УДК 004.056:003.26

ПРОТОКОЛЫ ВСТРАИВАНИЯ ОБЩЕЙ ПАМЯТИ В СИММЕТРИЧНЫЙ КАНАЛ СЕКРЕТНОЙ СВЯЗИ ДЛЯ ПРОТИВОДЕЙСТВИЯ MITM-АТАКАМ

Александров А.В., Сорокин И.И.

Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых, Владимир, e-mail: alex_izi@mail.ru

В статье предложены протоколы создания симметричного ключа и обмена сообщениями между Отправителем и Получателем, устойчивыми по отношению к атакам пассивного и активного противника в модели секретной связи, в частности к атакам типа «человек посередине». Для решения этого ряда задач вводится общая память у Отправителя и Получателя в модели симметричной секретной связи К. Шеннона. Для сеанса связи Отправитель и Получатель обмениваются предварительным ключом, содержащим ссылки на сегменты общей памяти. Симметричный криптографический ключ строится на основе предварительного. Показано, что, при перехвате этого вектора, противник в канале связи получает некоторую долю информации о значении симметричного ключа, но недостаточную для определения этого ключа. Для поддержания удаленной целостности общей памяти предложено использовать технику деревьев Дамгарда – Меркла, для этого разрабатывается криптографический протокол. Рассмотрены варианты статической и динамической общей памяти. Использование динамической общей памяти может снять проблему передачи ключа по каналу связи. Предложенные протоколы имеют универсальный характер и могут быть использованы как самостоятельно, так и в протоколах семейства TLS передачи данных.

Ключевые слова: криптографический протокол, протокол Диффи – Хеллмана, общая память, дерево Дамгарда – Меркла, хэш-функция

MEMORY INTEGRATION PROTOCOLS IN THE SYMMETRIC SECRET COMMUNICATION CHANNEL FOR COUNTERING MITM-ATTACK

Aleksandrov A.V., Sorokin I.I.

Vladimir State University named after Alexander Grigoryevich and Nikolai Grigoryevich Stoletov, Vladimir, e-mail: alex_izi@mail.ru

The article proposes protocols for creating a symmetric key and exchanging messages between Sender and Receiver stable in relation to attacks of a passive and active adversary in secret communication models, in particular, to the attacks of the type «man in the middle». To solve this series of problems, the common memory of the Sender and the Recipient is introduced in the model of K. Shannon's symmetric secret communication. For a communication session, the Sender and the Recipient exchange a preliminary key containing links to segments of shared memory. Symmetric cryptographic key is built on the basis of preliminary. It is shown that when intercepting this vector, the adversary in the communication channel receives a certain amount of information about the value of the symmetric key, but not enough to determine this key. To maintain the remote integrity of shared memory, it is proposed to use the Damgard-Merkle tree technique; for this, a cryptographic protocol is being developed. The options for static and dynamic shared memory are considered. Using dynamic shared memory can solve the problem of transmitting a key over a communication channel. The proposed protocols are universal in nature and can be used both independently and in protocols of the TLS family of data transmission. The proposed protocols are universal and can be used in the TLS data transmission protocols.

Keywords: cryptographic protocol, Diffie-Hellman protocol, shared memory, Damgard-Merkle tree, hash function

Один из первых двусторонних протоколов открытого распределения ключей предложен в 1976 г. У. Диффи, М. Хеллманом. Протокол основан на трудности решения задачи дискретного логарифма в мультипликативной группе большого порядка с образующей α.

1. A→B: $\alpha^x \bmod p$
2. B→A: $\alpha^y \bmod p$ (DH)
3. A: $k_{AB} = ((a^y)^x) \bmod p$
4. A: $k_{BA} = ((a^x)^y) \bmod p$

Несмотря на то, что задача дискретного логарифма NP-трудна, существует вариант атаки «человек посередине» активного противника (MITM), против которого DH-протокол бессилен, ввиду отсутствия в нем аутентификации сторон. Для проведения атаки противник С не обязан решать задачу дискретного логарифмирования, однако, пользуясь отсутствием аутентификации в (DH), встает посередине между А и В, представляясь для А абонентом В, для В соответственно абонентом А. Далее, не обнаруживая себя, он осуществляет подмену передаваемых значений на свои значения, не решая при этом задачу дискретного логарифма и контролируя после завершения протокола обе части разорванного канала связи между А и В.

1. A→C: $\alpha^x \bmod p$
2. C→B: $\alpha^x \bmod p$
3. B→C: $\alpha^y \bmod p$
4. C→A: $\alpha^y \bmod p$ (MITM DH)

5. A: $k_{AC} = (\alpha^{y^*})^x \bmod p$
6. B: $k_{BC} = (\alpha^{y})^{x^*} \bmod p$
7. C: $k_{AC} = (\alpha^{x})^{y^*} \bmod p$
8. C: $k_{CB} = (\alpha^{x^*})^{y} \bmod p$

В связи с востребованностью DH-протокола в приложениях, приведены многочисленные варианты усиления DH в направлении аутентификации сторон A и B, с относительным сохранением свойств протоколов открытого распределения колючей. Далее мы воспользуемся только описанием направлений усиления, пользуясь детальным обзором в книге [1].

Первые усиления основаны на использовании доверенного центра и введении доверенных сертификатов, связывающих участников протокола $X \in \{A, B\}$ со значениями a^x, a^y и подписью доверенного центра Sig_T:

$$cert_x = \left(X, \alpha^x \bmod p, Sig_T\left(X, \alpha^x \bmod p\right), X \in \{A, B\}\right).$$

В первоначальном варианте семействе протоколов MTI [1, с. 174] без участия доверенного центра предполагается у участников протокола секретных ключей, и формулы выработки секретного ключа мультипликативно зависят от их значений. В результате MITM-атаки противника, все стороны протокола получат различные ключи, что будет обнаружено в результате чтения переданных сообщений. В последующих усилениях MTI появляется зависимость и от открытых ключей участников работы протокола.

В нашей работе в рамках модели секретной связи К. Шеннона мы предлагаем аддитивные протоколы порождения симметричного ключа и обмена данными, противодействующие атаке человек посередине. Для этого нами используется предварительно созданная Отправителем A и Получателем B и разнесенная на соответствующие устройства общая память – целочисленное множество значений $D = \{d_1 \ldots d_n\}$. Строго говоря, использование общей памяти выходит за рамки протоколов открытого распределения ключей, и само использование общей памяти достаточно сильно идентифицирует канал связи между абонентами A и B. Однако с учетом того, что значения общей памяти, вообще говоря, произвольны и не являются дополнительными идентификаторами, а представленные протоколы не требуют привлечения третьей доверенной стороны – арбитра, а также секретных и публичных ключей абонентов секретной связи, мы считаем, что такой подход имеет право на существование.

Цель: разработка протоколов создания симметричного ключа и обмена сообщениями между Отправителем и Получателем на основе общей памяти, устойчивых по отношению к активному и пассивному противнику в канале связи.

Определения

Обозначим здесь и далее $m > 1$ размер симметричного криптографического ключа k, $f_k^{\pm 1}(S)$ функции симметричного шифрования/расшифрования и пусть $H_k(S)$ – ключевая хэш-функция, сильно сопротивляющаяся поиску коллизий.

Определение 1: Назовем общей памятью Отправителя (A) и Получателя (B) – согласованное между Отправителем и Получателем и разнесенное на устройства Отправителя и Получателя числовое множество $D = \{d_1 \ldots d_n\}$, $n > 1$, $n < m$.

В качестве элементов D могут выступать файлы, расположенные на информационных носителях Отправителя A и Получателя B, и только на них. Смысл значений $D = \{d_1 \ldots d_n\}$, $n > 1$, $n < m$ нас не интересует. После разнесения общей памяти на устройства пользователей множество $D = \{d_1 \ldots d_n\}$, $n > 1$, $n < m$ и его элементы, если это необходимо, снабжаем индексом $X \in \{A, B\}$ для того, чтобы подчеркнуть, на каком устройстве мы рассматриваем само множество и его элементы. Наличие общих элементов Отправителя A и Получателя B, конечно, выходит за рамки модели секретной связи К. Шеннона, однако не противоречит модели угроз Долев – Яо, предложенной в 1981 г. Д. Долевым и А. Яо для детального описания среды, в которой происходит обмен шифрованными сообщениями, при наличии в ней пассивного или активного противника C.

Процедура создания общей памяти, выработки первого значения симметричного ключа и первого ключевого хэша не предполагает использования канала связи и выполняется в момент создания множества D.

Определение 2. Свяжем со значениями общей памяти предварительный ключ

$$E = \left(e_1 \ldots e_n\right), e_i \in \left\{0,1\right\}, E \neq 0. \qquad (1)$$

Для выработки симметричного ключа используем аддитивную формулу

$$k_E = \sum_{i=1}^{n} e_i d_i \bmod 2^m, \qquad (2)$$

где, напомним, m – размер сеансового ключа шифрования.

Утверждение 1: Пусть $D = \{d_1 \ldots d_n\}$, $n > 1$, $n < m$ – общая память, созданная на устройствах отправителя и получателя ($D^A = D^B$) в модели секретной связи К. Шеннона, E – предварительный ключ, k_E – сеансовый ключ. Тогда противник С по значению предварительного ключа $E \neq 0$ не может построить симметричный ключ k_E. Это можно выразить в терминах условной энтропии К. Шеннона. $H(S|S_i)$: меры неопределенности значения S при условии обладания значением S_i

$$\begin{cases} H(k_E \mid \{E, D\}) = 0 \\ H(k_E \mid \{E\}) = H(k_E) - \delta(|E|) \end{cases}, \quad (3)$$

где $\delta(|E|) < H(k_E)$.

В терминах условной энтропии Шеннона, понимаемой относительно вероятностных распределений для значений k_E, E, D, первое равенство означает, что неопределенность значения ключа k_E по значениям теней E, D полностью снимается. Второе соотношение определяет меру неопределенности симметричного ключа для противника С при перехвате предварительного ключа и несовершенность схемы разделения секрета следует из работы [2].

Доказательство использует технику (2,2) пороговых схем разделения секрета [2], для которых секретным значением является ключ k_E. Долями секрета являются соответственно значения D, E.

В силу разбалансирования размеров долей секрета друг относительно друга, схема разделения, порождаемая формулой (2), является (2,2) пороговой, не идеальной, не совершенной. Математическая конструкция подобных схем исследована в работах Kurosawa. В частности, из неидеальности СРС вытекает последняя формула в (3).

Криптографические протоколы создания симметричного ключа и передачи данных

Пусть противник С в канале связи пассивный, перехватывает весь трафик в канале связи, но не имеет возможности подмены.

Приведем протокол создания симметричного ключа, безопасный по отношению к пассивному противнику.

1. $A \to B : E = \{e_1 \ldots e_n\} \neq 0$
2. $A : k_{AB} = \sum e_i d_i \bmod 2^m$ (4)
3. $B : k_{BA} = \sum e_i d_i \bmod 2^m$

В силу утверждения 1 противник С не имеет возможности построить значение $k_{BA} = k_{AB}$.

Для противодействия активному противнику, который может перехватывать и видоизменять передаваемый трафик, воспользуемся ключевой хэш-функцией $H_{d_e}(S)$ с ключом d_e – предыдущим сгенерированным сеансовым ключом. Приведем двусторонний протокол формирования нового симметричного ключа, безопасный по отношению к активному противнику

1. $A \to B : E = \{e_1 \ldots e_n\} \| H^A_{d_e}(E)$
2. $B : H^B_{d_e}(E)$; если $H^b_{d_e}(E) \neq H^A_{d_e}(E)$, то стоп
3. $A : k_{AB} = \sum e_i d_i \bmod 2^m ; H^A_{d_e}(k_{AB})$
4. $A : k_{BA} = \sum e_i d_i \bmod 2^m ; H^B_{d_e}(k_{BA})$ (5)
5. $A \to B : H^A_{d_e}(k_{AB})$
6. B: если $H^A_{d_e}(k_{AB}) = H^B_{d_e}(k_{BA})$, то $d_e \leftarrow k_{BA}$, иначе стоп
7. $B \to A : Ok$
8. A: если Ok, то $d_e \leftarrow k_{AB}$.

Нетрудно видеть, что такое усиление протокола (4), в силу свойств $H_{d_e}(S)$ и утверждения 1, эффективно противостоит активному противнику в канале связи. В самом деле, противник на шаге (5.1) может легко изменить вектор E, однако не может построить функцию $H_{d_e}(S)$, так как не имеет доступа к общей памяти в модели Долев – Яо, хотя и знаком с ее описанием. В то же время абонент B, зная $H_{d_e}(S)$, легко может обнаружить факт подмены предварительного ключа Е, на шаге (5.2) протокола передачи. Противник также может совершить перехват и подмену $H_{d_e}(k_{AB})$ на шаге (5.5), разрушая протокол создания симметричного ключа, однако на шаге (5.6) этот факт также будет обнаружен без выработки значения *OK*.

При обмене сообщениями отправитель посылает получателю сообщение, зашифрованное на выработанном ключе, при этом совместно с зашифрованным сообщением отправляет хэш-функцию на выработанном ключе,

1. $A \to B : f_k(S) \| H^A_k(S)$
2. $B : f_k^{-1}\left(f_k(S) \| H^A_k(S)\right)$ (6)
3. B: если $H^B_k(S) = H^A_k(S)$, то Ok.

При корректном завершении работы протокола и равенстве значений хеш-функций на шаге (6.3), сообщение S считаем успешно переданным.

Дерево Дамгарда – Меркла и контроль целостности общей памяти

Существует принципиальная возможность удаленного контроля целостности общей памяти отправителя и получателя, использующая технику деревьев Дамгарда – Меркла. Дерево Дамгарда – Меркла – бинарное дерево, листовыми вершинами которого являются хэш-значения от элементов общей

памяти, а внутренние вершины являются результатом попарной конкатенации пар смежных хэшей на соседнем нижнем этаже бинарного дерева (рис. 1). Структура деревьев Дамгарда – Меркла предложена в работе «A Digital Signature Based on a Conventional Encryption Function» Ральфом Мерклом, и сегодня находит свое применение в контроле целостности больших массивов цифровых данных, в частности в электронных валютах стандарта Bitcoin [3].

Преимущество структур Деревьев Дамгарда – Меркла следующие:

1. Деревья Дамгарда – Меркла позволяют обеспечивать целостность и достоверность данных.

2. Деревья Дамгарда – Меркла требуют небольшого объема памяти или дискового пространства, поскольку доказательства Дамгарда – Меркла являются логарифмическими по вычислительной сложности, а поэтому легкими и быстрыми.

3. Доказательства Дамгарда – Меркла требуют малого количества информации, которая должна передаваться по сетям связи.

4. Может использоваться любая из доверенных функций хэширования, которая сильно сопротивляется коллизиям.

Обозначим дерево Дамгарда – Меркла $RDM^X(D)$, $X = \{A, B\}$, построенное над общей памятью соответственно Отправителя для $X = A$ и Получателя для $X = B$. Представленные выше старты протоколов порождения ключа и передачи данных можно завязать на успешное сравнение мастер-хешей $RootHash(RDM^A(D)) = RootHash(RDM^B(D))$ в противном случае, проигрывается парный протокол доказательства Меркла, позволяющий выловить различия в общей памяти Отправителя и Получателя.

Процедура порождения симметричного ключа на основе предварительного – критически важна для работы всех предложенных протоколов, несмотря на ее безопасность в смысле утверждения 1. Активный противник имеет возможность постоянно вмешиваться в протокол, на стадиях передачи данных в режиме подмены сообщений, разрушая работу протокола и обнаруживая себя. Существует конструкция, позволяющая избежать частого применения этого протокола создания симметричного ключа, основанная на выделении динамической компоненты в области общей памяти. Этот подход кратко обсудим в следующем разделе.

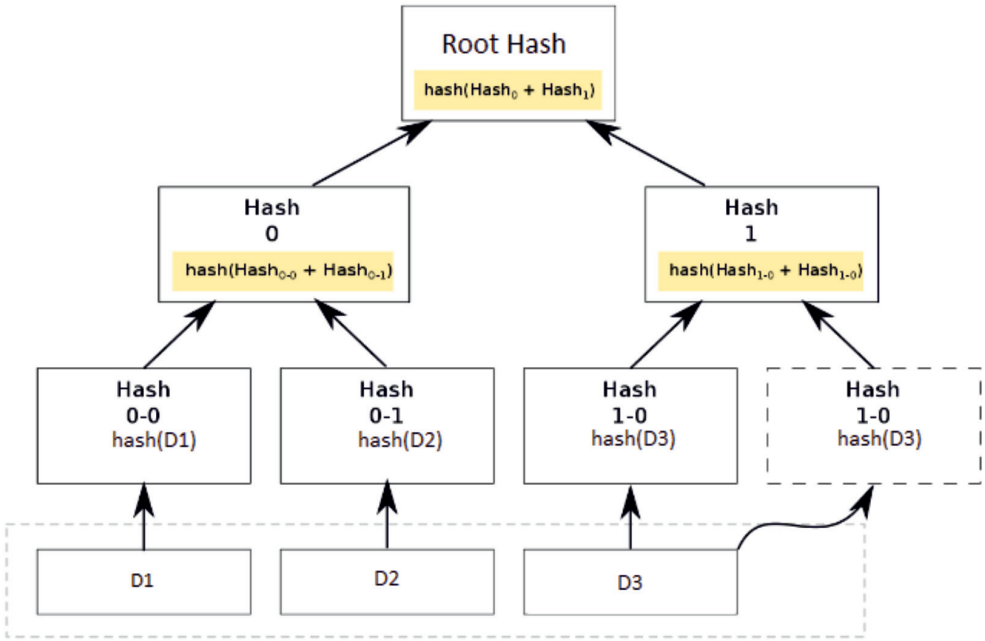

Рис. 1. Дерево Дамгарда – Меркла

Где $D_1 \ldots D_n$ – элементы общей памяти

Hash 0-0, 0-1 … N-N – листовые вершины дерева Дамгарда – Меркла

Hash 0, 1, N – вершины дерева Дамгарда – Меркла

RootHash – корневой хэш всего дерева Дамгарда – Меркла – мастер-хэ

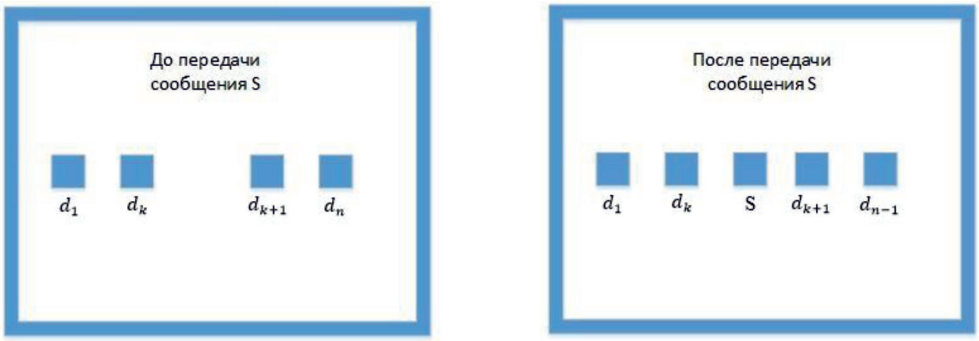

Рис. 2. Общая память с динамической компонентой

Динамическая общая память

Выделим в общей памяти статическую и динамические области $D = \{d_1 \ldots d_k, d_{k+1} \ldots d_n\}$, $\{d_1 \ldots d_k\}$ – статическая компонента общей памяти, и $\{d_{k+1} \ldots d_n\}$ – динамическая компонента общей памяти, представленная в виде очереди. Пусть S удачно переданное сообщение. Под удачно переданным сообщением подразумевается, что при передаче в протоколе (5) имеет место равенство значений на шаге (6.3), корректно завершающее работу протокола передачи. Перемещаем значение S в общую память на место значения d_{k+1} и перемещаем на одну позицию все значения общей памяти динамического сегмента $\{d_{k+1} \ldots d_n\}$. Более точно изменение значения общей памяти выглядит следующим образом (рис. 2):

$$D = \{d_1 \ldots d_k, d_{k+1} \ldots d_n\} \to D = \\ = \{d_1 \ldots d_k, S, d_{k+1} \ldots d_{n-1}\},$$

где d_i – элемент общей памяти,
S – передаваемое сообщение.

Такое динамическое перестроение общей памяти завершается автоматическим пересчетом значений общей памяти, пересчетом элементов деревьев Меркла $RDM^X(D)$, $X = \{A, B\}$ и соответствующим изменением значения сеансового ключа (2) без передачи по протоколам (4–5) и изменения предварительного ключа.

Из такой конструкции перестроения общей памяти следует, что с использованием динамической компоненты общей памяти, можно не создавать новый симметричный ключ по протоколу (4–5), а просто пересчитать значения $k_{AB} = k_{BA}$ в (2) по измененным значениям общей памяти, в том случае если элементы динамической области отмечены в предварительном ключе единицей.

Заключение

Выделенный подход использования общей памяти, конечно, выходит за границы протоколов открытого распределения ключей, так как заранее отправитель и получатель в канале связи договариваются об общей числовой базе. Однако отметим то, что значения общей памяти не обязаны служить идентификаторами пользователей, или их секретными и открытыми ключами, и имеют общий порядок. Кроме того, приведенные протоколы на завязаны на стандарты шифрования и хэширования. В частности, с использованием уникальности хэш-функции для Отправителя и Получателя $H_k^B(S) = H_k^A(S)$ с неизвестным значением ключа для противника С, созданной на стадии протоколов (4–5) легко преобразовать сам протокол Диффи – Хеллмана, сделав его устойчивым к MITM атаке противника С.

4. $A \to B : \alpha^x \bmod p \| H_k(\alpha^x \bmod p)$
5. $B \to A : \alpha^y \bmod p \| H_k(\alpha^y \bmod p)$
6. $A : k_{AB} = (\alpha^y)^x \bmod p$ (*DH**)
7. $A \to B : H_k(k_{AB})$
8. $B : k_{BA} = (\alpha^x)^y \bmod p$
9. *B*: если $H_{d_e}^A(k_{AB}) = H_{d_e}^B(k_{BA})$, то *Ok*, иначе СТОП.

Для этого к передаваемым значениям в DH-протоколе на шаге (*DH**.1-2) прикрепляем значение хэш-функции от передаваемых данных, где хэш-функция построена на ключе, сформированном по протоколу (5). Кроме того, на шаге (*DH**.4) передается значение хэш от построенного ключа на стороне А. На шаге (*DH**.6) сторона В либо подтверждает создание нового симметричного ключа в случае равенства значений хэшей, либо обнаруживает MITM-атаку противника С.

По терминологии документов международной организации Internet Engineering Task Force (IETF), существует современная классификация свойств, характеризующих безопасность криптографических протоколов, в которой выделены 20 пунктов, разделенные по 10 группам [1, с. 13–21].

Свойства безопасности, характеризующие основные протоколы

Протокол	Свойство G_i														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
TLS	×	×	×				×			×	×		×		
TLS-v1.1	×	×	×				×			×	×		×		
TLS-SPR	×	×	×				×			×	×		×		
TLS-sharedkeys	×	×	×				×			×	×		×		

Атака типа «человек посередине» нарушает свойства протокола безопасности G1, G2, G4, G5, G7, G13. Далее приведена таблица, в которой отмечено, какими свойствами безопасности обладает ряд криптографических протоколов семейства TLS (таблица).

Протокол Диффи – Хеллмана встроен в семейство TLS-протоколов [4], поэтому в таблице обращаем внимание на тотальную уязвимость протоколов семейства: TLS, TLS-v1.1, TLS-SRP, TLS-sharedkeys по отношению к свойствам G1, G2, G7, G13 в силу уязвимости DH к атаке активного противника «человек посередине». Представленные протоколы (5), (*DH**) закрывают перечисленные уязвимости.

В заключение отметим, что протоколы (4), (5), (*DH**), предложенные в работе, удовлетворяют требованиям безопасных аутентифицированных протоколов обмена ключами.

Понятие безопасного аутентифицированного протокола обмена ключами было введено У. Даффи, П. ван Ооршоном и М. Вайнером в 1992 г. в работе «Authentication and Authenticated Key Exchanges». По определению, такой аутентифицированный протокол безопасен, если при каждом выполнении протокола двумя участниками гарантируется выполнение двух условий:

1) если сторона A принимает идентификатор стороны B, записи в сообщениях, передаваемых обеими сторонами, осуществляются правильно;

2) никто кроме сторон A и B не может определить передаваемый ключ.

Протоколы (4), (5), (*DH**), предложенные в работе, удовлетворяют этим свойствам, поскольку общая память для сторон A и B едина, то в вышеописанных свойствах 1–2 A и B можно поменять местами. Использование общей памяти в криптографических протоколах встречается не впервые и, в частности, использовано в работе [5].

С учетом использования общей памяти в канале секретной связи, протоколы (3), (4), (5), (*DH**) ограничены в использовании, имеют парный, долговременный и безопасный характер секретной передачи данных для абонентов A, B, по отношению к противнику C в канале связи, как пассивном, так и активном.

Благодаря использованию динамической общей памяти удается снять одну из традиционных проблем симметричной криптографии – процедуру создания и передачи симметричного ключа. С динамическим изменением и синхронизацией общей памяти D с привлечением конструкции дерева Дамгарда – Меркла симметричный ключ k_E по формуле (2) автоматически изменяется на устройствах A и B.

Работа выполнена при поддержке и финансировании Российского фонда фундаментальных исследований, грант № 18-01-00596А.

Список литературы

1. Черемушкин А.В. Криптографические протоколы. Основные свойства и уязвимости: учеб. пособие для студ. учреждений высш. проф. образования. М.: Издательский центр «Академия», 2009. 272 с.

2. Kurosawa K., Okada K., Sakano K., Ogata W., Tsujii S. Nonperfect secret sharing schemes and matroids, LNCS 765, Advances in Cryptology, Proceedings of Eurocrypt'93, Springer Verlag. 1993. P. 126–141.

3. Israa Alqassem, Davor Svetinovic. Towards Reference Architecture for Cryptocurrencies: Bitcoin Architectural Analysis (англ.). IEEE International Conference on Internet of Things (iThings), and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom): журнал. 2014. Сентябрь. ISBN 978-1-4799-5967-9.

4. Oppliger, Rolf. Introduction. SSL and TLS: Theory and Practice. 2nd. Artech House, 2016. P. 13. ISBN 978-1-60807-999-5.

5. Александров А.В., Сорокин И.И. Симметричная рюкзачная криптографическая система с общей памятью, основанная на рекуррентных базисах в задаче об укладке рюкзаков // Известия высших учебных заведений. Приборостроение. 2019. Т. 62. № 4. С. 320–330.