ТЕМА 4. Идентификация и проверка подлинности

4.1 Аутентификация сообщений

Аутентификация – процедура проверки подлинности.

Это понятие применимо и к сеансу связи, и к сторонам, передающим сообщения, и к сообщениям и др.

Аутентификация сообщения - проверка того, что данные, переданные по каналу связи, являются подлинными по своему содержанию, источнику, по времени создания, времени пересылки и т.д.

Целостность сообщения - свойство, позволяющее убедиться в том, что сообщение не было изменено несанкционированным лицом с тех пор, как было создано, передано или сохранено автором сообщения.

Под изменениями обычно понимают *пропуски, вставки, замены и перестановки фрагментов* сообщения.

Принципы аутентификации

– Аутентификация источника сообщения предполагает и проверку целостности, так как если данные подверглись модификации, то они уже имеют другой источник.

– При аутентификации сообщений не требуется проверка времени создания и единственности документа.

– Нарушение единственности подразумевает его повторную передачу или повторное использование.

Замечания:

Если источник сообщений один, то вместо термина аутентификация источника сообщений используют термин аутентификация сообщений.

Целостность данных и аутентификация данных тесно связаны друг с другом:

– *Если данные подверглись модификации, то у них автоматически изменился источник.*

– *Если же известен источник, то без ссылки на него нельзя установить целостность.*

– *Следовательно, аутентификация источника* сообщений включает проверку целостности данных.

Задача аутентификации и обеспечения целостности данных не решается простым шифрованием. С одной стороны в шифротексте трудно сделать какие-либо осмысленные изменения, поскольку они легко потом выявляются при расшифровании. Только пользователи, обладающие секретным ключом шифрования, могут изготовить зашифрованное сообщение.

Однако только шифрование неспособно обеспечить их аутентичности по следующим причинам:

✓ Изменения, внесенные шифротекст, становятся очевидными после

расшифрования только в случае большой избыточности исходных данных;

✓ Факт успешного (в смысле предыдущего пункта) расшифрования сообщения на секретном ключе может подтвердить его авторство только для самого получателя.

Имитозащита данных

При передаче и хранении данных информация неминуемо оказывается вне зоны непосредственного контроля за ней *следовательно* в реальной жизни физически защитить данные от несанкционированных изменений часто не возможно.

Под защитой данных от несанкционированных изменений в криптографии понимают не исключение самой возможности таких изменений, а набор методов, позволяющих надежно зафиксировать их факты, если они имели место.

Имитозащита данных – это защита от навязывания злоумышленником ложных данных, т.е. выдачи их за подлинные.

MAC-коды

MAC (message authentication code) - код аутентичности сообщения - определенным способом вычисленный набор символов, добавляемый к сообщению и предназначенный для проверки его целостности и аутентификации источника данных.

MAC (K, M) зависит от передаваемого сообщения M и секретного ключа K отправителя A, и обладает свойствами:

– получатель B, используя такой же или другой ключ, имеет возможность проверить целостность и доказать принадлежность информации отправителю A;

– код аутентификации невозможно фальсифицировать.

Порядок работы с MAC кодом

MAC вычисляется в тот момент, когда известно, что сообщение корректно. После этого MAC присоединяется к сообщению и передается вместе с ним получателю. Получатель вычисляет MAC, используя тот же самый секретный ключ, и сравнивает вычисленное значение с полученным. Если эти значения совпадают, то с большой долей вероятности можно считать, что при пересылке изменения сообщения не произошло.

Свойства функции MAC

Свойства, которыми должна обладать функция MAC = f (M) для вычисления MAC:

1. Должно быть вычислительно трудно, зная M и $MAC = f(M)$ найти сообщение $M' \neq M$ с тем же *MAC*-кодом.
2. Значения $MAC = f(M)$ должны быть равномерно распределены в том смысле, что для любых сообщений M и M' вероятность того, что они будут иметь равные *MAC*-коды ($f(M) = f(M')$) должна быть равна 2^{-n}, где n - длина значения *MAC*.

Безопасность вычислений MAC:

Если длина ключа, используемого при вычислении MAC, равна k, то потребуется выполнить 2^k попыток для перебора всех ключей.

Если длина значения MAC-кода, равна n, то всего существует 2^n различных значений MAC.

ПРИМЕР

Пусть противник имеет доступ к открытому сообщению и его MAC-коду.

Если k > n, то зная сообщение M1 и MAC1= f (M1), он может найти MACi = fki (M1) для всех возможных ключей ki.

При этом хотя бы для одного ключа он получит

MACi = MAC1.

При этом усилия противника – вычисления 2^k значений MAC.

Так как по предположению k > n, то $2^k > 2^n$ и правильное значение MAC будет получено для нескольких ключей.

Поэтому для нахождения единственного ключа противнику требуется знать несколько пар сообщений и соответствующих им MAC-кодов.

Таким образом, простой перебор всех ключей требует не меньше, а больше усилий, чем поиск ключа симметричного шифрования той же длины.

Построение кода аутентификации:

Код аутентификации может быть построен:

1) на симметричной криптосистеме, в таком случае обе стороны имеют один общий секретный ключ,

2) на криптосистеме с открытым ключом, в которой А использует свой секретный ключ, а В - открытый ключ отправителя А.

Наиболее универсальный способ аутентификации сообщений через схемы ЭЦП на криптосистемах с открытым ключом состоит в том, что сторона А отправляет стороне В сообщение

M || ЭЦП (K, H (M)),

где H(M) – криптографическая хэш-функция в схеме ЭЦП.

Для аутентификации большого объема информации этот способ не подходит из-за медленной операции вычисления подписи.

Построение кода аутентификации

Например, вычисление одной ЭЦП на криптосистемах с открытым ключом занимает порядка 10 мс на ПК.

При средней длине IP-пакета 1Кб, для каждого из которых требуется вычислить код аутентификации, получим максимальную пропускную способность в 1 Кб/10 мс = 100Кб/с.

Поэтому для большого объема данных, которые нужно аутентифицировать, пользователи А и В создают общий секретный ключ аутентификации К.

Далее код аутентификации вычисляется либо с помощью блокового

шифра, либо с помощью криптографической хэш-функции.

Замечание: отечественный вариант названия MAC – имитовставка.

Выработка *MAC-кода* сообщений с использованием криптографического преобразования данных официально или полуофициально закреплена во многих стандартах шифрования.

Например, в комментариях к стандарту шифрования США для выработки *MAC-кода* рекомендуется использовать AES.

Российский стандарт шифрования ГОСТ 28147-89 явным образом предусматривает режим выработки имитовставки.

Выработка имитовставки ГОСТ 28147-89

Обеспечение имитозащиты предусмотрено ГОСТ 28147-89:

1) При выработке имитовставки используются 16 циклов шифрования в *режиме простой замены*.

2) Выработка имитовставки производится с использованием *той же* ключевой информации, что применяется для шифрования исходного текста.

3) Вся входная открытая информация разбивается на 64−битовые блоки, которые перед *выработкой имитовставки* делятся на 32−битовые блоки N_1, N_2. (Если последний 64−битовый блок неполон, то он дополняется нулями).

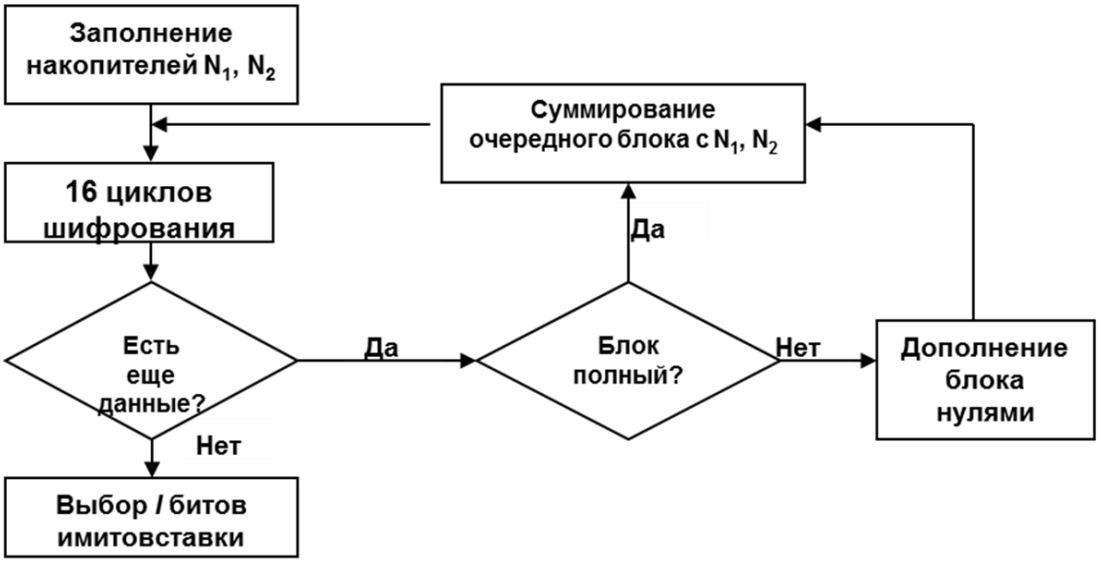

Рисунок 4.1. Схема выработки имитовставки

Длина имитовставки от 1 до 32 битов.

Открытый текст разбивается на блоки длиной 64 бита.

Последний блок в случае необходимости дополняется нулями.

$T_0 = T_0^{(1)} T_0^{(2)} T_0^{(3)} \dots T_0^{(n)}$

Первый блок $T_0^{(1)}$ шифруется в режиме простой замены ГОСТ 28147-89 тем же ключом, что и сообщение, но с применением 16 циклов вместо 32.

Результат по битам по модулю 2 складывается с вторым блоком $T_0^{(2)}$ и так же шифруется. Результат складывается с третьим блоком и так далее.

$I = E'_k(T_0^{(n)} \oplus E'_k(T_0^{(n-1)} \oplus E'_k(\dots E'_k(T_0^{(2)} \oplus E'_k(T_0^{(1)}))) \dots)$

Первые 32 бита получившегося блока составляют имитовставку.

Спецификация шифра предусматривает использование в качестве имитовставки и меньшее количество битов по желанию, но не большее.

Имитовставка обычно передаётся в конце сообщения и может вычисляться либо отдельно от шифрования/расшифрования, либо в процессе оного.

Хэш-функции

Хэш-функция предназначена для сжатия подписываемого документа M до нескольких десятков или сотен бит. Хэш-функция h принимает в качестве аргумента сообщение (документ) произвольной длины и возвращает хэш-значение h(M)=H фиксированной длины.

Обычно хэшированная информация h(M) является сжатым двоичным представлением основного сообщения M произвольной длины. Следует отметить, что значение h(M) хэш-функции сложным образом зависит от документа и не позволяет восстановить сам документ .

Хэш-функция должна удовлетворять целому ряду условий:

– хэш-функция должна быть чувствительна к всевозможным изменениям в тексте M, таким как вставки, выбросы, перестановки и т.п.;

– хэш-функция должна обладать свойством необратимости, то есть задача подбора документа, который обладал бы требуемым значением хэш-функции, должна быть вычислительно неразрешима;

– вероятность того, что значения хэш-функции двух различных документов (вне зависимости от их длин) совпадут, должна быть ничтожно мала.

Большинство хэш-функций строится на основе однонаправленной функции.

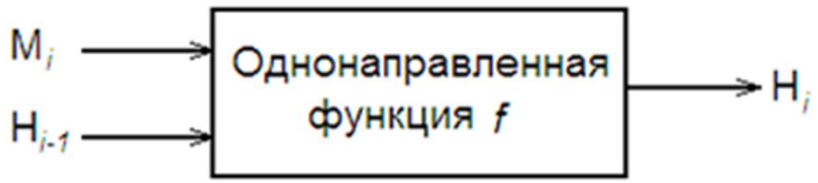

Рисунок 4.2. Схема хэширования на основе блочного алгоритма шифрования

Хэш-значение, вычисляемое при вводе последнего блока текста, становится хэш-значением всего сообщения M. В результате однонаправленная хэш-функция всегда формирует выход фиксированной длины n (независимо от длины входного текста).

Часто функции хэширования строят, используя в качестве однонаправленной функции – симметричный блочный алгоритм шифрования (DES, ГОСТ 28147-89) в режиме с обратной связью, принимая последний блок шифротекста за хэш-значение всего документа.

Так как длина блока в указанных алгоритмах невелика (64 бита), то часто в качестве хэш-значения используют два блока шифротекста.

Отличие Хэш-функции от имитовставки.

Хэш-функция:

1) контроль целостности файлов;
2) контроль передаваемых данных по каналам связи;
3) аутентификации источника данных (не во всех случаях).

Функция имитовставки:

1) контроль целостности файлов;
2) контроль передаваемых данных по каналам связи;
3) контроль целостности при считывании данных (например: с HDD);
4) хеши паролей;
5) аутентификация.

4.2. Электронная цифровая подпись.

Свойства электронной цифровой подписи

1) ЭЦП удостоверяет, что подписанный текст исходит от лица, поставившего подпись;

2) ЭЦП не даёт этому лицу отказаться от обязательств, связанных с подписанным текстом;

3) ЭЦП гарантирует целостность и подлинность подписанного текста.

4) В процедуре постановки подписи используется секретный ключ отправителя, в процедуре проверки – его открытый ключ.

Особенности документов с цифровой подписью.

1) Цифровая подпись позволяет организовать защищённый обмен документами с проверкой их подлинности;

2) Электронным документам может быть придана юридическая значимость;

3) При похищении секретного ключа невозможно доказать подделку электронной цифровой подписи (в отличие от ручной);

4) Электронная подпись гарантирует отсутствие искажений в тексте документа (в отличие от ручной).

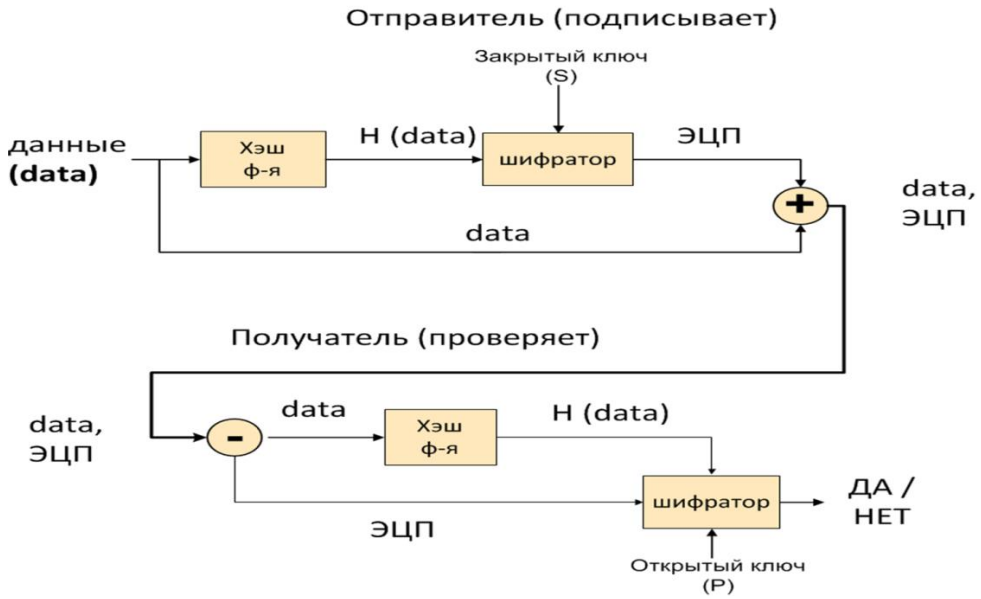

Рисунок 4.3. Структура выработки Электронной цифровой подписи

Метод формирования электронной подписи RSA

1. Отправитель вычисляет секретный ключ X и открытый ключ (E, N) по следующему правилу: $N=PQ$, $N_1=(P-1)(Q-1)$;

подбираются значения E и X, так что

$E<=N_1$, НОД$(E, N_1)=1$, $X<N$, $EX \bmod N_1 = 1$.

2. Вычисляется хэш-функция $m=h(M)$;
3. Вычисляется электронная цифровая подпись в виде числа

$t=m^x \bmod N$

4. Получателю направляется сообщение M, цифровая подпись t и открытый ключ в виде пары чисел (E, N).
5. Используя открытый ключ, получатель вычисляет число $m''=t^E \bmod N$ и хэш-функцию полученного сообщения M.
6. Получатель сравнивает результаты вычислений m и m''. Если они равны, то считается что сообщение M и подпись t являются подлинными

Авторы метода RSA рекомендуют следующие размеры длины открытого ключа:

768 бит для частных лиц,

1024 бит – для коммерческой информации,

2048 бит – для особо секретной информации

Подпись ElGamal.

Для генерации ключевой пары выбираются большое простое число p и примитивный элемент g мультипликативной группы GF(p).

Выбирается случайное число x такое, что $x < p$-1.

Открытым ключом является $y = g^x (mod\ p)$; секретным ключом является

x.

Стойкость основана на сложности дискретного логарифмирования.

Пусть *A* должен подписать сообщение *M*.

Выбирается случайное число *k*, взаимно-простое с *p*-1: НОД(*k; p*-1) = 1. Затем вычисляется a= $g^k \pmod p$

Рассмотрим уравнение

M =(*x a* + *k b*) (mod(*p* - 1)).

По теореме о вычетах ∃ *k^-1*: (*M-x a*) *k^-1* ≡ *b*(mod(*p*-1)).

Подписью под *M* является пара (*a, b*).

Проверка подписи:

Вычисляем $g^M(p)$ и $y^a \cdot a^b(p)$.

Проверяем

$$y^a \cdot a^b \pmod p = g^{a\cdot x} \cdot g^{k\cdot b} \pmod p = g^{a\cdot x + k\cdot b} \pmod p =$$

$$= g^{ax+kk^{-1}(M-xa)+(p-1)nk} \pmod p = g^{M+(p-1)nk} \pmod p = g^M \pmod p.$$

Сертификация электронных цифровых подписей.

Ключевое место в законодательстве, регламентирующем вопросы использования цифровых подписей, отводится Удостоверяющему центру.

Удостоверяющие центры сертифицируют открытый ключ электронной цифровой подписи с целью обеспечения возможности доказательства принадлежности подписи конкретному лицу для придания юридической силы электронным документам.

Корневой удостоверяющий центр.

Корневой удостоверяющий центр (КУЦ) является базовым компонентом ГосСУОК и занимает высшее положение в единой иерархической инфраструктуре доверия открытых ключей, реализуемой ГосСУОК.

Порядок функционирования КУЦ и процедура издания самоподписанного СОК определяются политикой применения СОК (далее — ППС) КУЦ, утвержденной ОАЦ.

Основными функциями КУЦ являются:

– генерация личных и открытых ключей КУЦ;

– издание, распространение, предоставление информации о статусе, отзыв, хранение самоподписанного СОК как начала маршрута сертификации (точки доверия) ГосСУОК;

– издание, распространение, предоставление информации о статусе, отзыв и хранение СОК (далее — управление СОК) РУЦ;

– кросс-сертификация (установление отношений доверия) с внешними инфраструктурами открытых ключей, в том числе с иностранными.

Регистрационные центры

Республиканский удостоверяющий центр

В областных регистрационных центрах, как и в Республиканском удостоверяющем центре столицы (пр-т Машерова, 25), осуществляется выпуск сертификатов открытого ключа с выдачей средств электронной цифровой подписи (ЭЦП).

Основные функции республиканского удостоверяющего центра:

– генерация личных и открытых ключей РУЦ;

– управление СОК РЦ, центра атрибутных сертификатов, физических и юридических лиц, сервисов (приложения, серверы или устройства);

– функции РЦ.

4.3 Идентификация и аутентификация пользователей.

Аутентификация означает установление подлинности. Аутентификация пользователей обеспечивает работу в сети только санкционированных пользователей. Аутентификация проводится при входе в сеть, но может проводиться и во время работы. Обычно проводится после процесса идентификации, во время которого пользователь сообщает свой идентификатор (называет себя).

В процедуре аутентификации участвуют две стороны: пользователь доказывает свою подлинность, а сеть проверяет это доказательство и принимает решение.

Методы аутентификации.

Пароль.

Наиболее распространенное средство аутентификации – пароль. Используется как при входе в систему, так и в процессе работы.

Необходимые требования к паролям:

– правила генерации (длина, случайность символов);

– хранения (хранить в защищенном месте).

Взаимная аутентификация.

Процесс аутентификации может носить обоюдный характер. Обе стороны должны доказать свою подлинность. В этом случае процедура называется – взаимная аутентификация.

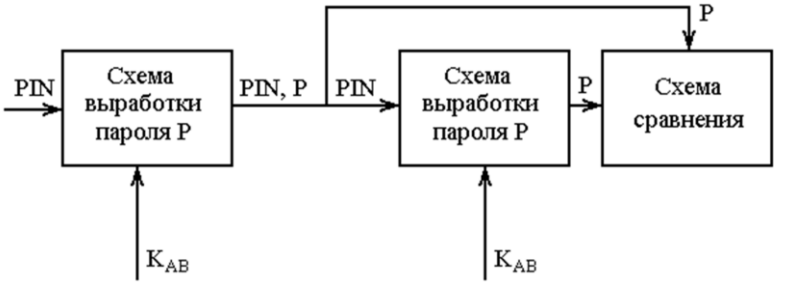

Рисунок 4.4. Схема удаленной аутентификации с использованием пароля

Временной штемпель.

Рассмотренная схема имеет существенный недостаток: злоумышленник может перехватить пароль P и PIN и позднее использовать их для своей аутентификации. Для устранения этого недостатка используют механизм отметки времени ("временной штемпель"). При выработке пароля наряду с ключом используется текущее время в виде некоторого интервала, в пределах которого пароль действителен, аналогично вырабатывается пароль на стороне B, в этом случае устаревшим паролем нельзя воспользоваться.

Удаленная аутентификация пользователей с использованием механизма запроса-ответа.

Если A хочет быть уверенным, что сообщения, получаемые им от 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: Имеется возможность установления закономерности между запросом и ответом.

Процедура "рукопожатия".

Стороны признают друг друга законными партнерами, если докажут друг другу, что обладают правильными ключами.

Процедура рукопожатия для двух пользователей.

Пусть применяется симметричная криптосистема. **A** и **B** разделяют один и тот же секретный ключ K_{AB}.

1) **A** инициирует процедуру рукопожатия, отправляя пользователю **B** свой идентификатор ID_A в открытой форме.

2) **B**, получив идентификатор ID_A, находит в базе данных секретный ключ K_{AB} и вводит его в свою криптосистему.

3) **A** генерирует случайную последовательность **S** с помощью псевдослучайного генератора **PG** и отправляет ее пользователю **B** в виде криптограммы $E_{K_{AB}}(S)$

4) Пользователь **B** расшифровывает эту криптограмму и раскрывает исходный вид последовательности **S**.

5) **A** и **B** преобразуют последовательность **S**, используя открытую одностороннюю функцию a(*).

6) **B** шифрует сообщение a(S) и отправляет эту криптограмму пользователю **A**.

7) **A** расшифровывает эту криптограмму и сравнивает полученное

сообщение a’(S) с исходным a(s).

8) Если эти сообщения равны, пользователь **A** признает подлинность пользователя **B**.

9) **B** проверяет подлинность пользователя **A** таким же способом.

Преимущество модели рукопожатия

Производится 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. Процедура рукопожатия была рассмотрена в предположении, что пользователи A и B доверяют друг другу и имеют общий секретный сеансовый ключ.

Протоколы идентификации с нулевой передачей знаний

Нередки ситуации когда пользователи должны осуществить взаимную аутентификацию, не доверяя друг другу, и не обмениваясь никакой конфиденциальной информацией.

Во многих приложениях главная проблема заключается в том, чтобы при предъявлении интеллектуальной карты оперативно обнаружить обман и отказать обманщику в допуске, ответе или обслуживании.

Секретный ключ владельца карты становится неотъемлемым признаком его личности.

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.

1) Выбирают случайное значение модуля n, который является произведением двух больших простых чисел.

2) Модуль n должен иметь длину 512 - 1024 бит.

3) Это значение может быть представлено группе пользователей, которым придется доказывать свою подлинность.

4) В процессе идентификации участвуют две стороны:

– сторона A, доказывающая свою подлинность;

– сторона B, проверяющая представляемое стороной A доказательство.

5) Для того чтобы сгенерировать открытый и секретный ключи для A, доверенный арбитр (Центр) выбирает некоторое число V, которое является квадратичным вычетом по модулю n.

Иначе говоря, выбирается такое число V, что сравнение

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

имеет решение и существует целое число

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

6) Выбранное значение V является открытым ключом для A.

7) Вычисляют наименьшее значение S, для которого:

$S = sgrt\ (V^{-1})\ (mod\ n)$

8) Это значение S является секретным ключом для A.

Протокол аккредитации.

Теперь можно приступить к выполнению протокола идентификации.

1) A выбирает некоторое случайное число r, где r<n .
2) A вычисляет $x = r^2\ mod\ n$ и отправляет x стороне B.
3) B посылает A случайный бит b .
4) Если b=0, тогда A отправляет r стороне B.
5) Если b=1, то A отправляет $y = r * S\ mod\ n$ стороне B.
6) Если b=0, то сторона B проверяет, что $x = r^2\ mod\ n$,чтобы убедиться, что A знает $sqrt(x)$.
7) Если b=1, сторона B проверяет, что $x = y^2 V\ mod\ n$, чтобы быть уверенной, что A знает $sgrt\ (V^{-1})$.

Безопасность протокола аккредитации

Описанные выше 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 отправит ей b=1. Но этого невозможно сделать в обоих случаях по следующим причинам:

Вероятность того, что A обманет B в одном цикле, составляет *1/2*

Вероятность обмануть B *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 V1,V2 ,…Vk,

где каждое Vi, является квадратичным вычетом по модулю n.

Иначе говоря, выбирают значение Vi, таким, что сравнение

$x^2 \equiv V_i mod\ n$

имеет решение и существует

$$V_i^{-1} mod\ n$$

Полученная строка V1,V2 ,…Vk является открытым ключом.

Затем вычисляют такие наименьшие значения Si, что

$S = sgrt\ V_i^{-1} mod\ n$

Эта строка S1,S2 ,…Sk является секретным ключом стороны A.

Процесс идентификации

1) A выбирает некоторое случайное число r, где r<n
2) A вычисляет $x = r^2 mod\ n$ и посылает x B.
3) B отправляет стороне A некоторую случайную двоичную строку из K бит: b1,b2,…bk.
4) A вычисляет $y = r(S^{b_1} * S^{b_2} * ... * S^{b_k}) mod\ n$.
5) Перемножаются только те значения Si, для которых bi=1.

Например, если bi=1, то сомножитель Si входит в произведение, если же bi=0, то Si не входит в произведение, и т.д..

6) Вычисленное значение y отправляется стороне B.
7) B проверяет, что $x = y^2(V_1^{b_1} * V_2^{b_2} * ... * V_k^{b_k})\ mod\ n$
8) Фактически B перемножает только те значения Vi, для которых bi=1.
9) A и B повторяют этот протокол t раз, пока B не убедится, что A знает S1,S2,…Sk.

Безопасность протокола

Вероятность того, что A может обмануть B, равна $1/2^{kt}$.

Рекомендуется в качестве контрольного значения брать вероятность обмана B равной $(1/2)^{20}$ при *K=5* и *t=4*.

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.

ПРИМЕР

Рассмотрим работу этого протокола для небольших числовых значений.

Если n=35 (n - произведение двух простых чисел 5 и 7), то возможные квадратичные вычеты будут следующими:

1: $x^2 = 1$ (mod 35) имеет решения: x =1, 6, 29, 34;
4: $x^2 = 4$ (mod 35) имеет решения: x = 2, 12, 23, 33;
9: $x^2 = 9$ (mod 35) имеет решения: x = 3,17,18, 32;
11: $x^2 = 11$ (mod 35) имеет решения: x = 9,16,19, 26;
14: $x^2 = 14$ (mod 35) имеет решения: x = 7, 28;
15: $x^2 = 15$ (mod 35) имеет решения: x = 15, 20;

16: $x^2 = 16 \pmod{35}$ имеет решения: x = 4,11,24,31;

21: $x^2 = 21 \pmod{35}$ имеет решения: x =14, 21;

25: $x^2 = 25 \pmod{35}$ имеет решения: x = 5, 30;

29: $x^2 = 29 \pmod{35}$ имеет решения: x = 8,13, 22, 27;

30: $x^2 = 30 \pmod{35}$ 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.

Итак, сторона А получает открытый ключ, состоящий из К=4 значений V: [4,11,16,29].

Соответствующий секретный ключ, состоящий из К = 4 значений S: [3, 4, 9, 8].

Таблица 4.1. Квадратичные вычеты по модулю 35, обратных к ним значеня по модулю 35 и их квадратные корни

V	$\mathbf{V^{-1}}$	$\mathbf{S = sqrt(V^{-1}) \bmod 35}$
1	1	1
4	9	3
9	4	2
11	16	4
16	11	9
29	29	8

Рассмотрим один цикл протокола.

1. Сторона А выбирает некоторое случайное число *r = 16*, вычисляет $x = 16^2 \bmod 35 = 11$ и посылает это значение x стороне В.

2. Сторона В отправляет стороне А некоторую случайную двоичную строку *[1, 1, 0, 1].*

3. Сторона А вычисляет значение $y = r * (S_1^{b1} * S_2^{b2} * ... * S_K^{bK}) \bmod n = 16 * (3^1 * 4^1 * 9^0 * 8^1) \bmod 35 = 31$ и отправляет это значение y стороне В.

4. Сторона В проверяет, что

$x = y^2 * (V_1^{b1} * V_2^{b2} * ... * V_K^{bK}) \bmod n = 31^2 * (4^1 * 11^1 * 16^0 * 29^1) \bmod 35 = 11.$

Биометрические системы идентификации

Это системы контроля доступа, основанные на идентификации и аутентификации человека по биологическим признакам, таким как структура ДНК, рисунок радужной оболочки глаза, сетчатка глаза, геометрия и температурная карта лица, отпечаток пальца, геометрия ладони (статические методы).

Часто используются еще и уникальные динамические методы биометрической аутентификации – подпись, клавиатурный почерк, голос и походка, которые основаны на поведенческих характеристиках людей.

Идентификация в биометрической системе.

Регистрация идентификатора – сведение о физиологической или поведенческой характеристике преобразуется в форму, доступную компьютерным технологиям, и вносятся в память биометрической системы;

Выделение – из вновь предъявленного идентификатора выделяются уникальные признаки, анализируемые системой;

Сравнение – сопоставляются сведения о вновь предъявленном и ранее зарегистрированном идентификаторах;

Решение – выносится заключение о том, совпадают или не совпадают идентификаторы.

Надежность систем.

Одна из самых важных характеристик систем защиты информации, основанных на биометрических технологиях, является высокая надежность, то есть способность системы достоверно различать биометрические характеристики, принадлежащие разным людям, и надежно находить совпадения.

В биометрии эти параметры называются ошибкой первого рода (False Reject Rate, FRR) и ошибкой второго рода (False Accept Rate, FAR).

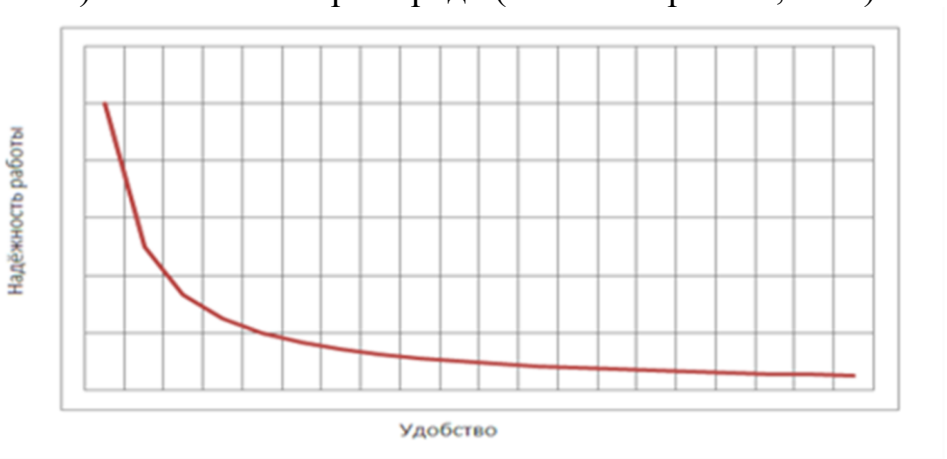

Рисунок 4.5. Зависимость надежности системы от удобства использования

Модуль регистрации

«Обучает» систему идентифицировать конкретного человека.

На этапе регистрации видеокамера или иные датчики сканируют человека для того, чтобы создать цифровое представление его облика. В результате сканирования формируются несколько изображений. В идеальном случае, эти изображения будут иметь слегка различные ракурсы и выражения лица, что позволит получить более точные данные.

Модуль идентификации

Модуль идентификации получает от видеокамеры изображение человека и преобразует его в тот же цифровой формат, в котором хранится шаблон. Полученные данные сравниваются с хранимым в базе данных шаблоном для того, чтобы определить, соответствуют ли эти изображения друг другу. Степень подобия, требуемая для проверки, представляет собой некий порог, который может быть отрегулирован для различного типа персонала, мощности PC, времени суток и ряда иных факторов.