Paweł Rajba
pawel@ii.uni.wroc.pl
http://kursy24.eu/

Application Security Certificates

Agenda

- Introduction
- Certificate structure
- Extensions
- Usages
- PKCS
- Encodings & formats
- Revoking certificates
- Check if certificate if trusted
- Certificates on the market
- Qualified signatures
- Certificate Signing Request

Introduction

- Certificate is an electronic document which includes:
  - Public key of the subject
  - Identity description of the subject
  - Digital signature of the **trusted** third party
  - Expiration date
- Main goal
  - Having a document which proves your identity in transactions
    - Something similar to driving licence in real life

Introduction

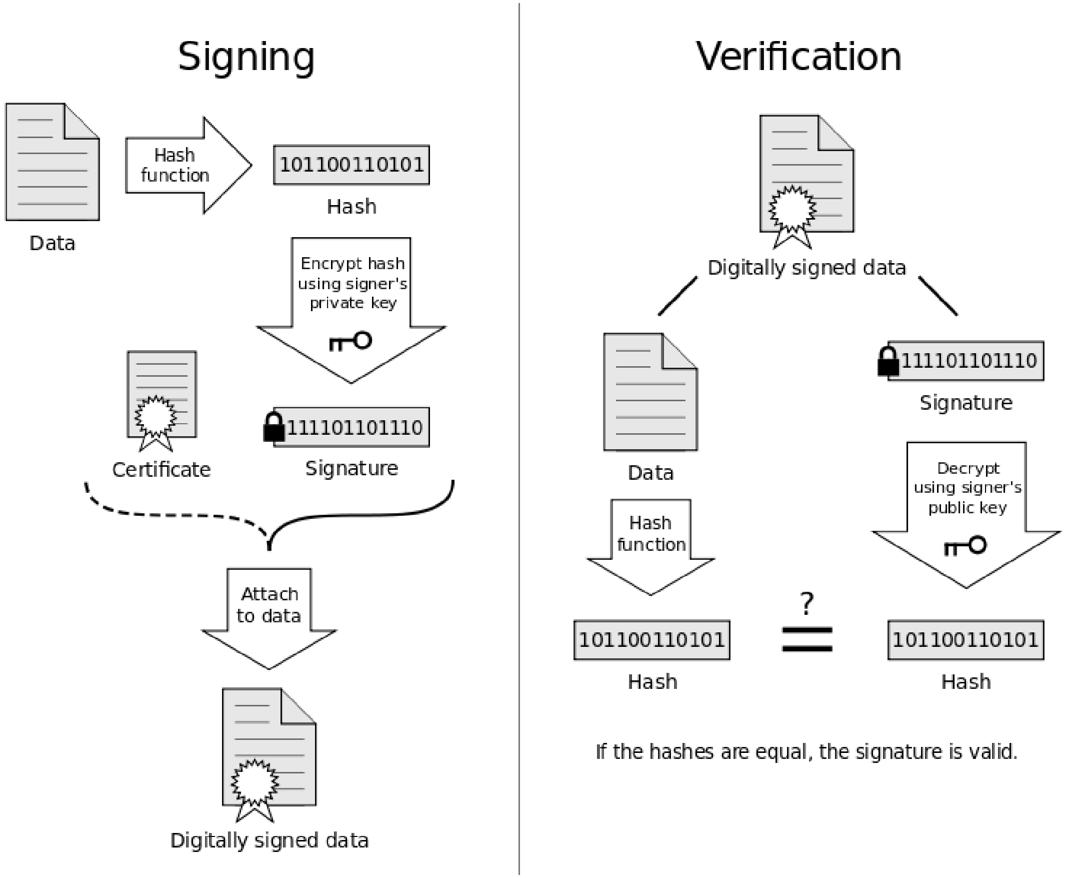
- PKI
  - Stands for Public Key Infrastructure
  - Goal: system for distribution public keys
  - Main constituents
    - Certificates
    - Certificates Authorities (CAs)
    - Method for revoking certificates
    - Method of evaluating chain of certificates
    - General operations
      - Signing: private key to sign, public key to verify signature
      - Encryption: public key to encrypt, private key to decrypt

Introduction

- CA
  - Stands for Certificate Authority
  - A unit which everyone „trust”
  - Every CA has a set of Root CA’s
    - https://www.symantec.com/page.jsp?id=roots
    - https://www.symantec.com/content/en/us/about/media/repository/root-certificates.pdf
  - We consider
    - Root CA – self-signed certificates
    - Intermediate CA – certficate signed by another CA

Introduction

- Digital signature – quick recap



Introduction

- Certificates systems
  - PGP, SPKI/SDSI
    - decentralized, based on WOT
  - X.509
    - based on hierarchy of certficate authorities

- In this presentation we will focus on X.509

X.509: Introduction

- ITU-T standard which allows to create a hierarchical Public Key Infrastructure (PKI)
- Built on top of X.500 family
  - http://pl.wikipedia.org/wiki/X.500
- Currently X.509 usually refers to IETF's PKIX Certificate and CRL Profile of the X.509 v3 certificate standard, as specified in RFC 5280
  - http://tools.ietf.org/html/rfc5280
- PKIX - Public Key Infrastructure X.509 Working Group (closed in 2013)

X.509: Structure

- Certificate
  - Version
  - Serial Number
  - Algorithm ID
  - Issuer
  - Validity
    - Not Before
    - Not After
  - Subject
  - Subject Public Key Info
    - Public Key Algorithm
    - Subject Public Key
  - Issuer Unique Identifier (optional)
  - Subject Unique Identifier (optional)
  - Extensions (optional)
    - ...
- Certificate Signature Algorithm
- Certificate Signature

X.509: Extensions

- Give additional information about certificate
- Uniquely identified by OIDs
  - Based on ASN.1 syntax
    - http://en.wikipedia.org/wiki/Abstract_Syntax_Notation_One
  - Registry one can find here:
    - http://www.alvestrand.no/objectid/
- Extension may be
  - Critical, then certificate system must reject certificate if it is
    - Not recognized or
    - Cannot be processed
  - Not-critical, then certificate system
    - May ignore extension if it is not recognized and
    - Must process if it is recognized

X.509: Extensions

- Examples
  - Subject Key Identifier (OID: 2.5.29.14)
    - A hash derived from the public key of certficate
  - Authority Key Identifier (OID: 2.5.29.35)
    - A hash based on public key of an issuer cert (SKI)
    - or based on issuer name and serial number
  - CRL Distribution Points (OID: 2.5.29.31)
    - A place when information about revocaton can be found
  - Netscape Certificate Type (OID:2.16.840.1.113730.1.1)
    - Define certficate subject to be SSL client, SSL server or CA
  - Basic Constraints (OID: 2.5.29.19)
    - Determine if subject can act as a CA
  - Key Usage (OID: 2.5.29.15)
    - Determine set of allowed usages
- Full list of extensions is defined in RFC:
  - http://tools.ietf.org/html/rfc5280#section-4.2.1

X.509: Usages

- Last 3 examples in previous slide define key usage limitation
- Let’s see the definitione of Key Usage field:
  - KeyUsage ::= BIT STRING {

```
        digitalSignature        (0),
        nonRepudiation          (1),
                        -- recent editions of X.509 have
                        -- renamed this bit to contentCommitment
        keyEncipherment         (2),
        dataEncipherment        (3),
        keyAgreement            (4),
        keyCertSign             (5),
        cRLSign                 (6),
        encipherOnly            (7),
        decipherOnly            (8) }
```

- Good summary from IBM
  - http://publib.boulder.ibm.com/infocenter/domhelp/v8r0/index.jsp?topic=%2Fcom.ibm.help.domino.admin.doc%2FDOC%2FH_KEY_USAGE_EXTENSIONS_FOR_INTERNET_CERTIFICATES_1521_OVER.html

Source: http://tools.ietf.org/html/rfc5280#section-4.2.1.3

X.509: Usages

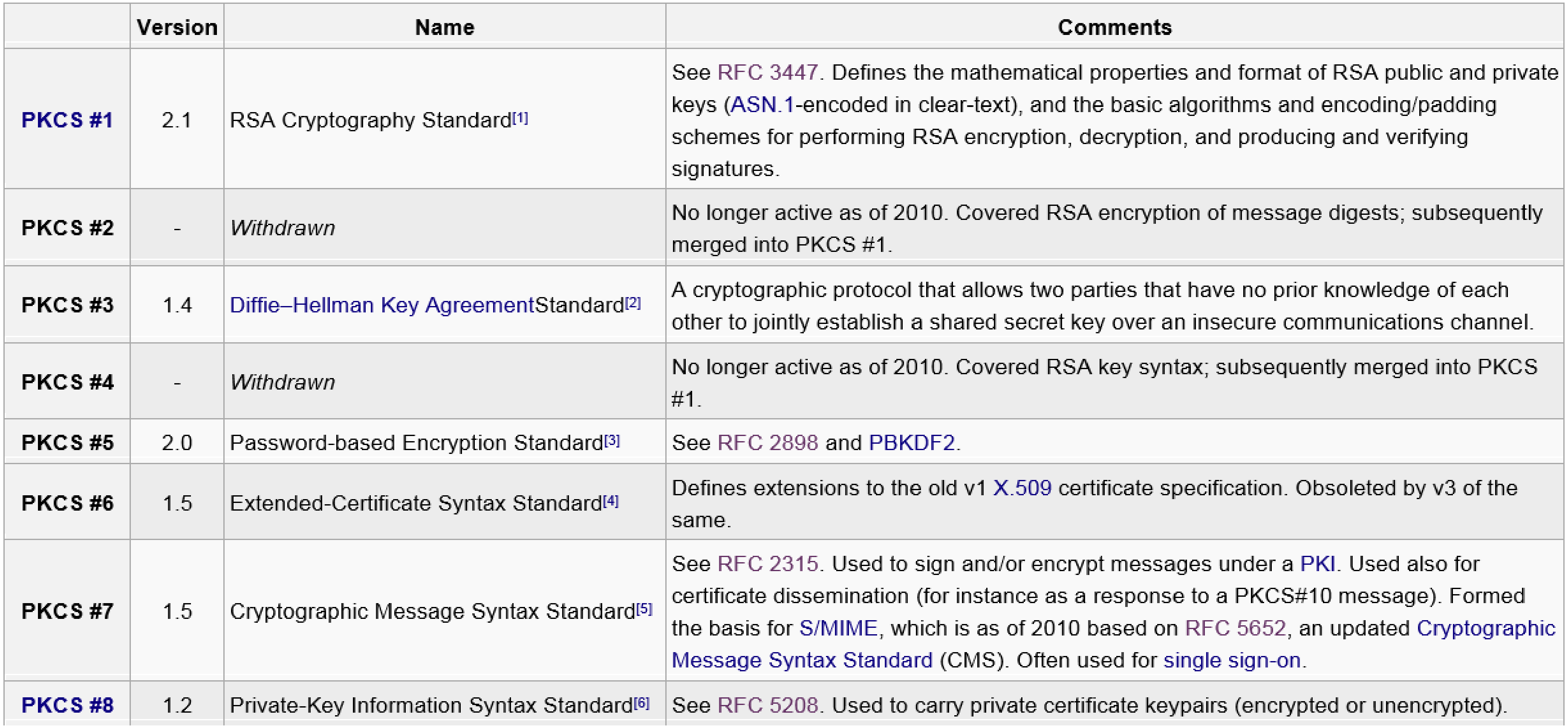
- Each certficate is intended to specific usages
  - E.g. Web servers, e-mails, code signing
- VeriSign introduces classes for types of certs:
  - Class 1 for individuals, intended for email.
  - Class 2 for organizations, for which proof of identity is required.
  - Class 3 for servers and software signing, for which independent verification and checking of identity and authority is done by the issuing certificate authority.
  - Class 4 for online business transactions between companies.
  - Class 5 for private organizations or governmental security.
    - https://www.symantec.com/page.jsp?id=roots

However, this is not a part of PKI standard

PKCS Standards

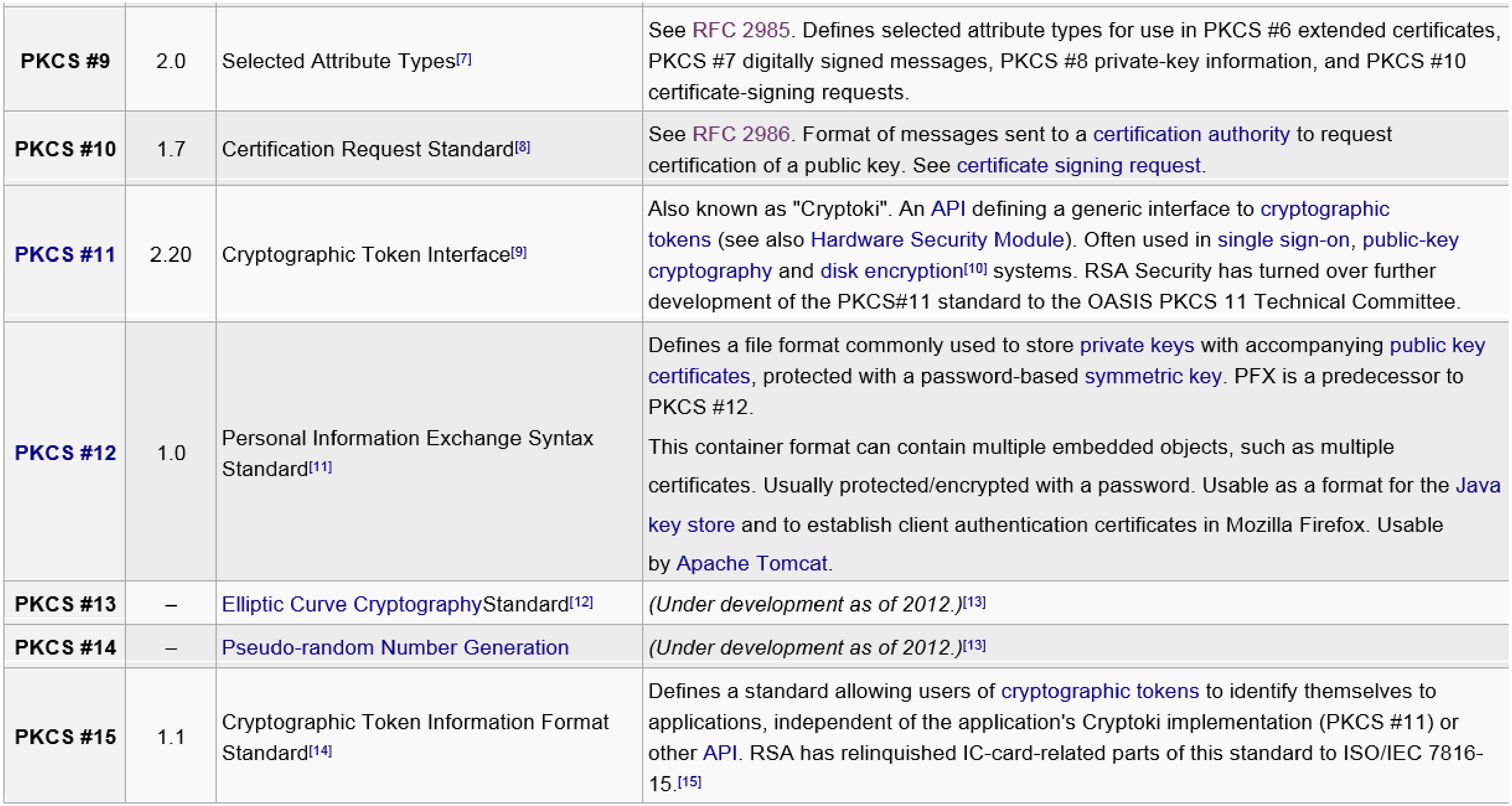
- A set of public-key cryptography standards
- Published by RSA Security Inc. in early 90s
  - Main goal was to promote cryptography techniques to which they had patents
  - Currently, most of them are in the public domain and taken care of organization like IETF and PKIX
- Let's review shortly standards on the next slides
  - List can be found: http://en.wikipedia.org/wiki/PKCS

PKCS Standards



	Version	Name	Comments
PKCS #1	2.1	RSA Cryptography Standard[1]	See RFC 3447. Defines the mathematical properties and format of RSA public and private keys (ASN.1-encoded in clear-text), and the basic algorithms and encoding/padding schemes for performing RSA encryption, decryption, and producing and verifying signatures.
PKCS #2	-	*Withdrawn*	No longer active as of 2010. Covered RSA encryption of message digests; subsequently merged into PKCS #1.
PKCS #3	1.4	Diffie–Hellman Key AgreementStandard[2]	A cryptographic protocol that allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure communications channel.
PKCS #4	-	*Withdrawn*	No longer active as of 2010. Covered RSA key syntax; subsequently merged into PKCS #1.
PKCS #5	2.0	Password-based Encryption Standard[3]	See RFC 2898 and PBKDF2.
PKCS #6	1.5	Extended-Certificate Syntax Standard[4]	Defines extensions to the old v1 X.509 certificate specification. Obsoleted by v3 of the same.
PKCS #7	1.5	Cryptographic Message Syntax Standard[5]	See RFC 2315. Used to sign and/or encrypt messages under a PKI. Used also for certificate dissemination (for instance as a response to a PKCS#10 message). Formed the basis for S/MIME, which is as of 2010 based on RFC 5652, an updated Cryptographic Message Syntax Standard (CMS). Often used for single sign-on.
PKCS #8	1.2	Private-Key Information Syntax Standard[6]	See RFC 5208. Used to carry private certificate keypairs (encrypted or unencrypted).

PKCS Standards



PKCS #9	2.0	Selected Attribute Types[7]	See RFC 2985. Defines selected attribute types for use in PKCS #6 extended certificates, PKCS #7 digitally signed messages, PKCS #8 private-key information, and PKCS #10 certificate-signing requests.
PKCS #10	1.7	Certification Request Standard[8]	See RFC 2986. Format of messages sent to a certification authority to request certification of a public key. See certificate signing request.
PKCS #11	2.20	Cryptographic Token Interface[9]	Also known as "Cryptoki". An API defining a generic interface to cryptographic tokens (see also Hardware Security Module). Often used in single sign-on, public-key cryptography and disk encryption[10] systems. RSA Security has turned over further development of the PKCS#11 standard to the OASIS PKCS 11 Technical Committee.
PKCS #12	1.0	Personal Information Exchange Syntax Standard[11]	Defines a file format commonly used to store private keys with accompanying public key certificates, protected with a password-based symmetric key. PFX is a predecessor to PKCS #12. This container format can contain multiple embedded objects, such as multiple certificates. Usually protected/encrypted with a password. Usable as a format for the Java key store and to establish client authentication certificates in Mozilla Firefox. Usable by Apache Tomcat.
PKCS #13	–	Elliptic Curve CryptographyStandard[12]	*(Under development as of 2012.)*[13]
PKCS #14	–	Pseudo-random Number Generation	*(Under development as of 2012.)*[13]
PKCS #15	1.1	Cryptographic Token Information Format Standard[14]	Defines a standard allowing users of cryptographic tokens to identify themselves to applications, independent of the application's Cryptoki implementation (PKCS #11) or other API. RSA has relinquished IC-card-related parts of this standard to ISO/IEC 7816-15.[15]

Encodings & formats

- PEM
  - Encoded with Base64
  - Doesn’t support storing the whole path of certficates
  - Doesn’t support storing combination of cerficate and private key
  - Extentions: .pem, .crt, .cer, .cert, .key
  - Popular in open source solutions (e.g. Apache uses PEM)
- DER
  - Binary representation
  - Doesn’t support storing the whole path of certficates
  - Doesn’t support storing combination of cerficate and private key
  - Extensions: .cer, .der
- PKCS#7
  - Supports storing whole chain of certificates
  - Doesn’t support storing private key
  - Extensions: .p7b, .p7c
- PKCS#12 (previously .pfx was predecessor of PKCS#12)
  - Supports storing whole path of certificates
  - Supports storing private key
  - Extensions: .p12, .pfx

X.509: Revoking certificates

- Certificate revocation allows to avoid certificates which shouldn't be trusted no longer
- There are 2 options: CRL and OCSP
- CRL
  - A file with a list of revoked certificates
  - Location included as an extension field in certificate
  - Signed by CA's private key
    - Let's see sample list from https://access.redhat.com/home
- OCSP (Online Certificate Status Protocol)
  - A service which can answer about the status of certificate
  - More efficient than parsing CRL lists
  - Read more
    - http://en.wikipedia.org/wiki/Online_Certificate_Status_Protocol
    - http://www.ietf.org/rfc/rfc2560.txt
  - On lab you will be asked to play with this more ☺

Check if certificate trusted

- There 2 parts of certificate validation process
  - Path Discovery
  - Path Validation
    - http://tools.ietf.org/html/rfc3379
  - Let's see main points of an algorithm
    - http://en.wikipedia.org/wiki/Certification_path_validation_algorithm
- Trust is based on Trusted Store Certificate in the system
- DEMO
  - Let's see the list of trusted certificates in IE
  - Let's see the chain of certificates for
    - https://www.symantec.com/index.jsp

Certificates on the market

- We consider 3 levels of validation
  - DV – Domain Validation
    - Only domain is checked in DNS systems
    - No information about organization in included
    - Available in a few minutes
  - OV – Full Organization Validation
    - Additionally organization is checked on the basis of organization documentation
    - Available in 1-2 days
  - EV – Extended Validation
    - More checks are performed: if company has a bank account, there is a phone call with set of questions, etc.
    - Available in 1-10 days
    - Only this type gives a green bar in a web browser

Certificates on the market

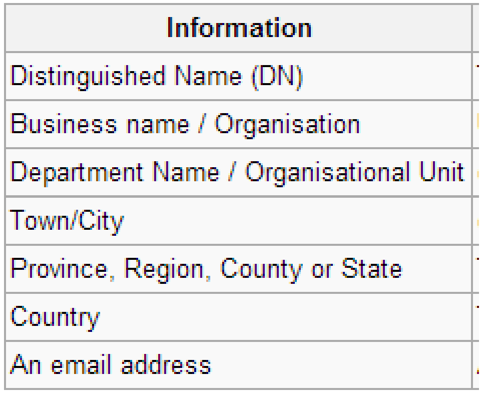
- What means a guarantee of certificate?
  - If something is wrong with a certificate or CA private key, an issuer is obliged to pay compensation
- There is a possibility to buy a wildcard certificate
  - *.domain.com
- Who sells certificates
  - VeriSign (Symantec ownership)
  - Thawte, Geotrust (part of VeriSign)
  - Comodo
  - GoDaddy
  - TrustWave
  - Certum (in Poland)

Qualified signatures

- Qualified signature (podpis kwalifikowany)
  - A digital dignature based on qualified certificate
    - Usually, if you buy a qualified signature, you get a package
      - Certificate
      - Device with private key
      - Software intended to make signatures
  - In Poland only National Certification Center is allowed to decide who should be able to issue such certificates
    - But it doesn't issue them on its own
    - Let's see their website: http://www.nccert.pl/
  - Read more: http://pl.wikipedia.org/wiki/Podpis_kwalifikowany
    - Let's see a short list o applications there

Certificate Signing Request

- Applicant generates public/private key pair
  - Private key keeps in secret
- Generates CSR (Certificate Signing Request)
  - A file with information about applicant
  - CSR file is signed by private key of applicant
- CSR file with additional documentation is sent to CA
- If everything is ok, CA sent back a certificate signed with a private key of CA



Information
Distinguished Name (DN)
Business name / Organisation
Department Name / Organisational Unit
Town/City
Province, Region, County or State
Country
An email address

Source: http://en.wikipedia.org/wiki/Certificate_signing_request

References

- SPKI
  - http://pl.wikipedia.org/wiki/SPKI
- PGP
  - http://pl.wikipedia.org/wiki/Pretty_Good_Privacy
- X.509 & PKI
  - http://en.wikipedia.org/wiki/X.509
  - http://technet.microsoft.com/en-us/library/cc737264(v=ws.10).aspx
- Encoding & formats
  - http://myonlineusb.wordpress.com/2011/06/19/what-are-the-differences-between-pem-der-p7bpkcs7-pfxpkcs12-certificates/
  - http://serverfault.com/questions/9708/what-is-a-pem-file-and-how-does-it-differ-from-other-openssl-generated-key-file
  - https://support.ssl.com/Knowledgebase/Article/View/19/0/der-vs-crt-vs-cer-vs-pem-certificates-and-how-to-convert-them
- Sample files
  - http://ospkibook.sourceforge.net/docs/OSPKI-2.4.7/OSPKI-html/sample-openssl-usage.htm
- Calculating hashes (very good)
  - http://certificateerror.blogspot.com/2011/02/how-to-validate-subject-key-identifier.html
- Good Knowledge Base
  - https://access.redhat.com/site/documentation/en-US/Red_Hat_Certificate_System/8.0/html/Admin_Guide/Standard_X.509_v3_Certificate_Extensions.html
  - https://certyfikatyssl.pl/faq.html
- Checking trust chain of certificates
  - http://www.oasis-pki.org/pdfs/Understanding_Path_construction-DS2.pdf
  - http://www.herongyang.com/PKI/HTTPS-IE-8-View-Server-Certificate-Path.html
  - http://technet.microsoft.com/en-us/library/cc962065.aspx
  - http://en.wikipedia.org/wiki/Extended_Validation_Certificate
  - http://blog.securism.com/2009/01/summarizing-pki-certificate-validation/

- Managing and obtaining certificates
  - http://msdn.microsoft.com/en-us/library/windowsazure/gg981929.aspx
- Related RFC documents
  - http://tools.ietf.org/html/rfc5280, http://tools.ietf.org/html/rfc3279, http://tools.ietf.org/html/rfc3280, http://tools.ietf.org/html/rfc4055, http://tools.ietf.org/html/rfc4491