

## CERTIFIKÁT X.509

# Kdopak se to podepsal?

Nedávno přijatý zákon o elektronickém podpisu rozhybal poněkud stojaté vody české „informační společnosti“ a státní správy. Nad digitálním (nebo chcete-li zaručeným elektronickým) podpisem však dokonce i v odborné veřejnosti dosud přetrvává řada otázek, mezi nimi i ona naznačená v titulku. Pokusíme se ji objasnit.

**K**líčovou část zákona o elektronickém podpisu představují certifikáty. Je tomu tak proto, že certifikát svazuje informace o signatáři s jeho veřejným klíčem. Pomocí veřejného klíče, který je v certifikátu podepsaného uveden, lze zkontrolovat správnost elektronického podpisu a z dalších údajů v certifikátu zjistit jeho identitu. Podíváme se proto blíže na obsah certifikátu, jeho formát a technickou stránku. I když formátů certifikátu existuje více, nejpoužívanějším je certifikát podle normy X.509 verze 3. Vzhledem k tomu, jak se vyvíjí situace na českém trhu, je víceméně jisté, že zde bude prakticky jediný používaný.

### STANDARD X.509

Norma X.509 je součástí třídy norem X.500 – X.599 mezinárodní telekomunikační unie (ITU), které se zabývají tzv. adresářem (DIRECTORY) a adresářovými službami. Adresář je vlastně globální seznam, podle něhož lze jakékoliv entitě na světě (například státu, osobě, pracovní stanici, serveru, ledničce,...) přiřadit jednoznačné jméno prostřednictvím hierarchických struktur jmen (viz obr. 2). X.509 se pak zabývá vzájemnou autentizací těchto entit. Definuje několik způsobů autentizace, z nichž tzv. *silná autentizace* je založena na digitálních podpisech a certifikátech. Obsah certifikátu podle normy X.509 v.3 je popsán pomocí abstraktního jazyka ASN.1 (viz infotypy) a je vidět v rámečku 1.

```
Version: 3 (0x2)
Serial Number: 1758 (0x6de)
Signature Algorithm: sha1WithRSAEncryption
Issuer:
  O=KPNQwest International, OU=(CZ) KPNQwest Czechia,
  CN=KPNQwest Czechia Public Test CA 2000
Validity
  Not Before: Oct 23 08:09:57 2000 GMT
  Not After : Nov 22 08:09:56 2000 GMT
Subject: C=CZ, CN=RNDr. Vlastimil KLIMA/Email=v.klima@decros.cz
Subject Public Key Info:
  Public Key Algorithm: rsaEncryption
  RSA Public Key: (1024 bit)
    Modulus (1024 bit):
      00:eb:86:d3:.....(zkráceno)..... 4a:0c:68:d8:08:9f:c6:ec:2d
    Exponent: 65537 (0x10001)
X509v3 extensions:
  X509v3 Basic Constraints:
    CA:FALSE
  X509v3 Subject Key Identifier:
    14:4A:C0:8D:CA:BE:99:87:C1:A8:C8:5B:A3:8D:20:8E:1D:42:96:57
  X509v3 Authority Key Identifier:
    keyid:
      86:A0:A7:36:C5:39:A5:B2:3C:19:EB:7F:93:F1:C7:26:BD:23:5E:20
    DirName:/O=KPNQwest International/
  OU=(CZ) KPNQwest Czechia/CN=KPNQwest Czechia Primary Test CA
  serial:01
  X509v3 Subject Alternative Name: email:v.klima@decros.cz
  Netscape Comment: For Testing Only. Pouze pro testovani.
Signature Algorithm: sha1WithRSAEncryption
49:b5:97:1f:15:.....(zkráceno)..... 4c:3e
```

### <1> Definice certifikátu podle X.509 v.3

```
Certificate ::= SIGNED { SEQUENCE {
  version          [0] Version DEFAULT v1,
  serialNumber     CertificateSerialNumber,
  signature        AlgorithmIdentifier,
  issuer           Name,
  validity         Validity,
  subject          Name,
  subjectPublicKeyInfo SubjectPublicKeyInfo,
  issuerUniqueID   [1] IMPLICIT UniqueIdentifier OPTIONAL,
  subjectUniqueID [2] IMPLICIT UniqueIdentifier OPTIONAL,
  extensions       [3] Extensions OPTIONAL }
```

V mnoha publikacích se uvádí definice certifikátu podle rámečku 2. I když zdánlivě vypadá úplně jinak, je to opravdu jenom jiný zápis téhož obsahu. Druhý zápis je více intuitivní, neboť říká, co je certifikát, jasněji: certifikát obsahuje to, co má být podepsáno (*tbsCertificate*), údaje o algoritmu, kterým certifikační autorita (CA) podepisuje tento obsah (*signatureAlgorithm*) a na konec vlastní podpis certifikační autority (*signature*).

V dalším textu si projdeme jednotlivé položky certifikátu podle rámečku 2.

### <2> Používanější zápis certifikátu X.509 v.3

(TBS znamená „To Be Signed“ čili podepisovaný obsah)

```
Certificate ::= SEQUENCE {
  tbsCertificate TBSCertificate,
  signatureAlgorithm AlgorithmIdentifier,
  signature      BIT STRING }
```

Obr. 1. Výpis certifikátu

```
TBSCertificate ::= SEQUENCE {
    version          [0] Version DEFAULT v1,
    serialNumber     CertificateSerialNumber,
    signature        AlgorithmIdentifier,
    issuer           Name,
    validity         Validity,
    subject          Name,
    subjectPublicKeyInfo SubjectPublicKeyInfo,
    issuerUniqueID   [1] IMPLICIT UniqueIdentifier OPTIONAL,
                    -- je-li použito, verze musí být v2 nebo v3

    subjectUniqueID [2] IMPLICIT UniqueIdentifier OPTIONAL,
                    -- je-li použito, verze musí být v2 nebo v3

    extensions      [3] Extensions OPTIONAL
                    -- je-li použito, verze musí být v3 }
```

#### PODPIS CERTIFIKAČNÍ AUTORITY

Na vlastním (digitálním) podpisu CA není nic zajímavého, je to řetězec typu BIT STRING (univerzální datový typ ASN.1, viz minulý díl), který uzavírá certifikát.

#### IDENTIFIKÁTOR PODEPISOVACÍHO ALGORITMU

Položka *signatureAlgorithm* je datovým typem **AlgorithmIdentifier** a je mnohem zajímavější. V certifikátu se vyskytuje na více místech a jako da-

tový typ se používá velmi často, neboť slouží k identifikaci jakéhokoliv algoritmu – šifrovacího, podepisovacího, verifikačního, asymetrického, klasického apod. Jeho obecná syntaxe (rámeček 3) umožňuje začlenit vše potřebné, a proto jen říká, že se tento datový typ skládá z identifikátoru algoritmu a jeho parametrů.

Protože každý algoritmus, který obdrží identifikátor, má definovanou syntaxi a sémantiku svých parametrů, z jeho identifikátoru se odvodí význam parametrů. V parametrech algoritmu se dají přenášet různá potřebná doplňková data – u asymetrických šifer to může být veřejný exponent, řád grupy, generátor, modul; u symetrických algoritmů počet rund, délka klíče, sůl; u hašovacích technik počet opakování hašovací funkce, sůl nebo náhodný seed apod. (viz např. články v Chipu 9/00 a 11/00 o formátování dat pro digitální podpis

```
<3> Identifikátor algoritmu
AlgorithmIdentifier ::= SEQUENCE{
    algorithm  ALGORITHM.&id({SupportedAlgorithms}),
    parameters ALGORITHM.&Type({SupportedAlgorithms}
                               {@algorithm})OPTIONAL}
```

```
SupportedAlgorithms ALGORITHM ::= { ... }
```

```
Příklad: SEQUENCE {
    OBJECT IDENTIFIER  sha1withRSAEncryption (1 2 840 113549 1 1 5)
    NULL }
```

pomocí PKCS#1). U certifikátů se často setkáme s podepisovacím algoritmem RSA v kombinaci s hašovací funkcí SHA-1 (viz tamtéž a příklad v rámečku 3).

#### VERZE

Položka version je datovým typem **Version**, který je ve stylu ASN.1 definován jako

```
Version ::= INTEGER{ v1(0), v2(1), v3(2) },
```

kde INTEGER je univerzální datový typ.

Znamená číslo verze certifikátu a pokud není uvedena, má hodnotu v1 (verze 1). Dnes je nejpoužívanější verze 3, tj. hodnota v3 (= 2), jak vidíte i na obrázku 1. Připomeňme, že verze 2 zavedla nově položky *issuerUniqueIdentifier* a *subjectUniqueIdentifier*, aby v nich mohla uvádět další údaje o vlastníkově a vydavateli certifikátu (alternativní jména, která nejsou podle X.5xx). To ale nestačilo, a tak verze 3 tyto položky ve skutečnosti přesunula do mnohem širší a bohatší struktury, která se nazývá rozšíření (*extensions*), viz dále.

#### SÉRIOVÉ ČÍSLO

Položka *serialNumber* je typu **CertificateSerialNumber** ::= INTEGER. Je to celé číslo, které certifikátu přiřazuje certifikační autorita. Ta musí zajistit, že je jedinečné v rámci jí přidělovaných čísel. Může se sice stát, že dvě různé certifikační autority vydají stejné sériové číslo, ale dvojice *jméno CA – sériové číslo* určuje jedinečný certifikát.

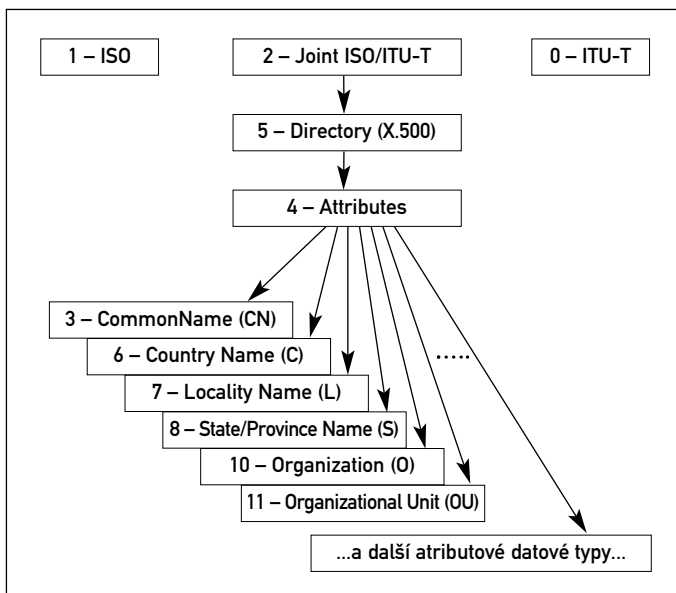
<4> Jméno (subjektu nebo certifikační autority)

```
Name ::= CHOICE{ distinguishedName RDNSequence }
RDNSequence ::= SEQUENCE OF RelativeDistinguishedName
RelativeDistinguishedName ::= SET OF AttributeValueAssertion
AttributeValueAssertion ::= SEQUENCE{ type AttributeType,
value AttributeValue }
AttributeType ::= OBJECT IDENTIFIER
AttributeValue ::= ANY DEFINED BY AttributeType
```

#### JMÉNO VYDAVATELE (CERTIFIKAČNÍ AUTORITY)

Jméno vydavatele certifikátu (*issuer*) je v ASN.1 definováno jako typ **Name** podle rámečku 4 a platí současně i pro jméno subjektu. Toto JMÉNO (viz X.500, X.501), jehož konstrukce se zdá být složitá, je v podstatě jen posloupností několika jmen *RelativeDistinguishedName* definovaných v globální databázi (DIRECTORY), které v certifikátu identifikují vydávající certifikační autoritu. Je to datový typ, který je určen příslušnými normami. Může jím být například jméno a příjmení (označen CN – Common Name), jméno státu (C – Country Name), jméno lokality (L – Locality Name), jméno organizace (O – Organization), jméno organizační jednotky (OU – Organizational Unit) apod.

*RelativeDistinguishedName*, jak vidíme z definice, je konečnou množinou hodnot **AttributeValueAssertion**, které jsou tvořeny posloupností složene z objektového identifikátoru a řetězce. Objektový identifikátor říká, jaký má uložený řetězec význam. Identifikátorů jsou stovky a postup přiřazování identifikátorů jménům je vidět na obrázku 2. Strom identifikátorů začíná od nejvyšších vydávajících autorit (jsou tři: ISO, ITU-T a společné normy obou), potom následuje číslo normy (zde X.500), dále označení třídy jmen (jedná se o tzv. atributové typy v rámci X.500) a poté následují už konkrétní jména (jsou to datové typy). Tak například identifikátor pro datový typ „organizace“ je 2.5.4.10.



Obr. 2. Strom objektových identifikátorů

(ve schématu rozvětvena až úroveň „jmen“; jejich identifikátory jsou konstruovány společnou částí 2.5.4.\*, kde \* znamená číslo ve 4. úrovni)

#### PLATNOST CERTIFIKÁTU

Platnost certifikátu je uvedena v položce *Validity*, definované jako posloupnost dvou časů,

```
Validity ::= SEQUENCE {notBefore Time, notAfter Time},
```

kteří označují počátek a konec platnosti certifikátu. Čas se udává buď jako univerzální (UTC), nebo jako Greenwich Mean Time (GMT):

```
Time ::= CHOICE {utcTime UTCTime, generalizedTime GeneralizedTime }.
```

V obou případech se jedná se o univerzální datový typ (blíže viz minulý díl).

<5> Definice rozšíření podle ASN.1

```
Extensions ::= SEQUENCE OF Extension
Extension ::= SEQUENCE{
    extnId      EXTENSION.&id({ExtensionSet}),
    critical    BOOLEAN DEFAULT FALSE,
    extnValue   OCTETSTRING }
```

Příklad standardního rozšíření (ukazuje možnosti použití certifikovaného klíče):

```
keyUsage EXTENSION ::= {
    SYNTAX      KeyUsage
    IDENTIFIED BY id-ce-keyUsage }
```

```
KeyUsage ::= BIT STRING {
digitalSignature (0),      nonRepudiation (1),
keyEncipherment (2),      dataEncipherment (3),
keyAgreement (4),         keyCertSign (5),
cRLSign (6),               encipherOnly (7),
decipherOnly (8) }
```

### JMÉNO SUBJEKTU

Jméno subjektu (*subject*) je datový typ **Name**, který už známe. Toto jméno identifikuje entitu (subjekt), jehož veřejný klíč je certifikován a uveden v položce *subjectPublicKeyInfo*. Pro každý subjekt musí být toto jméno jedinečné v rámci dané certifikační autority (CA). Je sice možné, aby CA vydala témuž subjektu (s tímž jménem) několik certifikátů, ale ty se budou lišit sériovými čísly. Také příslušné veřejné klíče budou pravděpodobně určeny k jiným účelům. Jméno subjektu nemusí být ve verzi 3 vyplněno a může být uvedeno v položce *subjectAltName* v *extensions*.

### VEŘEJNÝ KLÍČ SUBJEKTU

Veřejný klíč subjektu je společně se jménem subjektu tou nejpodstatnější informací v certifikátu. Má datový typ **SubjectPublicKeyInfo** a je definován jako

```
SubjectPublicKeyInfo ::= SEQUENCE{algorithm AlgorithmIdentifier,
subjectPublicKey BITSTRING}.
```

V tomto poli je jednak uložen vlastní veřejný klíč, a jednak identifikátor algoritmu, ke kterému patří, obojí viz obrázek 1.

### VOLITELNÉ IDENTIFIKÁTORY SUBJEKTU A VYDAVATELE

Položky *issuerUniqueID* a *subjectUniqueID* jsou volitelné ve verzi 2 a 3. Jsou datovým typem **UniqueIdentifier**, který je definován jako

```
UniqueIdentifier ::= BITSTRING
```

neboli (obecný) bitový řetězec. Měly obsahovat přídatnou informaci o subjektu a vydavateli, ale ve verzi 3 jejich úlohu přebírají rozšíření.

### ROZŠÍŘENÍ – CESTA K FLEXIBILITĚ

Pro různé účely certifikátů byla zavedena různá rozšíření, přičemž jejich množina zůstává stále otevřená. Ustálila se sice množina *standardních rozšíření*, pro různé specifické účely certifikátů je však možné vybrat jiné kombinace rozšíření, což umožňují tzv. *certifikační profily*. Ty potom určují, jaká rozšíření mají být přítomna, jaký mají obsah, jaký je jejich význam a jak se využívají v daném informačním systému (např. profil pro internetové použití, viz infotypy). V rozsáhlých organizacích tak lze například pomocí těchto rozšíření budovat řízený přístup (zaměstnanců, obchodních partnerů nebo zákazníků) k informačnímu systému organi-

zace (intranetu i extranetu). Postačí určovat práva každého podle obsahu příslušného rozšíření (například „funkce v organizaci“). Je-li tam kód zaměstnance odboru X, bude mu například zpřístupněn jen pohled na web organizace do jeho veřejné části a do části, týkající se odboru X a třeba databáze XY.

Rozšíření jsou konstruována velmi flexibilně. Umožňuje to jednak velká množina standardních rozšíření, ale zejména neomezená možnost přidávat nová *proprietární rozšíření*, aniž by se měnila definice certifikátu. Pochopitelně je tu otázka kompatibility, protože proprietárním rozšířením nebude kromě uzavřené komunity nikdo jiný rozumět. To v zásadě nemusí vadit, protože každé rozšíření má u sebe *příznak kritičnosti*, který říká, zda jde o závažné rozšíření; hlavním pravidlem přitom je, že aplikace, které certifikát zpracovávají a nerozpoznají některé kritické rozšíření, musí z dalšího zpracování certifikát vyloučit. Na druhé straně, pokud se jedná o nekritické rozšíření a aplikace mu nerozumí, lze ho ignorovat. Definice a příklad rozšíření podle ASN.1 je vidět v rámečku 5.

### ZÁVĚR

Seznámili jsme se ve stručnosti s nepoužívanějším formátem certifikátu podle normy X.509 verze 3. Zajímavou vlastností těchto certifikátů je jejich otevřenost prostřednictvím položky *extensions*. Díky ní certifikáty umožňují nejen realizovat elektronický podpis, ale i řadu dalších zajímavých funkcí pro informační systémy. | | | Vlastimil Klíma | v.klima@decros.cz

| infotypy   |
|--|
| <p><b>Standard X.509</b></p> <ul style="list-style-type: none"> <li>▶ ITU-T Recommendation X.509 (06/97) – Information technology – Open Systems Interconnection – The directory: Authentication Framework, ITU, 1997</li> </ul>   |
| <p><b>Profily certifikátů pro široké internetové použití – RFC 2459:</b></p> <ul style="list-style-type: none"> <li>▶ Internet X.509 Public Key Infrastructure: Certificate and CRL Profile, na <a href="ftp://ftp.isi.edu/in-notes/rfc2459.txt">ftp://ftp.isi.edu/in-notes/rfc2459.txt</a></li> </ul> |
| <p><b>O podpisu pomocí RSA:</b></p> <ul style="list-style-type: none"> <li>▶ Bezpečné použití RSA, Chip 11/00, str. 52 – 56.</li> </ul>  |
| <p><b>O ASN.1:</b></p> <ul style="list-style-type: none"> <li>▶ Jak popsat data, Chip 12/00, str 62 – 65.</li> </ul>   |
| <p><b>Články naleznete také na</b></p> <ul style="list-style-type: none"> <li>▶ <a href="http://www.decros.cz/Security_Division/Crypto_Research/archiv.htm">www.decros.cz/Security_Division/Crypto_Research/archiv.htm</a></li> </ul>  |

**profesionální CAD systém pro strojírenství**

# VariCAD®



**3D – objemový modelář**  
**Kompletní 2D dokumentace**  
**Editovatelné grafické knihovny**  
**Výpočty strojních součástí**  
**Automatická tvorba kusovníků**  
**Rozviny**  
**Prostředí WIN NT/95/98/2000, UNIX, LINUX**  
**Nejlepší poměr cena/výkon**  
**Množstevní slevy**  
**E-mail: [posta@varicad.cz](mailto:posta@varicad.cz)**  
**<http://www.varicad.cz>**  
**Obchodní odd.: VariCAD s. r. o.**  
**Tř. 1. máje 52, 461 74 Liberec 3**  
**Tel./fax: 048 511 3735**



**Tak snadno  
jste ještě netvořili!**

placená inzercie