), такой, что $e_2 d_2 = 1 \bmod p - 1$, и случайные значения $R < p$ и $r < p$. Затем вычисляет шифртекст $C^{(1)}$ как решение $C^{(1)} = \left(C_1^{(1)}, C_2^{(1)}\right)$ следующей системы уравнений относительно $C_1^{(1)}$ и $C_2^{(1)}$:

$$\begin{cases} RC_1^{(1)} + rC_2^{(1)} = 1 \bmod p, \\ KC_1^{(1)} + K^2 C_2^{(1)} = M^{e_2} \bmod p. \end{cases}$$

Затем направляет шифртекст $C^{(1)} = \left(C_1^{(1)}, C_2^{(1)}\right)$ пользователю B.

4. Пользователь B вычисляет общий секретный ключ $Z_{BA} = y_A{}^{x_B} \bmod p$ и сеансовый ключ $K = Z_{BA} R_A R_B \bmod p$. Затем формирует ответный шифртекст $C^{(2)} = \left(C_1^{(2)}, C_2^{(2)}\right)$:
   4.1. Вычисляет значение $C' = KC_1^{(1)} + K^2 C_2^{(1)} = M^{e_2} \bmod p$.
   4.2. Генерирует личный секретный ключ $(\varepsilon_2, \delta_2)$, согласно $\varepsilon_2 \delta_2 = 1 \bmod p - 1$, и случайные значения $R < p$ и $r < p$. Затем он вычисляет шифртекст $C^{(2)} = \left(C_1^{(2)}, C_2^{(2)}\right)$ как решение системы сравнений относительно $C_1^{(2)}$ и $C_2^{(2)}$:

$$\begin{cases} RC_1^{(2)} + rC_2^{(2)} = 1 \bmod p, \\ KC_1^{(2)} + K^2 C_2^{(2)} = C'^{\varepsilon_2} = M^{e_2 \varepsilon_2} \bmod p. \end{cases}$$

4.3. Направляет шифртекст $C^{(2)} = \left(C_1^{(2)}, C_2^{(2)}\right)$ пользователю А.

5. Пользователь *A* формирует ответный шифртекст $C^{(3)} = \left(C_1^{(3)}, C_2^{(3)}\right)$:

5.1. Вычисляет значение $C'' = KC_1^{(2)} + K^2 C_2^{(2)} = M^{e_2 \varepsilon_2} \bmod p$.

5.2. Генерирует случайные значения $R < p$ и $r < p$ и формирует $C^{(3)}$ как решение $C^{(3)} = \left(C_1^{(3)}, C_2^{(3)}\right)$ системы сравнений относительно $C_1^{(3)}$ и $C_2^{(3)}$:

$$\begin{cases} RC_1^{(3)} + rC_2^{(3)} = 1 \bmod p, \\ KC_1^{(3)} + K^2 C_2^{(3)} = C''^{d_2} = M^{e_2 d_2 \varepsilon_2} = M^{\varepsilon_2} \bmod p. \end{cases}$$

5.3. Вычисляет подпись $\mathrm{Sign}_A(C^{(3)})$ к шифртексту $C^{(3)} = \left(C_1^{(3)}, C_2^{(3)}\right)$ и направляет $C^{(3)}$ и $\mathrm{Sign}_A(C^{(3)})$ пользователю *B*.

6. Пользователь *B* проверяет подпись $\mathrm{Sign}_A(C^{(3)})$. Если подпись верная, он расшифровывает секретное сообщение *M* по формуле $M = \left(KC_1^{(3)} + K^2 C_2^{(3)}\right)^{\delta_2} = \left(M^{\varepsilon_2}\right)^{\delta_2} = M^{\varepsilon_2 \delta_2} \bmod p$.

Чтобы доказать, что из переданных шифртекстов можно вычислить не только *M*, но и другое осмысленное сообщение, атакующему потребуется решить задачу дискретного логарифмирования и найти один из разовых личных секретных ключей k_A или k_B (это позволит вычислить разовый общий секретный ключ *U*) и вычислить одну из пар (e_1, d_1) и $(\varepsilon_1, \delta_1)$, которые ему не были раскрыты. Ввиду вычислительной сложности такой задачи обман участников протокола ОШ остается нераскрытым.

Заключение

Представлен гибридный протокол псевдовероятностного коммутативного шифрования, в котором для выполнения аутентификации передаваемых сообщений используются открытые ключи участников защищенного коммуникационного протокола, а шифрование

передаваемого сообщения реализуется по схеме бесключевого шифрования.

Литература

1. Canetti R., Dwork C., Naor M., Ostrovsky R. Deniable Encryption // Proceedings Advances in Cryptology – CRYPTO 1997. Lectute Notes in Computer Science. Springer – Verlag. 1997. vol. 1294. pp. 90–104.
2. Moldovyan N.A., Al-Majmar N.A., Duc Tam Nguyen, Nam Hai Nguyen, Hieu Minh Nguyen. 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 Pte Ltd. 2018. vol. 672. pp. 209-218. (DOI 10.1007/978-981-10-7512-4_21).
3. Moldovyan A.A., Moldovyan N.A. Practical Method for Bi-Deniable Public-Key Encryption // Quasigroups and related systems. 2014. Vol. 22. P. 277-282.
4. Moldovyan N.A., Shcherbacov A.V., Eremeev M.A. Deniable-encryption protocols based on commutative ciphers // Quasigroups and related systems. 2017. vol. 25. no. 1. pp. 95–108.