A. Yu. Nesterenko, On an approach to the construction of secure connections, *Mat. Vopr. Kriptogr.*, 2013, Volume 4, Issue 2, 101–111

DOI: 10.4213/mvk86

МАТЕМАТИЧЕСКИЕ ВОПРОСЫ КРИПТОГРАФИИ
2013 Т. 4 № 2 С. 101–111

УДК: 519.719.2

Об одном подходе к построению защищенных соединений

А. Ю. Нестеренко

Московский институт электроники и математики НИУ ВШЭ, Москва

Получено 18.IX.2012

В работе рассматривается один из возможных подходов к построению защищенных соединений, позволяющих передавать конфиденциальную информацию по открытым каналам связи. Приводятся положения, необходимые для обоснования криптографических качеств разрабатываемых средств защиты. В качестве примеров использования предлагаемого подхода приводятся протоколы удаленного управления и открытого распределения ключей.

Ключевые слова: защита информации, открытые каналы связи, защищенные соединения, открытое распределение ключей

On an approach to the construction of secure connections

A. Yu. Nesterenko

Moscow Institute of Electronics and Mathematics NRU HSE, Moscow

Abstract. One of possible approaches to the construction of secure connections permitting the transmission of confidential information through public networks is considered. We discuss conditions which are necessary for the approval of cryptographic quality of the protection mechanism. Protocols of remote control and public key distribution are used as examples.

Key words: information security, public networks, secured connections, public key distribution

Citation: *Mathematical Aspects of Cryptography*, 2013, vol. 4, no. 2, pp. 101–111 (Russian).

В настоящее время наиболее распространенным способом обеспечения обмена конфиденциальной информацией между удаленными пользователями является организация VPN-сети. Такая сеть позволяет создавать логические защищенные соединения и передавать в их рамках информацию в зашифрованном виде.

Известно несколько готовых криптографических решений, позволяющих на практике реализовывать VPN-сети. Среди них можно отметить IPSec, OpenVPN, PPTP, L2TP и другие, см. обзор в книге [5]. Каждое из указанных решений при своей разработке было ориентировано на решение конкретной криптографической задачи, поэтому его адаптация под другие цели в большинстве случаев приводит к ухудшению эксплуатационных характеристик, а иногда и к нарушению конфиденциальности передаваемой информации. В связи с этим представляется важным рассмотреть вопрос о нахождении универсальных методов построения защищенных соединений, реализуемых в рамках VPN-сетей.

Мы рассматриваем защищенное соединение в качестве промежуточного уровня обработки передаваемой информации: доставка сообщений обеспечивается протоколами транспортного уровня, как правило, протоколами семейства TCP/IP. Защищенное соединение гарантирует конфиденциальность и целостность передаваемой информации, а процессы программного уровня, собственно, и формируют информацию, подлежащую засекречиванию. Если следовать традиционной модели OSI, см. ГОСТ Р ИСО/МЭК 7498-1-99, то мы рассматриваем прикладной уровень представления протокола.

Данное представление удаляет из нашего поля зрения протоколы, подобные IPSec и OpenVPN, поскольку они внедряют криптографические решения на уровень транспортного протокола. Вместе с тем, выбранное нами представление позволяет абстрагироваться от технических вопросов передачи сообщений и строить математические модели, позволяющие обеспечить высокий уровень криптографической защиты передаваемой информации.

Далее в работе предлагается общий подход к построению защищенных соединений и приводятся примеры его применения на основе протоколов удаленного управления и открытого распределения ключей.

1. Схема защищенного соединения

Защищенное соединение представляет собой последовательное выполнение нескольких процедур, позволяющих абонентам договориться об используемых ключах и обменяться зашифрованной информацией. Процесс

обмена информацией до момента смены ключей шифрования мы будем называть одной сессией защищенного соединения.

Мы предлагаем рассматривать защищенное соединение в виде последовательности следующих шагов.

1. Определяются ключевая система и криптографические примитивы: алгоритмы шифрования информации, алгоритмы контроля целостности и аутентичности передаваемых данных.
2. Выполняется протокол согласования. В ходе выполнения протокола согласовывается сессионный ключ CK, вырабатывается уникальный идентификатор сессии SID, а также инициируется один или несколько начальных элементов последовательности случайных значений $\{\xi_n\}_{n=0}^{\infty}$. Данная последовательность используется в протоколе обмена информацией.
3. Выполняется протокол обмена информацией, зашифрованной при помощи согласованного ключа CK. Протокол может выполняться последовательно несколько раз, что позволяет абонентам реализовывать схемы обмена информацией типа «запрос–ответ».

Обратим внимание на то, что на втором и третьем шагах используется одна и та же ключевая система.

Основным требованием при организации защищенного соединения является обеспечение конфиденциальности передаваемой информации. При этом различные особенности функционирования накладывают дополнительные требования, специфичные для решаемых в рамках сети задач. Например,

- использовать защищенное соединение могут только разрешенные пользователи (*аутентификация* пользователей);
- при обмене информацией необходимо взаимное доверие участников обмена (*взаимная аутентификация*);
- необходимо доказательное подтверждение того, что переданная информация не была искажена в процессе передачи (*целостность* информации);
- необходимо обеспечить уверенность в том, что обмен сообщениями производится в реальном времени, а не является последовательностью отправленных ранее сообщений (*уникальность* сессии);
- доказательное отсутствие возможности *навязывания* ключевой информации;

- некоторым участникам обмена может быть не доступен процесс генерации ключевой информации (необходимо обеспечить *передачу* ключевой информации). Данное требование, очевидно, не может накладываться одновременно с предыдущим требованием.

Для каждой конкретной VPN-сети приведенный набор требований может уточняться в зависимости от эксплуатационных особенностей.

Функциональные требования означают, что должны выполняться дополнительные криптографические условия, необходимые для обеспечения конфиденциальности передаваемой информации. Среди таких требований целесообразно отметить следующие.

- *Сложность компрометации сессионных ключей.* Сложность определения ключа, используемого для засекречивания передаваемой информации, должна быть высокой и сравнимой со сложностью компрометации алгоритма шифрования, для которого вырабатывается ключ.
- *Сложность определения долговременных ключей.* Сложность определения долговременных ключей должна быть высокой и обеспечивать невозможность их определения в течение достаточно длительного периода времени. Если долговременные ключи используются для аутентификации участников протокола, этот срок должен быть не менее срока эксплуатации средства защиты. Если долговременные ключи используются для генерации сессионных ключей, срок сохранения долговременных ключей в секрете должен быть не менее времени хранения защищаемой информации. Конкретный перечень долговременных ключей определяется эксплуатационными особенностями VPN-сети, использующей защищенное соединение.
- *Защита от чтения вперед/назад.* Каждая сессия защищенного соединения должна вырабатывать уникальный ключ. Ситуация, при которой один или несколько общих ключей, выработанных в разных сеансах выполнения протокола, станут известны нарушителю, не должна приводить к компрометации ключей, вырабатываемых в других сеансах.
- *Защита при компрометации долговременных ключей.* Компрометация долговременных ключей не должна приводить к компрометации сеансовых ключей ключей, выработанных ранее.
- *Подтверждение ключа*. После выработки сессионного ключа каждый из абонентов должен быть уверен в том, что второй абонент обладает тем же общим ключом.

- *Защита от подмены владельца ключа*. Предположим, что долговременный ключ одного участника протокола, скажем $\mathfrak{A}$, скомпрометирован и известен нарушителю. В этом случае нарушитель может выдавать себя за абонента $\mathfrak{A}$ перед другими пользователями. Вместе с тем подобная ситуация не должна приводить к тому, чтобы нарушитель мог выдавать себя за другого пользователя перед абонентом $\mathfrak{A}$.
- *Защита от выработки ключа с третьим участником протокола*. Если абонент, скажем, $\mathfrak{A}$ хочет выработать сессионный ключ с абонентом $\mathfrak{B}$, то не должна возникать ситуация, при которой абонент $\mathfrak{A}$ вырабатывает сессионный ключ с некоторым другим, отличным от $\mathfrak{B}$, абонентом или нарушителем.

Приведенный перечень требований также должен уточняться перед разработкой и проведением анализа конкретного средства защиты. Фиксация перечня требований приводит к различным вариантам протокола согласования сессионного ключа. Вместе с тем протокол обмена информацией может быть унифицирован. Это позволяет применять не совсем обычный для криптографических протоколов модульный подход при разработке и эксплуатации VPN-сетей, используя различные протоколы согласования ключа в зависимости от решаемых задач.

Далее мы приведем примеры реализации данного подхода к построению конкретных криптографических решений. Мы будем обозначать символом Enc алгоритм блочного шифрования, $Sign$ — процедуру выработки цифровой подписи, Kdf — функцию выработки ключа, $Hash$ — функцию хеширования, Mac — алгоритм выработки кода аутентификации сообщений. В последнем разделе будет изложен унифицированный протокол обмена информацией.

2. Протокол согласования на основе гибридного шифрования

Настоящий протокол был предложен автором в докладе [7] для использования в системах управления удаленными объектами. Задача протокола заключается в доставке абоненту $\mathfrak{B}$ сессионного ключа CK, выработанного абонентом $\mathfrak{A}$.

Для данного протокола важным функциональным требованием является возможность передачи сформированного ранее сессионного ключа от одного абонента к другому. При этом требования невозможности навязывания ключа, а также его взаимного подтверждения оказываются излишними.

Ключевая система защищенного соединения заключается в следующем. Абонент $\mathfrak{A}$ обладает парой ключей (x_A, Y_A) цифровой подписи и использует в качестве функции вычисления кода аутентификации процедуру выработки цифровой подписи. Пара ключей и алгоритмы выработки и проверки цифровой подписи выбираются в соответствии с отечественным стандартом ГОСТ Р 34.10-2012.

Абонент $\mathfrak{B}$ обладает открытыми параметрами: эллиптической кривой $\mathcal{E}$ и точкой $P \in \mathcal{E}$ порядка q. Абонент $\mathfrak{B}$ обладает парой асимметричных ключей (x_B, Y_B), где $x_B \in \mathbb{Z}_q^*$, а Y_B — точка эллиптической кривой, для которой $Y_B = [x_B]P$.

Абонент $\mathfrak{A}$ вырабатывает сессионный ключ и зашифровывает его на открытом ключе абонента $\mathfrak{B}$ в соответствии с алгоритмом из [1], т. е.:

1. вырабатывает два случайных числа k, ξ_0, где $0 < k < q$, а $\xi_0 \in \mathbb{Z}_m$, где m — длина сессионного ключа CK в битах,
2. вырабатывает две точки на эллиптической кривой

$$U = [k]P, \quad W = [k]Y_B, \quad U, W \in \mathcal{E},$$

3. вырабатывает разовый ключ[1] $K = Kdf(W)$,
4. формирует заголовок сообщения $H_0 = U||Cert_A$, где $Cert_A$ — сертификат открытого ключа Y_A,

и направляет абоненту $\mathfrak{B}$ сообщение M_0, содержащее сессионный ключ CK и, при необходимости, дополнительную информацию T_0:

$$M_0 = H_0||Enc(K, CK||\xi_0||T_0)||Sign(x_A, H_0||CK||\xi_0||T_0).$$

Абонент $\mathfrak{B}$ получает сообщение и проверяет цифровую подпись под ним. Если цифровая подпись верна, то абонент $\mathfrak{B}$ вычисляет точку кривой $W = [x_B]U$ и разовый ключ $K = Kdf(W)$. После чего он расшифровывает сессионный ключ CK и дополнительную информацию T_0.

Поскольку абоненту $\mathfrak{A}$ необходимо быть уверенным в том, что абонент $\mathfrak{B}$ корректно расшифровал направленный ему ключ, абонент $\mathfrak{B}$ направляет ответ, для чего

1. вычисляет случайное значение $\xi_1 \in \mathbb{Z}_m$ и формирует идентификатор сессии $SID = Kdf(\xi_0||\xi_1)$;

[1] Функция $Kdf()$ используется для выработки разового ключа длины m из случайной последовательности произвольной длины. В качестве данной функции может выступать, например, функция хэширования.

2. формирует заголовок ответного сообщения $H_1 = SID||Hash(\xi_0)$

и направляет абоненту $\mathfrak{A}$ сообщение

$$M_1 = H_1||Enc(CK, \xi_1||T_1)||Mac(CK, H_1||\xi_1||T_1),$$

содержащее, при необходимости, дополнительную информацию T_1. В качестве функции вычисления кода аутентификации Mac абонент $\mathfrak{B}$ использует алгоритм HMAC с сеансовым ключом CK, см. [2] или [8].

Отметим, что данный протокол может быть легко трансформирован для передачи сессионного ключа от абонента $\mathfrak{B}$ к абоненту $\mathfrak{A}$. Для этого достаточно передавать сессионный ключ в качестве параметра T_1. Добавим, что вопросы компрометации изложенного протокола рассматривались в работе [3].

Изложенный протокол имеет некоторое сходство с регламентированным в стандарте ISO/IEC 11770-3 [6, п. 12.2] механизмом передачи ключей. Вместе с тем он обладает следующими отличительными особенностями.

- Рассматриваемый протокол включает в себя только два раунда обмена сообщениями. Двухраундовая процедура обмена сообщениями не позволяет абоненту $\mathfrak{A}$ проверить, выработан ли сессионный ключ CK в реальном времени, что, впрочем, не предполагалось при разработке протокола.
- Все коды аутентификации сообщений вычисляются перед зашифрованием информации. Обратим внимание на то, что мы используем понятие «код аутентификации» как для электронной цифровой подписи, так и для MAC — хэш-кода сообщения, вычисленного при помощи ключевой функции хэширования, поскольку оба преобразования обеспечивают аутентификацию отправителя данных.
- Коды аутентификации вычисляются для всей передаваемой в сообщении информации, включая заголовок сообщения.
- Используется последовательность случайных чисел $\xi_0, \xi_1, \ldots$, в отличие от предлагаемых в ISO/IEC временных меток и счетчиков. Кроме того, все элементы случайной последовательности передаются в зашифрованном виде.

3. Протокол согласования на основе открытого распределения ключей

В отличие от предыдущего протокола, в протоколе открытого распределения ключей сессионный ключ CK вырабатывается абонентами в процессе выполнения протокола. К данному протоколу предъявляются все перечисленные ранее требования, за исключением возможности передачи предварительно сформированного ключа.

Оба участника протокола обладают секретными и открытыми ключами электронной подписи. Секретные ключи x_A, x_B являются долговременными и известны только их владельцам. Открытые ключи Y_A, Y_B считаются общеизвестными и должны быть подтверждены сертификатами $Cert_A$ и $Cert_B$, выданными абонентам доверенным удостоверяющим центром. Как и ранее, пары ключей и алгоритмы выработки и проверки цифровой подписи регламентируются отечественным стандартом ГОСТ Р 34.10-2012.

Для реализации протокола необходимо определить общеизвестные открытые параметры.

- Оба абонента обладают собственными идентификаторами, соответственно, I_A и I_B.
- Абонентам должна быть известна эллиптическая кривая $\mathcal{E}$ и точка $P \in \in \mathcal{E}$ порядка q.

Помимо упомянутой ранее функции $Kdf()$, при реализации протокола используется функция выработки ключевой посылки $Kex()$. Данная функция позволяет выработать случайное число $k \in \mathbb{Z}_q^*$ и определить точку $R = [k]P \in \mathcal{E}$. Результатом действия функции $Kex()$ является пара (k, R).

Протокол состоит из последовательного обмена шестью сообщениями между абонентами $\mathfrak{A}$ и $\mathfrak{B}$. Каждое сообщение имеет вид $M = H||Body||S$, где H — заголовок сообщения, содержащий его номер и служебную информацию, $Body$ — тело сообщения, а S — электронная подпись отправителя сообщения. Схема выполнения протокола выглядит следующим образом.

Абонент $\mathfrak{A}$ Абонент $\mathfrak{B}$

1. Вырабатывает случайное число N_A, $T_1 = AlgList||Cert_A$,
$S_1 = Sign(x_A, H_1||I_A||N_A||T_1||I_B)$,
$M_1 = (H_1||N_A||T_1||S_1)$.

2. Вырабатывает случайное число N_B,
$T_2 = AlgResult||Cert_B$,
$S_2 = Sign(x_B, H_2||I_A||N_A||T_1||I_B||N_B||T_2)$,
$M_2 = (H_2||N_B||T_2||S_2)$.

3. Вычисляет значения $(k_A, R_A) = Kex()$,
формирует $SID = Kdf(N_A||N_B||I_A||I_B)$,
$S_3 = Sign(x_A, H_3||I_A||N_A||I_B||N_B||R_A)$,
$M_3 = (H_3||R_A||S_3)$.

4. Вычисляет значения $(k_B, R_B) = Kex()$ и $Q = [k_B]R_A$,
формирует $CK = Kdf(Q||N_A||N_B)$ и $SID = Kdf(N_A||N_B||I_A||I_B)$,
вырабатывает случайное сообщение T_3 и $ET_1 = Enc(CK, T_3)$,
$S_4 = Sign(x_B, H_4||I_A||N_A||I_B||N_B||R_A||R_B||T_3)$,
$M_4 = (H_4||R_B||ET_1||S_4)$.

5. Вычисляет значения $Q = [k_A]R_B$,
$CK = Kdf(Q||N_A||N_B)$ и $T_3 = Dec(CK, ET_1)$,
вырабатывает случайное сообщение T_4 и $ET_2 = Enc(CK, T_4||Request)$,
$S_5 = Sign(x_A, H_5||I_A||N_A||I_B||N_B||R_A||R_B||T_3||T_4||Request)$,
$M_5 = (H_5||ET_2||S_5)$.

6. Вычисляет $T_4 = Dec(CK, ET_2)$,
вырабатывает случайное число ξ_1 и формирует $Reply = \xi_1||\ldots$,
$S_6 = Sign(H_6||I_A||N_A||I_A||N_B||R_A||R_B||T_3||T_4||Request||Reply)$,
$M_6 = (H_6||Enc(CK, Reply)||S_6)$.

Сделаем некоторые замечания о приведенной схеме. Цель первого раунда протокола — идентификация абонента $\mathfrak{A}$. При инициализации протокола абонент $\mathfrak{A}$ направляет перечень доступных криптографических алгоритмов $AlgList$, а также сертификат своего открытого ключа Y_A. Мы предполагаем, что сертификат содержит в себе идентификатор I_A абонента $\mathfrak{A}$. В противном случае необходимо включение данного идентификатора в тело сообщения M_1.

Далее абонент $\mathfrak{B}$ утверждает используемые далее алгоритмы параметром $AlgResult$, а также предоставляет абоненту $\mathfrak{A}$ свой сертификат открытого ключа. Цель второго раунда — идентификация абонента $\mathfrak{B}$.

Цель третьего раунда заключается в выработке абонентом $\mathfrak{A}$ ключевой посылки и идентификатора сессии. Посылка вырабатывается на согласованных ранее параметрах эллиптической кривой.

В четвертом раунде абонент $\mathfrak{B}$ последовательно вырабатывает собственную ключевую посылку, вычисляет точку эллиптической кривой Q,

являющуюся общей ключевой информацией, вычисляет сессионный ключ шифрования CK, а также идентификатор сессии.

Далее выполняется процедура подтверждения ключа. Абонент $\mathfrak{B}$ зашифровывает на сессионном ключе случайное сообщение T_3 и отправляет его абоненту $\mathfrak{A}$. Тот последовательно вырабатывает общую ключевую информацию Q, вычисляет сеансовый ключ CK и, для его подтверждения, зашифровывает случайное сообщение T_4. В последнем раунде абонент $\mathfrak{B}$ подтверждает общий ключ и направляет клиенту ответное сообщение $Reply$ о завершении протокола и переходе в режим обмена зашифрованной информацией. Вопросы компрометации изложенного протокола рассматривались в работе [4].

Следует отметить, что приведенный протокол позволяет реализовывать защищенный обмен сообщениями типа «запрос–ответ», при которых соединение между абонентами длится только на момент передачи одного запроса ($Request$) и ответа на запрос ($Reply$). Примером такого обмена сообщениями может служить протокол HTTP (HTTPS).

4. Протокол обмена информацией

Теперь опишем заключительный этап организации защищенного соединения. Протокол обмена информацией использует параметры, выработанные либо ранее, в протоколе согласования сессионного ключа, либо в ходе выполнения предыдущего выполнения протокола, а именно идентификатор данной сессии SID, а также случайное число $\xi_{2n-1} \in \mathbb{Z}_m$, полученное ранее абонентом $\mathfrak{A}$ от абонента $\mathfrak{B}$.

Протокол представляет собой процедуру обмена двумя сообщениями, в ходе которой абонент $\mathfrak{A}$ формирует запрос абоненту $\mathfrak{B}$ и дожидается от него ответа. Возможно n-кратное последовательное выполнение протокола для $n = 1, 2, \ldots$.

Мы будем считать, что абонент $\mathfrak{A}$ направляет данные $Data_{2n} \in \mathbb{V}_\infty$, а абонент $\mathfrak{B}$ — данные $Data_{2n+1} \in \mathbb{V}_\infty$. Формат сообщений, передаваемых в каждом раунде протокола, одинаков, и имеет вид

$$M(x, m) = SID||Enc(CK, \xi_m||Data_m)||Mac(x, SID||\xi_{m-1}||\xi_m||Data_m).$$

В начале протокола абонент $\mathfrak{A}$ вырабатывает случайное число $\xi_{2n} \in \mathbb{Z}_m$ и направляет сообщение $M_{2n} = M(x_A, 2n)$. Абонент $\mathfrak{B}$ расшифровывает полученное сообщение, проверяет его код целостности, вырабатывает случайное число $\xi_{2n+1} \in \mathbb{Z}_m$ и направляет абоненту $\mathfrak{A}$ ответное сообщение $M_{2n+1} = M(x_B, 2n + 1)$.

При этом абоненты используют алгоритмы вычисления кода целостности сообщения $Mac(\ldots)$, фиксированные на этапе выполнения протокола согласования ключа. Эти алгоритмы, как видно из приведенных примеров, могут быть различны для различных абонентов и зависят от выбранной ранее ключевой системы.

Суммируя вышеизложенное, отметим, что предварительная формализация эксплуатационных требований к защищенному соединению существенно влияет на конкретную реализацию протокола согласования сессионного ключа. Вместе с тем использование общего подхода к предъявлению криптографических требований позволяет унифицировать не только процесс разработки данных протоколов, но и процесс обоснования уровня их криптографической защиты.

Список литературы

1. *Аносов В.Д., Нестеренко А.Ю.* Схема асимметричного шифрования, основанная на отечественных криптографических примитивах // Труды IX Международной конференции «Интеллектуальные системы и компьютерные науки». — 2006. — Т. 1.
2. *Нестеренко А.Ю.* Об одной реализации ключевой функции хеширования // Труды XXXIII Дальневосточной математической школы-семинара им. академика Золотова, 29 августа–4 сентября 2008 г. — Владивосток: Изд-во «Дальнаука», 2008. — С. 134–136.
3. *Нестеренко А.Ю.* О криптографических протоколах удаленного управления // Проблемы информационной безопасности. Компьютерные системы. — 2012. — № 2. — С. 76–82.
4. *Нестеренко А.Ю.* Новый протокол выработки общего ключа // Системы высокой доступности. — 2012. — № 2. — С. 81–90.
5. *Столлингс В.* Основы защиты сетей. Приложения и стандарты. — М.: «Вильямс», 2002. — 432 с.
6. ISO/IEC 11770-3. Information technology — Security techniques — Key management — Part 3: Mechanisms using asymmetric techniques. — Working Draft. — 2012.
7. *Nesterenko A. Yu.* Key transport protocol based on hybrid encryption scheme // The 7th Int. Comp. Sci. Symp. in Russia. Workshop «Current Trends in Cryptology». — 2012. — P. 20–21.
8. RFC 4357. Additional cryptographic algorithms for use with GOST 28147-89, GOST R 34.10-94, GOST R 34.10-2001 and GOST R 34.11-94 Algorithms.