Защита почтовых сообщений и коммуникаций

Криптографические Методы Защиты

Позволяют решать следующие задачи:

- закрытие (шифрование) данных при хранении или передаче по каналам связи
- контроль целостности данных при хранении или передаче по каналам связи
- аутентификация абонентов (взаимодействующих сторон)
- разграничение ответственности сторон за счет обеспечения неотказуемости (доказательства авторства или источника сообщения)

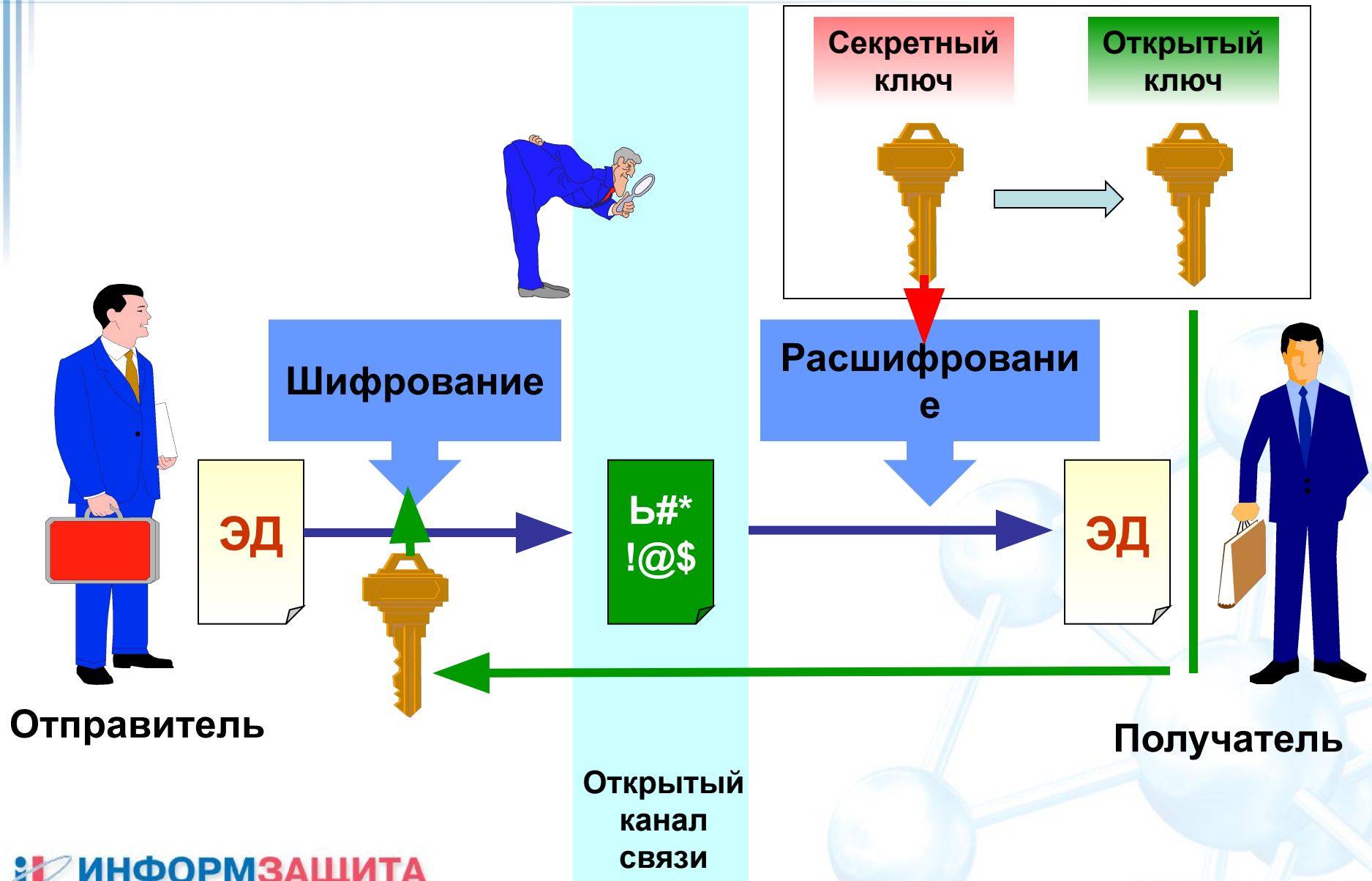
Криптографические методы защиты

Основные криптографические схемы:

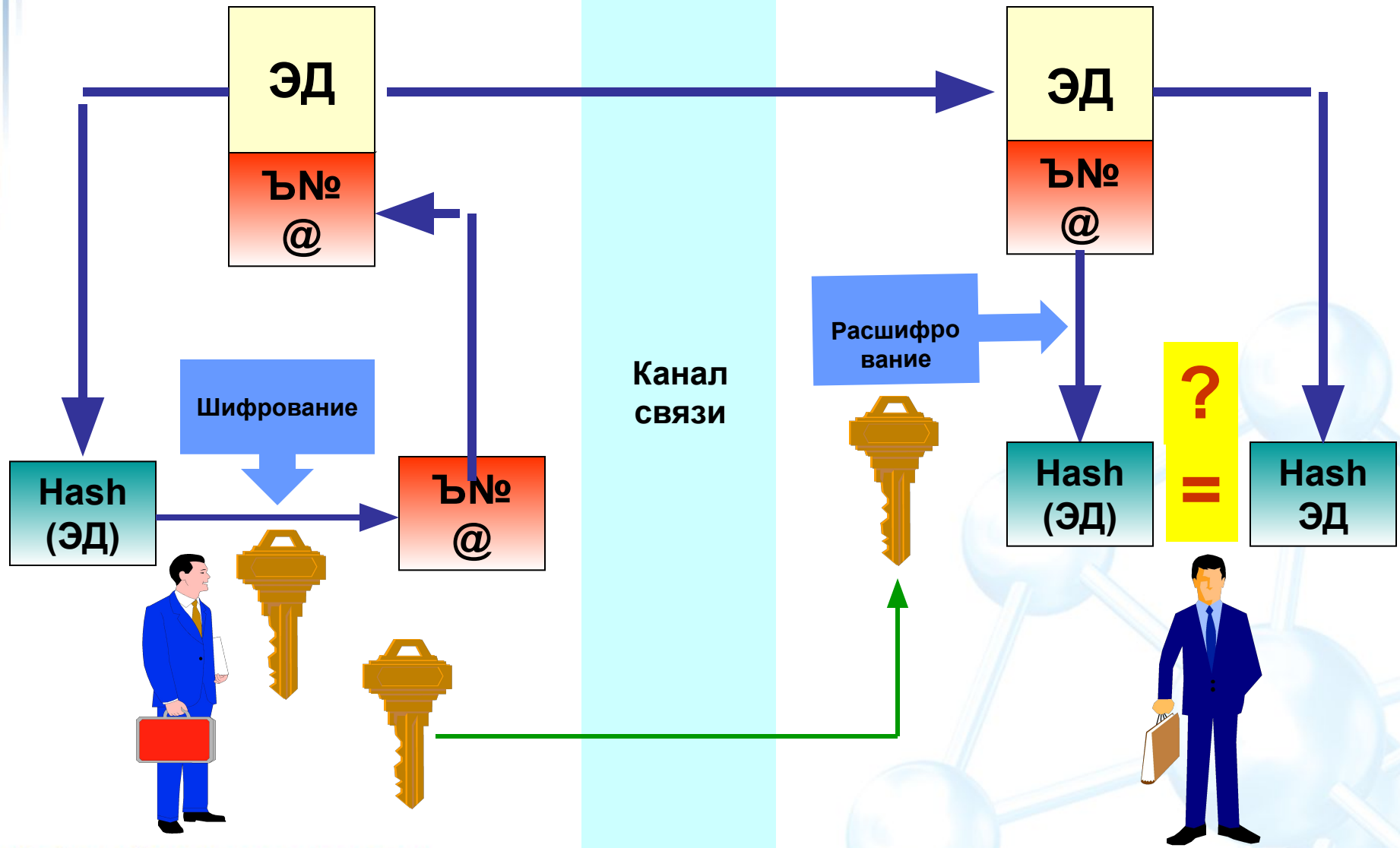
- Симметричное шифрование данных
- Асимметричное шифрование данных
- Комбинированные схемы шифрования
- Электронная цифровая подпись

Симметричное шифрование

Асимметричное шифрование



Электронная цифровая подпись



Криптографические алгоритмы

- **Симметричное шифрование**
  - Data Encryption Standard (DES)
    - DES: 56 бит
    - DESX: 128 бит
    - Triple DES: 112 бит, 168 бит
  - Rivest's Cipher (RC)
    - RC2, RC4: 40 бит, 56 бит, 128 бит
  - ГОСТ 28147-89
    - 256 бит
- **Хеширование**
  - Message Digest (MD)
    - MD2, MD4, MD5
  - Secure Hash Algorithm (SHA-1)
  - Hashed Message Authentication Code (HMAC)
  - ГОСТ Р 34.11-94
- **Цифровая подпись**
  - Digital Signature Algorithm (DSA)
  - RSA Digital Signature
  - ГОСТ Р 34.10-94 (2001)
    - 256 и 512 (1024) бит
- **Обмен ключами**
  - Diffie-Hellman Key Agreement
  - RSA Key Exchange

Три способа аутентификации (доказательства прав)

1. Знать что-либо, чего не знают другие
   (секретный пароль или ключ)
2. Иметь что-либо, чего нет у других
   (что сложно отнять или передать)
3. Иметь рекомендации от доверенного посредника
   (которому верит проверяющий)

Третий способ аутентификации

3. Иметь рекомендации от доверенного посредника (которому верит проверяющий)

- Версия сертификата, серийный номер и алгоритм подписи
- Имя уполномоченного по выпуску сертификатов
- Информация о держателе сертификатов: имя, организация, адрес
- Открытый ключ держателя
- Расширения X.509 версии 3
- Цифровая подпись выдавшего сертификат уполномоченного

Концепция PKI

В 1978 году Кохфельдер (Kohnfelder, Loren M.) предложил использовать сертификаты для распространения открытых ключей с тем, чтобы их аутентичность можно было проверить. Фактически он предложил концепцию инфраструктуры открытых ключей.

Назначение PKI - облегчение использования криптографии с открытым ключом посредством создания и распространения сертификатов открытых ключей и списков отозванных сертификатов

Сертификат - решение проблемы доверия

X509 -промышленный стандарт сертификатов

Сертификат может выпущен только уполномоченным эмитентом (CA) и содержит единственную ЭЦП эмитента

- **Номер версии X.509**
- **Открытый ключ пользователя**
- **Серийный номер сертификата**
- **Идентификатор пользователя**
- **Период действия сертификата**
- **Идентификатор издателя**
- **ЭЦП издателя**
- **Идентификатор алгоритма ЭЦП**

В настоящее время наиболее часто используются сертификаты на основе стандарта Международного союза телекоммуникаций ITU-T X.509 версии 3 и рекомендаций IETF (Internet Engineering Task Force) RFC 2459

Проблема отзыва сертификатов

Список отозванных сертификатов

Минимизировать возможный ущерб при компрометации индивидуального (секретного) ключа можно путем запрета его дальнейшего использования.
Но кроме этого необходимо запретить использование всех копий соответствующего открытого ключа, то есть отозвать сертификат, сообщив всем его потенциальным пользователям, что верить ему больше нельзя.

Certificate Revocation List (CRL) – подписанный набор записей, соответствующий отозванным открытым ключам, где каждая запись указывает серийный номер соответствующего сертификата, время отзыва, причину отзыва и др.

Сертификат – не воробей, -
издашь не отзовешь !
(Пословица нового времени)

Инфраструктура открытых ключей

Орган сертификации

Орган сертификации (**Certification Authority, CA**) – основной компонент PKI

CA - доверенная третья сторона, чья подпись под сертификатом подтверждает подлинность связи открытого ключа с представляемым объектом (стороной).

CA выполняет четыре основных функции PKI:

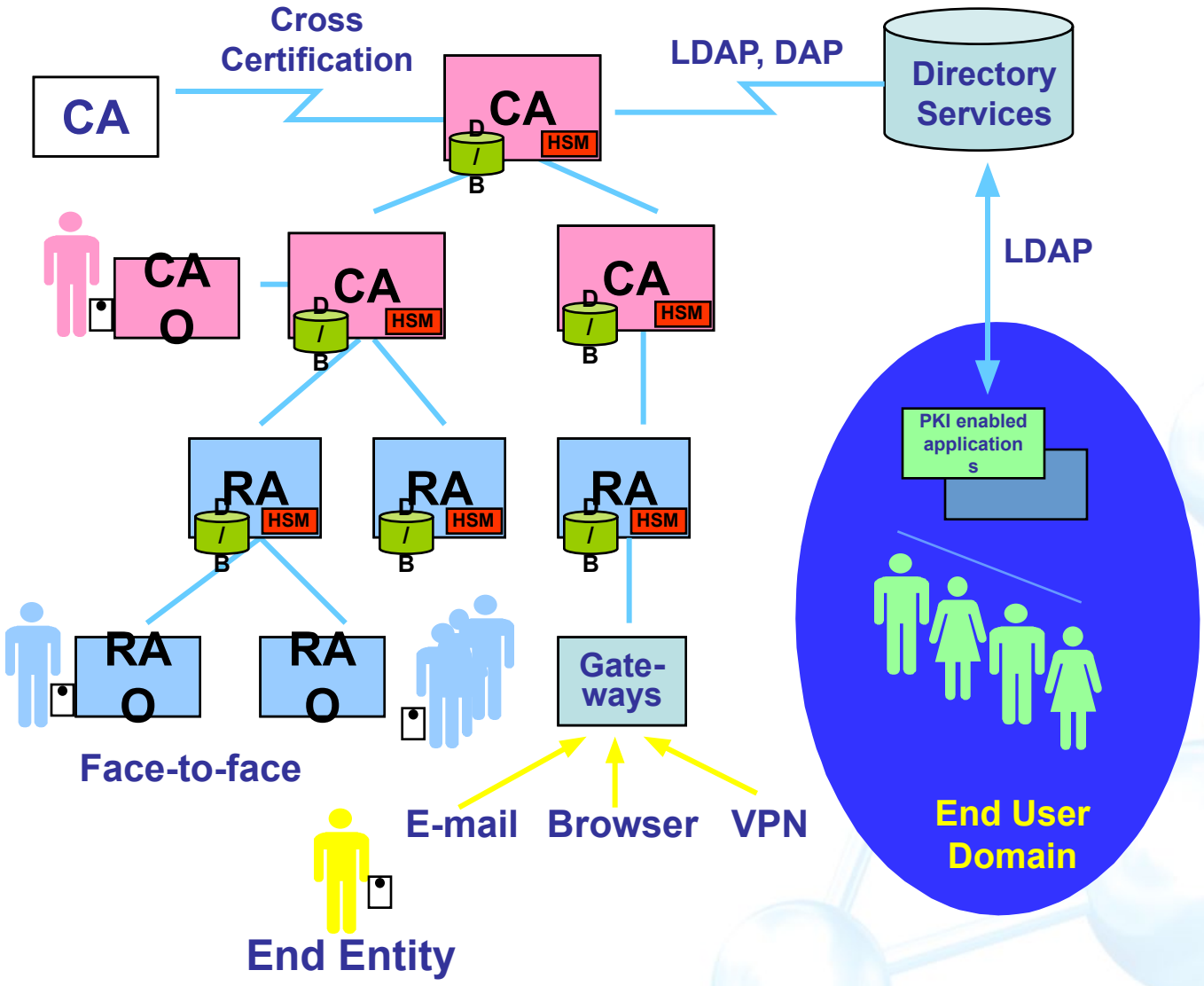
- **Издает (создает и подписывает) сертификаты**
- **Поддерживает информацию о статусе сертификатов и издает CRL**
- **Публикует свои и изданные им сертификаты и CRL**
- **Архивирует информацию об истекших или отозванных сертификатах**

Делегирование обязанностей CA

Пять основных функциональных компонентов PKI:

1. Орган сертификации (Certification Authority, CA)
2. Орган регистрации (Registration Authority - RA) проверяет контекст сертификатов
3. Хранилище (repository) распространяет сертификаты и CRL
4. Архив (archive) хранит истекшие сертификаты
5. Клиентское ПО реализует криптографические услуги для владельцев и пользователей ключей и сертификатов

Инфраструктура открытых ключей



Microsoft PKI

Microsoft Certificate Authority

- Два класса CA (Центров Сертификации)
  - Enterprise CA (предприятия)
  - Stand-Alone CA (автономный)
- Два типа CA (в иерархии)
  - Root CA (корневой)
  - Subordinate CA (подчиненный)

Enterprise и Standalone CA

Enterprise CA

- Целесообразно использовать во внутренней сети организации, так как он автоматически (прозрачно для объектов домена) выполняет большую часть работы

Standalone CA

- Целесообразно использовать в качестве корневого и промежуточных CA, так как в целях более надежной защиты он может быть изолирован от сети
- Может быть использован в качестве издающего CA, размещенного в DMZ (откуда нет доступа к AD)

Практическая работа 19

Установка служб сертификации Microsoft Certificate Services

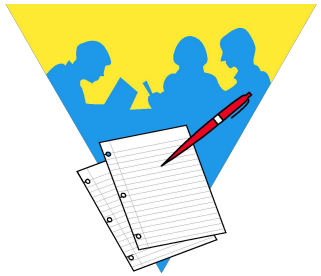
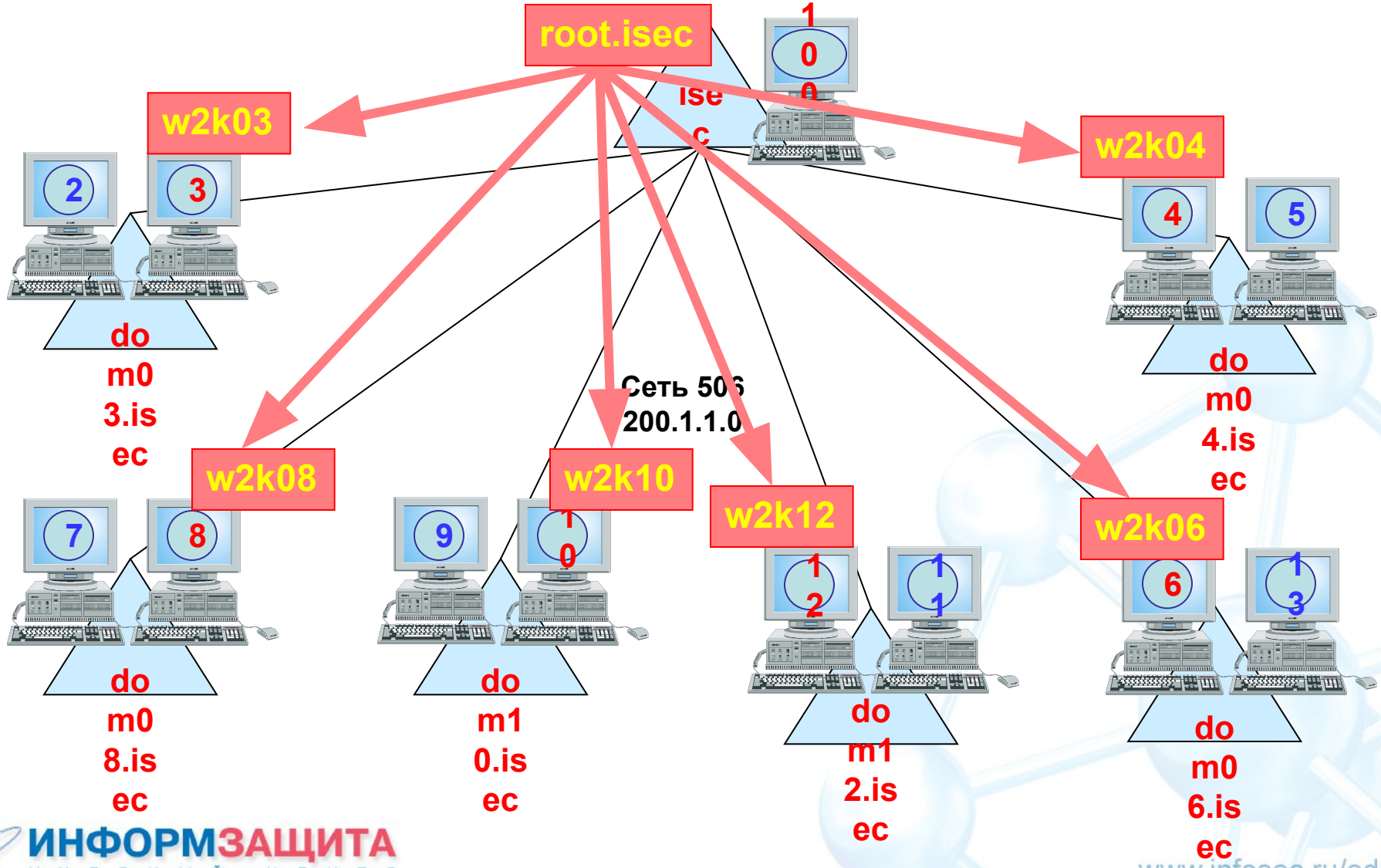


Схема сети класса



Управление CA

Папки в оснастке для каждого Certification Authority:

- **Revoked Certificates** («отозванные сертификаты») - список всех отозванных на данный момент сертификатов (CRL)
- **Issued Certificates** («изданные сертификаты») - список сертификатов, изданных данным CA
- **Pending Requests** («ожидающие запросы») - показывает «ожидающие» запросы, то есть запросы, оставленные CA в состоянии ожидания вашего решения
- **Failed Requests** («неудавшиеся запросы») - список неудавшихся или отвергнутых запросов
- **Policy Settings** («установки политики») - список шаблонов сертификатов, доступных на данном сервере

Шаблоны сертификатов

- Определяют фиксированные наборы атрибутов и расширений, которые будет иметь выдаваемый сертификат
- Windows 2000 поддерживает 19 шаблонов, позволяющих легко изготовлять сертификаты для конкретных задач

Примеры шаблонов сертификатов:

Шаблон	Назначение шаблона	Получатели сертификатов
Компьютер	Аутентифицирует клиентский компьютер по отношению к серверу и наоборот	Компьютеры, входящие в состав доменов
Пользователь	• Аутентифицирует сообщения от клиента к серверу • Подписывает и шифрует электронную почту • Шифрует данные EFS	Одиночные пользователи, не имеющие других специальных полномочий

Защита сообщений с использованием S/MIME

Что такое защита сообщений?

- Основная защита сообщений
  - Аутентификация источника данных
  - Конфиденциальность
  - Целостность
  - Неотказуемость с доказательством источника
- Дополнительные услуги
  - Подписывание квитанции о приеме: неотказуемость доставки
  - Метки безопасности
  - Защищенные перечни рассылки

Схемы защиты почтовых сообщений

- PEM: privacy enhanced mail
- PGP: pretty good privacy
- S/MIME: secure MIME

История и разработка

- После разработки PEM и в параллель с разработкой MOSS рабочая группа руководимая RSA Security, Inc. приступила к разработке другой спецификации для передачи цифровым образом подписанных и/или зашифрованных (в “конверте”) сообщений в соответствии с форматом сообщений MIME и некоторыми ранее опубликованными стандартами (PKCS)
- Подход и спецификация протокола была названа **Secure Multipurpose Internet Mail Extensions (S/MIME)**
- Имеется три версии S/MIME:
  - S/MIME версии 1 была специфицирована и официально опубликована в 1995 году RSA Security, Inc.
  - S/MIME версии 2 был специфицирован парой RFC - RFC 2311 и RFC 2312 – в марте 1998
  - Работы продолжавшиеся в IETF S/MIME Mail Security (SMIME) WG дали в результате S/MIME версии 3 специфицированной в RFC от 2630 до 2634 в июне 1999

Используемые стандарты

- Цель S/MIME защитить объекты MIME
- Объект MIME, в свою очередь, может быть частью сообщения, множеством частей, или полным сообщением e-mail
- Во всех случаях, S/MIME определяет как криптографически защитить объект MIME
- S/MIME основывается на синтаксисе криптографического сообщения (cryptographic message syntax, CMS) определенного в RFC 2630
- CMS, в свою очередь, получен из PKCS #7 версии 1.5 определенного в RFC 2315
- CMS величины генерируются с использованием ASN.1 и кодируются как байтовые строки согласно BER.

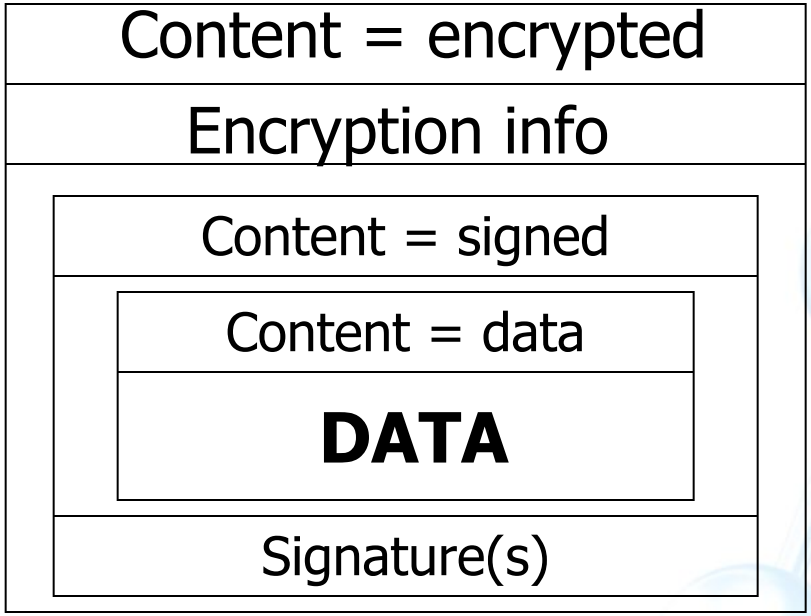
Синтаксис S/MIME

- В RFC 2630 определен только один тип защиты контекста
- Цель этого **ContentInfo** типа – инкапсулировать определенный тип контекста, и этот определенный тип контекста может обеспечить дальнейшую инкапсуляцию
- RFC 2630 определяет шесть типов контекста:
  - **Data**
  - **SignedData**
  - **EnvelopedData**
  - **DigestedData**
  - **EncryptedData**
  - **AuthenticatedData**

Синтаксис S/MIME

- Формат Type + value

- Контексты произвольно вкладываются

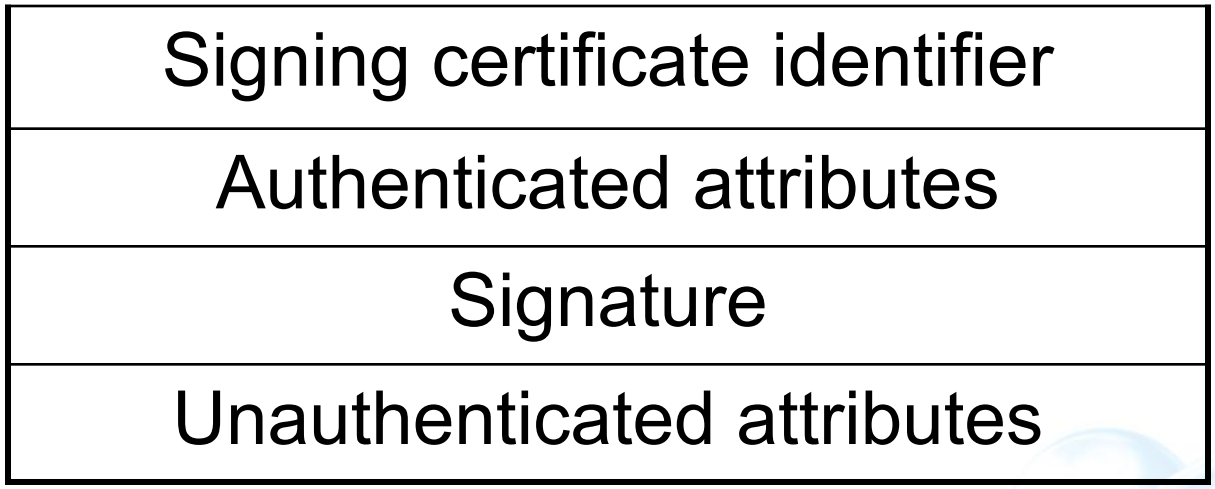


Формат Signed Data

Digest (hash) algorithm(s)
Encapsulated data
Signer certificate chain(s) Signature(s)

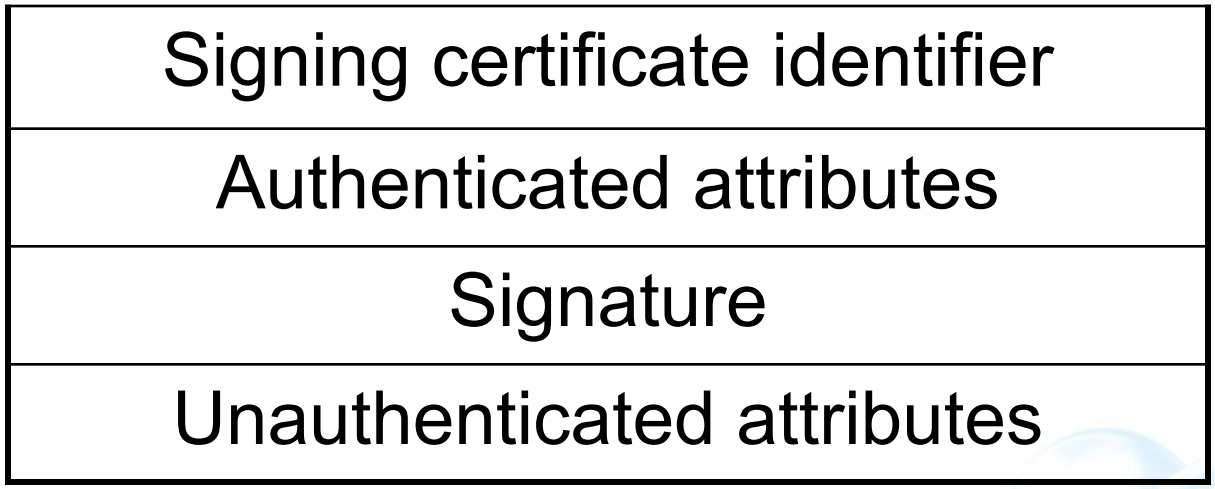
- Однопроходная обработка

Формат Signature



- Используются неаутентифицированные атрибуты для предоставления дополнительной информации
- Поддержка множественных подписей

Формат Signature



- Используются неаутентифицированные атрибуты для предоставления дополнительной информации
- Поддержка множественных подписей

Формат Enveloped Data

Per-recipient information
Key management certificate identifier Encrypted session key

- Поддержка заранее распределенных ключей
- Поддержка алгоритмов согласования ключей
- Поддержка алгоритмов транспортирования ключей

Поддерживаемые типы MIME

- Application/pkcs7-mime тип MIME с параметром smime-type установленным в signed-data
- Поддержка типов MIME: multipart/signed и application/pkcs7-signature

Процесс обработки S/MIME

Согласно RFC 2633, процесс отправления криптографически защищенного сообщения S/MIME состоит в следующем:

- Объект MIME подготавливается согласно обычных правил подготовки сообщения для MIME
- Получающийся MIME объект преобразуется в каноническую форму (детали процедуры канонизации зависят от фактического типа и подтипа MIME)
- MIME объект плюс некоторая относящаяся к защите сообщения информация, такая как идентификаторы алгоритмов или сертификаты, обрабатывается S/MIME и получается PKCS объект
- PKCS объект интерпретируется как контекст сообщения и заключается в оболочку MIME (в начале могут быть добавлены дополнительные MIME заголовки)
- Получившееся сообщение передается намечаемому получателю (лям)

Зашифрованный объект MIME

From: ...
To: ...
Subject: ...

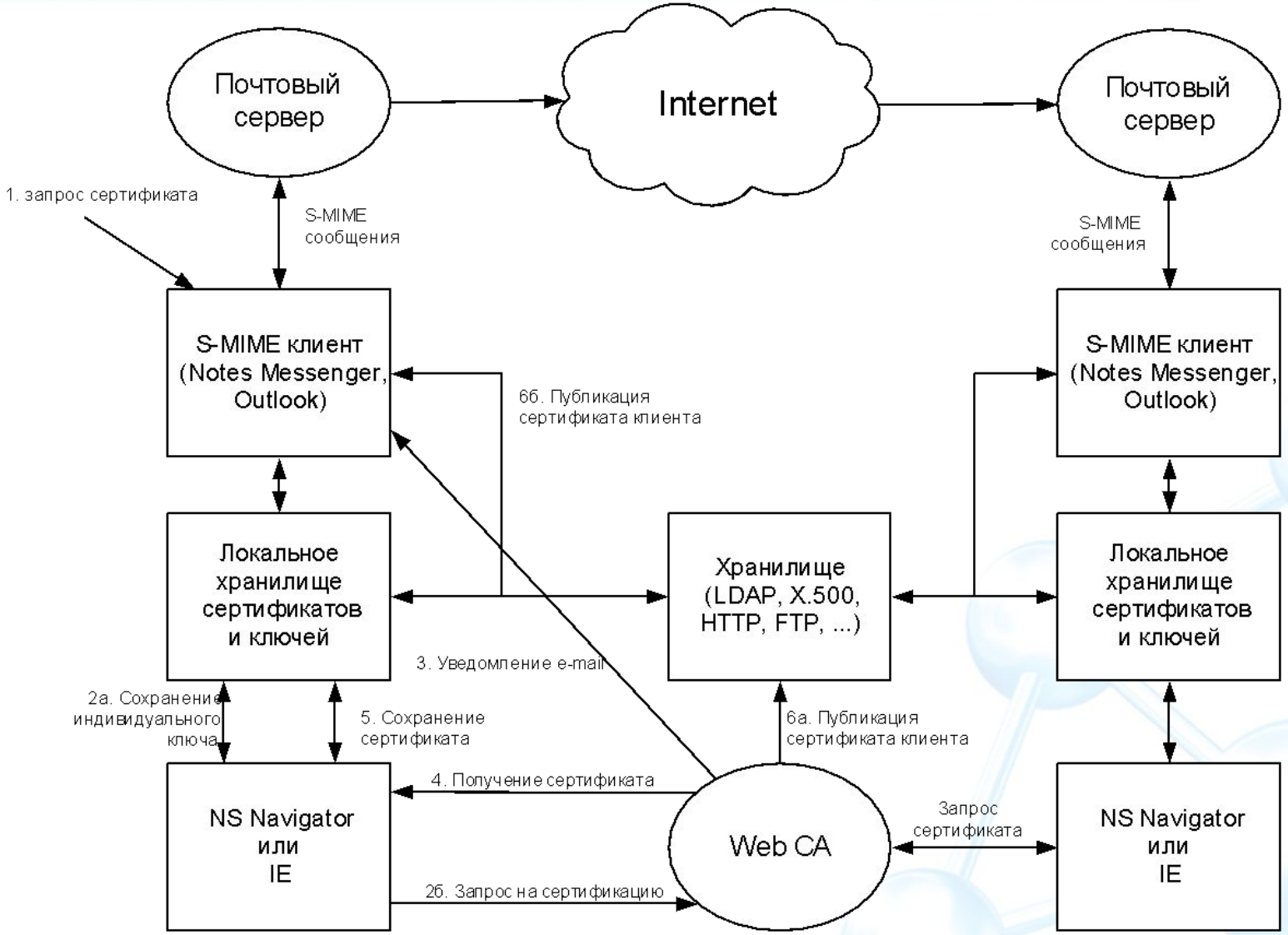
MIME-Version: 1.0
Content-Type: application/pkcs7-mime;
smime-type=enveloped-data;
name="smime.p7m"

Content-Transfer-Encoding: base64
Content-Disposition: attachment;
filename="smime.p7m
…

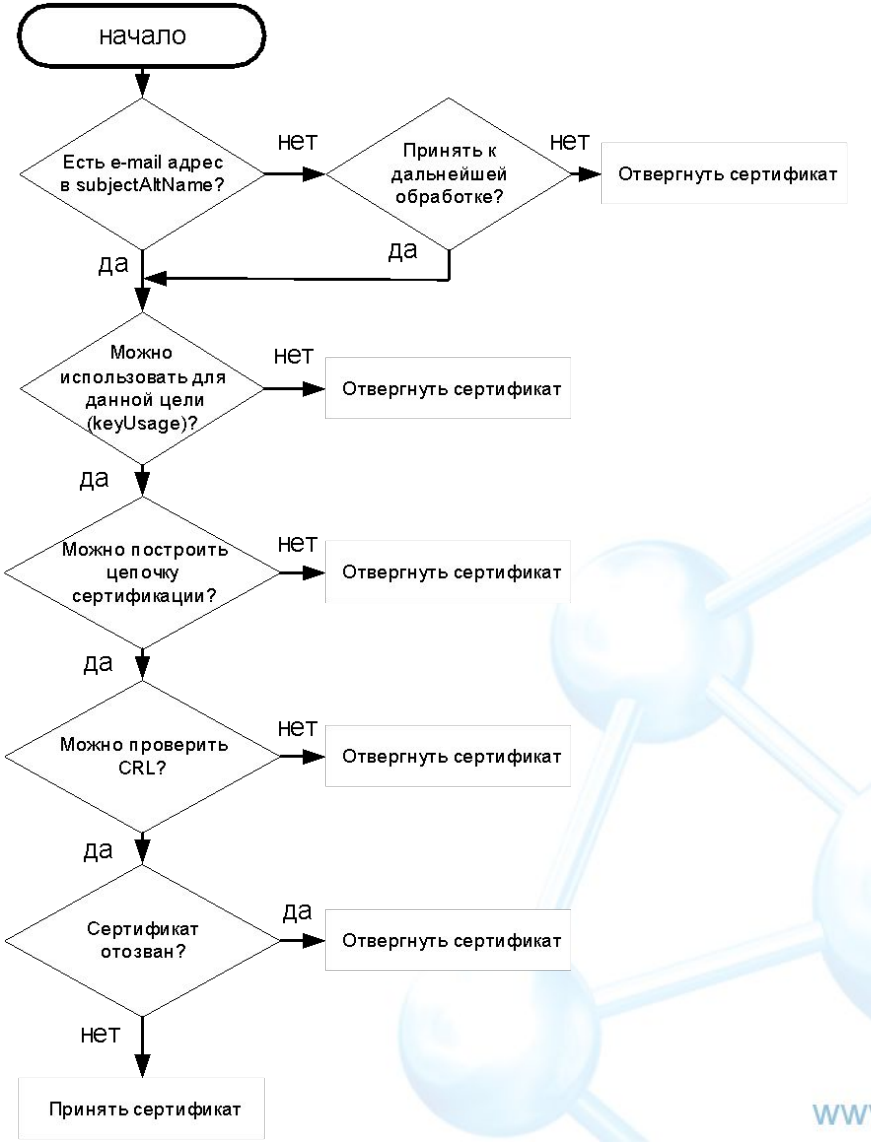
Криптоалгоритмы

- Криптографические алгоритмы:
  - S/MIME v2 требует поддержки для SHA-1, MD5, 40-битного RC2, и RSA (U.S. Версии дополнительно реализуют DES и 3DES)
  - S/MIME v3 требует дополнительной поддержки для DES, 3DES, 128-битного RC2, DH и DSA
- С криптографической точки зрения, алгоритмы используемые S/MIME идентичны или совместимы с используемыми PGP (и OpenPGP)

Процесс защищенного почтового обмена



Процедура построения и проверки цепочки сертификации по RFC 2632



S/MIME против PGP

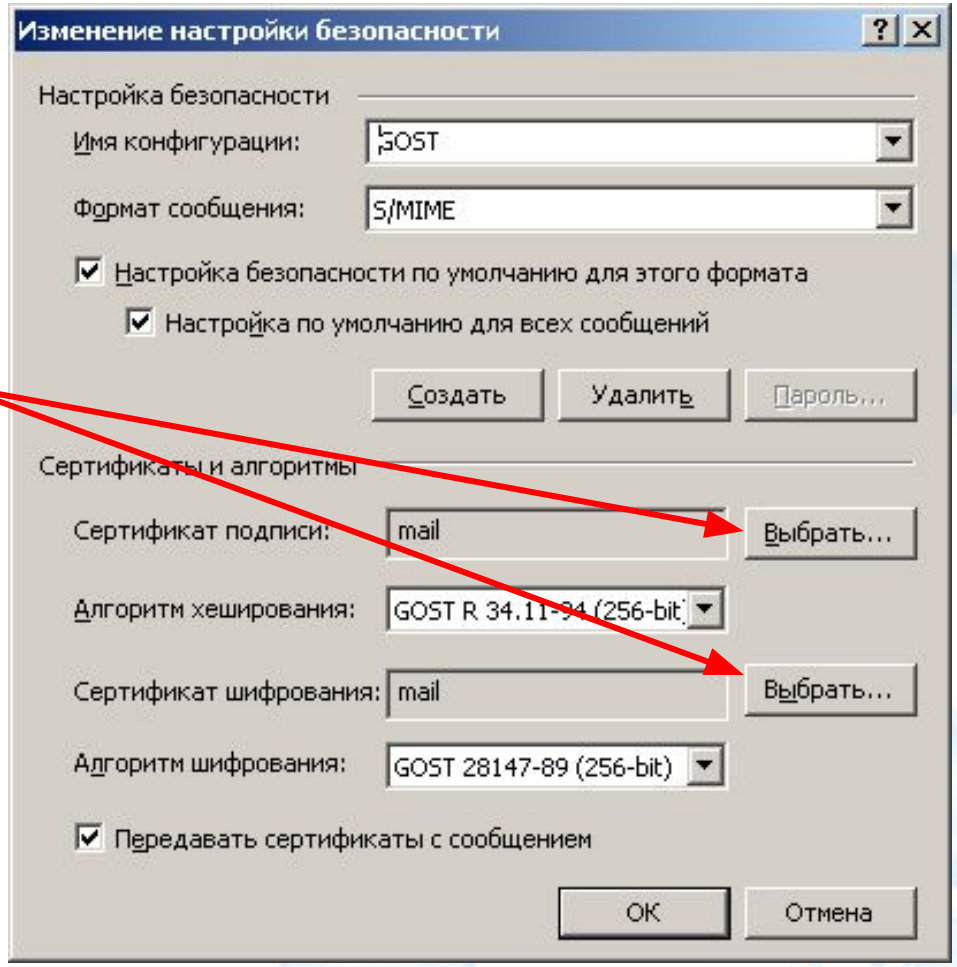
Обязательные возможности	S/MIME v3	OpenPGP
Формат сообщения	Двоичный, основанный на CMS	Двоичный, основанный на previous PGP
Формат сертификата	Двоичный, основанный на X.509v3	Двоичный, основанный на ранних версиях PGP
Алгоритмы симметричного шифрования	TripleDES (DES EDE3 CBC)	TripleDES (DES EDE3 CFB)
Алгоритмы подписи	DH (X9.42) с DSA	ElGamal (DH) с DSA
Алгоритм хэширования	SHA-1	SHA-1
MIME инкапсуляция подписанных данных	Выбор формата multipart/signed или CMS	multipart/signed с защитной оболочкой ASCII
MIME инкапсуляция зашифрованных данных	application/pkcs7-mime	multipart/encrypted

S/MIME против PGP

- PGP: сетевая модель
  - Сертификат может быть подписан несколько раз (различными объектами)
  - Личное доверие (установление степени доверия)
- S/MIME: Модель доверия не определена
  - Реализация почти любой модели доверия
  - Сертификаты X.509
  - Сертификат подписывается один раз

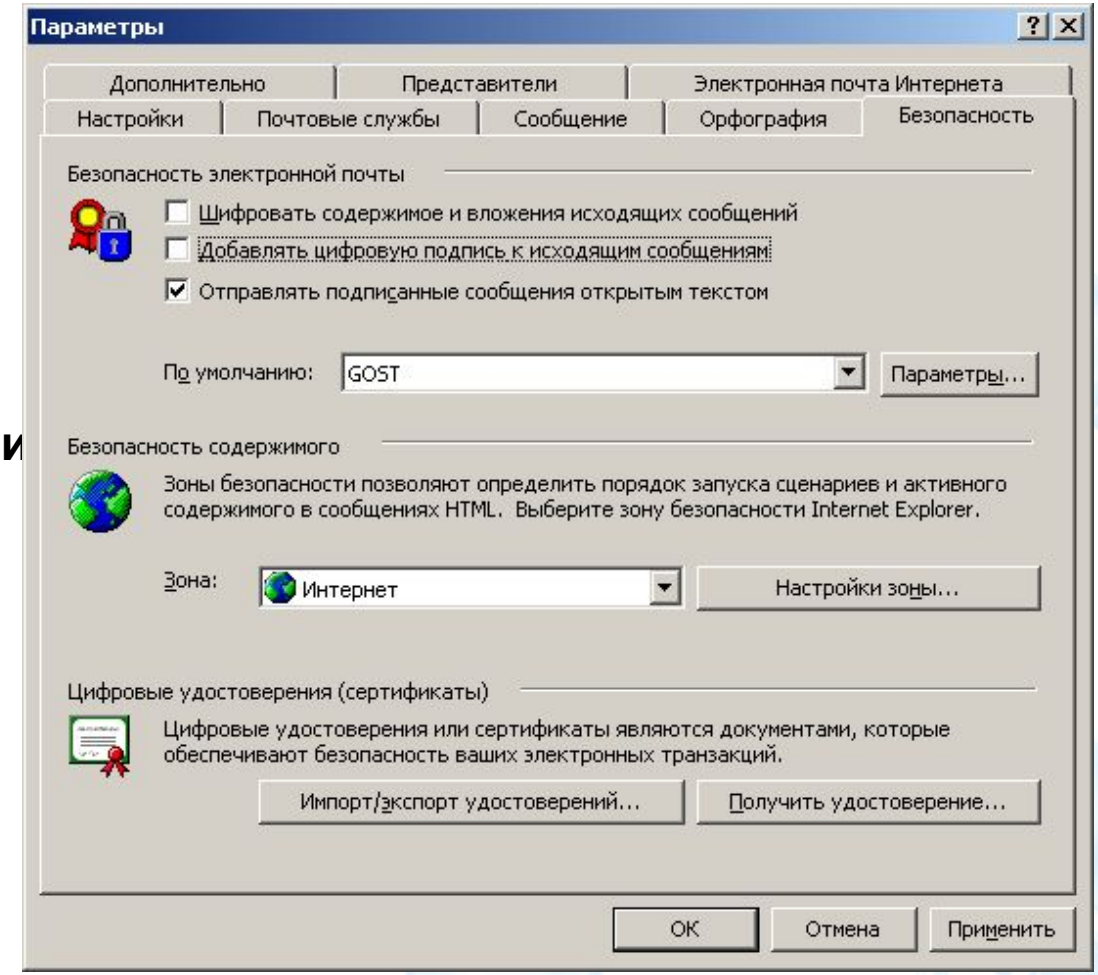
Конфигурация Outlook

1. Выберите пункт меню **Сервис, Параметры** и нажмите на закладку **Безопасность**. Нажмите кнопку **Параметры**.
2. Используя кнопки **Выбрать**, выберите личные сертификаты, соответствующие ключам подписи и шифрования.



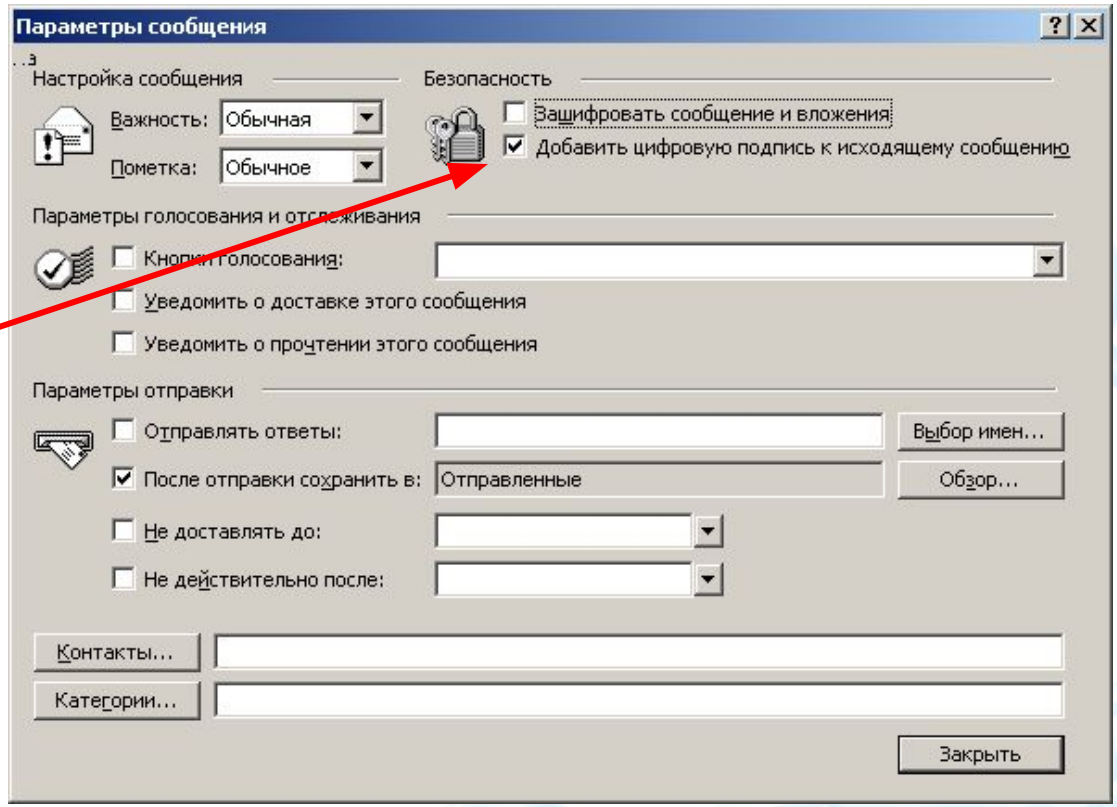
Конфигурация Outlook

3. Выберите пункт меню Сервис, Параметры и нажмите на закладку Безопасность.
4. В отображаемом диалоге можно включить режимы **Шифровать содержимое и вложения исходящих сообщений** и **Добавлять цифровую подпись к исходящим сообщениям**



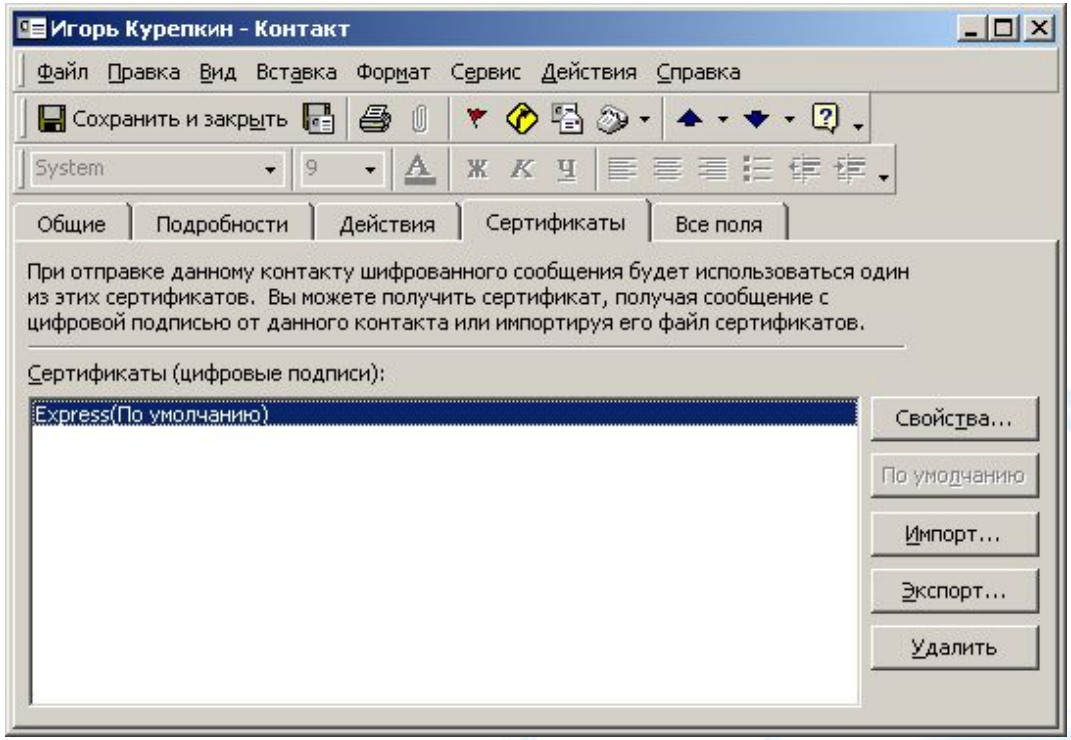
Отправка подписанных сообщений в Outlook

1. В окне **Сообщение** нажмите кнопку **Параметры**.
2. В появившемся окне диалога установите флаг **Добавить цифровую подпись к исходящему сообщению**.
3. После того, как сообщение подготовлено к отправке, нажмите кнопку **Отправить**.



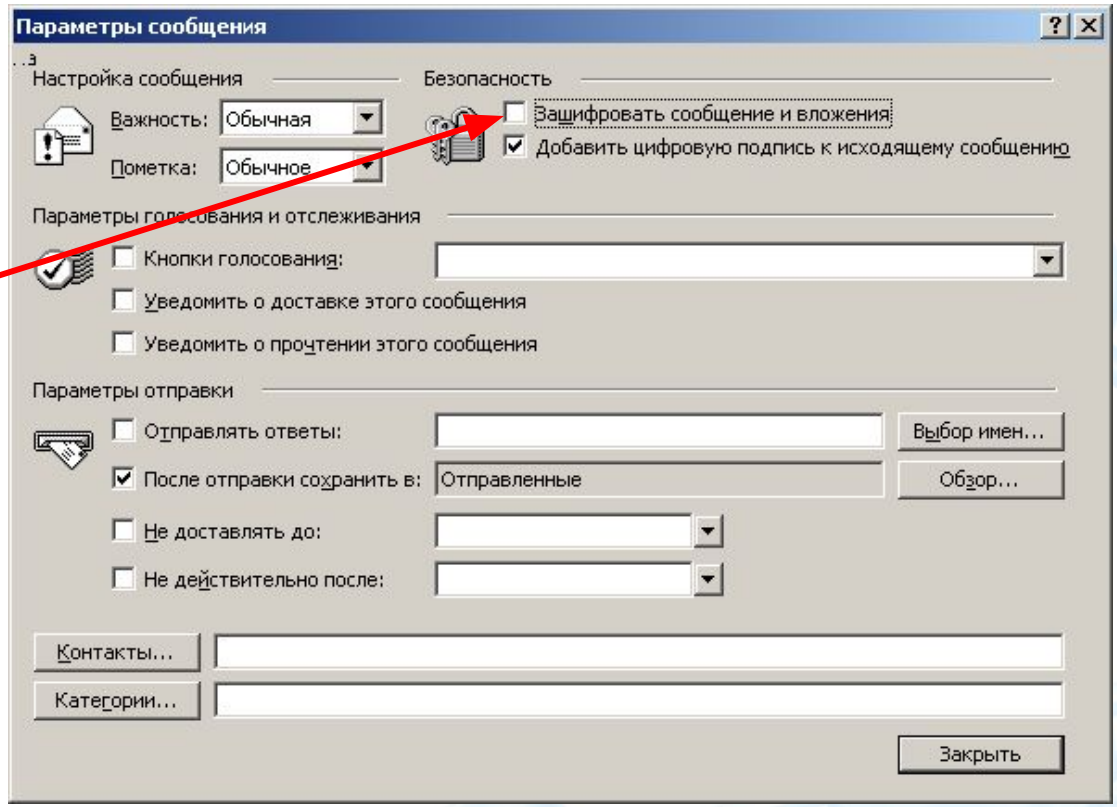
Получение сертификата для шифрования сообщений в Outlook

1. Откройте полученное подписанное письмо.
2. Установите курсор на адрес отправителя и, нажав правую кнопку мыши, выберите пункт **Добавить к контактам**.
3. В отображаемом диалоге нажмите на закладку **Сертификаты** и убедитесь в наличие сертификата отправителя.



Отправка шифрованных сообщений в Outlook

1. Нажмите кнопку **Параметры** и в отображаемом диалоге установите флаг **Зашифровать сообщение и вложение**.
2. После того, как сообщение подготовлено к отправке, нажмите кнопку **Отправить**.



Практическая работа 20

Запрос сертификатов и настройка защищенного почтового обмена