Certifikáty, ověřování, certifikační autority. Jak postavit vlastní CA (openssl)

Základy

- Šifrování
- Otisk
- Elektronický podpis

Šifrování

Šifrování

- Jednoduchý příklad – tabulka se záměnou písmen
- Základní dělení kryptografických metod
  - Symetrické (DES 3DES, AES, ...)
  - Nesymetrické (RSA, DSA, ...)
- Klíče, algoritmy
  - Uchování klíčů (klíč x pár klíčů)
  - Náročnost na zdroje
  - Délky klíčů ...

Otisk

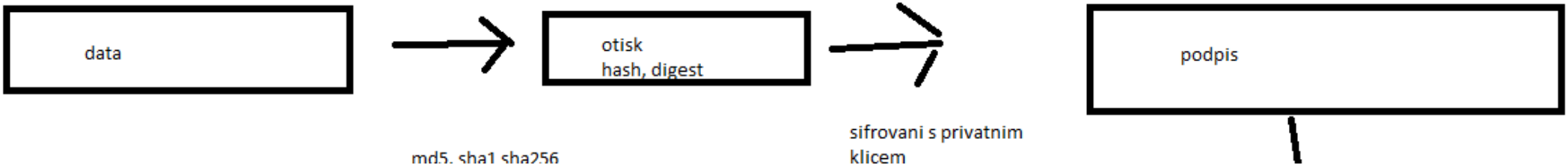
- Otisk dokumentu (hash, digest)
  - Vlastnosti ...

Otisk

- Otisk dokumentu (hash, digest)
  - Vlastnosti
    - Krátké pole bitů. V souřasnosti v řádu desítek [bytů] (bitů)
      - MD5 [16]
      - SHA-1 [20] (160)
      - SHA-2
        - SHA-224, SHA-256 [32], SHA-384, SHA-512 [64] (bity v názvu)
    - Na malou změnu dokumentu chceme “velkou změnu” v otisku
    - jednosměrný
  - “Unikátnost” ;)

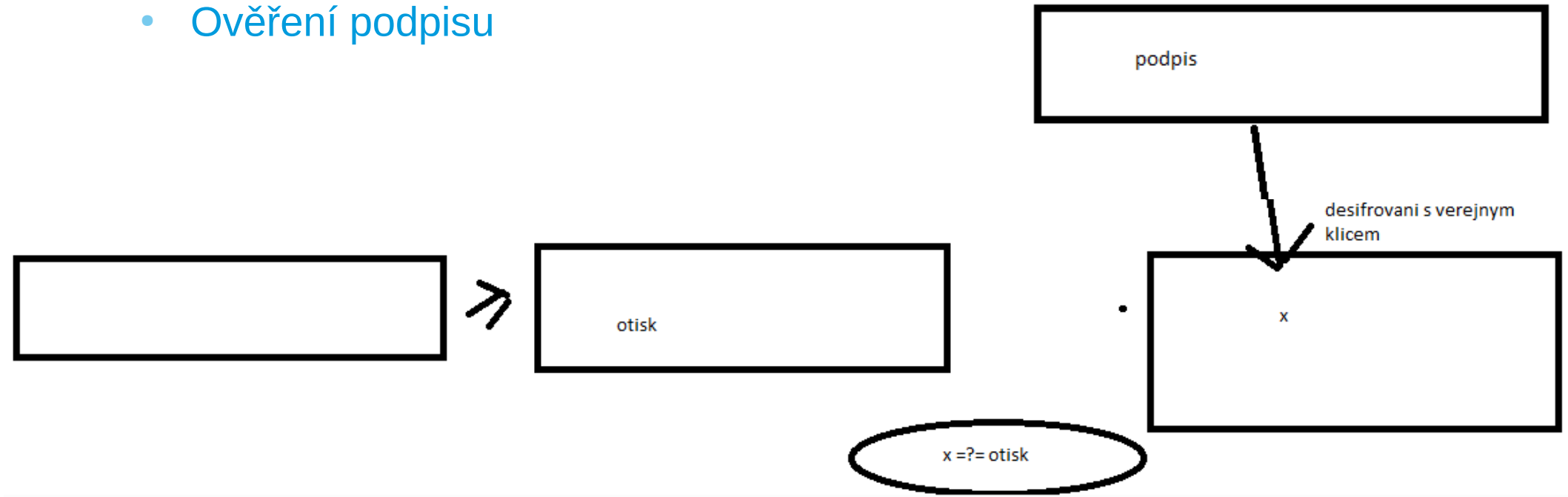
Elektronický podpis

- Elektronický podpis
  - Nesymetrická šifra, pár klíčů
  - Vytvoření podpisu

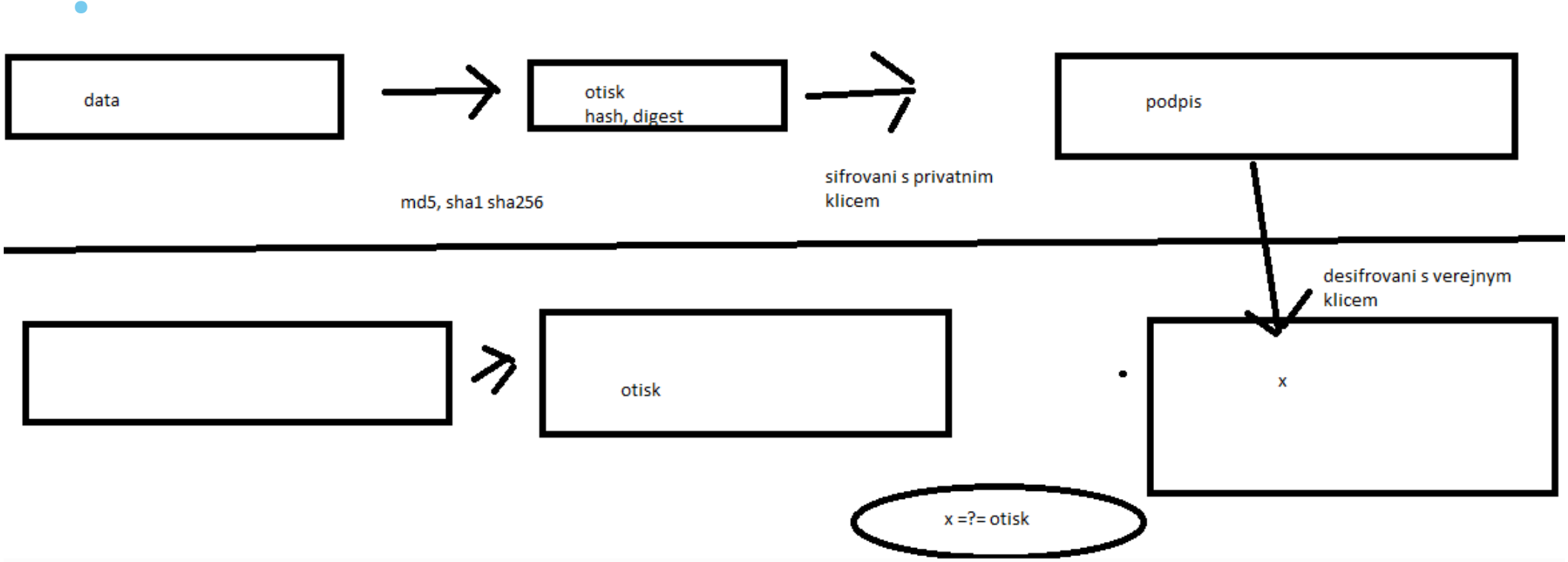


Elektronický podpis

- Elektronický podpis
  - Ověření podpisu



Elektronický podpis



ASN.1

- ASN.1, Abstract Syntax Notation #1
  - Vizuálně čitelný jazyk pro definici datových struktur, data lze de/serializovat z/do binární podoby
    - Komerčně dostupný standard X.208 ITU
  - Použití
    - pro výměnu dat mezi adresářovými servery (LDAP, …)
    - certifikáty
    - ...

ASN.1

- Typ
  - Začíná velkým písmenem
    - 4 druhy
      1. jednoduchý
      2. strukturovaný; zápis s {}, oddělovač čárka
      3. odvozený
      4. jiný (ANY, CHOISE)

ASN.1

- Typ – příklady

```
Version::=INTEGER
Year::=INTEGER

Student::=STRUCTURE OF {
    jmeno IA5String,
    prijmeni IA5String,
    rok Year,
    …
}
```

ASN.1

- Kódování (serializace)
  - Přepis mezi vizuální a binární podobou dat
  - X.209
    - Basic Encoding Rules (BER)
    - Distinguished Encoding Rules (DER)
      - 'normalizace' - zkrácení do minimálního tvaru (viz dále)

ASN.1

- Kódování
  - TYP DAT | DÉLKA | HODNOTA

    - TYP DAT (byte+)
      - Tag (bity 4-0)
        - 0-30 viz tabulka
        - > 31: 31 zde a následují byty 0x80+7bitů pro id tagu, poslední 0x00+7bitů
      - Třida (bity 7-6)
        - 00 univerzální tag
        - 01 aplikační tag
        - 10 kontextově specifický tag
        - 11 privátní tag
      - jednoduchý/strukturovaný typ (bit 5, 0/1)

ASN.1

typ tag

- BOOLEAN 1
- INTEGER 2
- BIT STRING 3
- OCTET STREAM 4
- NULL 5
- OBJECT IDENTIFIER 6
- REAL 9
- ENUMERATED 10
- SEQUENCE, SEQUENCE OF 16
- SET, SET OF 17
- PrintableString 19 (A-Z, 0-9, space a-z, '()+,-./:=?)
- T61String 20
- IA5String 22
- UTCTime 23 (20060323082133.5-0100)
- VisibleString 26

ASN.1

- Kódování
  - DÉLKA (byte+)
    - 0x00 + 7 bitů délky (rozsah 0-127)
    - 0x80 + 7 bitů obsahuje počet následujících bytů s délkou

ASN.1

- Kódování
  - HODNOTA v příkladech
    - INTEGER 0x1234
      - 02|02|12|34
    - INTEGER 128 0x0080 (je znaménkový)
      - 02|02|00|80
    - INTEGER 127
      - 02|01|7f

ASN.1

- Kódování
  - Pro připomenutí – dvojková doplňková soustava

DEC	HEX
-129	FF7F, FFFFFFFFFFFFFF7F
-128	80, FFFFFFFFFFFFFF80, FF80
-2	fe
-1	FFFFFFFFFFFFFFff
0	00
1	01
127	7f
128	0080
129	0081

ASN.1

- Kódování
  - HODNOTA v příkladech
    - INTEGER

DEC varianty kódování

0	|02|01|00	
1	|02|01|01	
127	|02|01|7f	
128	|02|02|00|80	|02|04|00|00|00|80
256	|02|02|01|00	
-128	|02|01|80	|02|03|FF|FF|80
-129	|02|02|FF|7F	|02|03|FF|FF|7F

    - Pravidla DER – minimální “normalizovaný” tvar, ten prvý výše

ASN.1

- Kódování příklady
  - UTCTime 23 (20060323082133.5-0100)
    - RRRRMMDDhhmm[ss]Z
    - RRRRMMDDhhmm[ss]+hhmm
    - RRRRMMDDhhmm[ss]-hhmm
    - … s necelými sekundami

    200603230821Z

    17|0D|32|30|30|36|30|33|32|33|30|38|32|31|5A

ASN.1

- Kódování příklady

```
Student::=SEQUENCE OF {
    jmeno IA5String,
    prijmeni IA5String,
    rok Year,
    …
}
```

  - Jan Novak 1980

ASN.1

- Kódování příklady
  - Jan Novak 1980

30|10| struktura
(tag 16 == 10h za STRUCTURE OF, 20h za strukturovaný typ)
16|03|..|..|.. Jan
16|05|..|..|..|..|.. Novak
02|02|07|BC| 1980

ASN.1

- Identifikace objektů
  - registrační úřady ITU (dříve CCITT nebo je to část) a ISO
  - stromová struktura

    OBJECT_IDENTIFIER

    1.3.6.1.2.1

    iso(1).identified organization(3).dod(6).internet(1).mgmt(2).mib(1)

  - https://oidref.com/1.3.6.1.2.1

ASN.1

- ccitt(0)
-- recomendation(0)
-- question(1)
-- administration(2)
-- network operator(3)
- iso(1)
-- standard(0)
-- registration authority(1)
-- member body(2)
-- identified organization(?)
---dod(6)
...

ASN.1

...
- iso(1)
-- standard(0)
-- registration authority(1)
-- member body(2)
-- identified organization(3)
---dod(6)
----internet(1)
-----directory(1)
-----mgmt(2)
------**mib(1)**
-----experimental(3)
-----private(4)
------enterprises(1)
--- …
-- …
- joint-iso-ccitt(2)

ASN.1

- Jiné typy – v příkladech
  - CHOICE

```
Certificate
ExtendedCertificate

ExtendedCertificateOrCertificate ::= CHOICE {
    certificate Certificate,
    extendedCertificate [0] IMPLICIT ExtendedCertificate
}
```

  - ANY

```
AlgorithmIdentifier ::= SEQUENCE {
    algorithm OBJECT_IDENTIFIER,
    parameters ANY DEFINED BY algorithm OPTIONAL
}
```

ASN.1

- Mám data, a jak to rozluštím
  - Obecně ASN.1

    openssl asn1parse -dump -in MYFILE -inform PEM

    # nebo format DER (text base64 s hranicemi x binární)
  - Certifikáty dle X.509
    - ze souboru

      openssl x509 -in MYFILE -inform PEM -text -noout

      # -subject -issuer -enddate -startdate -ext subjectAltName
    - pipou

      openssl s_client -servername www.fjfi.cvut.cz -connect www.fjfi.cvut.cz:443 | \

      openssl x509 -subject -issuer -startdate -enddate -noout

      (sorry za případně slité --, dokumentové - či “” ‘’)

Certifikáty

- Certifikát
  - datová struktura obsahující údaje o uživateli
  - mj. veřejný klíč
  - podepsaná vydavatelem
  - Srovnání s OP, PAS
    - Jméno, vydavatel, ...

Certifikáty

- Certifikát
  - Srovnání s OP, PAS
    - Jméno
    - Vydal
    - Platnost
    - ID číslo
    - Razítko
    - Formát
    - Foto (veřejný klíč)

Certifikáty

- Komponenty certifikátu

```
Version ::= INTEGER;    0 1 2
SerialNumber ::= INTEGER
AlgorithmIdentifier   (prakticky string registrovaný pod svým OID)
Validity ::= SEQUENCE {
    notBefore UTCTime,
    notAfter UTCTime
}
SubjectPublicKeyInfo ::= SEQUENCE {
    algorithm AlgorithmIdentifier,
    subjectPublicKey BIT_STRING
}
Subject ::= Name (viz dále)
Issuer ::= Name (viz dále)
```

Certifikáty

- Podpis certifikátu

```
SIGNED_SEQUENCE ::= SEQUENCE {
    coMaBytPodepsano OCTETSTRING,
    identifikatorAlgoritmu AlgorithmIdentifier,
    podpis OCTETSTRING
}
```

Certifikáty

- Typ Name

```
Name ::= SEQUENCE OF RelativeDistinguishedName
RelativeDistinguishedName ::= SET OF AttributeValueAssertion
AttributeValueAssertion ::= SEQUENCE {
    attributeType AttributeType,
    attributeValue AttributeValue
}
AttributeType ::= OBJECT_IDENTIFIER
AttributeValue ::= ANY
```

Certifikáty

- Hodnoty typu AttributeType pro Name

  C Country
  O Organization
  OU Organizational unit
  CN Common Name
  DN Distinguished Name, Domain Name
  Email e-mail address
  serialNumber
  …

  - Vaše uživatelské jméno … (řada standardů X.500 o adresářových službách)

    CN=Miroslav Minarik,OU=Domain Users,OU=LMC Users,DC=ad,DC=lmc,DC=cz
    cn=minarmir,ou=km,ou=people, DC=fjfi,DC=cvut,DC=cz

Certifikáty

- Rozšířené certifikáty mohou obsahovat další vlastnosti ...
  - Omezení působnosti klíče
    - identifikátor

      keyUsage OBJECT_IDENTIFICATOR ::= { joint-iso-ccitt(2).ds(5).id-ce(29).15 }
    - hodnota

```
KeyUsage ::= BIT_STRING {
    digitalSignature( 0),
    nonRepudation( 1),
    keyEncypherment( 2),
    dataEncypherment( 3),
    keyAgreement( 4),
    keyCertSign( 5),
    crlSign( 6)
}
```

Certifikáty

- Rozšířené certifikáty mohou obsahovat další vlastnosti ...
  - Omezení délky zřetězení certifikátů
    - identifikátor

      basicConstraints OBJECT_IDENTIFICATOR ::= { joint-iso-ccitt(2).ds(5).id-ce(29).19 }
    - Hodnota

```
BasicConstraints ::= SEQUENCE {
    cA BOOLEAN DEFAULT FALSE
    pathLenConstraints INTEGER OPTIONAL
}
```

Certifikáty

- Rozšířené certifikáty mohou obsahovat další vlastnosti ...
  - Odkaz na CRL (certificate revocation list, X.511)
    - Hodnota (přibližně :))

```
CRL ::= SIGNED SEQUENCE {
    signature AlgorithmIdentifier,
    issuer Name,
    lastUpdate UTCTime,
    nextUpdate UTCTime,
    revocedCertificates  SIGNED SEQUENCE OF SEQUENCE {
        userCertificate SerialNumber,
        revocationDate UTCTime OPTIONAL
     }
}
```

Certifikáty

- Rozšířené certifikáty mohou obsahovat další vlastnosti ...
  - Alternativní DNS jména
    - identifikátor

      subjectAltName OBJECT_IDENTIFICATOR ::= { joint-iso-ccitt(2).ds(5).id-ce(29).17 }
  - … další
    - https://oidref.com/2.5.29

Certifikáty

To be continued

Certifikáty

- Koncový certifikát
  - “Samostatný”
  - Má vydavatele, který ho podepsal (CA)

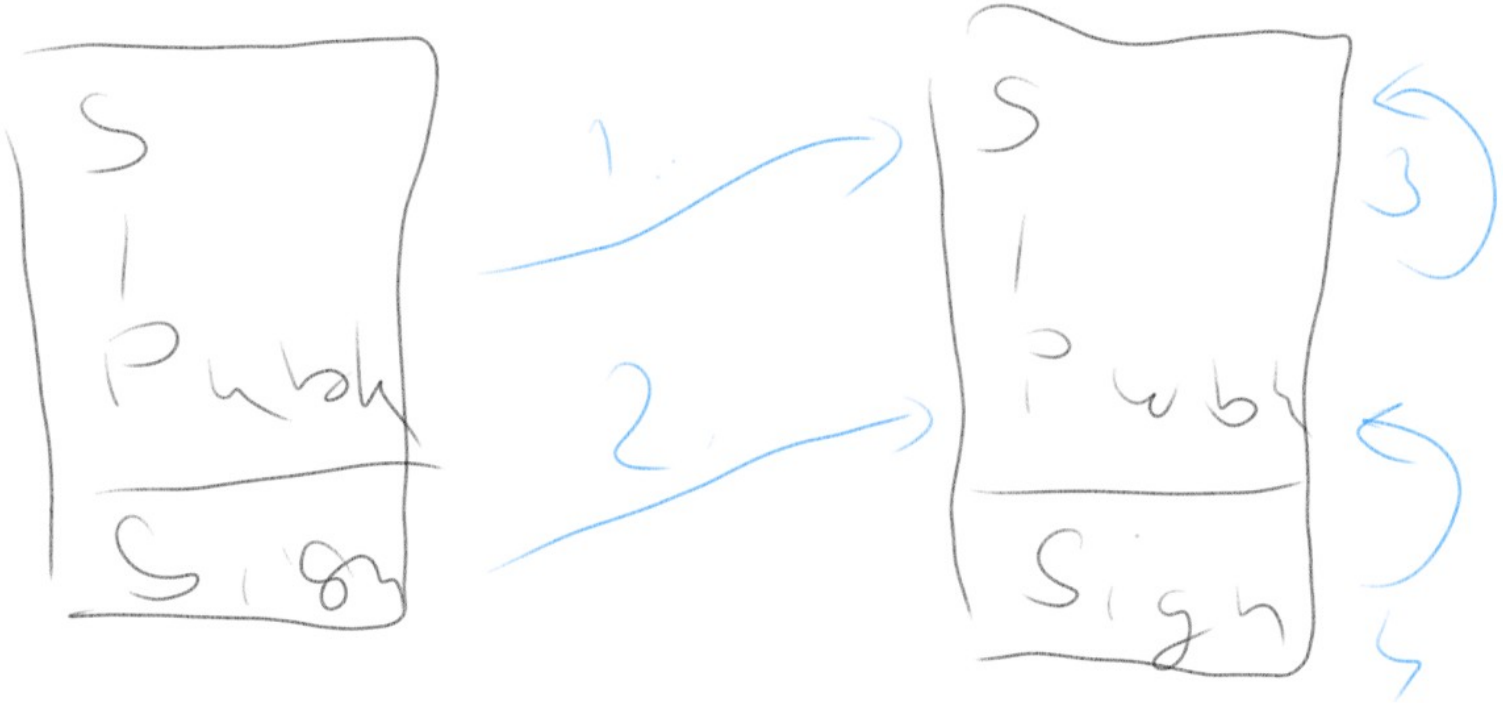
- obrázek vydá za tisíce slov ...

Certifikáty

- Kořenový, root, self signed, “podepsaný sám sebou”,

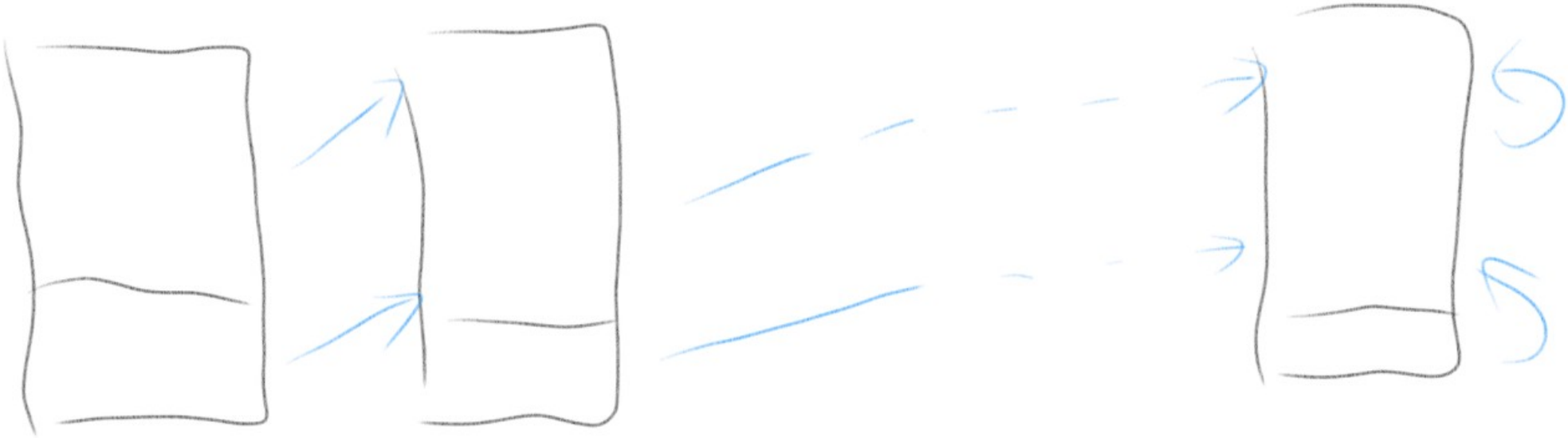
Certifikáty

- Koncový a jeho vydavatel (zde CA s kořenovým certifikátem)



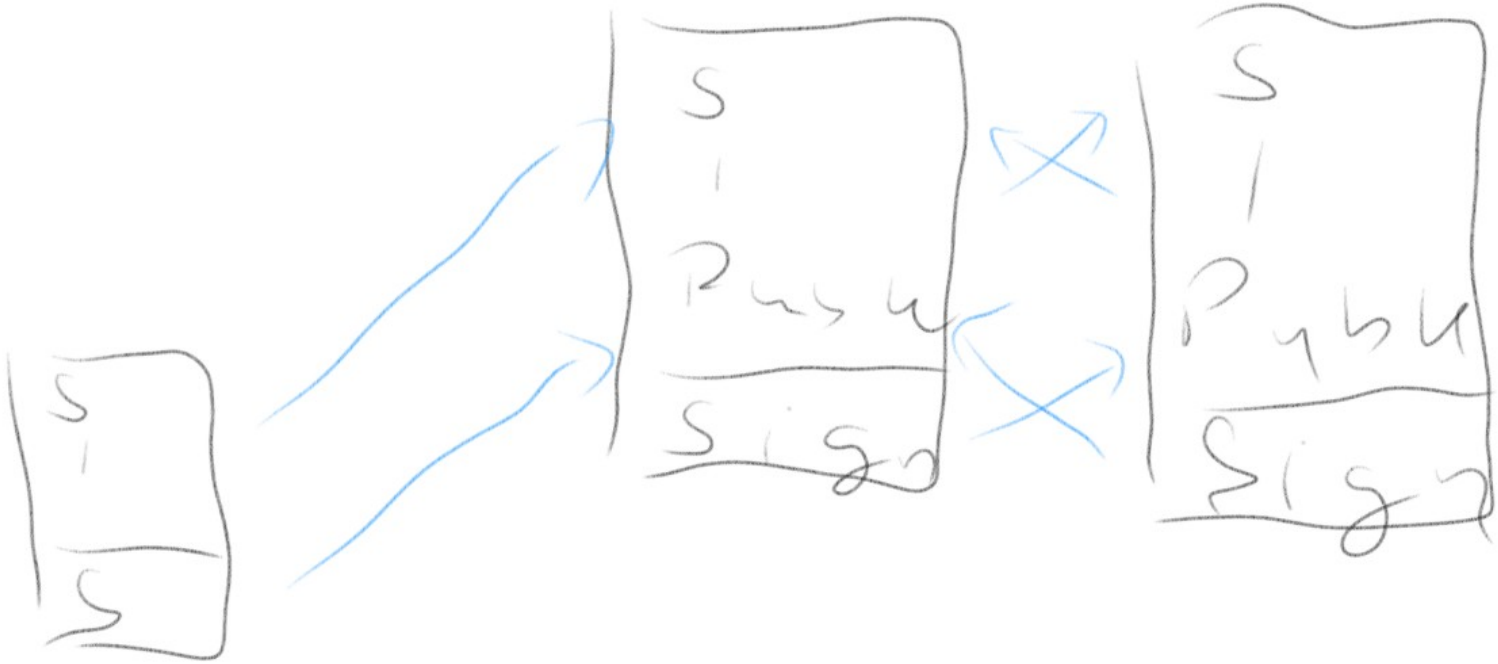
Certifikáty

- Řetízek certifikátů, certificate chain
  - Vydavatelé: Intermediate CA, mezilehlá CA



Certifikáty

- Křížem podepsané CA



Certifikáty

- Koncový certifikát
  - Kořenový, root, “self signed” (Subject a Issuer jsou stejné)
    - Podpis se ověří veřejným klíčem v certifikátu
    - obvykle pro testování; nebo je to CA
  - Vydán certifikační autoritou (CA)
    - Podpis se ověří veřejným klíčem z certifikátu vydavatele (CA)
    - chain (řetízek) CA - “rekurzivně”
      - Zřetězení podle vydavatele (ověřujeme jeho veřejným klíčem)
      - Poslední je
        - Kořenový, root, “self signed” (Subject a Issuer jsou stejné)
          - Podpis se ověří veřejným klíčem v certifikátu
        - 2 CA podepsaná křížem
          - Teoreticky by šel udělat kruh CA
          - Podpis se ověří veřejným klíčem v certifikátu druhé CA (nebo dle kruhu)

Certifikáty

- Ověření certifikátu
  - Podpis a neporušenost
  - Platnost (teď v čase šifrování; pro podpisy v čase vzniku dokumentu)
  - Ověření zneplatnění
    - Online (nepřítomnost na CRL, individuálně OCSP)
    - Offline (problém mj. současných web prohlížečů)
  - A dále stejně ověřit platnost každého certu v řetízku vydavatelů
  - Kořenový certifikát je/není nainstalovaný v důvěryhodných
    - Úložiště - pro aplikaci/subsystém/operační systém
    - Služba má posílat řetízek od koncového všechny intermediate, bez kořenového
      - varianta- důvěra nastavená pro některou intermediate
  - Vztah ke službě či uživateli (jméno web serveru apod.)
    - Subject a rozšíření subjectAltName (alternativní jména)

Certifikáty

- Pár klíčů
  - Po vygenerování “zaměnitelné”
  - Musí být doplněn identifikací (jméno-typ algoritmu)
  - Privátní (soukromý) – dobře uschovat
    - Pro šifrování je použit k odtajnění dokumentu
    - Pro podpis je použit k zašifrování otisku
  - Public (veřejný) – klidně na internet (web, mail, ...) a sdílet
    - Kdokoli s ním může šifrovat data
    - Kdokoli s ním může ověřit podpis (dešifruje jej a následuje porovná s otiskem)

Certifikační autority

- Certifikační autorita (rychlý náhled)
  - Funkce
    - vydává certifikáty
    - Skladba
      - Registrační autorita
      - Certifikační autorita
      - Správní autorita
  - Za co ručí
    - Technicky
      - jednoznačnost sériových čísel
      - (někdy jednoznačnost Subject)
    - Politicky
      - důvěryhodnost

Certifikační autority

- Certifikační autorita (rychlý náhled)
  - Proč jich je tolik, nestačila by jedna?
  - Třídy
    - 1 ručí za jednoznačnost
    - 2 plus kontroluje totožnost uživatele, bezp. HW
    - 3 jako 2, ale používají se pro konkrétní aplikaci/systém
  - Bezpečný HW
    - Izolace
    - Zajištení pro detekci narušení (podepsaná papírová páska...)
  - Privátní klíč
    - Ztráta/zničení … “OK”
    - Zcizení … 😈

Certifikační autority

- Implementujeme CA
  - Generování páru klíčů
  - Vytvoření žádosti o certifikát
  - Vydání certifikátu
  - Uložení ve vhodné formě

Certifikační autority

- Implementujeme CA
  - Kroky pro CA i klientský certifikát jsou podobné, proto budou následovat společně
  - Běžné parametry openssl
    - -in -out -key – vstupní výstupní soubor, soubor s privátním klíčem
    - -inform -outform -keyform (PEM DER) – formát souboru

Certifikační autority

- Implementujeme CA
  - Generování páru klíčů

```
# private keys
openssl genrsa -out MojeCA.key -des3 1024
openssl genrsa -out TrustedClient.key -des3 1024
```

Certifikační autority

- Implementujeme CA
  - Vytvoření žádosti o certifikát

```
#certificate requests
openssl req -outform PEM -out MojeCA.csr -new -key MojeCA.key
openssl req -outform PEM -out TrustedClient.csr -new -key TrustedClient.key
```

Certifikační autority

- Implementujeme CA
  - Vydání certifikátu

```
#certificates
openssl x509 -req -inform PEM -in MojeCA.csr \
        -outform PEM -out MojeCA.crt \
        -signkey MojeCA.key -keyform PEM \
        -CAcreateserial -CAserial MojeCA.srl -set_serial 1 \
        -days 365
echo "02" > MojeCA.srl  # fallback
openssl x509 -req -inform PEM -in TrustedClient.csr \
        -outform PEM -out TrustedClient.crt \
        -CA MojeCA.crt -CAkey MojeCA.key \
        -days 365
```

Certifikační autority

- Implementujeme CA
  - Uložení ve vhodné formě

```
#p12
openssl pkcs12 -export -out TrustedClient.p12 \
        -in TrustedClient.crt -inkey TrustedClient.key -name TrustedClient
#PEM
openssl pkcs12 -in TrustedClient.p12 -out TrustedClient.pem
```

Certifikační autority

- Dotazy, diskuse