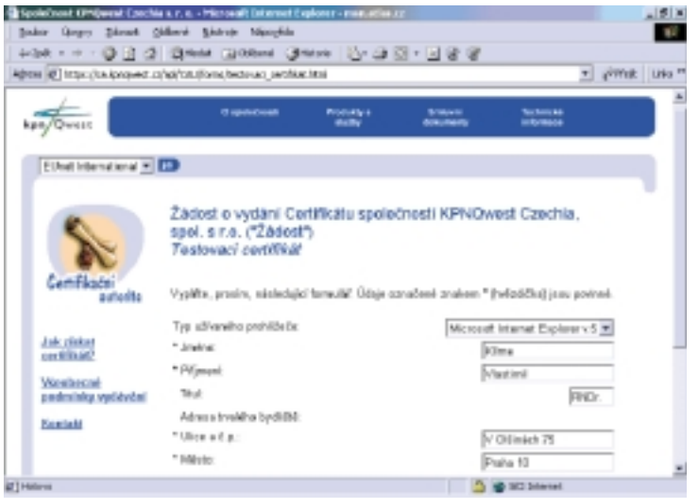
PKCS#10

Žádáme o certifikát

Abychom mohli používat zaručený elektronický podpis, musíme mít potřebný certifikát (popsali jsme jej v minulém dílu našeho seriálu). Znamená to spojit se nejprve s certifikační autoritou (poskytovatelem certifikačních služeb) a o certifikát zažádat. Jak žádost o certifikát v nejpoužívanějším formátu podle normy PKCS#10 vlastně vypadá, si vysvětlíme nyní.

Při vytváření žádosti o certifikát používáme vždy nějaký program, jehož pomocí zadáváme příslušné údaje, které se mají objevit v certifikátu, a zároveň hned můžeme vygenerovat svůj pár klíčů (tajný – podepisovací a veřejný – ověřovací). Žádost o certifikát za nás program připraví sám; předtím nás ovšem vyzve, abychom svým – třeba právě „novopečeným" – podepisovacím klíčem žádost podepsali. Výsledkem našeho úsilí a činnosti zmíněného programu pak je (námi podepsaný) soubor dat reprezentující žádost o certifikát.

Některé certifikační autority (CA), které už u nás zahájily činnost, nabízejí vydání **testovacích certifikátů** zdarma, což oceníme při zkoušení vlastností naší budoucí „podepsané" komunikace. Na obrázku 1 vidíte příklad uživatelského rozhraní pro získání takového certifikátu. Po odeslání vyplněného formuláře (prohlížeč, v tomto případě MSIE, po stisku příslušného tlačítka sám odešle žádost o certifikát v on-line režimu) nám certifikační autorita vydá testovací certifikát. Téměř okamžitě po odeslání žádosti obdržíme e-mail s pokyny, jak certifikát nainstalovat (viz obr. 2); po několika klepnutích myší je pak celá záležitost ukončena. Snadné, že? U „ostrých" certifikátů to ovšem už tak jednoduché nebude – jak celá procedura vydání certifikátu vypadá, hodně záleží na certifikační autoritě a její certifikační politice.



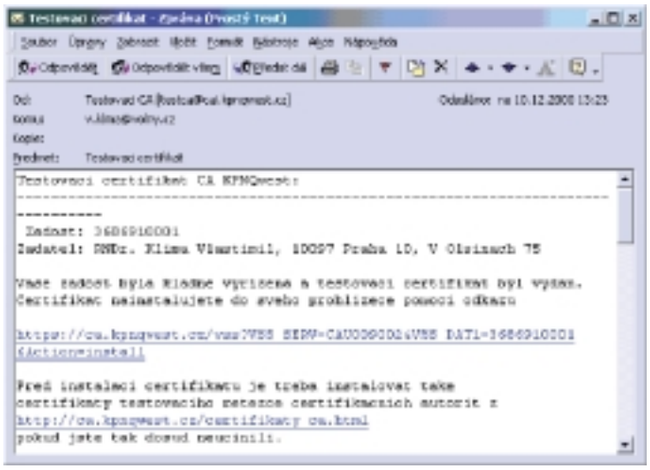
Obr. 1. Žádost o certifikát

Standard PKCS#10, verze 1.7

V dalším textu se seznámíme s velmi „čerstvou" (přijatou poměrně nedávno – 26. 5. 2000) verzí standardu PKCS#10 pro tvorbu žádosti o certifikát. Jedná se o verzi 1.7, která nahradila sedm let platnou verzi 1.0. Než přejdeme k jejímu obsahu, řekneme si, v čem byly nutné změny. Jde zejména o změny v odkazu na nové normy ITU-T a X.680 – X.690, které aktualizují definici jazyka ASN.1 a jeho kódovací pravidla (o ASN.1 jsme psali v Chipu 11/00, viz infotipy). Z normy byly dále odstraněny veškeré odkazy na PKCS#6, která už není podporována – PKCS#6 totiž zaváděla vlastní cestu pro tzv. rozšíření (extensions) v certifikátech, ta však byla vytlačena normou X.509 v.3, jak jsme o ní psali minule (Chip 1/01, viz infotipy).

Pevná pravidla, proměnný obsah

Standard popisuje **syntaxi** žádosti o certifikát. Žádost vždy obsahuje *jméno subjektu*, jeho *veřejný klíč* (s identifikátorem algoritmu, pro nějž byl vytvořen) a volitelně množinu *atributů*. Za těmito údaji následuje (zaručený) elektronický podpis, jímž žadatel o certifikát tyto údaje stvrzuje. Podpis přitom vytváří svým privátním klíčem, který patří k veřejnému klíči uvedenému v žádosti. Certifikační autorita, která žádost převezme, tak má jistotu, že žadatel měl k veřejnému klíči, o jehož cer-



Obr. 2. Certifikát přijde elektronickou poštou a podle pokynů se nainstaluje do systému.

tifikaci žádá, k dispozici i jeho privátní doplněk, neboť platnost podpisu může ověřit uvedeným veřejným klíčem.

Certifikační autorita pak ze žádosti vybere příslušná data (zejména jméno subjektu a jeho veřejný klíč), doplní své jméno, sériové číslo a další položky (blíže viz minulý díl), celé to podepíše a převede na certifikát podle normy X.509. Obsah žádosti o certifikát je vidět v rámečku 1. Zavedení položky atributů v žádosti má dvojí smysl. Sem je totiž možné přidat nové informace o žadateli, které se mohou objevit v položce *extensions* v certifikátu, a také informace, jež žadateli později umožní tento certifikát odvolat (revokovat). Řada používaných standardních rozšíření je definována v normě PKCS#9.

‹1› Syntaxe žádosti o certifikát

```
CertificationRequest ::= SEQUENCE {
certificationRequestInfo          CertificationRequestInfo,
signatureAlgorithm          AlgorithmIdentifier{{ SignatureAlgorithms }},
signature          BIT STRING}

CertificationRequestInfo ::= SEQUENCE {
version          INTEGER { v1(0) } (v1,...),
subject          Name,
subjectPKInfo          SubjectPublicKeyInfo{{ PKInfoAlgorithms }},
attributes          [0] Attributes{{ CRIAttributes }}}

AlgorithmIdentifier {ALGORITHM:IOSet } ::= SEQUENCE {
algorithm          ALGORITHM.&id({IOSet}),
parameters          ALGORITHM.&Type({IOSet}{@algorithm}) OPTIONAL}

SubjectPublicKeyInfo { ALGORITHM : IOSet} ::= SEQUENCE {
algorithm          AlgorithmIdentifier {{IOSet}},
subjectPublicKey          BIT STRING}

PKInfoAlgorithms ALGORITHM ::= { ...-- libovolný definovaný algoritmus --}

Attributes { ATTRIBUTE:IOSet } ::= SET OF Attribute{{ IOSet }}

CRIAttributes  ATTRIBUTE  ::= {... -- libovolné definované atributy --}

Attribute { ATTRIBUTE:IOSet } ::= SEQUENCE {
type          ATTRIBUTE.&id({IOSet}),
values          SET SIZE(1..MAX) OF ATTRIBUTE.&Type({IOSet}{@type})}
```

Obsah žádosti

Jak vidíme v rámečku 1, žádost se skládá z vlastní „informační části“ *(certificationRequestInfo)*, dále z identifikátoru podpisového algoritmu *(signatureAlgorithm)* a z vlastního podpisu *(signature)*. Položka *certificationRequestInfo* obsahuje jednoznačné jméno subjektu *(distinguished name)*, jeho veřejný klíč a volitelně množinu atributů, které obsahují doplňující informace k subjektu (držiteli). Všechny tyto položky se vytvářejí při vyplňování žádosti nebo jsou v této době k dispozici. Popíšeme si je jednotlivě:

Jedním z **nejpoužívanějších formátů** žádosti o certifikát je PKCS#10, který kromě standardních údajů obsahuje i velmi flexibilní **položku atributů**.

- ***Version*** je číslo verze (pro námi popisovaný standard by to měla být 0).
- ***Subject*** je jednoznačné jméno žadatele o certifikát. Připomeňme, že to není jen jméno a příjmení, ale celá množina jmen (bližší popis datové struktury **Name** viz normu X.501 a minulý díl seriálu). Příklad je vidět ve výpisu žádosti o certifikát v rámečku 2 (položka subject se zde skládá ze čtyř jmen – C, O, CN, T).
- ***SubjectPublicKeyInfo*** obsahuje informaci o veřejném klíči, který je certifikován. V rámečku 1 vidíme, že obsahuje jak identifikátor algoritmu *(AlgorithmIdentifier)*, tak vlastní hodnotu certifikovaného klíče. V identifikátoru algoritmu (jemuž jsme se také podrobně věnovali v minulém dílu) jsou uvedeny i parametry algoritmu. Například v rámečku 2 se jedná o algoritmus RSA s 1024bitovým modulem, parametry jsou modul algoritmu RSA a veřejný exponent. Pokud není algoritmus uveden,

přednastavená hodnota (default) je *md5WithRSAEncryption* (tj. RSA ve spojení s hašovací funkcí MD5).

- ***Attributes*** jsou výše zmiňované atributy. V rámečku 2 nejsou vyplněny (záznam a0:00 znamená prázdné pole), neboť u testovacího certifikátu nebyly žádné atributy požadovány. Atributy jsou, stejně jako identifikátor algoritmu, definovány velmi flexibilně.

Jedním z atributů je tzv. *challengePassword attribute*, který specifikuje heslo, jímž může žadatel žádat revokaci certifikátu (například při ztrátě nebo kompromitaci svého privátního klíče). Data, která jsou v této položce uvedena, pochopitelně nejsou otevřenou hodnotou hesla, které CA vyžaduje na uživateli při odvolávání certifikátu – certifikát je totiž veřejně dostupný a heslo by si pak kdokoliv mohl přečíst. Na druhé straně se zde nekladou žádné omezující podmínky a specifikace obsahu tohoto atributu je plně v rukou certifikační autority. Tomu odpovídá i obecná syntaxe tohoto atributu

```
challengePassword ATTRIBUTE ::= {
WITH SYNTAX DirectoryString {pkcs-9-ub-challengePassword}
EQUALITY MATCHING RULE caseExactMatch
SINGLE VALUE TRUE
ID pkcs-9-at-challengePassword
},
```

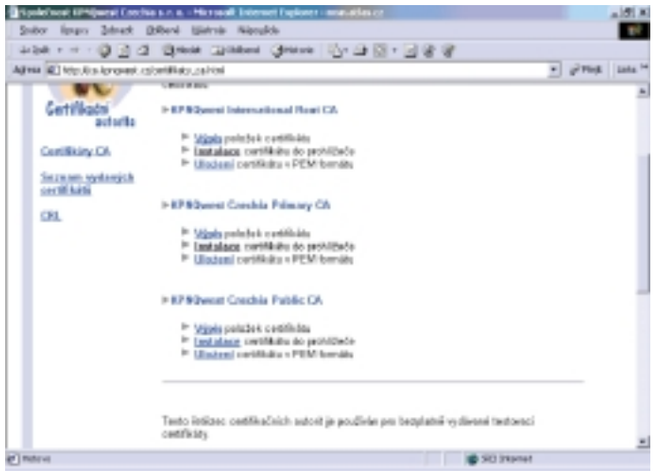
přičemž typ **DirectoryString** je definován v normě X.520.

Požaduje se přitom pouze to, aby každá aplikace zpracovávající tato data byla schopna rozeznat a zpracovat všechny řetězcové typy použité v typu **DirectoryString**, a v něm aby se používaly pokud možno pouze typy **PrintableString**.

Dalším příkladem atributů jsou informace, které se nakonec objeví jako rozšíření vydaného certifikátu (viz *extensions* certifikátu X.509 v. 3). Jedná se o atribut *extensionRequest*, který je definován v PKCS#9, a to tak, že přebírá definici z X.509 v. 3. Díky tomu je množina atributů otevřena dalším definicím, které se mohou vyskytnout v budoucnu.

‹2› Výpis obsahu žádosti o certifikát

```
Data:
  Version: 0 (0x0)
  Subject:
      C=CZ,
      O=DECROS spol. s r.o. - ICO14499894 - DIC077-14499894,
      CN=RNDr. Vlastimil Klima/Email=v.klima@decros.cz,
      T=kryptolog
  Subject Public Key Info:
      Public Key Algorithm: rsaEncryption
      RSA Public Key: (1024 bit)
          Modulus (1024 bit):
              00:95:62:af:17:8a:28:41:90:
.....zkráceno.....
              bc:cc:78:16:0a:2a:2f:38:37
          Exponent: 65537 (0x10001)
    Attributes:
        a0:00
Signature Algorithm: md5WithRSAEncryption
6d:3b:cc:34:37:11:4d:fe:d5:1c:b0:e9:4e:fb:08:2e:3f:26:
.....zkráceno.....
b9:c5:df:78:89:c4:ae:6b:26:3c:21:1b:c6:1d:0e:94:af:85
```



Obr. 3. Instalace certifikátu

Vydání certifikátu

Jakmile CA ověří správnost podpisu žádosti o certifikát a další údaje, vydá certifikát a vhodným způsobem ho také zveřejní. Obsah certifikátu jsme popsali v minulém dílu, ale standard PKCS#10 se už nezabývá tím, v jakém formátu jej certifikační autorita vydává. Většinou se nabízejí tři formáty exportu: jeden podle PKCS#7, druhý ve formátu DER podle X.509 a třetím je týž formát, na nějž je navíc aplikováno překódování do šestibitových znaků (tzv. kódování *base64*).

Pokud je certifikát vydán podle PKCS#7, objeví se jako soubor s koncovkou *p7b* (PKCS#7 binary); jde pak o datový typ **signedData**, v němž lze kromě vlastního uživatelova certifikátu předávat i množství certifikátů dalších. Může to být například celá certifikační cesta, vedoucí od uživatele přes řadu nadřízených autorit až ke kořenové certifikační autoritě, nebo různé křížové certifikáty a také tzv. CRL *(Certificate-Revocation List)*, což je seznam zneplatněných (revokovaných) certifikátů.

Poznamenejme ještě, že nová verze žádosti o certifikát už není kompatibilní se žádostí ve formátu pro PEM (Privacy-Enhanced Mail, viz RFC 1424).

Závěr

Právě jsme se seznámili se standardem PKCS#10 verze 1.7, který definuje syntaxi a sémantiku jednoho z nejpoužívanějších formátů žádosti o certifikát. Tato žádost obsahuje kromě standardních údajů i atributy přenášené jako rozšíření do následně vydaného certifikátu podle normy X.509 verze 3. Tyto atributy umožňují flexibilní použití normy PKCS#10.

Vlastimil Klíma | v.klima@decros.cz

infotipy

O ASN.1:
- Jak popsat data, Chip 12/00, str. 62 – 65.

Formát certifikátů podle normy X.509 v.3:
- Kdopak se to podepsal?, Chip 1/01, str. 130 – 133.

Standard PKCS#10:
- *www.rsasecurity.com/rsalabs/pkcs/pkcs-10/index.html*

Uvedené články naleznete také na *www.decros.cz/Security_Division/Crypto_Research/archiv.htm*