Задания СРС:

**Цель: у**глубление теоретических знаний и развитие практических навыков по обеспечению кибербезопасности, формирование компетенций в области защиты информации, освоение принципов функционирования систем защиты, а также анализ современных киберугроз и методов противодействия им.

**Лекция №6: Электронная цифровая подпись**

### ****Цель:****

Изучить назначение, принципы работы и алгоритмы электронной цифровой подписи, а также ознакомиться с её юридической значимостью и применением в ИКТ.

### ****Задание:****

#### ****1. Теоретическая часть:****

Изучить и законспектировать:

* Понятие и назначение ЭЦП
* Отличие от обычной подписи
* Принцип работы ЭЦП:
  + генерация ключей
  + создание подписи
  + проверка подписи
* Алгоритмы, применяемые для ЭЦП:
  + RSA
  + DSA
  + ГОСТ Р 34.10-2012
* Роль хеш-функций в ЭЦП
* Юридическая значимость ЭЦП в Республике Казахстан (на основе законодательства)

#### ****2. Практическая часть:****

* Составить блок-схему процесса формирования и проверки ЭЦП
* Рассмотреть пример применения ЭЦП в eGov (портал электронного правительства)
* Подготовить обзор одного программного средства для работы с ЭЦП (например, NCALayer, CryptoPro и т.д.)
* (Дополнительно) Выполнить создание и проверку ЭЦП в выбранной системе (если доступна)

#### ****3. Подготовить:****

* Презентацию (8–12 слайдов) с краткими тезисами и схемами
* Краткий отчёт (1–2 страницы) по изученным вопросам и практике

**Цель лекции**: познакомиться с некоторыми алгоритмами формирования и проверки электронной цифровой подписи (*ЭЦП*, ЭП).

### Электронная подпись на основе алгоритма RSA

Рассмотренная нами в Лекции 11 схема использования алгоритма *RSA* при большом модуле N практически не позволяет злоумышленнику получить закрытый *ключ* и прочитать зашифрованное сообщение. Однако она дает возможность злоумышленнику подменить сообщение от абонента А к абоненту Б, так как *абонент* А шифрует свое сообщение открытым ключом, полученным от Б по открытому каналу связи. А раз *открытый ключ* передается по открытому каналу, любой может получить его и использовать для подмены сообщения. Избежать этого можно, используя более сложные протоколы, например, следующий.

Пусть, как и раньше, *пользователь* А хочет передать пользователю Б сообщение, состоящее из нескольких блоков mi. Перед началом сеанса связи абоненты генерируют открытые и закрытые ключи, обозначаемые, как указано в следующей таблице:

|  |  |  |
| --- | --- | --- |
|  | **Открытый ключ** | **Закрытый ключ** |
| Пользователь А | NA, dA | eA |
| Пользователь Б | NБ, dБ | eБ |

В результате каждый *пользователь* имеет свои собственные открытый (состоящий из двух частей) и закрытый ключи. Затем пользователи обмениваются открытыми ключами. Это подготовительный этап протокола.

Основная часть протокола состоит из следующих шагов.

1. Сначала пользователь А вычисляет числа Описание: c_i=m_i^{e_A} \: mod \:N_A, то есть шифрует сообщение своим закрытым ключом. В результате этих действий пользователь А подписывает сообщение.
2. Затем пользователь А вычисляет числа Описание: g_i=c_i^{d_Б} \: mod \: N_Б, то есть шифрует то, что получилось на шаге 1 открытым ключом пользователя Б. На этом этапе сообщение шифруется, чтобы никто посторонний не мог его прочитать.
3. Последовательность чисел gi передается к пользователю Б.
4. Пользователь Б получает gi и вначале вычисляет последовательно числа Описание: c_i=g_i^{e_Б} \: mod \: N_Б, используя свой закрытый ключ. При этом сообщение расшифровывается.
5. Затем Б определяет числа Описание: m_i=c_i^{d_A} \: mod \: N_A
   , используя открытый ключ пользователя А. За счет выполнения этого этапа производится проверка подписи пользователя А.

В результате *абонент* Б получает исходное сообщение и убеждается в том, что его отправил именно *абонент* А. Данная схема позволяет защититься от нескольких видов возможных нарушений, а именно:

* пользователь А не может отказаться от своего сообщения, если он признает, что секретный ключ известен только ему;
* нарушитель без знания секретного ключа не может ни сформировать, ни сделать осмысленное изменение сообщения, передаваемого по линии связи.

Данная схема позволяет избежать многих конфликтных ситуаций. Иногда нет необходимости зашифровывать передаваемое сообщение, но нужно его скрепить электронной подписью. В этом случае из приведенного выше протокола исключаются шаги 2 и 4, то есть текст шифруется закрытым ключом отправителя, и полученная последовательность присоединяется к документу. Получатель с помощью открытого ключа отправителя расшифровывает прикрепленную подпись, которая, по сути, является зашифрованным повторением основного сообщения. Если расшифрованная подпись совпадает с основным текстом, значит, подпись верна.

Существуют и другие варианты применения алгоритма *RSA* для формирования *ЭЦП*. Например, можно шифровать (то есть подписывать) открытым ключом не само сообщение, а хеш-код от него.

Возможность применения алгоритма *RSA* для получения электронной подписи связана с тем, что секретный и открытый ключи в этой системе равноправны. Каждый из ключей, d или e, могут использоваться как для шифрования, так и для расшифрования. Это свойство выполняется не во всех криптосистемах с открытым ключом.

*Алгоритм* *RSA* можно использовать также и для обмена ключами.

### Цифровая подпись на основе алгоритма Эль-Гамаля

#### Принцип создания и проверки подписи

Алгоритм Эль-Гамаля также можно использовать для формирования цифровой подписи. Группа пользователей выбирает общие параметры Р и А. Затем каждый абонент группы выбирает свое секретное число Хi, 1 < Хi< Р-1, и вычисляет соответствующее ему открытое число Описание: Y_i \: : \: Y_i=A^{X_i} \: mod \: P. Таким образом, каждый пользователь получает пару (закрытый ключ; открытый ключ) = (Хi, Yi). Открытые ключи пользователей могут храниться в общей базе системы распределения ключей и при необходимости предоставляться всем абонентам системы.

Сообщение, предназначенное для подписи, должно быть представлено в виде числа, меньшего модуля Р. При большом размере сообщение разбивается на блоки необходимого размера. В некоторых случаях подписывается не само сообщение, а значение хеш-функции от него. В любом варианте цифровая подпись вычисляется в зависимости от некоторого числа m (m < P).

Пусть пользователь 1 хочет подписать свое сообщение цифровой подписью и передать его пользователю 2. В этом случае алгоритм действий следующий.

1. Первый пользователь выбирает случайное секретное число k, взаимно простое с Р-1, и вычисляет число Описание: a=A^k \: mod \: P
2. Затем с помощью расширенного алгоритма Евклида необходимо найти значение b в следующем уравнении:

m = (X1 \* a +k \* b) mod (P-1)

Пара чисел (a, b) будет цифровой подписью сообщения m.

1. Сообщение m вместе с подписью (a, b) отправляется пользователю 2.
2. Пользователь 2 получает сообщение m и с использованием открытого ключа первого абонента Y1 вычисляет два числа по следующим формулам: Описание: c_1=Y_1^a \times a^b \: mod \: P \\ c_2=A^m \: mod \: P  Если с1 = с2, то цифровая подпись первого пользователя верная. Для подписывания каждого нового сообщения должно каждый раз выбираться новое значение k.

Подписи, созданные с использованием алгоритма Эль-Гамаля, называются *рандомизированными*, так как для одного и того же сообщения с использованием одного и того же закрытого ключа каждый раз будут создаваться разные подписи (a,b), поскольку каждый раз будет использоваться новое значение k. Подписи, созданные с применением алгоритма RSA, называются *детерминированными*, так как для одного и того же сообщения с использованием одного и того же закрытого ключа каждый раз будет создаваться одна и та же подпись.

#### Пример вычисления и проверки цифровой подписи

Пусть абоненты, обменивающиеся через Интернет зашифрованными сообщениями, имеют следующие общие параметры: Р = 11, А = 7.

Один из пользователей этой системы связи хочет подписать свое сообщение m=5 цифровой подписью, сформированной по алгоритму Эль-Гамаля. Вначале он должен выбрать себе закрытый ключ, например, Х1=3 и сформировать открытый ключ Y1 = 73 mod 11 = 2. Открытый ключ может быть передан всем заинтересованным абонентам или помещен в базу данных открытых ключей системы связи.

Затем пользователь выбирает случайное секретное число k, взаимно простое с Р-1. Пусть k=9 ( 9 не имеет общих делителей с 10 ). Далее необходимо вычислить число

Описание: a=A^k \: mod \: P=7^9\: mod \: 11=8

После этого с помощью расширенного алгоритма Евклида находится значение b в уравнении:

Описание: m=(X_1 \times a + k \times b) \: mod \: (P-1),\\5=(3 \times 8 + 9 \times b) \: mod \: 10

Решением последнего уравнения будет значение b=9.

Таким образом, пара чисел (8, 9) будет цифровой подписью сообщения m=5.

Если любой другой пользователь сети желает проверить цифровую подпись в сообщении, он должен получить из базы данных открытый ключ первого пользователя (он равен 2 ), вычислить два числа с1 и с2 и сравнить их.

Описание: c_1-Y_1^a \times a^b \: mod \: P=2^8 \times 8^9 \: mod \: 11=10,\\c_2=A^m \: mod \: 11=10

Так как с1 = с2, то цифровая подпись первого пользователя в сообщения m=5 верная.

### Стандарты на алгоритмы цифровой подписи

#### Стандарт цифровой подписи DSS

Во многих странах сегодня существуют стандарты на *электронную (цифровую) подпись*. *Стандарт цифровой подписи* DSS (*Digital Signature Standard* – DSS) был принят в США в 1991 году и пересмотрен в 1994 году. В основе стандарта лежит алгоритм, называемый *DSA* (*Digital Signature Algorithm*) и являющийся вариацией подписи Эль-Гамаля. В алгоритме используется однонаправленная хеш-функция H(m). В качестве хэш-алгоритма стандарт DSS предусматривает использование алгоритма SHA-1.

Рассмотрим сам алгоритм генерации ЭЦП. Вначале для группы абонентов выбираются три общих (несекретных) параметра р, q и a:

* параметр р должен быть простым числом длиной от 512 до 1024 бит.
* q – простое число длиной 160 бит; между p и q должно выполняться соотношение p = bq + 1 для некоторого целого b. Старшие биты в р и q должны быть равны единице (таким образом 2159 < q < 2160 ).
* число а, удовлетворяющее неравенству 1 < a < p-1 и являющееся корнем уравнения aq mod p = 1.

Зная эти числа, каждый абонент системы случайно выбирает число х, удовлетворяющее неравенству 0 < х < q, и вычисляет

y = ax mod p.

Число х будет секретным ключом пользователя, а число у — открытым ключом. Вычислить у по известному х довольно просто. Однако, имея открытый ключ у, вычислительно невозможно определить х, который является *дискретным логарифмом* у по основанию a.

Предполагается, что открытые ключи всех пользователей указываются в некотором несекретном, но "сертифицированном" справочнике, который должен быть у всех, кто собирается проверять подписи. На этом этап выбора параметров заканчивается, и абоненты готовы к тому, чтобы формировать и проверять подписи.

Пусть имеется сообщение m, которое один из пользователей желает подписать. Для генерации подписи пользователь должен выполнить следующие действия:

1. Вычислить значение хеш-функции h = H(m) для сообщения m. Значение хеш-функции должно лежать в пределах 0 < h < q.
2. Затем сгенерировать случайное число k, 0 < k < q.
3. Вычислить r = (ak mod p) mod q.
4. Определить s = [k-1}(H(m) + x \* r)] mod q

В результате пользователь получит для сообщения m подпись, состоящую из пары чисел (r,s). Сообщение вместе с подписью может быть послано любому другому абоненту системы. Проверить подпись можно следующим образом:

1. Вычислить значение хеш-функции h = H(m) для сообщения m.
2. Проверить выполнение неравенств 0 < r < q, 0 < s < q.
3. Вычислить w = s-1 mod q ;
4. u1 = [H(m) \* w] mod q
5. u2 = r \* w mod q

v = [(аu1 \* yu2) mod p] mod q

1. Проверить выполнение равенства v = r. Если v = r, то подпись считается подлинной, иначе подпись считается недействительной.

В силу сложности вычисления дискретных логарифмов злоумышленник не может восстановить k из r или х из s, а следовательно, не может подделать подпись. По той же самой причине автор сообщения не сможет отказаться от своей подписи, так как никто кроме него не знает закрытого ключа х.

#### Стандарт цифровой подписи ГОСТ Р34.10-94

В России принят стандарт ГОСТ Р34.10-94 "Информационная технология. Криптографическая защита информации. Процедуры выработки и проверки электронной цифровой подписи на базе асимметричного криптографического алгоритма". В этом стандарте используется алгоритм, аналогичный алгоритму, реализованному в стандарте DSS. Рассмотрим вначале полностью алгоритм, описанный в ГОСТ Р34.10-94, а затем отметим его отличия от алгоритма *DSA*.

Вначале, так же как и по стандарту DSS, для группы абонентов выбираются три общих (несекретных) параметра р, q и a:

* параметр р должен быть простым числом длиной от 512 до 1024 бит. Старший бит в р должен быть равен единице.
* q – простое число длиной 254-256 бит; так же как и в *DSA*, q должно быть делителем числа (р-1). Старший бит в q должен быть равен единице.
* число а, удовлетворяющее неравенству 1 < a < p-1 и являющееся корнем уравнения aq mod p = 1.
* Затем каждый пользователь может сформировать закрытый и открытый ключи. В качестве закрытого ключа выбирается произвольное число х, 0 < x < q. Открытым ключом является число y, получаемое по формуле

y = аx mod p

Для создания каждой новой подписи каждый раз выбирается новое случайное число k, 0 < k < q.

Подпись сообщения m состоит из двух чисел (r, s), вычисляемых по следующим формулам:

r = (аk mod p) mod q,

s = (k \* H(m) + x \* r) mod q,

где H(m) – результат вычисления хеш-функции для сообщения m.

На этом формирование подписи закончено, и сообщение m вместе с ЭЦП (r,s) может быть отправлено получателю. Теперь отметим отличия алгоритма формирования ЭЦП по ГОСТ Р34.10-94 от алгоритма DSS.

1. Перед вычислением подписи исходное сообщение обрабатывается разными функциями хеширования: в ГОСТ Р34.10-94 применяется отечественный стандарт на хеш-функцию ГОСТ Р34.11-94, в DSS используется SHA-1, которые имеют разную длину хеш-кода. Отсюда и разные требования на длину простого числа q: в ГОСТ Р34.10-94 длина q должна быть от 254 до 256 бит, а в DSS длина q должна быть от 159 до 160 бит.

2. По-разному вычисляется компонента s подписи. В ГОСТ Р34.10-94 компонента s вычисляется по формуле

s = (k \* H(m) + x \* r) mod q,

а в DSS компонента s вычисляется по формуле

s = [k-1 (H(m) + x \* r)] mod q.

Последнее отличие приводит к соответствующим отличиям в формулах для проверки подписи.

В результате процедура проверки подписи по ГОСТ Р34.10-94 заключается в следующем. Получив [m, (r, s)], получатель вычисляет

w = H(m)-1mod q,

u1 = w \* s mod q,

u2 = (q-r) \* w mod q,

v = [(аu1 \* yu2) mod p] mod q.

Затем проверяется равенство вычисленного значения v и полученного в составе ЭЦП параметра r. Подпись считается корректной, если v = r.

В алгоритме создания ЭЦП по ГОСТ Р34.10-94, так же как и в алгоритме DSS, производятся достаточно сложные вычисления, требующие затрат вычислительных ресурсов. Для ускорения процесса генерации подписей по этим алгоритмам можно заранее вычислять некоторое количество значений параметра r, не зависящего от подписываемого сообщения. Затем эти значения можно использовать по мере необходимости для подписи документов. Для алгоритма DSS заранее может вычисляться и значение k-1.

#### Пример создания и проверки подписи по стандарту ГОСТ Р34.10-94

Пусть p = 23, q = 11, a =6 (проверяем: 611 mod 23 = 1 )

*Создание подписи*.

Предположим, пользователь А выбрал в качестве закрытого ключа число х=8. После этого он вычисляет открытый ключ по формуле y = аx mod p. То есть y = 68 mod 23 = 18.

Для создания подписи пользователь А выбирает случайное число k = 5.

Пусть результат вычисления хеш-функции для сообщения H(m) = 9.

Подпись сообщения состоит из двух чисел (r, s):

r = (аkmod p) mod q = (65 mod 23) mod 11 = 2,

s = (k\* H(m) + x \* r) mod q = (5 \* 9 + 8 \* 2) mod 11 = 6,

Таким образом, подпись сообщения состоит из пары чисел (2, 6).

*Проверка подписи*.

Получив сообщение вместе с подписью (2, 6), получатель вычисляет

w = H(m)-1mod q = H(m)-1mod q = 9-1mod 11 = 5,

u1 = w \* s mod q = 5 \* 6 mod 11 = 8,

u2 = (q-r) \* w mod q = (11-2) \* 5 mod 11 = 1,

v = [(аu1 \* yu2) mod p] mod q =[(68 \* 181) mod 23] mod 11 = 2

Так как v = r, то подпись считается верной.

Подписи, созданные с использованием стандартов ГОСТ Р34.10 или DSS, так же, как и подписи, полученные по алгоритму Эль-Гамаля, являются рандомизированными, так как для одинаковых сообщений с использованием одного и того же закрытого ключа х каждый раз будут создаваться разные подписи (r,s) благодаря использованию разных случайных значений k.

#### Новый отечественный стандарт ЭЦП

В 2001 г. был принят новый отечественный стандарт на алгоритм формирования и *проверки ЭЦП*. Его полное название следующее: "ГОСТ Р34.10-2001. Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи".

Данный алгоритм был разработан главным управлением безопасности связи Федерального агентства правительственной связи и информации при Президенте Российской Федерации при участии Всероссийского научно-исследовательского института стандартизации. Новый стандарт разрабатывался с целью обеспечения большей *стойкости алгоритма* генерации ЭЦП.

В основе ГОСТ Р34.10-2001 лежат алгоритмы с использованием операций на эллиптических кривых. Стойкость ГОСТ Р34.10-2001 основывается на сложности взятия *дискретного логарифма* в группе точек эллиптической кривой, а также на стойкости хэш-функции по ГОСТ Р34.11-94. Размер формируемой цифровой подписи – 512 бит.

В целом алгоритм вычислений по алгоритму ГОСТ Р34.10-2001 аналогичен применяемому в предыдущем стандарте ГОСТ Р34.10-94. Сначала генерируется случайное число k, с его помощью вычисляется компонента r подписи. Затем на основе компоненты r, числа k, значения секретного ключа и хэш-значения подписываемых данных формируется s-компонента ЭЦП. При проверке же подписи аналогичным образом проверяется соответствие определенным соотношениям r, s, открытого ключа и хэш-значения информации, подпись которой проверяется. Подпись считается неверной, если соотношения неверны.

Старый ГОСТ Р34.10-94 не отменен, и в настоящее время параллельно действуют два отечественных стандарта на ЭЦП. Однако необходимо отметить, что для прежнего ГОСТа принято ограничение: при реализации ЭЦП по стандарту ГОСТ Р34.10-94 разрешено использовать только 1024-битные значения параметра p.

Использование математического аппарата группы точек эллиптической кривой в новом ГОСТ Р34.10-2001 позволяет существенно сократить порядок модуля p без потери криптостойкости. Так, в стандарте указано, что длина числа р может быть 256 или больше бит.

### Симметричная или асимметричная криптография?

Однозначного ответа на вопрос о том, какие алгоритмы - симметричные или асимметричные - предпочтительнее, конечно же, нет. Основным достоинством симметричной криптографии является высокая скорость обработки данных. Проблемы криптосистем с закрытым ключом обсуждались подробно в лекции 8. Попробуем теперь оценить особенности алгоритмов шифрования с открытым ключом.

Главным достоинством асимметричной криптографии является отсутствие необходимости в предварительном доверенном обмене ключевыми элементами при организации секретного обмена сообщениями. К основным недостаткам асимметричных криптосистем, мешающим им вытеснить симметричные методы шифрования, относят следующие.

1. Алгоритмы с открытым ключом работают намного (в сотни раз) медленнее классических алгоритмов с закрытым ключом. Это их самый главный недостаток. Связан он с тем, что основной операцией в системах с открытым ключом является возведение в степень по большому модулю 500-1000 битовых чисел, что при программной реализации производится намного медленнее, чем шифрование того же объема данных классическими способами.
2. Алгоритмы с открытым ключом требуют обеспечения достоверности открытых ключей, что порой превращается в довольно сложную задачу. То же самое относится и к протоколам цифровой подписи. Для управления открытыми ключами используют специальную инфраструктуру открытых ключей, обеспечивающую функции управления открытыми ключами.
3. Алгоритмы с открытым ключом чувствительны к атакам по выбранному открытому тексту.

Таким образом, с практической точки зрения системы с открытым ключом и *асимметричным шифрованием* целесообразно использовать лишь для распределения секретных ключей и организации цифровых подписей, так как для решения этих задач не требуется шифрования больших блоков данных.

Использование *асимметричных алгоритмов* позволяет создавать сеансовые ключи шифрования, которые удаляются после окончания сеанса связи. Это значительно снижает риск вскрытия зашифрованных сообщений, так как, если каждое передаваемое сообщение шифруется уникальным сеансовым ключом, задача взломщика существенно усложняется. Причем пользователям совсем необязательно выполнять *протокол обмена ключом* перед симметричным шифрованием. Вот возможный вариант протокола передачи зашифрованных данных одновременно с передачей ключа.

1. Пользователь А генерирует случайный *сеансовый ключ* К и зашифровывает им с помощью *симметричного алгоритма* Fсим свое сообщение M:

Cт = Fсим(M, K)

1. Пользователь А получает из базы данных открытый ключ U пользователя Б и зашифровывает им *сеансовый ключ* К:

Ck = Fасим(К, U)

1. Пользователь А посылает своему абоненту зашифрованное сообщение Cт и зашифрованный *сеансовый ключ* Ck. Для защиты от вскрытия "человек-в-середине" передаваемые данные могут быть дополнены цифровой подписью.
2. Пользователь Б расшифровывает полученный *сеансовый ключ* Ck с помощью своего закрытого ключа R:

K = Fасим-1(Ck, R)

1. Пользователь Б расшифровывает сообщение с помощью сеансового ключа К:

M = Fсим-1 (Cт, K)

Такая *криптографическая система* называется *смешанной*, так как в ней используется и асимметричное, и *симметричное шифрование*. Смешанные криптосистемы широко применяются на практике: в банковских и платежных сетях передачи данных, в мобильной связи, в системах электронной почты и др. Для лучшего обеспечения безопасности они могут быть дополнены цифровыми подписями пользователей и удостоверяющего центра, метками времени. *Цифровая подпись* в сочетании с открытым распределением ключей позволяют организовывать защищенный обмен электронными документами.

### Управление открытыми ключами

Благодаря асимметричной криптографии проблема распределения секретных ключей, рассмотренная нами в лекции 8, была решена, вернее, ликвидирована, однако, появилась новая проблема – проблема подтверждения подлинности открытых ключей. Эта проблема заключается в том, что, получая *открытый ключ* некоторого абонента А, *пользователь* должен быть уверен, что *ключ* принадлежит именно абоненту А, а не кому-то другому. Дело в том, что в исходном виде система распределения ключей, предлагавшаяся Диффи и Хеллманом, давала возможность проведения различного рода атак, основанных на перехвате и подмене открытых ключей абонентов. Так, например, в лекции 11 нами была рассмотрена *атака* "человек-в-середине", позволявшая злоумышленнику осуществлять полный *контроль* над передаваемой в системе связи информацией. На практике возможны и другие более сложные варианты атак, связанные с подменой открытых ключей абонентов, отказом от закрытого ключа, дублированием сообщений и т.д.

Большую роль в решении проблемы *сертификации открытых ключей* сыграло создание цифровой подписи. В системах связи с большим количеством абонентов, применяющих асимметричные криптосистемы, стали использовать специальные организационные структуры, выполняющие функции управления ключами абонентов и занимающиеся сертификацией открытых ключей. Эти организационные структуры играют роль доверенной третьей стороны и заверяют открытые ключи абонентов своими цифровыми подписями. Таким образом, в распределенных системах связи, использующих криптосистемы с открытыми ключами, вводится понятие **инфраструктуры открытых ключей** (*Public* Key *Infrastructure* - *PKI*), включающей комплекс программно-аппаратных средств, а также организационно-технических и административных мероприятий, обеспечивающих абонентам системы связи необходимый сервис для управления их открытыми ключами.

Основным элементом *инфраструктуры открытых ключей* является **центр сертификации** ( *удостоверяющий центр* ) (*Certification authority*, *CA*), который обеспечивает *контроль* за выполнением всех процедур, связанных с изготовлением, регистрацией, хранением и обновлением ключей, **сертификатов открытых ключей** и списков отозванных сертификатов. Сертификат представляет собой информацию, заверенную цифровой подписью центра, и включающую *открытый ключ* и другие данные об абоненте. Такими данными являются, например, *идентификатор* алгоритма электронной подписи, имя удостоверяющего центра, срок годности сертификата, *имя пользователя*, которому принадлежит сертификат. *Международный стандарт* *ISO* X.509 определяет структуру сертификатов открытых ключей и правила их использования для аутентификации в распределенных системах связи.

Сертификат обладает следующими свойствами:

* каждый пользователь центра сертификации, имеющий доступ к открытому ключу центра, может извлечь открытый ключ, включенный в сертификат;
* ни одна сторона, помимо центра сертификации, не может изменить сертификат так, чтобы это не было обнаружено (сертификаты нельзя подделать).

Так как сертификаты не могут быть подделаны, их можно опубликовать, поместив в общедоступный справочник.

Каждый *пользователь* системы связи может быть владельцем одного или нескольких сертификатов, сформированных удостоверяющим центром. *Открытый ключ* абонента может быть извлечен из сертификата любым пользователем, знающим *открытый ключ* администратора удостоверяющего центра. В качестве администратора центра сертификации выступает обычно не физическое лицо (человек), а высокопроизводительная *автоматизированная система*.

В распределенных системах связи с большим числом абонентов может быть создано несколько центров сертификации. Центры сертификации объединяются в древовидную структуру, в корне которой находится главный *удостоверяющий центр*. Главный центр выдает сертификаты подчиненным ему центрам сертификации, тем самым заверяя открытые ключи этих центров.

*Открытый ключ* пользователя формируется на основе закрытого ключа. Каждый *пользователь* должен хранить свой закрытый *ключ* таким образом, чтобы никто другой не смог узнать его *значение*. Если же у владельца ключа есть основания полагать, что *ключ* стал известен кому-либо еще, то такой закрытый *ключ* считается *скомпрометированным*, и потерпевший, допустивший компрометацию своего ключа, должен оповестить всех остальных абонентов системы связи, что его *открытый ключ* следует считать недействительным.

Сертификаты открытых ключей имеют *период действия*, однако любой сертификат может быть выведен из обращения (отозван) до истечения этого периода, если соответствующий сертификату *секретный ключ* пользователя скомпрометирован или скомпрометирован *ключ* удостоверяющего центра, использованный при формировании сертификата. *Удостоверяющий центр* должен информировать своих абонентов об отозванных сертификатах. Для этой цели он поддерживает *список* отозванных сертификатов или *список* отмены.

При такой организации *администратор* удостоверяющего центра не имеет доступа к секретным ключам пользователей, а значит, и к защищаемой с их помощью информации. *Администратор* может лишь подменить в справочнике сертификатов открытые ключи одного из абонентов или включить фиктивного абонента, от его имени войти в контакт и получить предназначенное ему сообщение. Для исключения таких конфликтов может применяться следующая схема подготовки и рассылки ключей.

1. Администратор удостоверяющего центра генерирует пару (закрытый ключ, открытый ключ) и сообщает свой открытый ключ всем своим абонентам.
2. Пользователь А выбирает закрытые ключи для выполнения операций шифрования и формирования ЭЦП, а также вычисляет соответствующие открытые ключи.
3. Открытые ключи шифрования и подписи зашифровываются открытым ключом администратора и предъявляются в удостоверяющий центр для регистрации.
4. Администратор удостоверяющего центра проверяет (расшифровывает своим закрытым ключом) открытые ключи пользователя А; изготавливает и подписывает сертификаты открытых ключей пользователя А и помещает их в справочники открытых ключей шифрования и открытых ключей подписей. Каждый из справочников предоставляется в распоряжение абонентов удостоверяющего центра.
5. Любой пользователь системы может извлечь из справочника сертификат необходимого абонента, проверить подпись администратора под сертификатом (расшифровать его открытым ключом администратора) и извлечь открытый ключ.

Такая схема подготовки и распределения открытых ключей выглядит несколько тяжеловесной, однако она защищает абонентов системы связи от разнообразных конфликтных ситуаций. На практике рассмотренная схема дополняется метками времени в цифровых подписях, проверками дополнительных полей в сертификатах (например, срока действия) и другими проверками, повышающими *безопасность* функционирования всей системы в целом.

Необходимо отметить, что в настоящее время в связи с широким использованием асимметричных криптоалгоритмов для банковских, платежных и других систем *инфраструктура* открытых ключей постоянно совершенствуется.

### Ключевые термины

**DSS** (*Digital Signature Standard*) – стандарт США на цифровую подпись. В основе стандарта лежит *алгоритм*, называемый *DSA* (*Digital Signature Algorithm*) и являющийся вариацией подписи Эль-Гамаля.

**ГОСТ Р34.10-2001** – новый российский стандарт на *алгоритм* формирования и *проверки ЭЦП*. Основан на сложности взятия *дискретного логарифма* в группе точек эллиптической кривой, а также на стойкости хэш-функции по ГОСТ Р34.11-94. Размер формируемой цифровой подписи – 512 *бит*.

**ГОСТ Р34.10-94** – российский стандарт на *алгоритм* формирования и *проверки ЭЦП*, действующий с 1995 года. В стандарте используется модификация схемы шифрования с открытым ключом Эль-Гамаля и *алгоритм* выработки хэш-функции по ГОСТ Р34.11-94.

**Инфраструктура открытых ключей** – комплекс программно-аппаратных средств, организационно-технических и административных мероприятий, обеспечивающих абонентам системы связи необходимый сервис для управления их открытыми ключами.

**Сертификат открытого ключа** – *информация*, заверенная цифровой подписью центра, и включающая *открытый ключ* и другие данные об абоненте (*идентификатор* алгоритма электронной подписи, имя удостоверяющего центра, срок годности сертификата, *имя пользователя*, которому принадлежит сертификат и др).

**Центр сертификации** – организация или подразделение организации, которая выпускает сертификаты ключей электронной цифровой подписи и отвечает за управление ключами пользователей. Открытые ключи и другая *информация* о пользователях хранится удостоверяющими центрами в виде цифровых сертификатов.

### Краткие итоги

Для формирования электронной цифровой подписи могут быть применены алгоритмы *RSA*, Эль-Гамаля и другие. Во многих странах сегодня существуют стандарты на электронную цифровую подпись. Так, в Российской Федерации действуют стандарты на алгоритмы формирования и *проверки ЭЦП* ГОСТ Р34.10-94 и ГОСТ Р34.10-2001.

В системах связи с большим количеством абонентов, применяющих асимметричные криптосистемы, стали использовать специальные организационные структуры, выполняющие функции управления ключами абонентов и занимающиеся сертификацией открытых ключей. Эти организационные структуры играют роль доверенной третьей стороны и заверяют открытые ключи абонентов своими цифровыми подписями. Основным элементом *инфраструктуры открытых ключей* является *центр сертификации* (*удостоверяющий центр*), который обеспечивает *контроль* за выполнением всех процедур, связанных с изготовлением, регистрацией, хранением и обновлением ключей, сертификатов открытых ключей и списков отозванных сертификатов.

*Сертификат открытого ключа* представляет собой информацию, заверенную цифровой подписью центра, и включающую *открытый ключ* и другие данные об абоненте.

В целом, системы с открытым ключом и *асимметричным шифрованием* целесообразно использовать лишь для распределения секретных ключей и организации цифровых подписей, так как для решения этих задач не требуется шифрования больших блоков данных. *Шифрование* больших массивов данных обычно выполняется симметричными алгоритмами, скорость работы которых значительно выше, чем скорость алгоритмов с открытым ключом. Такие криптографические системы называются смешанными, так как в них используется и асимметричное, и *симметричное шифрование*. Смешанные криптосистемы широко применяются на практике: в банковских и платежных сетях передачи данных, в мобильной связи, в системах электронной почты и др. Для лучшего обеспечения безопасности они могут быть дополнены цифровыми подписями пользователей и удостоверяющего центра, метками времени. *Цифровая подпись* в сочетании с открытым распределением ключей позволяют организовывать защищенный обмен электронными документами.

#### Вопросы для самопроверки

1. Какие асимметричные алгоритмы и как могут применяться для формирования и проверки электронной цифровой подписи?
2. Опишите процесс создания и проверки цифровой подписи с использованием различных *асимметричных алгоритмов*.
3. Какие стандарты действуют на алгоритмы формирования и проверки электронной цифровой подписи в России?
4. Какие цифровые подписи называются рандомизированными?
5. В чем заключается проблема сертификации открытых ключей?
6. Что включается в понятие инфраструктуры открытых ключей?
7. Каковы функции центра сертификации открытых ключей?
8. Что такое сертификат открытого ключа?
9. Какая схема распределения открытых ключей абонентов может использоваться в системе связи, имеющей в своем составе центр сертификации открытых ключей?