ГЛАВА 5. ИДЕНТИФИКАЦИЯ И ПРОВЕРКА ПОДЛИННОСТИ

5.1. Основные понятия и концепции

С каждым объектом компьютерной системы (КС) связана некоторая информация, однозначно идентифицирующая его. Это может быть *число, строка символов, алгоритм,* определяющий данный объект. Эту информацию называют *идентификатором объекта.* Если объект имеет некоторый идентификатор, зарегистрированный в сети, он называется законным (легальным) объектом; остальные объекты относятся к незаконным (нелегальным).

Идентификация объекта – одна из функций подсистемы защиты. Эта функция выполняется в первую очередь, когда объект делает попытку войти в сеть. Если процедура идентификации завершается успешно, данный объект считается законным для данной сети.

Следующий шаг – аутентификация объекта (проверка подлинности объекта). Эта процедура устанавливает, является ли данный объект именно таким, каким он себя объявляет.

После того как объект идентифицирован и подтверждена его подлинность, можно установить сферу его действия и доступные ему ресурсы КС. Такую процедуру называют *предоставлением полномочий (авторизацией).*

Перечисленные три процедуры инициализации являются процедурами защиты и относятся к одному объекту КС [48].

При защите каналов передачи данных *подтверждение подлинности* (аутентификация) *объектов* означает взаимное установление подлинности объектов, связывающихся между собой по линиям связи. Процедура подтверждения подлинности выполняется обычно в начале сеанса в процессе установления соединения абонентов. (Термин "соединение" указывает на логическую связь (потенциально двустороннюю) между двумя объектами

сети. Цель данной процедуры – обеспечить уверенность, что соединение установлено с законным объектом и вся информация дойдет до места назначения.

После того как соединение установлено, необходимо обеспечить выполнение требований защиты при обмене сообщениями:

(а) получатель должен быть уверен в подлинности источника данных;

(б) получатель должен быть уверен в подлинности передаваемых данных;

(в) отправитель должен быть уверен в доставке данных получателю;

(г) отправитель должен быть уверен в подлинности доставленных данных.

Для выполнения требований (а) и (б) средством защиты является *цифровая подпись.* Для выполнения требований (в) и (г) отправитель должен получить *уведомление* о *вручении* с помощью удостоверяющей почты (certified mail). Средством защиты в такой процедуре является цифровая подпись подтверждающего ответного сообщения, которое в свою очередь является доказательством пересылки исходного сообщения.

Если эти четыре требования реализованы в КС, то гарантируется защита данных при их передаче по каналу связи и обеспечивается функция защиты, называемая функцией подтверждения (неоспоримости) передачи. В этом случае отправитель не может отрицать ни факта посылки сообщения, ни его содержания, а получатель не может отрицать ни факта получения сообщения, ни подлинности его содержания.

5.2. Идентификация и механизмы подтверждения подлинности пользователя

Прежде чем получить доступ к КС, пользователь должен идентифицировать себя, а затем средства защиты сети должны подтвердить подлинность этого пользователя, т.е. проверить, является ли данный пользователь действительно тем, за кого он себя выдает. Компоненты механизма защиты легальных пользователей размещены на рабочей ЭВМ, к которой подключен пользователь через его терминал (или каким-либо иным способом). Поэтому процедуры идентификации, подтверждения подлинности и наделения полномочиями выполняются в начале сеанса на местной рабочей ЭВМ.

Когда пользователь начинает работу в КС, используя терминал, система запрашивает его имя и идентификационный номер. В зависимости от ответов пользователя компьютерная система проводит его идентификацию. Затем система проверяет, является ли пользователь действительно тем, за кого он себя выдает. Для этого она запрашивает у пользователя пароль. Пароль – это лишь один из способов подтверждения подлинности пользователя.

Перечислим возможные способы подтверждения подлинности.

- Предопределенная информация, находящаяся в распоряжении пользователя: пароль, персональный идентификационный номер, соглашение об использовании специальных закодированных фраз.
- Элементы аппаратного обеспечения, находящиеся в распоряжении пользователя: ключи, магнитные карточки, микросхемы и т.п.
- Характерные личные особенности пользователя: отпечатки пальцев, рисунок сетчатки глаза, тембр голоса и т.п.
- Характерные приемы и черты поведения пользователя в режиме реального времени: особенности динамики и стиль работы на клавиатуре, приемы работы с манипулятором и т.п.
- Навыки и знания пользователя, обусловленные образованием, культурой, обучением, воспитанием, привычками и т.п.

Применение пароля для подтверждения подлинности пользователя. Традиционно каждый законный пользователь компьютерной системы получает идентификационный номер и/или пароль. В начале сеанса работы на терминале пользователь указывает свой идентификационный номер (идентификатор пользователя) системе, которая затем запрашивает у пользователя пароль.

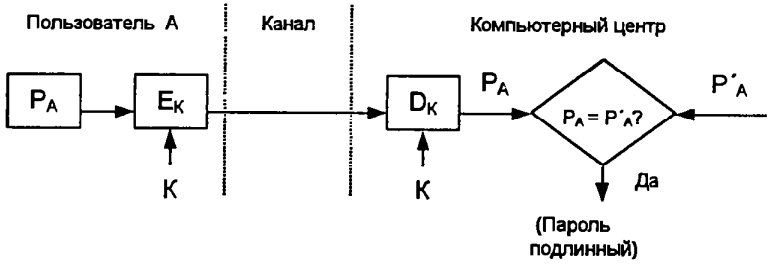

Рис. 5.1. Схема простой аутентификации с помощью пароля

Простейший метод подтверждения подлинности с использованием пароля основан на сравнении представляемого пользователем пароля P_A с исходным значением P_A', хранящимся в компьютерном центре (рис. 5.1) [104]. Поскольку пароль должен храниться в тайне, его следует шифровать перед пересылкой по незащищенному каналу. Если значения P_A и P_A' совпадают, то пароль P_A считается подлинным, а пользователь – законным.

Если кто-нибудь, не имеющий полномочий для входа в систему, узнает каким-либо образом пароль и идентификационный номер законного пользователя, он получит доступ в систему.

Иногда получатель не должен раскрывать исходную открытую форму пароля. В этом случае отправитель должен пересылать вместо открытой формы пароля отображение пароля, получаемое с использованием односторонней функции $\alpha(\cdot)$ пароля. Это преобразование должно гарантировать невозможность раскрытия противником пароля по его отображению, так как противник наталкивается на неразрешимую числовую задачу.

Например, функция $\alpha(\cdot)$ может быть определена следующим образом:

$$\alpha\,(P) = E_P\,(ID),$$

где P – пароль отправителя; ID – идентификатор отправителя; E_P – процедура шифрования, выполняемая с использованием пароля P в качестве ключа.

Такие функции особенно удобны, если длина пароля и длина ключа одинаковы. В этом случае подтверждение подлинности с помощью пароля состоит из пересылки получателю отображения $\alpha(P)$ и сравнения его с предварительно вычисленным и хранимым эквивалентом $\alpha'(P)$.

На практике пароли состоят только из нескольких букв, чтобы дать возможность пользователям запомнить их. Короткие пароли уязвимы к атаке полного перебора всех вариантов. Для того чтобы предотвратить такую атаку, функцию $\alpha(P)$ определяют иначе, а именно:

$$\alpha\,(P) = E_{P \oplus K}\,(ID),$$

где K и ID – соответственно ключ и идентификатор отправителя.

Очевидно, значение $\alpha(P)$ вычисляется заранее и хранится в виде $\alpha'(P)$ в идентификационной таблице у получателя (рис. 5.2). Подтверждение подлинности состоит из сравнения двух отображений пароля $\alpha(P_A)$ и $\alpha'(P_A)$ и признания пароля P_A, если эти отображения равны. Конечно, любой, кто получит доступ к идентификационной таблице, может незаконно изменить ее содержимое, не опасаясь, что эти действия будут обнаружены.

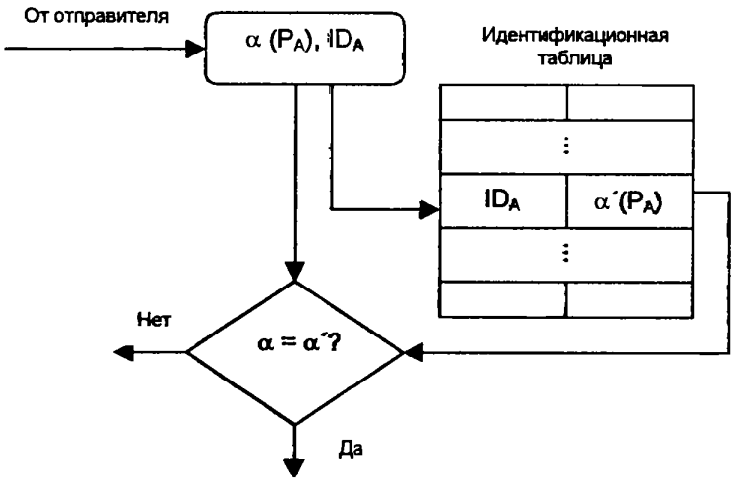

Рис. 5.2. Схема аутентификации с помощью пароля с использованием идентификационнои таблицы

Применение для целей идентификации и аутентификации персонального идентификационного номера PIN рассматривается в гл. 9.

5.3. Взаимная проверка подлинности пользователей

Обычно стороны, вступающие в информационный обмен, нуждаются во взаимной проверке подлинности (аутентификации) друг друга. Этот процесс взаимной аутентификации выполняют в начале сеанса связи.

Для проверки подлинности применяют следующие способы [48]:

- механизм запроса-ответа;
- механизм отметки времени ("временной штемпель").

Механизм запроса-ответа состоит в следующем. Если пользователь A хочет быть уверенным, что сообщения, получаемые им от пользователя B, не являются ложными, он включает в посылаемое для B сообщение непредсказуемый элемент – запрос X (например, некоторое случайное число). При ответе пользователь B должен выполнить над этим элементом некоторую операцию (например, вычислить некоторую функцию f (X)). Это невозможно осуществить заранее, так как пользователю B неизвестно, какое случайное число X придет в запросе. Получив ответ с результатом действий B, пользователь A может быть уверен, что B – подлинный. Недостаток этого метода – возможность установления закономерности между запросом и ответом.

Механизм отметки времени подразумевает регистрацию времени для каждого сообщения. В этом случае каждый пользователь сети может определить, насколько "устарело" пришедшее сообщение, и решить не принимать его, поскольку оно может быть ложным.

В обоих случаях для защиты механизма контроля следует применять шифрование, чтобы быть уверенным, что ответ послан не злоумышленником.

При использовании отметок времени возникает проблема *допустимого временного интервала задержки* для подтверждения подлинности сеанса. Ведь сообщение с "временным штемпелем" в принципе не может быть передано мгновенно. Кроме того, компьютерные часы получателя и отправителя не могут быть абсолютно синхронизированы. Какое запаздывание "штемпеля" является подозрительным?

Для взаимной проверки подлинности обычно используют *процедуру "рукопожатия"* [48, 104]. Эта процедура базируется на указанных выше механизмах контроля и заключается во взаимной проверке ключей, используемых сторонами. Иначе говоря, стороны признают друг друга законными партнерами, если докажут друг другу, что обладают правильными ключами. Процедуру рукопожатия обычно применяют в компьютерных сетях при организации сеанса связи между пользователями, пользователем и хост-компьютером, между хост-компьютерами и т.д.

Рассмотрим в качестве примера процедуру рукопожатия для двух пользователей A и B. (Это допущение не влияет на общность рассмотрения. Такая же процедура используется, когда вступающие в связь стороны не являются пользователями). Пусть применяется симметричная криптосистема. Пользователи A и B разделяют один и тот же секретный ключ K_{AB}. Вся процедура показана на рис. 5.3.

- Пусть пользователь A инициирует процедуру рукопожатия, отправляя пользователю B свой идентификатор ID_A в открытой форме.
- Пользователь B, получив идентификатор ID_A, находит в базе данных секретный *ключ* K_{AB} и вводит его в свою криптосистему.

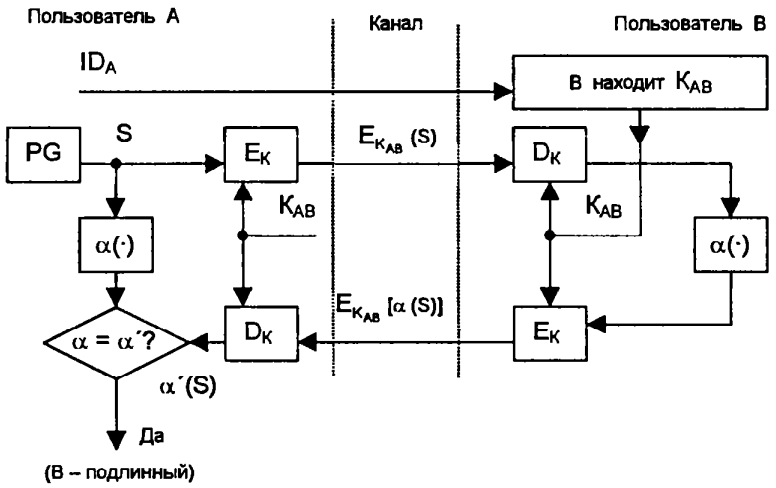

Рис. 5.3.Схема процедуры рукопожатия (пользователь А проверяет подлинность пользователя В)

- Тем временем *пользователь А* генерирует случайную *последовательность S* с помощью псевдослучайного генератора PG и отправляет ее *пользователю В* в виде криптограммы
$$E_{K_{AB}}(S).$$
- Пользователь В расшифровывает эту криптограмму и раскрывает исходный вид последовательности S.
- Затем оба пользователя А и В преобразуют последовательность S, используя открытую одностороннюю функцию α(·).
- Пользователь В шифрует сообщение α(S) и отправляет эту криптограмму *пользователю А*.
- Наконец, пользователь А расшифровывает эту криптограмму и сравнивает полученное сообщение α′(S) с исходным α(S). Если эти сообщения равны, пользователь А признает подлинность пользователя В.

Очевидно, пользователь В проверяет подлинность пользователя А таким же способом. Обе эти процедуры образуют процедуру рукопожатия, которая обычно выполняется в самом начале любого сеанса связи между любыми двумя сторонами в компьютерных сетях.

Достоинством модели рукопожатия является то, что ни один из участников сеанса связи не получает никакой секретной информации во время процедуры подтверждения подлинности.

Иногда пользователи хотят иметь непрерывную проверку подлинности отправителей в течение всего сеанса связи. Один из простейших способов непрерывной проверки подлинности показан на рис. 5.4 [104]. Передаваемая криптограмма имеет вид

$$E_K(ID_A, M),$$

где ID_A – идентификатор отправителя А; М – сообщение.

Получатель В, принявший эту криптограмму, расшифровывает ее и раскрывает пару (ID_A, M). Если принятый идентификатор ID_A совпадает с хранимым значением ID_A', получатель В признает эту криптограмму.

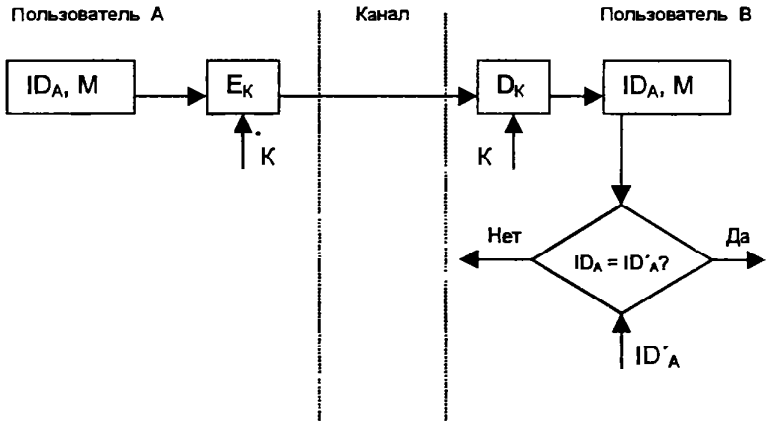

Рис. 5.4. Схема непрерывной проверки подлинности отправителя

Другой вариант непрерывной проверки подлинности использует вместо идентификатора отправителя его секретный пароль. Заранее подготовленные пароли известны обеим сторонам. Пусть P_A и P_B – пароли пользователей А и В соответственно. Тогда пользователь А создает криптограмму

$$C = E_K(P_A, M).$$

Получатель криптограммы расшифровывает ее и сравнивает пароль, извлеченный из этой криптограммы, с исходным значением. Если они равны, получатель признает эту криптограмму.

Процедура рукопожатия была рассмотрена в предположении, что пользователи А и В уже имеют общий *секретный сеансовый ключ*. Реальные процедуры предназначены для распределения ключей между подлинными партнерами и включает как этап распределения ключей, так и этап собственно подтверждения подлинности партнеров по информационному обмену. Такие процедуры будут рассмотрены в гл. 7.

5.4. Протоколы идентификации с нулевой передачей знаний

Широкое распространение интеллектуальных карт (смарт-карт) для разнообразных коммерческих, гражданских и военных применений (кредитные карты, карты социального страхования, карты доступа в охраняемое помещение, компьютерные пароли и ключи, и т.п.) потребовало обеспечения безопасной идентификации таких карт и их владельцев. Во многих приложениях главная проблема заключается в том, чтобы при предъявлении интеллектуальной карты оперативно обнаружить обман и отказать обманщику в допуске, ответе или обслуживании.

Для безопасного использования интеллектуальных карт разработаны протоколы идентификации с нулевой передачей знаний [102]. Секретный ключ владельца карты становится неотъемлемым признаком его личности. Доказательство знания этого секретного ключа с нулевой передачей этого знания служит доказательством подлинности личности владельца карты.

Упрощенная схема идентификации с нулевой передачей знаний

Схему идентификации с нулевой передачей знаний предложили в 1986 г. У.Фейге, А.Фиат и А.Шамир. Она является наиболее известным доказательством идентичности с нулевой передачей конфиденциальной информации.

Рассмотрим сначала упрощенный вариант схемы идентификации с нулевой передачей знаний для более четкого выявления ее основной концепции. Прежде всего выбирают случайное значение модуля n, который является произведением двух больших простых чисел. Модуль n должен иметь длину 512...1024 бит. Это значение n может быть представлено группе пользователей, которым придется доказывать свою подлинность. В процессе идентификации участвуют две стороны:

- сторона A, доказывающая свою подлинность,
- сторона B, проверяющая представляемое стороной A доказательство.

Для того чтобы сгенерировать открытый и секретный ключи для стороны A, доверенный арбитр (Центр) выбирает некоторое число V, которое является квадратичным вычетом по модулю n. Иначе говоря, выбирается такое число V, что сравнение

$$x^2 \equiv V \pmod{n}$$

имеет решение и существует целое число

$$V^{-1} \bmod n.$$

Выбранное значение V является *открытым ключом* для A. Затем вычисляют наименьшее значение S, для которого

$$S \equiv \text{sqrt}\,(V^{-1}) \pmod{n}.$$

Это значение S является *секретным ключом* для A

Теперь можно приступить к выполнению протокола идентификации.

1. Сторона A выбирает некоторое случайное число r, $r < n$. Затем она вычисляет

$$x = r^2 \bmod n$$

и отправляет x стороне B.

2. Сторона B посылает A случайный бит b

3. Если b=0, тогда A отправляет r стороне B. Если b=1, то A отправляет стороне B

$$y = r * S \bmod n.$$

4. Если b = 0, сторона B проверяет, что

$$x = r^2 \bmod n,$$

чтобы убедиться, что A знает sqrt (x). Если b=1, сторона B проверяет, что

$$x = y^2 * V \bmod n,$$

чтобы быть уверенной, что A знает $\text{sqrt}\,(V^{-1})$.

Эти шаги образуют один цикл протокола, называемый *аккредитацией*. Стороны A и B повторяют этот цикл t раз при разных случайных значениях r и b до тех пор, пока B не убедится, что A знает значение S.

Если сторона A не знает значения S, она может выбрать такое значение r, которое позволит ей обмануть сторону B, если B отправит ей b=0, либо A может выбрать такое r, которое позволит обмануть B, если B отправит ей b=1. Но этого невозможно сделать в обоих случаях. Вероятность того, что A обманет B в одном цикле, составляет 1/2. Вероятность обмануть B в t циклах равна $(1/2)^t$.

Для того чтобы этот протокол работал, сторона A никогда не должна повторно использовать значение r. Если A поступила бы таким образом, а сторона B отправила бы стороне A на шаге 2 другой случайный бит b, то B имела бы оба ответа A. После этого B может вычислить значение S, и для A все закончено.

Параллельная схема идентификации с нулевой передачей знаний

Параллельная схема идентификации позволяет увеличить число аккредитаций, выполняемых за один цикл, и тем самым уменьшить длительность процесса идентификации.

Как и в предыдущем случае, сначала генерируется число n как произведение двух больших чисел. Для того, чтобы сгенерировать открытый и секретный ключи для стороны A, сначала выбирают K различных чисел $V_1, V_2, ..., V_K$, где каждое V_i является квадратичным вычетом по модулю n. Иначе говоря, выбирают значение V_i таким, что сравнение

$$x^2 \equiv V_i \bmod n$$

имеет решение и существует $V_i^{-1} \bmod n$. Полученная строка $V_1, V_2, ..., V_K$ является *открытым ключом*.

Затем вычисляют такие наименьшие значения S_i, что

$$S_i = \text{sqrt}\,(V_i^{-1}) \bmod n.$$

Эта строка $S_1, S_2, ..., S_K$ является *секретным ключом* стороны A.

Протокол процесса идентификации имеет следующий вид:

1. Сторона A выбирает некоторое случайное число r, r<n. Затем она вычисляет $x=r^2 \bmod n$ и посылает x стороне B.

2. Сторона B отправляет стороне A некоторую случайную двоичную строку из K бит: $b_1, b_2, ..., b_K$.

3. Сторона A вычисляет

$$y = r * (S_1^{b_1} * S_2^{b_2} * ... * S_K^{b_K}) \bmod n.$$

Перемножаются только те значения S_i, для которых $b_i=1$. Например, если $b_1=1$, то сомножитель S_1 входит в произведение, если же $b_1=0$, то S_1 не входит в произведение, и т.д. Вычисленное значение y отправляется стороне B.

4. Сторона B проверяет, что

$$x = y^2 * (V_1^{b_1} * V_2^{b_2} * ... * V_K^{b_K}) \bmod n.$$

Фактически сторона B перемножает только те значения V_i, для которых $b_i=1$. Стороны A и B повторяют этот протокол t раз, пока B не убедится, что A знает $S_1, S_2, ..., S_K$.

Вероятность того, что A может обмануть B, равна $(1/2)^{Kt}$. Авторы рекомендуют в качестве контрольного значения брать вероятность обмана B равной $(1/2)^{20}$ при K=5 и t=4.

Пример. Рассмотрим работу этого протокола для небольших числовых значений [102]. Если n = 35 (n – произведение двух простых чисел 5 и 7), то возможные квадратичные вычеты будут следующими:

1: $x^2 \equiv 1 \pmod{35}$ имеет решения: x =1, 6, 29, 34;
4: $x^2 \equiv 4 \pmod{35}$ имеет решения: x = 2, 12, 23, 33;
9: $x^2 \equiv 9 \pmod{35}$ имеет решения: x = 3, 17, 18, 32;
11: $x^2 \equiv 11 \pmod{35}$ имеет решения: x = 9, 16, 19, 26;
14: $x^2 \equiv 14 \pmod{35}$ имеет решения: x = 7, 28;
15: $x^2 \equiv 15 \pmod{35}$ имеет решения: x = 15, 20;
16: $x^2 \equiv 16 \pmod{35}$ имеет решения: x = 4, 11, 24, 31;
21: $x^2 \equiv 21 \pmod{35}$ имеет решения: x =14, 21;
25: $x^2 \equiv 25 \pmod{35}$ имеет решения: x = 5, 30;
29: $x^2 \equiv 29 \pmod{35}$ имеет решения: x = 8, 13, 22, 27;
30: $x^2 \equiv 30 \pmod{35}$ имеет решения: x =10, 25.

Заметим, что 14, 15, 21, 25 и 30 не имеют обратных значений по модулю 35, потому что они не являются взаимно простыми с 35. Следует также отметить, что число квадратичных вычетов по модулю 35, взаимно простых с $n = p * q = 5 * 7 = 35$ (для которых НОД (x, 35) =1), равно

$$(p-1)(q-1)/4 = (5-1)(7-1)/4 = 6.$$

Составим таблицу квадратичных вычетов по модулю 35, обратных к ним значений по модулю 35 и их квадратных корней.

V	V^{-1}	$S = \text{sqrt}\,(V^{-1})$
1	1	1
4	9	3
9	4	2
11	16	4
16	11	9
29	29	8

Итак, сторона A получает открытый ключ, состоящий из K=4 значений V:

[4, 11, 16, 29].

Соответствующий секретный ключ, состоящий из K = 4 значений S:

[3 4 9 8].

Рассмотрим один цикл протокола.

1. Сторона A выбирает некоторое случайное число r = 16, вычисляет

$$x = 16^2 \bmod 35 = 11$$

и посылает это значение x стороне B.

2. Сторона B отправляет стороне A некоторую случайную двоичную строку

[1, 1, 0, 1].

3. Сторона A вычисляет значение

$$y = r * (S_1^{b_1} * S_2^{b_2} * ... * S_K^{b_K}) \bmod n = 16 * (3^1 * 4^1 * 9^0 * 8^1) \bmod 35 = 31$$

и отправляет это значение y стороне B.

4. Сторона B проверяет, что

$$x = y^2 * (V_1^{b_1} * V_2^{b_2} * ... * V_K^{b_K}) \bmod n = 31^2 * (4^1 * 11^1 * 16^0 * 29^1) \bmod 35 = 11.$$

Стороны A и B повторяют этот протокол t раз, каждый раз с разным случайным числом r, пока сторона B не будет удовлетворена.

При малых значениях величин, как в данном примере, не достигается настоящей безопасности. Но если n представляет собой число длиной 512 бит и более, сторона B не сможет узнать ничего о секретном ключе стороны A, кроме того факта, что сторона A знает этот ключ.

В этот протокол можно включить идентификационную информацию [104].

Пусть I – некоторая двоичная строка, представляющая идентификационную информацию о владельце карты (имя, адрес, персональный идентификационный номер, физическое описание) и о карте (дата окончания действия и т.п.). Эту информацию I формируют в Центре выдачи интеллектуальных карт по заявке пользователя A.

Далее используют одностороннюю функцию f (·) для вычисления f (I, j), где j – некоторое двоичное число, сцепляемое со строкой I. Вычисляют значения

$$V_j = f(I, j)$$

для небольших значений j, отбирают K разных значений j, для которых V_j являются квадратичными вычетами по модулю n. Затем для отобранных квадратичных вычетов V_j вычисляют наименьшие квадратные корни из $V_j^{-1} \pmod n$. Совокупность из K значений V_j образует открытый ключ, а совокупность из K значений S_j – секретный ключ пользователя A.

Схема идентификации Гиллоу – Куискуотера

Алгоритм идентификации с нулевой передачей знания, разработанный Л.Гиллоу и Ж.Куискуотером [102], имеет несколько лучшие характеристики, чем предыдущая схема идентификации. В этом алгоритме обмены между сторонами A и B и аккредитации в каждом обмене доведены до абсолютного минимума – для каждого доказательства требуется только один обмен с одной аккредитацией. Однако объем требуемых вычислений для этого алгоритма больше, чем для схемы Фейге–Фиата–Шамира.

Пусть сторона A – интеллектуальная карточка, которая должна доказать свою подлинность проверяющей стороне B. Идентификационная информация стороны A представляет собой битовую строку I, которая включает имя владельца карточки, срок действия, номер банковского счета и др. Фактически идентификационные данные могут занимать достаточно длинную строку, и тогда их хэшируют к значению I.

Строка I является аналогом открытого ключа. Другой открытой информацией, которую используют все карты, участвующие в данном приложении, являются модуль n и показатель степени V. Модуль n является произведением двух секретных простых чисел.

Секретным ключом стороны A является величина G, выбираемая таким образом, чтобы выполнялось соотношение

$$I * G^V \equiv 1 \pmod n.$$

Сторона A отправляет стороне B свои идентификационные данные I. Далее ей нужно доказать стороне B, что эти идентификационные данные принадлежат именно ей. Чтобы добиться этого, сторона A должна убедить сторону B, что ей известно значение G.

Вот протокол доказательства подлинности A без передачи стороне B значения G:

1. Сторона A выбирает случайное целое r, такое, что $1 < r \le n - 1$. Она вычисляет

$$T = r^V \bmod n$$

и отправляет это значение стороне B.

2. Сторона B выбирает случайное целое d, такое, что $1 < d \le n - 1$, и отправляет это значение d стороне A.

3. Сторона A вычисляет

$$D = r * G^d \bmod n$$

и отправляет это значение стороне B.

4. Сторона B вычисляет значение

$$T' = D^V I^d \bmod n.$$

Если $T \equiv T' \pmod n$,

то проверка подлинности успешно завершена.

Математические выкладки, использованные в этом протоколе, не очень сложны:

$$T' = D^V I^d = (r G^d)^V I^d = r^V G^{dV} I^d = r^V (I G^V)^d = r^V \equiv T \pmod n,$$

поскольку G вычислялось таким образом, чтобы выполнялось соотношение

$$IG^V \equiv 1 \pmod n.$$