Математические структуры и моделирование
2011, вып. 22, с. 101–113

УДК 004.056:004.057.4

СКРЫТЫЕ КАНАЛЫ ПЕРЕДАЧИ ИНФОРМАЦИИ В АЛГОРИТМЕ ЭЛЕКТРОННОЙ ЦИФРОВОЙ ПОДПИСИ ГОСТ Р 34.10-2001

М. И. Атмашкин, С. В. Белим

Приведены различные способы организации скрытых каналов в алгоритме цифровой подписи ГОСТ Р 34.10-2001. Рассмотрены примеры использования скрытых каналов для обмена сообщениями, для тайной передачи ключа подписи, а также для создания криптографических протоколов. Представлен способ ликвидации скрытых каналов.

Введение

Многие схемы цифровой подписи имеют «особенность», позволяющую подписывающему спрятать в подписи некоторую информацию, которая может быть затем извлечена при знании дополнительного секрета. Причём получаемые таким образом подписи неотличимы от «обычных». Это свойство было обнаружено Симмонсом и названо [7] им *скрытым каналом* (англ. *subliminal channel*).

Данным свойством обладают схемы, в которых получаемая подпись зависит не только от подписываемого сообщения, но и от некоторого «случайного» числа. Перебирая различных кандидатов на место этого числа, подписывающий может управлять видом подписи и закодировать в ней некоторую информацию.

Такими, например, являются подписи: Ong-Schnorr-Shamir, ElGamal, DSA, ESIGN и другие [3]. Не является исключением [1] и действующий российский стандарт цифровой подписи ГОСТ Р 34.10-2001.

Симмонс приводит такую [5] классификацию скрытых каналов:

- *широкополосные* (англ. *broadband*) — количество бит в скрытом сообщении сравнимо с числом бит в случайном числе. Обычно скрываемое сообщение и является этим числом либо легко из него получается;

- *узкополосные* (англ. *narrowband*) — количество бит в скрытом сообщении значительно меньше количества бит в случайном числе. Как правило, в этом случае размер сообщения зависит не от размеров случайного числа, а от вычислительной мощности подписывающего или проверяющего.

Омский государственный университет им. Ф. М. Достоевского.
E-mail: AtmashkinMI@gmail.com, sbelim@mail.ru

Не стоит, однако, считать, что широкополосные каналы лучше. В большинстве случаев они обладают недостатком, в результате которого проверяющему становится известен закрытый ключ подписи. Узкополосные каналы этого недостатка лишены.

В данной работе предложены различные типы широкополосных и узкополосных каналов для алгоритма цифровой подписи ГОСТ Р 34.10-2001. Рассмотрены примеры организации криптографических протоколов на основе широкополосных скрытых каналов, а также представлена модификация алгоритма подписи, позволяющая ликвидировать скрытые каналы.

1. Алгоритм ЭЦП ГОСТ Р 34.10-2001

Российский стандарт [2] ГОСТ Р 34.10-2001 является одним из самых молодых алгоритмов цифровой подписи. Стандарт принят и введён в действие Постановлением Госстандарта России от 12 сентября 2001 года взамен ГОСТ Р 34.10-94 и основан на эллиптических кривых. Его стойкость основывается на сложности вычисления дискретного логарифма в группе точек эллиптической кривой, а также на стойкости функции хэширования ГОСТ Р 34.11-94.

1.1. Параметры схемы цифровой подписи

- простое число p — модуль эллиптической кривой такой, что $p > 2^{255}$;
- эллиптическая кривая E, задаваемая своим инвариантом $J(E)$ или коэффициентами $a, b \in F_p$, где F_p — конечное поле из p элементов;
- целое число m — порядок группы точек эллиптической кривой, m должно быть отлично от p;
- простое число q — порядок некоторой циклической подгруппы группы точек эллиптической кривой, т. е. $q \mid m$. Кроме того, $2^{254} < q < 2^{256}$;
- точка P эллиптической кривой E, являющаяся генератором подгруппы порядка q, т. е. $qP = O$ и $kP \neq O$ для всех $k = 1, 2, \ldots, q-1$, где точка O является нейтральным элементом группы;
- $h(M)$ — функция хэширования (ГОСТ Р 34.11-94), которая отображает сообщения M в двоичные вектора длины 256 бит.

1.2. Используемые ключи

- ключ подписи — целое число d, удовлетворяющее неравенству $0 < d < q$;
- ключ проверки — точка эллиптической кривой Q, удовлетворяющая равенству $dP = Q$.

1.3. Дополнительные требования

- должно быть выполнено условие $p^t \neq 1 \ (mod\ q)$, для всех целых чисел $t = 1, 2, \ldots, B$, где B удовлетворяет неравенству $B \geq 31$;
- инвариант кривой должен удовлетворять условию $J(E) \neq 0$ или 1728.

1.4. Формирование цифровой подписи

1. Вычисление хэш-функции от сообщения M: $z = h(M)$.
2. Вычисление $e = z \ (mod\ q)$, и если $e = 0$, положить $e = 1$.
3. Генерация случайного числа k такого, что $0 < k < q$.
4. Вычисление точки эллиптической кривой $C = kP$ и по ней нахождение $r = x_C \ (mod\ q)$, где x_C — это координата x точки C. Если $r = 0$, то возвращаемся к предыдущему шагу.
5. Нахождение:
$$s = rd + ke \ (mod\ q), \tag{1}$$
если $s = 0$, то возвращаемся к шагу (3).
6. Формирование цифровой подписи $\zeta = (r \ || \ s)$.

1.5. Проверка цифровой подписи

1. Вычисление по цифровой подписи ζ чисел r и s. Если хотя бы одно из неравенств $0 < r < q$ и $0 < s < q$ неверно, то подпись неправильная.
2. Вычисление хэш-функции от сообщения M: $z = h(M)$.
3. Вычисление $e = z \ (mod\ q)$, и если $e = 0$, положить $e = 1$.
4. Вычисление $\nu = e^{-1} \ (mod\ q)$.
5. Вычисление $z_1 = s\nu \ (mod\ q)$ и $z_2 = -r\nu \ (mod\ q)$.
6. Вычисление точки эллиптической кривой $C = z_1P + z_2Q$. И определение $R = x_C \ (mod\ q)$, где x_C — координата x точки C.
7. В случае равенства $R = r$ подпись правильная, иначе — неправильная.

2. Скрытые каналы

Поскольку в алгоритме формирования цифровой подписи присутствует случайное число k, то подписывающий может надлежащим выбором этого числа придать подписи ζ некоторый «особый» вид, закодировав в ней сообщение.

Рассмотрим скрытые каналы согласно классификации, данной Симмонсом.

2.1. Широкополосные каналы

В простейшем случае можно подставить само сообщение вместо числа k. Алгоритмы формирования и проверки подписи принципиально не изменятся, и тот, кто знает ключ проверки, может по-прежнему проверить правильность подписи. Но если проверяющему известен также ключ подписи d, то он может вычислить k по формуле, вытекающий из (1):

$$k = e^{-1}(s - rd) \ (mod\ q),$$

а если ключ подписи неизвестен, то для нахождения k нужно решить проблему дискретного логарифма в группе точек эллиптической кривой.

Несмотря на максимальную пропускную способность, данный канал обладает серьёзными недостатками:

1. Проверяющему необходимо знать ключ подписи. Это является очень серьёзным недостатком, потому что в этом случае для подписывающего существует угроза компрометации его ключа подписи, и он должен полностью довериться своему собеседнику. Кроме того, в ряде случаев заранее невозможно узнать ключ подписи. Стоит также отметить, что в общем случае число k будет некоторой обратимой функцией от скрываемого сообщения, и знание сообщения влечёт за собой знание числа k, из которого легко получается ключ подписи:

$$d = r^{-1}(s - ke) \ (mod\ q), \tag{2}$$

   т. е. обмен «большими» сообщениями влечёт знание ключа подписи.

2. Стойкость алгоритма ГОСТ Р 34.10-2001 сильно зависит от качества используемого ГПСП, потому что при повторе числа k можно легко вычислить ключ подписи d. Так как число r зависит только от k, то оно будет в обоих случаях одинаковым, и получается такая система:

$$s_1 = rd + ke_1 \ (mod\ q), s_2 = rd + ke_2 \ (mod\ q), \tag{3}$$

   из которой следует формула для k:

$$k = (s_2 - s_1)(e_2 - e_1)^{-1} \ (mod\ q),$$

   а зная k, можно уже найти d по формуле (2).

   В случае использования скрытого канала данная ситуация становится очень вероятной — примером может быть обмен регулярными дежурными сообщениями. Чтобы исправить этот недостаток, можно использовать приём «подсаливания» числа k — замены части его бит на случайные. При извлечении скрытого сообщения собеседник будет просто игнорировать эти биты. Данный приём плох тем, что жертвуется пропускная способность скрытого канала, причём число подменяемых бит должно быть довольно велико для защиты от атаки перебором. Если злоумышленник

знает, что два случайных числа в системе (3) отличаются на небольшое число n бит в известных позициях, то он может попытаться решить 2^n систем линейных уравнений. Для усложнения схемы можно постоянно менять позиции заменяемых бит по некоторому заранее известному закону.

Но есть и более безопасный метод, который не поглощает пропускной способности канала. Можно вместо числа k использовать: $k'_i = k + h_i \ (mod\ q)$, где число h_i является действительно случайным. Получатель для восстановления исходного сообщения вычислит: $k = k'_i - h_i \ (mod\ q)$. Понятно, что последовательность h_i должна быть известна обоим. Этого можно достичь, например, используя общие секретные параметры для криптографически стойкого ГПСП. В качестве такого генератора может выступать генератор на основе функций хэширования. Пусть $H(x)$ — это 256-битная хэш-функция, например, это может быть ГОСТ Р 34.11-94 или SHA-256. Тогда ПСП будет выглядеть так:

$$h_0 = H(d), h_i = H(h_{i-1} \parallel d), i = 1, 2, \ldots \tag{4}$$

Очевидно, последовательность легко порождается по известному ключу подписи, но без его знания вычислительно трудно получить как следующие, так и предыдущие члены последовательности. Кроме знания ключа подписи собеседникам не нужно иметь других секретов. Получающиеся в результате подписи будут неотличимы от «обычных».

3. Не всякое сообщение, которому соответствует некоторое число $0 < k < q$, можно встроить в подпись. Это связано с ограничениями (4) и (5) шагов алгоритма формирования подписи: $r \neq 0 \ (mod\ q)$ и $s \neq 0 \ (mod\ q)$. Подобные подписи отбрасываются на шаге (1) алгоритма проверки подписи. Причём заранее сказать, для каких k не получится создать подпись, тяжело из-за сложной зависимости r и s от k.

Используя некоторое обобщение системы (3), можно реализовать схему генерации чисел k, которая позволит при знании некоторого секрета узнать ключ подписи. Например, можно использовать такую последовательность k_i:

$$k_0, k_i = a_i k_{i-1} + b_i \ (mod\ q), i = 1, 2, \ldots,$$

где k_0 — зерно, полученное из некоторого внешнего источника энтропии, а последовательности a_i и b_i построены аналогично (4) на основе ключей d_a и d_b соответственно. Для тех, кто не знает этих ключей, последовательность k_i является стойкой криптографической ПСП. Но тот, кому известны ключи d_a и d_b, может перехватить две последовательные подписи и, зная номер i, вычислить a_i и b_i, а затем решить систему:

$$s_{i-1} = r_{i-1} d + k_{i-1} e_{i-1} \ (mod\ q), s_i = r_i d + (a_i k_{i-1} + b_i) e_i \ (mod\ q). \tag{5}$$

Она линейная, и в ней только два неизвестных: d и k_{i-1}, которые легко находятся. Для «простых» перехватчиков неизвестных в этой системе слишком много. Имея даже большое количество перехваченных подписей, вычислительно трудно решить систему из-за сложной связи между членами в ПСП a_i и b_i.

3. Узкополосные каналы

Основным преимуществом узкополосных каналов является отсутствие необходимости раскрывать ключ подписи. К недостаткам же можно отнести повышенные требования к вычислительным ресурсам — памяти и процессорному времени. Именно ограниченностью этих ресурсов и обусловлена низкая пропускная способность этих каналов. Узкополосные скрытые каналы можно условно разделить на вероятностные и детерминированные.

3.1. Вероятностные каналы

Ключевой особенностью этих каналов является то, что требуемые скрытые сообщения можно получить только с некоторой вероятностью. Чем больше бит будет в сообщении, тем меньше вероятность его получить. В общем случае алгоритм создания вероятностного канала выглядит так [4]:

1. Подписывающий и проверяющий договариваются о некоторой секретной ключевой функции $h_K()$, которая отображает подпись ζ или её часть (r или s) в некоторую последовательность бит M, а также о некотором способе кодирования сообщения такой последовательностью.

2. Подписывающий генерирует случайные числа k, формирует для каждого из них цифровую подпись по стандартному алгоритму и вычисляет значения функции $h_K()$ от полученных подписей до тех пор, пока не получит последовательность бит M, соответствующую встраиваемому сообщению, либо пока не кончится отведённое на подпись время — в этом случае встроить сообщение не удастся.

3. После проверки подписи получатель, зная секретный ключ K и алгоритм декодирования последовательности M, извлекает секретное сообщение.

В качестве примера можно привести канал, предложенный Симмонсом, для DSA [6]. Его адаптация для подписи ГОСТ Р 34.10-2001 будет выглядеть так. Секретным ключом K являются n различных больших простых чисел. Функция $h_K()$ ставит в соответствие части r подписи последовательность M, состоящую из n бит, где i-й бит равен «1», если r является квадратичным вычетом по модулю i-го простого числа, и «0» — в противном случае. Последовательность M без дополнительных преобразований и будет являться секретным сообщением. Вероятность того, что r будет квадратичным вычетом по модулю некоторого простого числа, составляет $\frac{1}{2}$, для n чисел соответственно — $\frac{1}{2^n}$. Чем больше подписей в единицу времени может генерировать подписывающий, тем большую длину скрытого канала он может себе позволить.

На шаге (1) алгоритма создания вероятностного канала действительно удобнее использовать часть r подписи, потому что она вычисляется первой. В качестве функции $h_K()$ можно, например, использовать следующие функции:

- хэш-функцию ГОСТ Р 34.11-94 со стартовым вектором хэширования, равным K, либо другую ключевую хэш-функцию;

- возведение числа r в секретную степень, равную K, по модулю числа q. Этот способ основан на проблеме дискретного логарифмирования в кольце вычетов по модулю простого числа.

- умножение точки эллиптической кривой C, получаемой на шаге (4) алгоритма формирования подписи, на секретное число K, в качестве значения функции будет выступать координата x полученной точки. Этот способ основан на проблеме дискретного логарифмирования в группе точек эллиптической кривой.

В качестве способа кодирования значения предложенных функций $h_K()$ в скрытое сообщение можно использовать остаток от деления на степень 2, который равен младшим битам числа. Если допустить, что все остатки будут равновероятны, то вероятность получения конкретного сообщения будет такая же, как в примере Симмонса.

На шаге (2) алгоритма создания вероятностного канала можно использовать следующую оптимизацию. Вместо генерации нескольких случайных чисел k можно сгенерировать одно, а затем прибавлять к полученной точке $C = kP$ точку P, пока не получим нужного значения функции $h_K()$. Этот способ не уменьшает безопасность подписи, но позволяет заменить многократное умножение точек на более простое сложение.

3.2. Детерминированные каналы

Данный тип скрытых каналов позволяет гарантированно создавать требуемые скрытые сообщения, но требует от лица, извлекающего сообщения, большой процессорной мощности и дополнительной памяти. Алгоритм выглядит так:

1. Подписывающий и проверяющий договариваются о размере скрываемых сообщений. Подписывающий генерирует ключ для ПСП, построенной по принципу (4), и сохраняет его в своей постоянной памяти. Затем он генерирует большое число n членов этой ПСП, где n — число сообщений, которые он сможет отправить по этому каналу. Для каждого из этих чисел подписывающий вычисляет точку C из шага (4) алгоритма формирования подписи. Все n полученных точек подписывающий отправляет проверяющему.

2. Подписывающий начинает формировать подписи по стандартному алгоритму, но случайные числа он берет из ПСП, восстановленной по сохранённому ключу. Вместо точки C он использует точку $C' = C + mP$, где $0 \leq m < 2^n$ является скрываемым сообщением, длиной n бит.

3. После проверки подписи проверяющий начинает прибавлять к очередной точке C из точек, переданных ему на шаге (1), точку P до тех пор, пока не получит точку C'. По числу операций сложения он восстановит скрытое сообщение m.

В данном алгоритме проверяющий так и не узнает ни одного случайного числа k из секретной ПСП. Следовательно, он не сможет вычислить ключ подписи по формуле (2). Проверяющему сложно установить зависимость между переданными ему точками, а также узнать по конкретной точке использованное случайное число. Недостатком данного алгоритма является необходимость передавать большой объем информации на шаге (1), а также неравномерность во времени, затрачиваемом на декодирование получателем различных сообщений. Эту неравномерность можно сгладить, если начинать проверку не с точки C, а с точки, сдвинутой от неё на некоторое случайное число точек P, и в дальнейшем осуществлять сложение с точками P циклически с учётом границ сообщения m.

4. Ликвидация скрытых каналов

Все представленные скрытые каналы основаны на том, что подписывающий может самостоятельно выбирать случайное число k. С другой стороны, нельзя доверить кому-то генерацию этого числа, так как это неминуемо раскроет ключ подписи. Очевидным и наиболее подходящим решением будем совместная генерация числа k таким образом, что подписывающий не сможет контролировать ни один бит числа k, а контроллер не сможет узнать ни один бит числа k. В конце протокола контроллер должен убедиться, что подписывающий использовал именно то число, которое они сгенерировали вместе. Адаптация протокола, предложенного Симмонсом для DSA [5], для подписи ГОСТ Р 34.10-2001 будет выглядеть так:

1. Подписывающий генерирует число k' и отправляет контроллеру точку $U = k'P$.

2. Контроллер выбирает случайное число k'' и отправляет подписывающему.

3. Подписывающий использует для создания подписи число $k = k'k'' \ (mod\ q)$ и отправляет полученную подпись контроллеру.

4. Контроллер проверяет подпись, а также то, что координата x точки $k''U$ сравнима с r по модулю q.

Протокол Симмонса имеет серьёзный недостаток — на шаге (2) контроллер может сам управлять частью r подписи и встроить в неё собственное скрытое сообщение. Для предотвращения этого можно сделать следующую модификацию протокола. На шаге (1) подписывающий вместо точки U должен отправить значение хэш-функции от этой точки. Контроллер умножает точку C, получаемую на шаге (6) алгоритма проверки цифровой подписи, на $k''^{-1} \ (mod\ q)$, считает от неё значение хэш-функции и сверяет с полученным от подписывающего. В таком варианте протокола ни подписывающий, ни контроллер не смогут заранее узнать, каким будет число r. Но применение контроллера для создания подписей возможно далеко не всегда, и от скрытых каналов полностью избавиться нельзя.

5. Криптографические протоколы

В то время как узкополосные каналы удобнее для организации утечки ключа подписи в непроверенных реализациях цифровой подписи, широкополосные каналы можно использовать в криптографических протоколах для легальной передачи сеансовых ключей, причём сам факт передачи ключей может быть скрыт. Скрытый канал может выполнять функции шифра, при этом сохранятся все «обычные» свойства цифровой подписи. Таким образом в подписи можно зашифровать сеансовый ключ, а проверяющий сможет по-прежнему проверить целостность сообщения и его авторство. В широкополосном канале можно передать сеансовый ключ размером 256 бит, что достаточно для таких симметричных шифров, как ГОСТ 28147-89 или AES-256.

Будем использовать стандартные [3] имена для участников протоколов: Алиса — сторона, инициирующая сеанс, Боб — сторона, с которой устанавливается сеанс, Трент — доверенная промежуточная сторона.

Подпись будем обозначать так: $S_d(k, M)$, где d — ключ подписи; k — случайное число, используемое для формирования подписи; M — подписываемое сообщение. Вместе с подписью в протоколах подразумевается передача самого сообщения M. Представленные протоколы предполагают синхронизацию часов всех участников. Погрешность синхронизации компенсируется запоминанием истории передач в течение некоторого времени. Передача меток времени вместе с временем жизни ключей защищает от повторной передачи сообщений.

5.1. Протокол № 1

Рассмотрим первый протокол, который использует простейший широкополосный канал с общим ключом подписи. Он является модифицированным вариантом базовой версии протокола *Kerberos* [3]. Вместо шифрования здесь используется цифровая подпись со скрытым каналом.

Пусть Трент имеет общий ключ подписи с Алисой — d_A и с Бобом — d_B. Данный протокол позволяет обменяться сеансовым ключом, который сгенерировал Трент, а также провести взаимную аутентификацию.

1. Алиса отправляет Тренту сообщение с идентификаторами: A, B.

2. Трент создаёт два сообщения, состоящих из метки времени T_T, времени жизни ключа L и идентификаторов. В сообщении для Алисы используется идентификатор Боба, а для Боба наоборот. Затем он подписывает эти сообщения ключами d_A и d_B. В качестве случайного числа в обеих подписях используется сгенерированный им сеансовый ключ K, зашифрованный соответственно ключами d_A и d_B. Полученные подписи Трент отправляет Алисе: $S_{d_A}(E_{d_A}(K), B \,||\, T_T \,||\, L), S_{d_B}(E_{d_B}(K), A \,||\, T_T \,||\, L)$.

3. Алиса проверяет первую подпись Трента, убеждается, что сообщение текущее, извлекает и расшифровывает сеансовый ключ. Затем она создаёт сообщение, состоящее из её идентификатора и метки времени, шифрует

его сеансовым ключом и отправляет Бобу. Алиса также посылает Бобу вторую подпись Трента: $E_K(A \,||\, T_T), S_{d_B}(E_{d_B}(K), A \,||\, T_T \,||\, L)$.

4. Боб проделывает с подписью Трента те же операции, что и Алиса, затем он расшифровывает сообщение Алисы и убеждается в том, что оно текущее. Создаёт сообщение, состоящее из метки времени плюс единица, шифрует его сеансовым ключом и отправляет Алисе: $E_K(T_T + 1)$.

5. Алиса расшифровывает сообщение Боба и проверяет метку времени.

6. Алиса и Боб шифруют свои сообщения, используя сеансовый ключ K.

Шифрование сеансового ключа необходимо для того, чтобы Алиса не смогла вычислить ключ d_B по формуле (2). Кроме того, это делает части r передаваемых подписей различными. Злоумышленник, перехватывающий сообщения, может даже не узнать, что в этом протоколе, кроме аутентификации, происходит обмен сеансовым ключом. Главным преимуществом по сравнению с *Kerberos* является аутентификация Трента, обеспечиваемая цифровыми подписями.

5.2. Протокол № 2

Рассмотрим теперь протокол, основанный на «недостатке» широкополосных каналов, связанном с повтором случайного числа и последующим вычислением ключа подписи. Цифровая подпись может использоваться здесь для разделения секрета, в роли которого будет выступать сеансовый ключ.

Пусть Тренту известны открытые ключи Алисы — A_{open} и Боба — B_{open}, используемые в некоторой схеме асимметричного шифрования. Алисе и Бобу известен ключ проверки подписи $S_{main}()$ Трента. Эта подпись будет использоваться для аутентификации Трента и может вычисляться по другому алгоритму. Скрытый канал в этой подписи, если он есть, использоваться не будет.

1. Алиса отправляет Тренту сообщение с идентификаторами: A, B.

2. Трент генерирует сеансовый ключ K, а также ключи подписи и проверки для алгоритма со скрытым каналом — T_{close} и T_{open}. Трент составляет сообщение из подписи $S_{main}()$ ключа T_{open}, а также подписи сообщения, состоящего из идентификатора Боба, открытого ключа Боба, метки времени и времени жизни сеансового ключа, сделанной на ключе T_{close}. В качестве случайного числа в подписи используется ключ K. Полученное сообщение Трент шифрует открытым ключом Алисы и отправляет ей: $E_{A_{open}}(S_{main}(T_{open}) \,||\, S_{T_{close}}(K, B \,||\, B_{open} \,||\, T_T \,||\, L))$.

3. Трент проделывает те же действия для Боба и отправляет ему аналогичное сообщение: $E_{B_{open}}(S_{main}(T_{open}) \,||\, S_{T_{close}}(K, A \,||\, A_{open} \,||\, T_T \,||\, L))$.

4. Алиса расшифровывает сообщение Трента, проверяет подписи, убеждается в том, что сообщение текущее. Затем она зашифровывает открытым ключом Боба подпись Трента со скрытым каналом и отправляет Бобу: $E_{B_{open}}(S_{T_{close}}(K, B \,||\, B_{open} \,||\, T_T \,||\, L))$.

5. Аналогично Боб отправляет Алисе: $E_{A_{open}}(S_{T_{close}}(K, A \mid\mid A_{open} \mid\mid T_T \mid\mid L))$.

6. Алиса и Боб расшифровывают и проверяют принятые подписи. Зная две подписи Трента, каждый из них может восстановить сеансовый ключ K.

7. Алиса и Боб шифруют свои сообщения, используя сеансовый ключ K.

Ключ подписи со скрытым каналом является одноразовым, чтобы участники не могли узнать сеансового ключа после шагов (2) и (3). По той же причине используется шифрование с открытым ключом вместо симметричного шифрования. Без подписи $S_{main}()$ Алиса или Боб могли бы выдавать себя в дальнейшем за Трента. Ключ для этой подписи может быть долговременным.

5.3. Протокол № 3

Не всегда приемлем тот факт, что сеансовый ключ генерирует третья сторона, в дальнейшем ей будет доступна вся переписка. Более привлекательной является совместная генерация ключа Алисой и Бобом. Алиса генерирует ключ K_A, Боб соответственно — K_B, сеансовый ключ получается в результате сложения по модулю 2 этих ключей: $K = K_A \oplus K_B$. Совместную генерацию ключа можно организовать, например, используя формулу (5). Обменявшись одной подписью, участники протокола не смогут в дальнейшем отказаться от выбранного ключа. После обмена второй подписью можно будет вычислить сеансовый ключ собеседника и убедиться в его подлинности.

Пусть Алиса знает ключи, используемые Бобом для генерации последовательностей a_i^B и b_i^B, а Боб знает ключи Алисы для a_i^A и b_i^A. Также они имеют общий ключ шифрования K_{AB}, используемый для взаимной аутентификации.

1. Алиса генерирует свой ключ K_A, вычисляет ключ подписи $d_A = E_{K_{AB}}(K_A)$ и соответствующий ему ключ проверки Q_A. Затем она формирует сообщение, состоящее из ключа проверки, текущего номера в ПСП i_A и подписи своего идентификатора, метки времени T_A и времени жизни L_A ключа K_A. В качестве случайного числа в этой подписи используется ключ K_A. Алиса отправляет Бобу: $Q_A, i_A, S_{d_A}(K_A, A \mid\mid T_A \mid\mid L_A)$.

2. Боб проверяет подпись Алисы и проделывает те же действия. Он отправляет Алисе: $Q_B, i_B, S_{d_B}(K_B, B \mid\mid T_B \mid\mid L_B)$.

3. Алиса проверяет подпись Боба и создаёт новую подпись сообщения, состоящего из ключа проверки Боба и его номера в ПСП. В качестве случайного числа она использует $K'_A = a^A_{i_A} K_A + b^A_{i_A}$ и отправляет полученную подпись Бобу: $S_{d_A}(K'_A, Q_B \mid\mid i_B)$.

4. Боб вычисляет $K'_B = a^B_{i_B} K_B + b^B_{i_B}$ и отправляет Алисе аналогичную подпись: $S_{d_B}(K'_B, Q_A \mid\mid i_A)$.

5. Алиса и Боб проверяют полученные подписи. Зная номера текущих членов ПСП друг друга, они могут решить систему (5) и вычислить по ней ключ

подписи и использованное случайное число. Для взаимной аутентификации они проверяют, что ключ подписи — это зашифрованное случайное число. Затем они вычисляют сеансовый ключ: $K = K_A \oplus K_B$.

6. Алиса и Боб шифруют свои сообщения, используя сеансовый ключ K.

Злоумышленник не сможет решить систему (5), он также не сможет подменить подписи или номера в ПСП, потому что ему неизвестен ключ K_{AB}. Ни Алиса, ни Боб не смогут вычислить ключ собеседника до получения второй подписи и не смогут отказаться от выбранного сеансового ключа, потому что в случае подмены ключа не произойдёт аутентификации.

5.4. Протокол № 4

Можно провести совместную генерацию сеансового ключа и взаимную аутентификацию по аналогии с алгоритмом ликвидации скрытого канала.

Пусть Алиса и Боб имеют общий ключ подписи K_{AB}.

1. Алиса генерирует свой ключ K_A, вычисляет хэш-код ключа и отправляет его Бобу вместе со своим идентификатором: $A, H(K_A)$.

2. Боб генерирует свой ключ K_B и использует его в качестве случайного числа для создания подписи сообщения, состоящего из своего идентификатора, метки времени T_B и времени жизни L_B ключа K_B. Ключом подписи является общий ключ K_{AB}. Полученную подпись Боб отправляет Алисе: $S_{K_{AB}}(K_B, B \,||\, T_B \,||\, L_B)$.

3. Алиса проверяет подпись Боба, извлекает из подписи его ключ K_B, вычисляет сеансовый ключ $K = K_A \oplus K_B$ и отправляет Бобу аналогичную подпись: $S_{K_{AB}}(K, A \,||\, T_A \,||\, L_A)$.

4. Боб проверяет подпись Алисы, извлекает из подписи сеансовый ключ K, вычисляет по нему ключ Алисы K_A и убеждается, проверяя его хэш-код, что Алиса не отказалась от первоначального ключа.

5. Алиса и Боб шифруют свои сообщения, используя сеансовый ключ K.

Цифровая подпись обеспечивает взаимную аутентификацию Алисы и Боба. Злоумышленник не сможет подменить подписи, потому что он не знает ключа K_{AB}. Алисе вычислительно трудно найти другой ключ, дающий тот же хэш-код, а Бобу вычислительно трудно узнать ключ Алисы до того, как он отправит ей свой. Для этого ему нужно обратить хэш-функцию.

6. Заключение

Представленные в статье скрытые каналы предоставляют множество возможностей для тайной и безопасной передачи информации в стандартных цифровых подписях ГОСТ Р 34.10-2001. В случае передачи сообщений широкополосные

каналы стоит применять, когда источник и приёмник сообщений — это одно и то же лицо либо подразделения одной организации. В этих случаях не произойдёт компрометации ключа подписи. Узкополосные каналы предпочтительнее, когда приёмником является лицо с более низкой степенью доверия либо группа лиц. Нахождение ключа подписи для получателей будет в этом случае вычислительно трудной задачей. Оба типа скрытого канала при правильном применении доступны лишь авторизованным получателям и необнаружимы для посторонних лиц, не обладающих дополнительным секретом.

На широкополосных каналах можно строить криптографические протоколы. Цифровая подпись с таким каналом будет выполнять и функции подписи, и функции шифра. Также с помощью таких цифровых подписей можно проводить совместную генерацию ключа одновременно с взаимной аутентификацией. Размер скрытого канала достаточен для большинства симметричных шифров.

Узкополосные каналы можно использовать для создания реализаций подписи, организующих утечку конфиденциальной информации. Следует особенно тщательно проверять используемые программные и аппаратные средства для создания цифровых подписей и генерации случайных чисел, полученные от третьих лиц, возможно, даже имеющих лицензию на их изготовление. Предпочтительнее исследование исходных кодов используемых программ и их самостоятельная компиляция. В случае, когда исходные коды недоступны либо необходимы дополнительные гарантии проверяющему, можно провести совместную генерацию подписи с контроллером по предложенному протоколу.

Литература

1. Белим С. В., Федосеев А. М. Исследование скрытых каналов передачи информации в алгоритме цифровой подписи ГОСТ Р 34.10-2001 // Известия Челябинского научного центра. 2007. Вып. 2. С. 55–57.
2. ГОСТ Р 34.10-2001. Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи. М.: Изд-во стандартов, 2001. 16 с.
3. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. М.: Триумф, 2002. 816 с.
4. Kobara K., Imai H. On the Channel Capacity of Narrow-band Subliminal Channels // In Proc. of ICICS '99. 1999. Vol. 1726. P. 309–323.
5. Simmons G. J. Subliminal Channels: Past and Present // European Transactions on Telecommunications. 1994. Vol. 4. No. 4. P. 459–473.
6. Simmons G. J. Subliminal Communication is Easy Using the DSA // Advances in Cryptology — EUROCRYPT '93 Proceedings. Springer-Verlag. 1994. P. 218–232.
7. Simmons G. J. The Prisoner's Problem and the Subliminal Channel // Advances in Cryptology: Proceedings of CRYPTO '83. Plenum Press. 1984. P. 51–67.