УДК 004.056.55, 512.54 DOI: 10.14529/cmse200101

РАЗРАБОТКА И РЕАЛИЗАЦИЯ ГРУППОВОГО ПРОТОКОЛА ГЕНЕРАЦИИ КЛЮЧА НА БАЗЕ IKE

© 2020 А.А. Волохов, Ю.В. Косолапов

Южный федеральный университет

(344066 Ростов-на-Дону, ул. им. Большая Садовая, д. 105/42)

E-mail: sashavolohov@yandex.ru, itaim@mail.ru

Поступила в редакцию: 16.07.2019

В качестве основы информационного взаимодействия участников в недоверенной среде часто выступает протокол выработки общего секретного ключа. С помощью такого ключа в дальнейшем может быть построен защищенный канал или защищенная сеть связи. В настоящее время актуальна задача разработки протоколов генерации общего ключа для группы участников. Одним из способов построения таких протоколов является обобщение протокола для двух участников на случай нескольких участников. В работе строится протокол генерации общего секретного ключа для группы участников (для конференции). В основе разработанного протокола лежит протокол IKE (Internet Key Exchange) из семейства протоколов IPSec для двух участников, обеспечивающий выполнение таких свойств безопасности, как аутентификация субъекта и сообщения, генерация новых ключей, защита от чтения назад, защита от повтора и ряда других. Стойкость разработанного протокола генерации ключа основана на сложности задачи дискретного логарифмирования в циклической группе. В работе исследуются свойства безопасности, обеспечиваемые построенным протоколом, в частности, исследуется стойкость к коалиционным атакам, актуальным для групповых протоколов. Также отмечаются некоторые особенности практического применения построенного протокола.

Ключевые слова: генерация секретного ключа, IKE, конференция.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Волохов А.А., Косолапов Ю.В. Разработка и реализация группового протокола генерации ключа на базе IKE // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2020. Т. 9, № 1. С. 5–19. DOI: 10.14529/cmse200101.

Введение

Развитие средств связи и совершенствование технологий коммуникации приводит к упрощению объединения различных субъектов связи в группы [1]. Например, средства видеонаблюдения, контроля доступа и охранной сигнализации объединяются в единую систему физической защиты помещений; устройства бытовой техники, объединенные с системой защиты помещений, формируют систему «умный дом»; видеокамеры и радары объединяются в систему «умный город»; ученые, эксперты, специалисты объединяются в группы для обсуждения интересующих вопросов; рядовые пользователи сети Интернет объединяются в группы для общения, игр и т.п. Часто каналы связи между участниками группы не защищены физически от пассивного или активного вмешательства, что создает угрозу конфиденциальности и целостности данным, передаваемым по каналам. Для нейтрализации этой угрозы могут применяться криптографические протоколы, базовым среди которых является протокол генерации общего для группы секретного ключа. Возможным способом построения протокола генерации ключа является обобщение использующихся на практике двухточечных протоколов (протоколов для двух участников). Такой подход обоснован, например, тем, что для использующихся на практике двухточечных протоколов известны обеспечиваемые этими протоколами свойства безопасности, сформулированные инженерным советом интернета IETF (Internet Engineering Task Force).

В настоящее время для защиты данных, передаваемых по сетевому протоколу IP (Internet Protocol), часто используется семейство двухточечных протоколов IPSec (IP security). В семейство протоколов IPSec кроме протоколов аутентификации сторон и шифрования передаваемых пакетов входит протокол генерации ключей IKE. В [2] отмечено, что протокол IKE из набора двадцати свойств безопасности, сформулированных организацией IETF, обеспечивает выполнение 10-ти свойств: G1-G3, G7, G9-G11, G13-G15.

В настоящей работе ставится задача построения и реализации группового протокола генерации ключей на основе протокола IKE. Кроме введения и заключения, работа содержит три раздела. Первый раздел посвящен обзору работ в области генерации групповых секретных ключей. Во втором разделе приводятся необходимые сведения о протоколе IKE и его стойкости. В третьем разделе на базе IKE строится групповой протокол генерации ключей, обосновывается его стойкость и рассматриваются свойства безопасности, обеспечиваемые этим протоколом в рамках модели угроз, предложенной в 1981 году Д. Долевым и А. Яо [3]. В конце второго раздела анализируется коммуникационная сложность протокола и рассматриваются особенности реализации разработанного протокола, в частности, проводится экспериментальная оценка времени генерации ключа конференцией в зависимости от числа участников.

1. Обзор работ

Способы генерации группового ключа можно разделить на сетевые протоколы и канальные протоколы. Канальные протоколы характеризуются тем, что для генерации ключа могут использоваться особенности среды передачи сигнала. В частности, в [4] и [5] предложен канальный протокол генерации общего секретного ключа для группы беспроводных устройств. Так как на канальном уровне для каждой пары устройств помеховая обстановка отличается от помеховой обстановки любой другой пары, то каждая такая пара на первом этапе может сгенерировать общий секретный ключ пары. Такие ключи далее используются на втором этапе для генерации общего ключа группы. Теоретические основы генерации в зашумленной среде общего ключа между парами участников заложены А. Вайнером в работе [6] в 1975 г.

В сетевых протоколах не доступна информация о помеховой обстановке, поэтому для генерации ключа используются иные криптографические примитивы. В работе [7] рассматривается случай, когда участники группы имеют общий пароль, и их задачей является генерация с помощью группового протокола Диффи—Хэллмана общего секретного ключа. Основным результатом [7] является доказательство стойкости группового протокола к атаке по словарю. В [8] групповой протокол Диффи—Хэллмана модифицируется с целью защиты от нарушителя, имеющего доступ к долговременным ключам. Для аутентификации участников группы в [9] разработан протокол генерации ключа, где групповой ключ широковещательно рассылается сервером, а для аутентификации участников группы используется схема разделения секрета Шамира.

Отметим, что в работах [7] и [8] участники протокола при генерации протокола взаимодействуют непосредственно друг с другом, а в [9] — через центральный сервер. Такие способы взаимодействия участников используются для относительно небольших групп. В случае большого числа участников возможно применение древовидной сети, где выделяются участники, ответственные за генерацию и передачу ключевого материала для других

субъектов. В [10] на основе протокола Диффи—Хэллмана строится протокол генерации ключа в децентрализованной древовидной сети; там же доказывается его стойкость.

В работе [11], с целью упрощения вычислений, групповой протокол генерации ключа строится на основе обобщенных полиномов Чебышева. При этом, как утверждается в [11] (без приведения доказательства), безопасность протокола не снижается по сравнению с протоколом на основе Диффи—Хэллмана. Отметим, что разработка доказуемо стойких криптографических протоколов, в том числе, протоколов генерации ключей, в основе стойкости которых не лежат задачи дискретного логарифмирования и факторизации, является актуальной в связи с развитием квантовых вычислений. К работам в этом направлении можно отнести работу [12], где строится схема распределения ключей на 3-дизайнах Адамара и доказывается стойкость разработанной схемы к атакам коалиций мощности не более двух.

Несмотря на актуальность разработки новых криптографических протоколов для групп участников (в частности, протоколов, стойких в постквантовую эпоху), в настоящее время, с целью упрощения перехода от двухточечных протоколов к групповым, актуальна задача доработки существующих двухточечных протоколов до групповых версий. В настоящей работе групповой протокол генерации ключа строится на основе применяемого на практике двухточечного протокола IKE.

2. Двухточечный протокол генерации ключа

2.1. Протокол Диффи—Хэллмана

Ядром протокола IKE является протокол Диффи—Хэллмана для выработки общего ключа в незащищенном от прослушивания канале связи [13]. В удобном виде приведем протокол Диффи—Хэллмана. Пусть $G = \langle g \rangle$ — группа порядка p, порожденная элементом g. При выполнении протокола обе взаимодействующие стороны, обозначаемые C_1 и C_2, выбирают большие случайные числа $i_1, i_2 (\in \mathbb{Z}_p)$ — секретные ключи Диффи—Хэллмана — и выполняют шаги, указанные ниже. Результатом выполнения этого протокола для двух

Протокол Диффи—Хеллмана

1: $C_1 \to C_2 : pkey_1 = g^{i_1}$
2: $C_2 \to C_1 : pkey_2 = g^{i_2}$
3: $C_1 : SK = pkey_{2,1} = (pkey_2)^{i_1}$
4: $C_2 : SK = pkey_{1,2} = (pkey_1)^{i_2}$

участников является общий ключ SK, который может быть, например, ключом для симметричного шифрования пересылаемых в дальнейшем сообщений. Числа $pkey_1$ и $pkey_2$ будем далее называть открытыми *полуключами* Диффи—Хэллмана.

Стойкость протокола Диффи—Хэллмана основана на том, что только по полуключам $pkey_1$ и $pkey_2$ вычислительно сложно найти ключ SK. Предположение о вычислительной сложности этой задачи также называется *предположением Диффи—Хэллмана* (Diffie-Hallman assumption, DH) или *предположением Диффи—Хэллмана о сложности вычисления* (Computational Diffie-Hallman assumption, CDH). Оценка сложности этой задачи тесно связана с оценкой вычислительной сложности задачи дискретного логарифмирования [14], когда по заданному $a \in G$ требуется найти целое неотрицательное решение x уравнения $g^x = a$ [13]. Предположение о сложности решения задачи дискретного логарифмирования называют *предположением о дискретном логарифме* (Discrete Logarithm assumption,

DL). Заметим, что если имеется эффективный алгоритм решения задачи дискретного логарифмирования, то имеется эффективный алгоритм нахождения ключа SK по полуключам $pkey_1$ и $pkey_2$. С другой стороны, если нет эффективного алгоритма для нахождения секретного ключа по открытым полуключам Диффи—Хэллмана, то нет эффективного алгоритма и для задачи вычисления дискретного логарифма. Иногда стойкость протокола Диффи—Хэллмана доказывается в рамках *предположения Диффи—Хэллмана о принятии решения* (Decisional Diffie-Hallman assumption, DDH) [15]. Согласно предположению DDH, случайные векторы $(g^{I_1}, g^{I_2}, g^{I_1 I_2})$ и $(g^{I_1}, g^{I_2}, g^{X})$ вычислительно *неразличимы* для достаточно большого числа p, где I_1, I_2, X — случайные величины, принимающие значения равновероятно из $\mathbb{Z}_p$.

Недостатком протокола Диффи—Хеллмана является отсутствие защиты от атаки «человек посередине», а также от засоряющей атаки, в ходе которой одна из сторон многократно отправляет свои полуключи. В частности, ни инициатор соединения, ни отвечающая сторона не могут достоверно определить, кем является их собеседник, что позволяет реализовывать атаку типа человек посередине. Для защиты от такой атаки может быть использована, например, взаимная аутентификация на основе асимметричных криптоалгоритмов (см. далее протокол IKE). При выполнении засоряющей атаки, например, стороной C_1, сторона C_2 должна многократно вычислять полуключи протокола, что может привести к отказу в обслуживании со стороны C_2. Чтобы предотвратить засоряющую атаку, к протоколу добавляются два раунда, в которых стороны обмениваются дополнительными наборами данных о клиентах, называемыми cookies. В общем случае, cookies представляют собой результат вычисления хеш-функции от уникальных идентификаторов, таких как IP-адрес, номер сетевого порта, номер протокола и т.п. Добавление cookies позволяет отвечающей стороне отсекать непрекращающиеся запросы от инициатора в начале протокола.

2.2. Протокол IKE

В упрощенном виде протокол IKE из семейства протоколов IPSec состоит из двух фаз, выполняемых друг за другом. Целью первой фазы является выработка для сеанса связи ключевого материала (см. (1)–(4)), на основе которого во второй фазе вырабатываются ключи шифрования сообщений, передаваемых в рамках сеанса (см. (9)). Обозначим $[\![m]\!]_{\mathcal{A}}$ шифрование сообщения m на открытом ключе $\mathcal{A}$ по алгоритму асимметричного шифрования, согласованному участниками протокола. Пусть $\mathcal{I}_j$ — открытый ключ участника C_j, $j = 1, 2$; $h : \mathcal{K} \times \{0,1\}^* \to \mathcal{H}_1$ — ключевая криптографическая хеш-функция с множеством ключей $\mathcal{K}$ и множеством значений $\mathcal{H}_1$, $\mathrm{hash} : \{0,1\}^* \to \mathcal{H}_2$ — бесключевая криптографическая хеш-функция с множеством значений $\mathcal{H}_2$. Фазы протокола IKE приведены в протоколах *IKE.Фаза 1* и *IKE.Фаза 2*.

На первом шаге (раунде) первой фазы инициатор C_1 отправляет заголовок HDR_1, содержащий, в том числе, *cookies* и идентификатор передаваемого сообщения Msg_{ID} (для второй фазы), а также предложенные параметры безопасности SA_{pr}, включающие алгоритмы и параметры симметричного и асимметричного шифрования, алгоритмы вычисления хеш-значений. Отвечающая сторона отправляет свой заголовок HDR_2 и выбранные параметры безопасности SA_s. Далее участники обмениваются случайными числами $N_j^{(1)}$, собственными идентификаторами ID_j и полуключами, шифруя передаваемые данные на открытых ключах получающей стороны с целью взаимной аутентификации (и защиты от атаки «человек посередине»). После четвертого раунда участники вычисляют ключевой

Протокол *IKE.Фаза 1*

$C_1 \to C_2 : HDR_1, SA_{pr}$
$C_2 \to C_1 : HDR_2, SA_s$
$C_1 \to C_2 : HDR_1, pkey_1, [\![N_1^{(1)}, ID_1]\!]_{\mathcal{I}_2}$
$C_2 \to C_1 : HDR_2, pkey_2, [\![N_2^{(1)}, ID_2]\!]_{\mathcal{I}_1}$
C_1, C_2 : вычисление S, S_d, S_a, S_e
$C_1 \to C_2 : HDR_1, [H_1]_{S_e}$

Протокол *IKE.Фаза 2*

1: $C_1 \to C_2 : HDR_1, [H_1^2, SA_s, N_1^{(2)}, pkey_1, ID_1, ID_2]_{S_e}$
2: $C_2 \to C_1 : HDR_2, [\tilde{H}_2^2, SA_s, N_2^{(2)}, pkey_2, ID_2, ID_1]_{S_e}$
3: $C_1 \to C_2 : HDR_1, [H^2]_{S_e}$

набор, состоящий из первоначального ключа S, ключа для создания других ключей S_d, ключа аутентификации S_a и ключа шифрования S_e:

$$S = h(\mathrm{hash}(\langle \textit{набор } N^{(1)} \rangle), \langle \textit{набор cookies} \rangle), \tag{1}$$

$$S_d = h(S, SK | \langle \textit{набор cookies} \rangle | 0), \tag{2}$$

$$S_a = h(S, S_d | SK | \langle \textit{набор cookies} \rangle | 1), \tag{3}$$

$$S_e = h(S, S_a | SK | \langle \textit{набор cookies} \rangle | 2). \tag{4}$$

Здесь $\langle$*набор* $N^{(1)}\rangle$ имеет вид $\langle N_1^{(1)} | N_2^{(1)} \rangle$, сгенерированный ключ SK равен $pkey_{1,2}$, а $\langle$*набор cookies*$\rangle$ равен $\langle cookies_1 | cookies_2 \rangle$. Здесь и далее $\mathbf{a}|\mathbf{b}$ обозначает конкатенацию векторов (строк) $\mathbf{a}$ и $\mathbf{b}$. Проверка правильности сгенерированного ключевого набора (аутентификация ключа) выполняется на шаге 6, где первым участником для $j = 1$ вычисляется значение

$$H_j = h(S, \langle \textit{набор полуключей, известных участнику } C_j \rangle | \langle \textit{набор cookies} \rangle | SA_s | ID_j), \tag{5}$$

которое шифруется с помощью сгенерированного ключа S_e (символом $[m]_a$ обозначается симметричное шифрование сообщения m на ключе a в рамках согласованной участниками симметричной криптосистемы) и отправляется второму участнику. Второй участник имеет необходимые данные для вычисления H_j (для $j = 1$) и проверки правильности общего ключа. Во второй фазе на первых двух шагах участники удостоверяются, что обе стороны используют один ключевой набор (1)–(4), сгенерированный на первой фазе, при этом:

$$H_j^2 = h(S_a, Msg_{ID} | SA_s | N_j^{(2)}), \tag{6}$$

$$\tilde{H}_j^2 = h(S_a, Msg_{ID} | N_j^{(1)} | SA_s | N_j^{(2)}), \tag{7}$$

$$H^2 = h(S_a, 0 | Msg_{ID} | \langle \textit{набор } N^{(1)} \rangle). \tag{8}$$

При этом для взаимной аутентификации участников используют случайные числа, которыми они обменялись в первой фазе. Взаимная аутентификация во второй фазе позволяет участникам убедиться, что собеседник не подменен. Третий шаг второй фазы используется для аутентификации ключевого материала, на основе которого генерируется общий ключ K шифрования сообщений в сеансе:

$$K = h(S_d, \langle \textit{набор } N^{(2)} \rangle). \tag{9}$$

3. Групповой протокол генерации ключа

Для построения группового протокола генерации ключа рассмотрим обобщение протокола Диффи—Хэллмана на случай $n(\in \mathbb{N})$ участников. Множество натуральных чисел от 1 до n обозначим $[n]$. В обобщенном протоколе генерируемый ключ SK имеет вид $g^{i_1,...,i_n}$, где i_j — секретный ключ Диффи—Хэллмана участника C_j, $j \in [n]$. Опишем процедуру генерации общего ключа. Пусть

$$pkey_{l_1,...,l_r} = g^{i_{l_1},...,i_{l_r}} \tag{10}$$

— полуключ порядка r, где числа $l_1, ..., l_r$ принадлежат множеству $[n]$ и попарно различны. Схема генерации общего ключа состоит в выполнении $n-1$ итераций: на r-ой итерации участники вычисляют и обмениваются полуключами r-ого порядка, $r \in [n-1]$ (для простоты полагается, что вместе с полуключом r-ого порядка передаются номера участников, на основе секретных ключей которых построен этот полуключ). При вычислении полуключей r-ого порядка используются полуключи $r-1$ порядка: участник C_j возводит в степень i_j все такие полученные на предыдущей итерации полуключи $pkey_{l_1,...,l_{r-1}}$ порядка $r-1$, для которых $j \notin \{l_1, ..., l_{r-1}\}$. Каждый полуключ $pkey_{l_1,...,l_r}$, вычисленный участником C_j на r-ой итерации, передается такому участнику C_k, для которого $k \notin \{l_1, ..., l_r\}$. Таким образом, на $(n-1)$-ой итерации каждый из участников отправит по одному полуключу порядка $n-1$, с помощью которого может быть найден общий ключ SK.

В рамках модели угроз Долева-Яо, пассивный наблюдатель перехватывает все полуключи всех порядков. В [16] доказано, что если протокол Диффи—Хэллмана для двух пользователей вычислительно стойкий в рамках предположения DDH, то случайные векторы

$$(g^{I_1}, ..., g^{I_n}, g^{I_1 I_2}, ..., g^{I_{n-1} I_n}, ..., g^{I_1 ... I_{n-1}}, ..., g^{I_2 ... I_n}, g^X), \tag{11}$$

$$(g^{I_1}, ..., g^{I_n}, g^{I_1 I_2}, ..., g^{I_{n-1} I_n}, ..., g^{I_1 ... I_{n-1}}, ..., g^{I_2 ... I_n}, g^{I_1 ... I_n}) \tag{12}$$

вычислительно неразличимы, когда случайные величины I_1,...,I_n, X принимают значения случайно и равновероятно из $\mathbb{Z}_p$. Это позволяет строить групповую версию протокола IKE.

3.1. Групповой протокол

В настоящей работе строится протокол с сервером, через который осуществляются пересылки сообщений, формируемых участниками в процессе генерации ключа. Ключом SK для многопользовательского протокола является значение $g^{i_1 ... i_n i_r}$, где g^{i_j} — полуключи участников для $j = 1, ..., n$, а g^{i_r} — полуключ сервера R. Значения $N^{(1)}_{sum} = \sum_{i=1}^{n} N^{(1)}_j$, $cookies_{hash} = \mathrm{hash}(cookies_1, ..., cookies_n)$ и $N^{(2)}_{sum} = \sum_{i=1}^{n} N^{(2)}_j$ используются при вычислении S, S_d, S_a, S_e, H_j, H^2 и K в формулах (1)–(5), (8) и (9) вместо соответственно $\langle$*набор* $N^{(1)}\rangle$, $\langle$*набор cookies*$\rangle$, $\langle$*набор* $N^{(2)}\rangle$. Пусть $N^{(i)}_{sum,j} = N^{(i)}_{sum} - N^{(i)}_j$ для $i \in [2]$, $\mathcal{R}$ — открытый ключ сервера R; для указания на то, что отправителем данных $\langle$данные$\rangle$ является сервер R, будем использовать обозначение $\langle$данные$\rangle_r$;

$$\hat{H}^{2,j}_r = h(S_a, Msg_{ID} | \tilde{N}^{(1)}_j | SA_s | N^{(2)}_{sum}). \tag{13}$$

В протоколе для $3 \le j \le n$ символом $pkey(C_{j-1})$ обозначен набор полуключей, полученных сервером от участника C_{j-1}, $pkey(C_1) = \{pkey_{1,r}, pkey_{n,r}\}$, а $pkey(C_{j-1}, j)$ — набор полуключей, получаемых путем возведения в степень i_j (секретный ключ Диффи—Хэллмана

участника C_j) полуключей из $pkey(C_{j-1})$: $pkey(C_{j-1}, j) = \{pkey^{i_j} : pkey \in pkey(C_{j-1})\}$. В частности, $pkey(C_j) = pkey(C_{j-1}, j)$ для $j = 2, ..., n$. Символом $\overline{pkey}(C_{n-1})$ обозначим следующий набор полуключей:

$$\overline{pkey}(C_{n-1}) = \{pkey_\tau : n \notin \tau, pkey_\tau \in \cup_{l=1}^{n-1} pkey(C_l)\},$$

а символом $\overline{pkey}(C_n)$ обозначим набор полуключей, отправляемых участником с номером n серверу (условие $pkey \neq pkey_{1,...,n-1,r}$ гарантирует, что участник C_n не отправит серверу общий ключ SK по открытому каналу):

$$\overline{pkey}(C_n) = \{(pkey)^{i_n} : pkey_\tau \in \overline{pkey}(C_{n-1}), pkey \neq pkey_{1,...,n-1,r}\}.$$

Протокол *Фаза 1*

1: $R \to C_j,\ j \in [n] : HDR_r, [\![SA_{pr}, N_j^{(1)}, ID_r]\!]_{\mathcal{I}_j}$
2: $C_j \to R,\ j \in [n] : HDR_j, h(N_j^{(1)}, ID_j), pkey_j, [\![SA_{s,j}, \tilde{N}_j^{(1)}, ID_j]\!]_{\mathcal{R}}$
3: $R \to C_2 : HDR_r, pkey_1, pkey(C_1), h(N_2^{(1)}, \tilde{N}_2^{(1)} | ID_2)$
4: $C_2 \to R : HDR_2, pkey_{1,2}, pkey(C_1, 2)$
5: ...
6: $R \to C_j : HDR_r, pkey_{1,...,j-1}, pkey(C_1), ..., pkey(C_{j-1}), h(N_j^{(1)}, \tilde{N}_j^{(1)} | ID_j)$
7: $C_j \to R : HDR_j, pkey_{1,...,j}, pkey(C_1, j), ..., pkey(C_{j-1}, j)$
8: ...
9: $R \to C_n : HDR_r, pkey_{1,...,n-1}, \overline{pkey}(C_{n-1}), h(N_n^{(1)}, \tilde{N}_n^{(1)} | ID_n)$;
10: $C_n \to R : HDR_n, pkey_{1,...,n}, \overline{pkey}(C_n)$;
11: $R \to C_j,\ j \in [n-1] : HDR_r, pkey_{1,2,...,j-1,j+1,...,n,r}, [\![N_{sum,j}^{(1)}, cookies_{hash}]\!]_{\mathcal{I}_j}$
12: $C_j, j \in [n]$: вычисление $S, S_d, S_a, S_e, h(H_j, \tilde{N}_j)$
13: $C_j \to R,\ j \in [n] : HDR_j, [h(H_j, \tilde{N}_j)]_{S_e}$

Протокол *Фаза 2*

1: $C_{init} \to R : [Msg_{ID}, ID_{init}, ID_r]_{S_e}$
2: $R \to C_j,\ j \in [n] : [Msg_{ID}, ID_r, ID_j]_{S_e}$
3: $C_j, j \in [n]$: вычисление H_j^2
4: $C_j \to R,\ j \in [n] : HDR_j, [\tilde{H}_j^2, SA_s, N_j^{(2)}, ID_j, ID_r]_{S_e}$
5: $R \to C_j,\ j \in [n] : HDR_r, [\hat{H}_r^{2,j}, SA_s, N_{sum,j}^{(2)}, ID_r, ID_j]_{S_e}$
6: $C_j, j \in [n]$: вычисление H^2, K
7: $C_j \to R,\ j \in [n] : HDR_j, [H^2, ID_j, ID_r]_{S_e}$

Сервер в первой фазе на шагах 1 и 11 выполняет по n пересылок, а на шагах 2–10 сервер участвует $n-1$ раз как инициатор пересылок. С другой стороны, каждый участник в первой фазе является инициатором пересылки три раза: шаги 2, 13 и один раз на одном из шагов с 3 по 12. Таким образом, суммарное число пересылок в первой фазе составляет порядка $6n$. На второй фазе сервер выполняет $2n$ пересылок (шаги 2 и 5), а участники совершают в совокупности $2n+1$ пересылок (шаги 1, 4 и 7). Общее количество пересылок составляет порядка $\mathcal{O}(8n)$. Заметим, что число пересылок и объем пересылаемых данных в разработанном протоколе существенно меньше, чем при широковещательной рассылке

участниками полуключей r-ого порядка, когда $r \in [n-1]$, так как в последнем случае требуется совершить порядка $\mathcal{O}(n^3)$ пересылок.

Для примера в табл. 1 приведены наборы пересылаемых полуключей при $n = 4$. В таблице выделены полуключи, на основе которых каждый участник C_j может вычислить общий секретный ключ SK, возведя полученные от сервера полуключи порядка n в степень, равную соответствующему секретному ключу i_j (так как в протоколе участвует сервер, то общий ключ будет порядка $n+1$).

Таблица 1

Наборы передаваемых полуключей для $n = 4$ на шагах 3–13 первой фазы протокола

Раунд протокола	Полуключи
$C_j \to R$, $j = 1, ..., n$	$pkey_j$
$R \to C_2$	$pkey_1$,$pkey_{1,r}$,$pkey_{4,r}$
$C_2 \to R$	$pkey_{1,2}$,$pkey_{1,2,r}$, $pkey_{2,4,r}$
$R \to C_3$	$pkey_{1,2}$,$pkey_{1,r}$,$pkey_{4,r}$, $pkey_{1,2,r}$, $pkey_{2,4,r}$
$C_3 \to R$	$pkey_{1,2,3}$, $pkey_{1,3,r}$,$pkey_{3,4,r}$, $pkey_{1,2,3,r}$, $pkey_{2,3,4,r}$
$R \to C_4$	$pkey_{1,2,3}$, $pkey_{1,r}$, $pkey_{1,2,r}$, $pkey_{1,3,r}$, $pkey_{1,2,3,r}$
$C_4 \to R$	$pkey_{1,2,3,4}$, $pkey_{1,4,r}$, $pkey_{1,2,4,r}$, $pkey_{1,3,4,r}$
$R \to C_1$	$pkey_{2,3,4,r}$
$R \to C_2$	$pkey_{1,3,4,r}$
$R \to C_3$	$pkey_{1,2,4,r}$

3.2. Анализ свойств безопасности группового протокола

В протоколе IKE предусмотрены опции, применение которых обеспечивает свойства безопасности G1-G3, G7, G9-G11, G13-G15. Ниже приводится анализ построенного протокола на предмет выполнения свойств безопасности. Напомним, что, согласно модели Долева-Яо, нарушитель может: 1) получить любое сообщение, которое передается по сети, 2) устанавливать соединение с любым другим пользователем от своего имени, 3) стать стороной, принимающей сообщения, 4) передавать сообщения от имени других пользователей. С другой стороны, нарушитель не может: 1) угадывать случайные числа из достаточно большого диапазона, 2) расшифровывать сообщения, не имея ключа, 3) найти секретный ключ по открытому ключу, 4) получить доступ к закрытым внутренним ресурсам средств связи, таким как оперативная память или жесткий диск других пользователей.

Утверждение 1. *Пусть $\mathcal{A}$ — множество возможных нарушителей, $R \notin \mathcal{A}$. Подмена сервера R до выполнения первой и/или до выполнения второй фазы может быть выявлена участниками протокола за полиномиальное время.*

Доказательство. Подмена сервера до выполнения первой фазы будет выявлена на третьем шаге первой фазы: для аутентификации серверу требуется расшифровать случайное число, отправленное каждым участником, и зашифрованное на открытом ключе сервера, чтобы отправить значение $h(N_j^{(1)}, \tilde{N}_j^{(1)} | ID_j)$. В рамках модели Долева-Яо, нарушитель не может по открытому ключу найти секретный ключ и угадывать случайные числа из достаточно большого диапазона, поэтому нарушитель не может подменить сервер до начала первой фазы. Во второй фазе для аутентификации сервера также используются случайные

числа $\tilde{N}_j^{(1)}$ (см. (13)), которые по каналам связи передаются только в зашифрованном на открытом ключе сервера виде. Так как вычисление значения хеш-функции и расшифрование выполняется за полиномиальное время, то и аутентификация сервера выполняется за полиномиальное время. □

Утверждение 2. *Пусть $\mathcal{A}$ — множество возможных нарушителей. Каждый нарушитель из $\mathcal{A}$ в рамках модели угроз Долева-Яо имеет доступ к полному набору полуключей, доступных легитимному участнику C_j для всех $j \in [n]$ после завершения первой фазы протокола.*

Доказательство. Доказательство следует из того, что все наборы полуключей в протоколе передаются в незашифрованном виде. □

Из утверждения 2 следует, что для аутентификации участников, то есть для доказательства субъектом соответствия своему идентификатору, набора полуключей для вычисления H_j не достаточно. Для аутентификации необходимо использовать информацию, доступную только самим участникам. Такой информацией может быть, например, случайное число $\tilde{N}_j^{(1)}$, сгенерированное участником на шаге 2 первой фазы для дальнейшей аутентификации сервера (в рамках модели Долева-Яо, нарушитель не может угадывать случайные числа из достаточно большого диапазона). Поэтому в протоколе на шаге 12 вычисляется хеш-значение $h(H_j, \tilde{N}_j^{(1)})$ (так как число $\tilde{N}_j^{(1)}$ известно только участнику C_j и серверу R).

Утверждение 3. *Пусть $\mathcal{P} = \{C_1, ..., C_n, R\}$ — множество легитимных участников протокола генерации группового ключа, $\mathcal{A}$ — множество нарушителей, $R \notin \mathcal{A}$. Любой нарушитель из множества $\mathcal{A}$, выдающий себя в первой фазе протокола за легитимного участника из множества $\mathcal{P}$, будет выявлен на одном из шагов первой фазы.*

Доказательство. Пусть $A(\in \mathcal{A})$ — нарушитель. Рассмотрим два случая: $A \in \mathcal{A} \backslash \mathcal{P}$ и $A \in \mathcal{P}$. В первом случае нарушитель не является легитимным участником, однако в рамках модели Долева-Яо, ему неизвестны секретные ключи участников. Тогда нарушитель, выдающий себя за участника C_j в начале первой фазы, будет выявлен на шаге 2, так как не сможет расшифровать случайное число $N_j^{(1)}$, отправленное сервером. Заметим, что нарушитель может попытаться выдать себя за участника C_j на одном из шагов 2–10. На этих шагах сервером не проверяется источник сообщений, поэтому нарушитель не будет выявлен. И в этом случае ключ Диффи—Хэллмана SK' будет известен нарушителю, а, следовательно, могут быть вычислены ключи S', S'_d, S'_a и S'_e (здесь штрихом обозначены ключи, которые получены с помощью секретного ключа Диффи—Хэллмана i'_j, выбранного нарушителем A). Однако на шаге 13 сервер проверяет правильность ключа, используя значение $h(H_j, \tilde{N}_j^{(1)})$. Так как случайное число $\tilde{N}_j^{(1)}$ известно только легитимному участнику (и не может быть угадано нарушителем), аутентифицированному к этому моменту сервером на шагах 1 и 2, то проверку на шаге 13 может пройти только легитимный участник. Аналогично показывается, что нарушитель из $\mathcal{P}$ также будет выявлен на шаге 2 или 13. □

Утверждение 4. *Пусть $\mathcal{P} = \{C_1, ..., C_n, R\}$ — множество легитимных участников протокола генерации группового ключа, $\mathcal{A}$ — множество нарушителей, $R \notin \mathcal{A}$. Любой нарушитель из множества $\mathcal{A}$, выдающий себя во второй фазе протокола за легитимного участника из множества $\mathcal{P}$, будет выявлен на одном из шагов второй фазы.*

Доказательство. Предполагается, что к началу второй фазы участники аутентифицированы, следовательно серверу известны случайные числа $\tilde{N}_1^{(1)},...,\tilde{N}_n^{(1)}$, сгенерированные в первой фазе участниками, а участникам — случайные числа $N_1^{(1)},...,N_n^{(1)}$, сгенерированные в первой фазе сервером. В рамках модели угроз Долева-Яо нарушитель не имеет доступа к этим случайным числам, поэтому использование во второй фазе случайных чисел с первой фазы обеспечивает взаимную аутентификацию сторон. Взаимная аутентификация во второй фазе выполняется с помощью хеш-значений $\tilde{H}_j^2$ и $\hat{H}_r^{2,j}$. □

Из утверждений 1, 3 и 4 следует, что для первой и второй фазы группового протокола выполняется свойство G1 — аутентификация субъекта.

Для аутентификации передаваемых в ходе протокола сообщений используются хеш-значения H_j (шаг 13 в первой фазе) и H^2 (шаг 7 во второй фазе), которые вычисляются с помощью криптографической хеш-функции с использованием случайных чисел, известных только серверу и легитимным участникам. Таким образом, выполняется свойство G2 — аутентификация сообщений. Использование случайных чисел, как в первой фазе, так и во второй фазе обеспечивает также защиту от повтора (свойство G3), защиту от чтения назад (свойство G9). Защита от чтения назад обеспечивается за счет того, что компрометация ключа SK не позволяет в случае использования криптографической хеш-функции h найти ключевой материал (1)–(4), зависящий не только от SK, но и от случайных чисел, передаваемых между клиентами и сервером в зашифрованном виде. Применение случайных чисел в каждой фазе обеспечивает формирование новых ключей (свойство G10). Однако во второй фазе не выполняется свойство совершенной прямой секретности (PFS, Perfect Forward Security), так как во второй фазе ключи генерируются на основе ключа Диффи—Хэллмана, полученного в первой фазе. Генерация во второй фазе ключей Диффи—Хэллмана, как показали эксперименты (см. раздел 3.3), существенно замедляет скорость обмена сообщениями: с ростом числа участников время генерации общего ключа растет нелинейно. Отметим, что в двухточечном протоколе IKE предусмотрен режим, обеспечивающий PFS.

Утверждение 5. *Пусть $\mathcal{P} = \{C_1, ..., C_n, R\}$ — множество легитимных участников протокола генерации группового ключа, $\mathcal{A}$ — множество нарушителей, $R \notin \mathcal{A}$. В завершении первой и/или второй фазы сервер либо обнаруживает вмешательство нарушителя, либо фиксирует, что только участникам из $\mathcal{P}$ известен секретный ключ.*

Доказательство. Из утверждений 1, 3 и 4 вытекает, что нарушитель может быть выявлен в течении первой или второй фазы. Неизвестность секретного ключа вытекает из стойкости группового протокола Диффи—Хэллмана в рамках предположения DDH (наборы (11) и (12) неотличимы за полиномиальное время). □

Из утверждения 5 вытекает свойство G7: сервер получает подтверждение того, что никакой другой участник, кроме заранее определенных участников конференции, не может получить доступа ни к одному секретному ключу.

Передача ассоциации безопасности SA_{pr} и SA_s в шифрованном виде (см. шаги 1 и 2 первой фазы) обеспечивает выполнение свойства G11 — стороны могут безопасно договориться о параметрах защищенной связи. Шифрование идентификаторов (ID) позволяет обеспечить свойство G13 — анонимность при прослушивании, а так как участники не подписывают сообщения электронной цифровой подписью, и идентификаторы, в общем случае, могут быть псевдонимами, меняющимися от сеанса к сеансу, то обеспечивается частичная

анонимность при работе с другими участниками (свойство G14). Частичная анонимность обеспечивается в том смысле, что серверу известны публичные ключи участников; с другой стороны, всем участникам известен идентификатор сервера, так как на втором шаге первой фазы участниками используется публичный ключ сервера. В протоколе предусмотрена ограниченная защищенность от атак типа отказ в обслуживании (свойство G15): вычисления общего секретного ключа по групповому протоколу Диффи—Хэллмана происходят после аутентификации клиентов (шаги 1 и 2), а также за счет *cookies*, передаваемых в заголовках сообщений. Частичная защита связана с тем, что ряд атак может проводиться после этапа аутентификации субъектов, например, подмена легитимного участника нарушителем.

Заметим, что из утверждений 1, 3, 4 и 5 вытекает необходимость в доверии серверу R, так как выполнение свойств G1 и G7 доказано при допущении, что сервер не входит в число нарушителей. Недоверенный сервер может выдать себя за любого участника конференции и посылать сообщения от его имени. В этом случае свойства G1 и G7 не выполняются: подмена сервером одного участника (или нескольких) останется незамеченной другими участниками конференции.

3.3. Особенности реализации

В работе для реализации пользовательской части использовался фреймворк Vue, который позволяет применять реактивные данные на языке JavaScript для удобного взаимодействия с пользователем. Серверная часть реализована с помощью сервера Node.js, фреймворка Express и пакета Cors, который позволяет абстрагироваться от низкоуровневой архитектуры программы и описывать непосредственно логику хранения и пересылок данных. Вычисления проводились на MacBook Pro (13-inch, 2018, Four Thunderbolt 3 Ports), процессор 2,3 GHz Intel Core i5, 8 ГБ 2133 MHz LPDDR3, видеокарта Intel Iris Plus Graphics 655 1536 МБ. После выполнения протокола у каждого из клиентов на странице браузера отображается итоговый ключ K, вычисляемый в соответствии с (9) и используемый для дальнейшего шифрования сообщений. При реализации были проведены замеры времени между первым отправляемым сообщением и выработкой ключа K на стороне участника с номером n, а также на стороне сервера между первым сообщением получаемым от участника с номером 1 и последним сообщением, отправляемым участнику с номером n. Для реализации группового протокола Диффи—Хэллмана выбрана мультипликативная группа конечного поля $\mathbb{F}_{2^{256}}$, а для работы с большими числами использована библиотека BigInteger.

Несмотря на то, что разработанный протокол с сервером ($\mathcal{O}(8n)$ пересылок) обладает меньшей коммуникативной сложностью, чем протокол без сервера ($\mathcal{O}(n^3)$ пересылок), использование тяжелой арифметики больших чисел в протоколе Диффи—Хэллмана приводит к тому, что с ростом числа участников конференции время генерации ключа растет нелинейно. Как видно из табл. 2, даже при небольшой длине ключа Диффи—Хэллмана для 14 клиентов ключ генерируется, без учета времени на пересылку, порядка 10 с.

Заключение

В работе на основе двухточечного протокола IKE построен протокол генерации ключа для группы участников. Представляется, что в разработанном протоколе может быть использован не только протокол Диффи—Хэллмана, но и другие криптографические примитивы, такие как полиномы Чебышева, а также примитивы на основе помехоустойчивых

Таблица 2

Время выполнения от количества клиентов n, мс. Длина ключ Диффи—Хэллмана — 256 бит

n	**На сервере**	**На клиенте**	n	**На сервере**	**На клиенте**
2	888	915	9	724	2189
3	701	2023	10	1947	1982
4	333	1557	11	2260	2299
5	938	2228	12	2528	3830
6	881	2340	13	4977	5021
7	644	1754	14	9130	9180
8	886	1980			

кодов. Отметим, что использование арифметики больших чисел приводит к низкой скорости генерации ключа даже для относительно небольших групп (см. табл. 2). Кроме того, согласно [17], для задачи дискретного логарифмирования имеется эффективный алгоритм для квантовых компьютеров. Поэтому криптографические алгоритмы, основанные, в частности, на предположениях DL, DH или DDH, в постквантовую эпоху не будут обеспечивать высокой стойкости. В связи с этим особенно актуальна адаптация разработанного протокола для применения в нем криптографических примитивов на основе некоторых кодовых криптосистем типа Мак-Элиса, обзор которых приведен, например, в [18].

Для разработанного протокола показано, что выполнение ряда свойств безопасности, которые обеспечивает протокол IKE, требует наличие доверия к серверу, через который происходит общение участников при генерации ключа. В случае использования недоверенного сервера выполнение, как минимум, свойств G1 (аутентификация субъекта) и G7 (гарантия неизвестности ключа нелегитимным участникам) не гарантируется. В связи с этим еще одной актуальной задачей для дальнейшего исследования является доработка протокола для использования недоверенного сервера.

Литература

1. Bilal M., Kang S.-G. A Secure Key Agreement Protocol for Dynamic Group // Journal Cluster Computing. 2017. Vol. 20, no. 3. P. 2779–2792. DOI: 10.1007/s10586-017-0853-0.

2. Черемушкин А.В. Криптографические протоколы: основные свойства и уязвимости // Прикладная дискретная математика. 2009. Приложение 2. С. 115–150.

3. Dolev D., Yao A.C. On the Security of Public Key Protocol // IEEE Transactions on Information Theory. 1983. Vol. 29, no. 2. P. 198–208. DOI: 10.1109/tit.1983.1056650.

4. Liu H., Yang J., Wang Y., Chen Y.J., Koksal C.E. Group Secret Key Generation via Received Signal Strength: Protocols, Achievable Rates, and Implementation // IEEE Transactions on Mobile Computing. 2014. Vol. 13, no. 12. P. 2820–2835. DOI: 10.1109/TMC.2014.2310747.

5. Xu P., Cumanan K., Ding Z., Dai X., Leung K.K. Group Secret Key Generation in Wireless Networks: Algorithms and Rate Optimization // IEEE Transactions on Information Forensics and Security. 2016. Vol. 11, no. 8. P. 1831–1846. DOI: 10.1109/TIFS.2016.2553643.

6. Wyner A.D. The wire-tap channel // The Bell System Technical Journal. 1975. Vol. 54, no. 8. P. 1355–1387. DOI: 10.1002/j.1538-7305.1975.tb02040.x.

7. Bresson E., Chevassut O., Pointcheval D. Group Diffie-Hellman Key Exchange Secure against Dictionary Attacks // 8th International Conference on the Theory and Application of Cryptology and Information Security (Queenstown, New Zealand, December, 1–5, 2002). Lecture Notes in Computer Science. 2002. P. 497–514. DOI: 10.1007/3-540-36178-2_31.

8. Bresson E., Manulis M. Securing Group Key Exchange against Strong Corruptions and Key Registration Attacks // International Journal of Applied Cryptography. 2008. Vol. 1, no. 2. P. 91–107. DOI: 10.1504/IJACT.2008.021083.

9. Baiju B.V. Secret Key Sharing Scheme Based On Key Generation Centre For Authenticated Exchange Of Messages // International Journal of Engineering Science Invention. 2013. Vol. 2, no. 11. P. 15–21.

10. Kim Y., Perrig A., Tsudik G. Tree-based Group Key Agreement // ACM Transactions on Information and System Security. 2004. Vol. 7, no. 1. P. 60–96. DOI: 10.1145/984334.984337.

11. Lin T.-H., Tsung C.-K., Lee T.-F., Wang Z.-B. A Round-Efficient Authenticated Key Agreement Scheme Based on Extended Chaotic Maps for Group Cloud Meeting // Sensors. 2017. Vol. 17, no. 12. P. 1–14. DOI: 10.3390/s17122793.

12. Деундяк В.М., Таран А.А. Система распределения ключей на дизайнах. // Моделирование и анализ информационных систем. 2019. Т. 26, № 2. С. 229–243. DOI: 10.18255/1818-1015-2019-2-229-243.

13. Diffie W., Hellman M.E. New Directions in Cryptography // IEEE Transactions on Information Theory. 1976. Vol. 22, no. 6. P. 644–654. DOI: 10.1109/TIT.1976.1055638.

14. ElGamal T. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms // IEEE Transactions on Information Theory. 1985. Vol. 31, no. 4. P. 469–472. DOI: 10.1109/TIT.1985.1057074.

15. Boneh D. The Decision Diffie–Hellman Problem // Third International Symposiun, ANTS-III (Portland, Oregon, USA, June, 21–25, 1998). Lecture Notes in Computer Science. 1998. Vol. 1423. P. 48–63. DOI: 10.1007/BFb0054851.

16. Steiner M., Tsudik G., Waidner M. Diffie-Hellman Key Distribution Extended to Group Communication // 3rd ACM conference on Computer and communications security (New Delhi, India, March, 14–15, 1996). New York, ACM. 1996. P. 31–37. DOI: 10.1145/238168.238182.

17. Sendrier N. Code-Based Cryptography: State of the Art and Perspectives // IEEE Security & Privacy. 2017. Vol. 15, no. 4. P. 44–50. DOI: 10.1109/MSP.2017.3151345.

18. Deundyak V.M., Kosolapov Yu.V. On the Berger–Loidreau Cryptosystem on the Tensor Product of Codes // Journal of Computational and Engineering Mathematics. 2018. Vol. 5, no. 2. P. 16–33. DOI: 10.14529/jcem180202.

Волохов Александр Александрович, магистрант, Южный федеральный университет, Институт математики, механики и компьютерных наук им. И.И. Воровича (Ростов-на-Дону, Российская Федерация)

Косолапов Юрий Владимирович, к.т.н., кафедра алгебры и дискретной математики, Южный федеральный университет, Институт математики, механики и компьютерных наук им. И.И. Воровича (Ростов-на-Дону, Российская Федерация)

DOI: 10.14529/cmse200101

DEVELOPMENT AND IMPLEMENTATION OF THE CONFERENCE SECRET KEY GENERATION PROTOCOL BASED ON IKE

© 2020 A.A. Volokhov, Y.V. Kosolapov

Southern Federal University

(Bolshaya Sadovaya 105/42, Rostov-on-Don, 344006 Russia)

E-mail: sashavolohov@yandex.ru, itaim@mail.ru

Received: 16.07.2019

The protocol for generating a shared secret key often acts as the basis for informational interaction of participants in an untrusted environment. With the help of such a key, a secure channel or a secure communication network can be built in further interactions. Currently, the task of developing protocols for generating a shared key for a group of participants is relevant. One way to build such protocols is to generalize the protocol for two participants to the case of several participants. In the paper a protocol for generating a shared secret key for a group of participants (for a conference) is developed. The developed protocol is based on the Internet Key Exchange (IKE) protocol from the IPSec family of protocols for two participants, which ensures the implementation of security properties, such as authentication of the subject and message, generation of new keys, protection against reading back, protection against repetition, and a number of others. The strength of the developed key generation protocol is based on the complexity of the discrete logarithm problem in a cyclic group. The work studies the security properties provided by the constructed protocol, in particular, it studies the resistance to coalition attacks that are relevant for group protocols. Some features of the practical application of the constructed protocol are also noted.

Keywords: private key generation, IKE, conference.

FOR CITATION

Volokhov A.A., Kosolapov Y.V. Development and Implementation of the Conference Secret Key Generation Protocol Based on IKE. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering.* 2020. Vol. 9, no. 1. P. 5–19. (in Russian) DOI: 10.14529/cmse200101.

References

1. Bilal M., Kang S.-G. A Secure Key Agreement Protocol for Dynamic Group. Journal Cluster Computing. 2017. Vol. 20, no. 3. P. 2779–2792. DOI: 10.1007/s10586-017-0853-0.
2. Cheremushkin A.V. Cryptographic Protocols: Basic Properties and Vulnerabilities. Applied discrete mathematics. Appendix. 2009. no. 2. P. 115–150. (in Russian)
3. Dolev D., Yao A.C. On the security of public key protocol. IEEE Transactions on Information Theory. 1983. Vol. 29, no. 2. P. 198–208. DOI: 10.1109/tit.1983.1056650.
4. Liu H., Yang J., Wang Y., Chen Y. J., Koksal C.E. Group Secret Key Generation via Received Signal Strength: Protocols, Achievable Rates, and Implementation. IEEE Transactions on Mobile Computing. 2014. Vol. 13, no. 12. P. 2820–2835. DOI: 10.1109/TMC.2014.2310747.
5. Xu P., Cumanan K., Ding Z., Dai X., Leung K.K. Group Secret Key Generation in Wireless

Networks: Algorithms and Rate Optimization. IEEE Transactions on Information Forensics and Security. 2016. Vol. 11, no. 8. P. 1831–1846. DOI: 10.1109/TIFS.2016.2553643.

6. Wyner A.D. The Wire-tap Channel. The Bell System Technical Journal. 1975. Vol. 54, no. 8. P. 1355–1387. DOI: 10.1002/j.1538-7305.1975.tb02040.x.
7. Bresson E., Chevassut O., Pointcheval D. Group Diffie-Hellman Key Exchange Secure against Dictionary Attacks. 8th International Conference on the Theory and Application of Cryptology and Information Security (Queenstown, New Zealand, December, 1–5, 2002). Lecture Notes in Computer Science. 2002. P. 497–514. DOI: 10.1007/3-540-36178-2_31.
8. Bresson E., Manulis M. Securing Group Key Exchange against Strong Corruptions and Key Registration Attacks. International Journal of Applied Cryptography. 2008. Vol. 1, no. 2. P. 91–107. DOI: 10.1504/IJACT.2008.021083.
9. Baiju B.V. Secret Key Sharing Scheme Based On Key Generation Centre For Authenticated Exchange Of Messages. International Journal of Engineering Science Invention. 2013. Vol. 2, no. 11. P. 15–21.
10. Kim Y., Perrig A., Tsudik G. Tree-based Group Key Agreement. ACM Transactions on Information and System Security. 2004. Vol. 7, no. 1. P. 60–96. DOI: 10.1145/984334.984337.
11. Lin T.-H. , Tsung C.-K., Lee T.-F., Wang Z.-B. A Round-Efficient Authenticated Key Agreement Scheme Based on Extended Chaotic Maps for Group Cloud Meeting. Sensors. 2017. Vol. 17, no. 12. P. 1–14. DOI: 10.3390/s17122793.
12. Deundyak V.M., Taran A.A. Key Distribution System Based on Hadamard Designs. Modeling and Analysis of Information Systems. 2019. Vol. 26, no. 2. P. 229–243. (in Russian) DOI: 10.18255/1818-1015-2019-2-229-243.
13. Diffie W., Hellman M.E. New Directions in Cryptography. IEEE Transactions on Information Theory. 1976. Vol. 22, no. 6. P. 644–654. DOI: 10.1109/TIT.1976.1055638.
14. ElGamal T. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. IEEE Transactions on Information Theory. 1985. Vol. 31, no. 4. P. 469–472. DOI: 10.1109/TIT.1985.1057074.
15. Boneh D. The Decision Diffie–Hellman Problem. Third International Symposiun, ANTS-III (Portland, Oregon, USA, June, 21–25, 1998). Lecture Notes in Computer Science. 1998. Vol. 1423. P. 48–63. DOI: 10.1007/BFb0054851.
16. Steiner M., Tsudik G., Waidner M. Diffie-Hellman key distribution extended to group communication. 3rd ACM conference on Computer and communications security (New Delhi, India, March, 14–15, 1996). New York, ACM. 1996. P. 31–37. DOI: 10.1145/238168.238182.
17. Sendrier N. Code-Based Cryptography: State of the Art and Perspectives. IEEE Security & Privacy. 2017. Vol. 15, no. 4. P. 44–50. DOI: 10.1109/MSP.2017.3151345.
18. Deundyak V.M., Kosolapov Yu.V. On the Berger–Loidreau Cryptosystem on the Tensor Product of Codes. Journal of Computational and Engineering Mathematics. 2018. Vol. 5, no. 2. P. 16–33. DOI: 10.14529/jcem180202.