
# Инфраструктура на публичния ключ

Цифрови подписи, сертификати, SSL

**Светлин Наков**

Българска асоциация на разработчиците на софтуер

Българска асоциация на разработчиците на софтуер

# Криптография с публични ключове

- Криптография – наука за кодиране на информацията
- Криптоанализ – наука за разбиване на криптографски кодове и алгоритми
- Криптография с публични (несиметрични) ключове (Public Key Cryptography)
  - Математическа наука, която осигурява конфиденциалност и автентичност при обмяната на информация чрез използване на криптографски алгоритми с публични и лични ключове
- Двойка криптографски ключове (public/private key pair)
  - Публичен и съответен на него личен ключ
  - Много е трудно по единия ключ да намерим другия
- Личен ключ
  - Тайно число, известно само на притежателя му
  - С него се кодира информация и се полагат цифрови подписи

Българска асоциация на разработчиците на софтуер

# Цифрови подписи

- Публичен ключ (public key)
  - Число (последователност от битове), което обикновено е свързано с дадено лице
  - С него се декодира информация и се проверяват цифрови подписи
  - Обикновено не е тайна за никого
- Цифрово подписване
  - Механизъм за удостоверяване на произхода и целостта на информация, предавана по електронен път
- Цифров подпис (цифрова сигнатура)
  - Число (последователност от битове), което се изчислява математически при подписването на даден документ (съобщение)
  - Гарантира цялостност и автентичност на информацията

Българска асоциация на разработчиците на софтуер

# Хеширащи функции

- Хеширане (хешираща функция)
  - Математическо преобразование, което изчислява хеш-стойност на дадено съобщение
- Хеш-стойност
  - Последователност от битове, обикновено с фиксирана дължина, извлечено по някакъв алгоритъм от съобщението
  - Тази стойност се нарича още message digest
- Криптографски силни хеш-функции (message digest алгоритми)
  - Необратими функции – по хеш-стойността не може лесно да се намери оригиналното съобщение
  - При промяна на само един бит в съобщението се получава тотално различна хеш-стойност
  - Типични примери: MD4, MD5, SHA1

Българска асоциация на разработчиците на софтуер

# Кодиращи алгоритми

- Кодиращите алгоритми
  - Преобразуват дадено съобщение в кодирано съобщение по даден ключ
  - Кодираното съобщение може да се декодира със същия алгоритъм и подходящ декодиращ ключ
  - Биват симетрични и несиметрични
  - Симетричните кодиращи алгоритми
    - Кодирането и декодирането става с един и същ ключ
    - Примери: DES, 3DES, AES, RC2, IDEA
  - Асиметричните кодиращи алгоритми
    - Използват двойка ключове за кодиране и за разкодиране
    - Примери: RSA, DSA, ECDSA, Diffie-Hellman

# Как работи цифровият подпис

Българска асоциация на разработчиците на софтуер

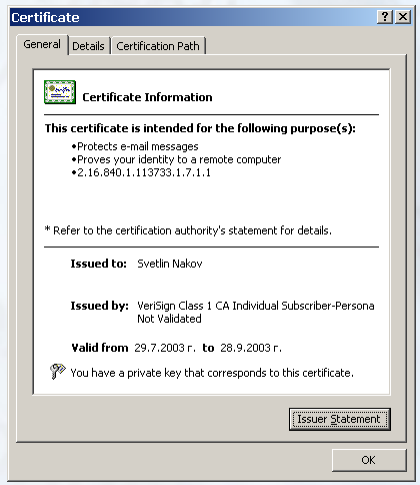
# PKI и сертификати

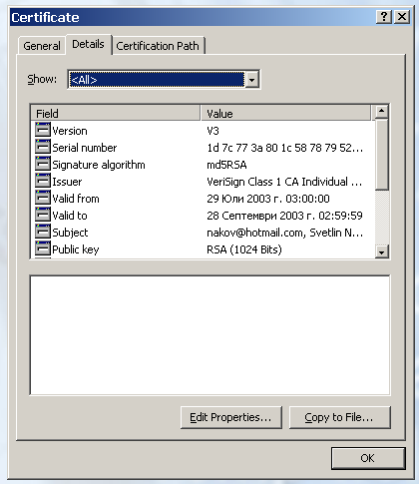
- Инфраструктурата на публичния ключ (Public Key Infrastructure – PKI)
  - Цялостната архитектура, организация, техники, практики и процедури, които подпомагат чрез цифрови сертификати приложението на криптографията, базирана на публични ключове (public key cryptography) за целите на сигурната обмяна на информация по несигурни мрежи и преносни среди
  - Включва сертифициращи организации и цифрови сертификати
- Цифрови сертификати
  - Съдържат публичен ключ и информация за неговия собственик
  - Свързват определен публичен ключ с определено лице
  - Могат да бъдат подписани от друг сертификат или да са саморъчно подписани (self-signed)

Българска асоциация на разработчиците на софтуер

# X.509 сертификати

- X.509
  - Широко-възприет стандарт за цифрови сертификати
  - Масово използван в Интернет
  - Дефинира следната информация:
    - Версия
    - Сериен номер
    - Алгоритъм за цифров подпис
    - Име на издателя
    - Период на валидност
    - Име на притежателя
    - Публичен ключ на притежателя
    - Разширения
    - Цифров подпис на издателя



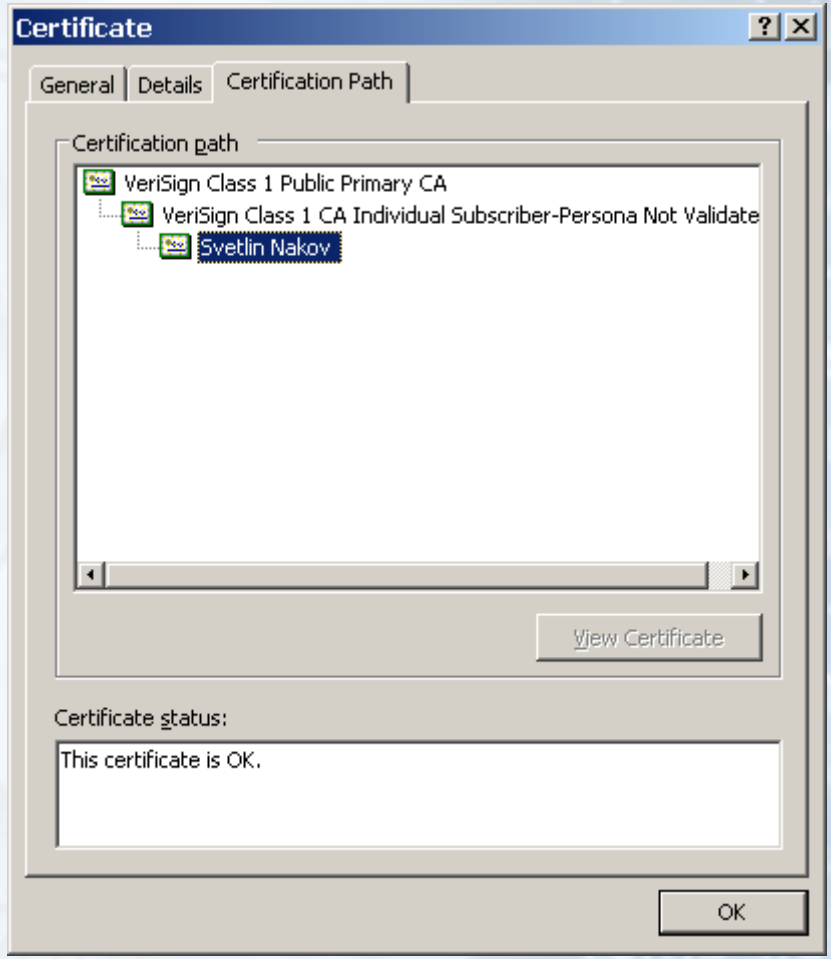



# Сертифициращи организации

- Сертифицираща организация (Certification Authority, CA)
  - Институция, която е упълномощена да издава цифрови сертификати и да ги подписва със своя личен ключ
  - Осигурява доверие между непознати страни
  - Може да е локална за организацията или глобална (VeriSign, GlobalSign, Entrust, Thawte, GTE CyberTrust...)
- Сертификатите биват
  - Саморъчно подписани (self-signed) сертификати
  - Сертификати на сертифициращи организации от първо ниво (root-сертификати)
  - Доверени root-сертификати (trusted root CA certificates)
  - Сертификати на междинни сертифициращи организации

# Вериги от сертификати

- Вериги от сертификати (certification chains)
  - Състоят се от няколко сертификата, като всеки от тях (без последния) е подписан с личния ключ на предходния
  - Използват се за проверка дали на даден сертификат може да се вярва (дали е trusted)
  - В началото на веригата най-често стои доверен root-сертификат на глобална сертифицираща организация



Българска асоциация на разработчиците на софтуер

# Проверка на сертификати

- Проверени сертификати – на които може да се вярва
- Процедура за проверка на сертификат
  - Проверява се срокът на валидност на сертификата
  - Проверява се сертификационната верига, която започва от него
    - дали всеки сертификат от веригата (без последния) е подписан от следващия
    - дали всеки сертификат от веригата (без първия) има право да подписва други сертификати
    - дали последният сертификат е доверен root-сертификат на някое CA, на което имаме доверие
    - дали някой от сертификатите по веригата не е анулиран
- Анулирани сертификати – всяко CA поддържа актуални списъци на анулираните сертификати (CRL)

Българска асоциация на разработчиците на софтуер

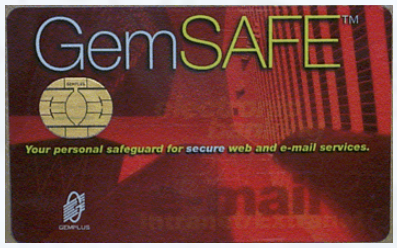
# Хранилища за сертификати

- Protected keystores – защитени хранилища за ключове и сертификати
  - Могат да съдържат един или повече X.509 цифрови сертификата, евентуално с пълната си сертификационна вериги и евентуално с личните ключове, които им съответстват
  - Хранилищата са защитени с парола
  - Личните ключове са защитени с още една парола
  - Съхраняват се в
    - .PFX и .P12 файлове – стандарт PKCS#12
    - JKS – Java Key Store файлове
    - Смарт-карти

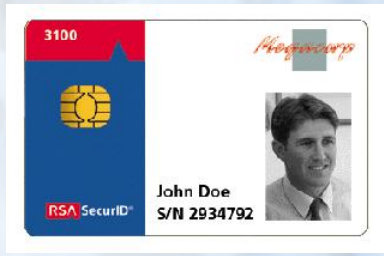
Българска асоциация
на разработчиците на софтуер

# Смарт-карти

- Смарт-картите представляват хардуерни устройства, които имат микропроцесор и памет и могат да генерират и съхраняват ключове и сертификати
- Могат да имат различна форма и интерфейс за достъп
- Криптографските функции като подписване и криптиране на информация се извършват от самата карта
- Достъпът до личните ключове е с PIN код
- Много трудно могат да бъдат прекопирани








Българска асоциация на разработчиците на софтуер

# Добиване на сертификати

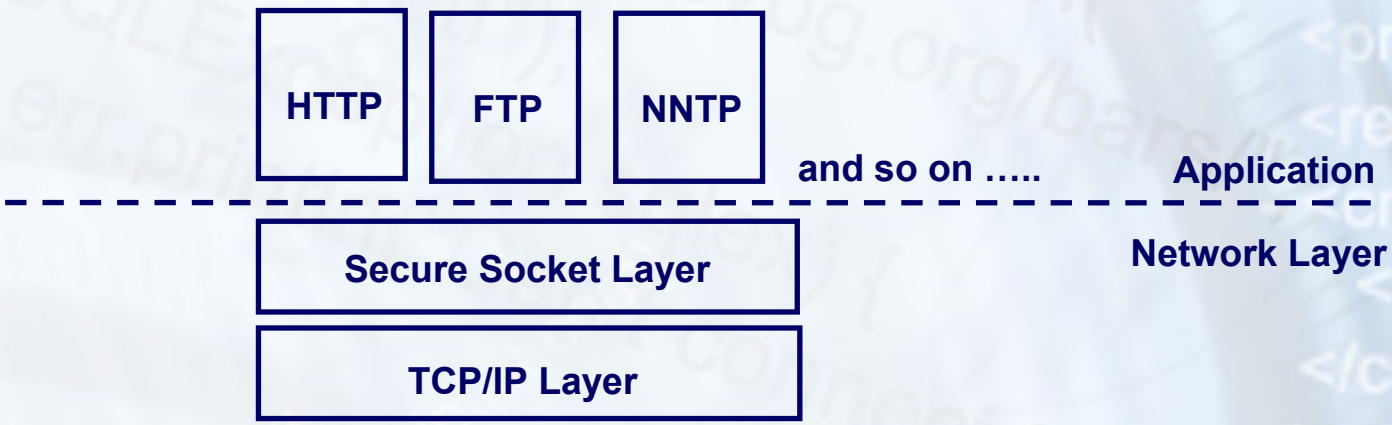
- Генериране на self-signed сертификат със средствата на Java 2 SDK

```
keytool -genkey -alias signFiles –keystore
SignApplet.jks -keypass !secret –dname
"CN=My Company" -storepass !secret
```

- Закупуване на сертификат от сертифицираща организация
- Сертификати за тестови цели – безплатно:
  - Сертификат за защитена e-mail кореспонденция от Thawte – **http://www.thawte.com/html/COMMUNITY/personal/index.html**
  - Тестов сертификат за 60 дни от VeriSign – **http://www.verisign.com/client/enrollment/index.html**
  - Тестов сертификат за 30 дни от GlobalSign – **http://secure.globalsign.net/**

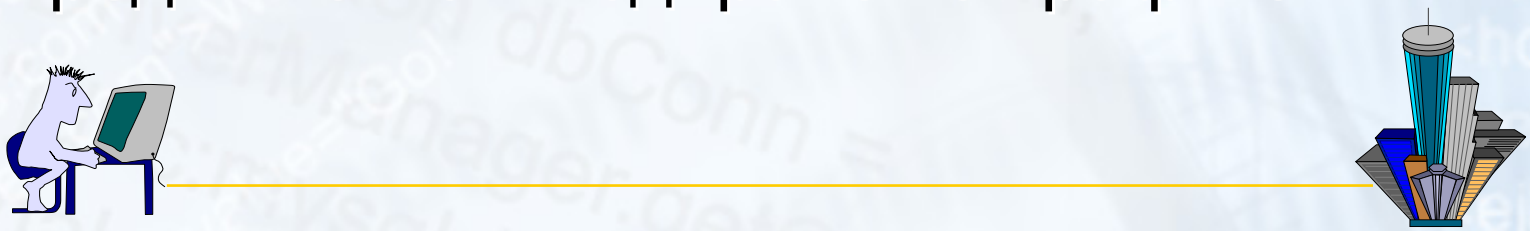
# SSL протоколът

- Secure Socket Layer (SSL) е мрежов протокол, който се използва за сигурен пренос на данни по TCP/IP мрежи
- Осигурява защитени от подслушване тунели за поточно-ориентиран пренос на информация
- Използва криптографски алгоритми, PKI и цифрови сертификати



# SSL 2.0

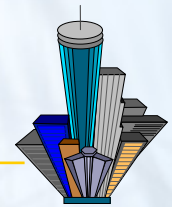
- SSL 2.0 предоставя автентикация на сървъра пред клиента и кодиране на трафика




• Client and Server use $SessKey_C$ to encrypt all data exchanged over the Internet

# SSL 3.0

- SSL 3.0 предоставя възможност и клиентите да се автентикират пред сървъра



- Client Connects to Secure Server
- Server sends copy of the Server certificate ($Cert_S$) to the Client, indicating that SSL 3.0 is enabled with client authentication

$Cert_S$ - SSL 3.0

- Client verifies signature on $Cert_S$
- Client generates session key ($SessKey_C$)
- Client encrypts $SessKey_C$ using $Cert_S$

$\{SessKey_C\}\ Cert_S + Cert_C$

- Server verifies signature on $Cert_C$ (Server can check other information as well)
- Server decrypts $SessKey_C$ using it's private key
- Client asks operator to select a Cleint certificate ($Cert_C$) to access server

$\{Data\}\ SessKey_C$

- Client and Server use $SessKey_C$ to encrypt all data exchanged over the Internet

Българска асоциация
на разработчиците на софтуер

# Инфраструктура на публичния ключ

Цифрови подписи, сертификати, SSL

# Въпроси?