UNIVERZA V LJUBLJANI
EKONOMSKA FAKULTETA

MAGISTRSKO DELO

DOLGOROČNA DOKAZILA O VERODOSTOJNOSTI ELEKTRONSKIH DOKUMENTOV V XML ZAPISU

LJUBLJANA, MAJ 2008 SVETLANA ŠALJIĆ

IZJAVA

Študentka Svetlana Šaljić izjavljam, da sem avtorica tega magistrskega dela, ki sem ga napisala pod mentorstvom prof. dr. Borke Jerman Blažič, in skladno s 1. odstavkom 21. člena Zakona o avtorskih in sorodnih pravicah dovolim objavo magistrskega dela na fakultetnih spletnih straneh.

V Ljubljani, maj 2008 Podpis:_____

Kazalo

Kazalo tabel

Kazalo slik

Kazalo primerov

Uvod

»Nadomestiti elektronski zapis z njegovo popravljeno kopijo, je zelo lahko, in … zamenjava lahko ima daljnosežne učinke.« (Lynch, 1993, str. 68)

Ustvarjanje dokumentov v elektronski obliki je odprlo vprašanje pravno-formalne enakovrednosti elektronske oblike dokumenta papirni. Razmnoževanje in spreminjanje elektronskih dokumentov sta enostavna postopka in zaradi tega predstavljata izkazovanje verodostojnosti in celovitosti velik problem. Kako sem lahko prepričana, da je to kar vidim, izvoren dokument? Kako vem, da dokument ni spremenjen? Na elektronskem dokumentu so možne nenamerne spremembe, na primer zaradi izgube podatkov pri presnemavanju, dobronamerne spremembe v smislu dopolnjevanja podatkov in slabonamerne spremembe, na primer z namenom ponarejanja informacij (Graham, 1994, str. 43).

Problemi arhiviranja elektronskega gradiva so predvsem fizična obstojnost nosilcev, zastarevanje strojne opreme, zastarevanje formata, dokazovanje avtentičnosti in celovitosti podatkov. Tema, ki je obdelana v magistrskem delu, je iz področja dolgoročnega zagotavljanja avtentičnosti in celovitosti elektronskih dokumentov. Poudarek je na kriterijih, ki jim moramo zadostiti, da lahko dolgoročno varno hranimo digitalno podpisano[1] arhivsko gradivo v izključno elektronski obliki. Ključni postopki v tem kontekstu so preverjanje digitalnih podpisov, ohranjanje veljavnosti digitalnih potrdil, možnost dokazovanja časa prejema gradiva v arhiv in časa preverjanja, veljavnosti podpisov ter obnavljanje dokazov o avtentičnosti in celovitosti, ki sčasoma izgubijo vrednost (EESSI, 1999).

V času obdobja arhiviranja elektronskega gradiva lahko postanejo uporabljeni zgostitveni in kriptografski algoritmi nezanesljivi, digitalnim potrdilom iz varnostnih razlogov po petih letih večinoma poteče veljavnost; vse to poveča možnost za poneverbe arhivskih gradiv. Zaradi tega potrebujemo dokaze, da je gradivo pred vstopom v arhiv zares obstajalo, da smo preverili njegovo avtentičnost in da se od takrat ni spremenilo. V ta namen najpogosteje uporabimo časovni žig. Znani mehanizmi za časovno žigosanje imajo podobne težave kot digitalni podpis, zato jih je potrebno znova in znova podaljševati. Podaljšamo ga lahko tako, da pred nastopom nevarnosti preteka veljavnosti dokazil pridobimo nov časovni žig, ki bo

[1] Digitalni podpis je v tej nalogi opredeljen tako, kot je v 2.členu Zakona o elektronskem poslovanju in elektronskem podpisu (ZEPEP) opredeljen varen elektronski podpis. Podrobneje na str. 5.

ščitil gradivo in vse predhodne časovne žige. Zaradi potrebe po izmenjavi dokazil o verodostojnosti stroka potrebuje tako standardiziran podatkovni format kot tudi standardizirane postopke za ustvarjanje časovnih žigov in njihovo podaljševanje.

Splošen standard na tem področju je RFC 4998 (Brandner, 2007a) z naslovom Sintaksa za evidenco dokazil (angl. Evidence Record Syntax), v nadaljevanju standard ERS, ki ga je pripravila mednarodna organizacija IETF (angl. Internet Engineering Task Force). IETF je organ Internet Society in je svetovno priznana organizacija za standardizacijo tehnologij Interneta, tesno sodeluje z organizacijami za standardizacijo W3C (angl. World Wide Web Consortium), ISO (angl. International Organization for Standardization) in IEC (angl. International Electrotechnical Commission). Standard ERS opredeljuje sintakso za zapis in pravila za ustvarjanje dodatnih varnostnih atributov k arhiviranemu gradivu. Ta zapis standard imenuje »evidenčni zapis«. Pravilno ustvarjen evidenčni zapis omogoča, da lahko na ravni tehnologije kadarkoli neizpodbitno dokažemo obstoj in veljavnost digitalnih vsebin ob določenem času v preteklosti ter nespremenljivost digitalnega gradiva za celotno obdobje arhiviranja. Standard ERS opredeljuje evidenčne zapise v formatu ASN.1 (angl. Abstract Syntax Notation One). Na področju elektronskega poslovanja postaja za opis podatkovnih struktur vse pomembnejši razširljivi označevalni jezik XML (angl. Extensible Markup Language), zato se je pojavila potreba po pripravi standarda, ki bi podpiral enakovredne mehanizme zagotavljanja dolgoročne verodostojnosti kot ERS, temeljil pa bi na XML formatu evidenčnih zapisov.

Potreba po sodelovanju različnih aplikacij in izmenjavi podatkov iz različnih sistemov se veča, še posebej z razmahom svetovnega spleta. V uporabi je več jezikov za opisovanje podatkovnih struktur, neodvisno od uporabljene tehnologije, kot so ASN.1 in EDI (angl. Electronic Data Interchange), oba se uporabljata za izmenjavo sporočil v komunikacijskih protokolih, SQL (angl. Structured Query Language) za relacijske baze podatkov, XML in drugi. V povezavi s svetovnim spletom je jedro povezljivih tehnologij postal jezik XML in okrog njega razvite tehnologije (za opredeljevanje strukture, transformacijo podatkov, povezovanje dokumentov, poizvedbe idr.). V zadnjem času vse več aplikacij in standardov na področju internetnih tehnologij temelji na XML strukturi zapisov.

Zaradi razlik v procesnih pravilih za delo z XML in drugih lastnostih XML jezika v primerjavi z ASN.1, je pri prilagoditvi standarda ERS potrebno ob nadomestitvi

ASN.1 podatkovnih struktur z ustreznimi XML strukturami tudi prilagoditi procese ustvarjanja in preverjanja veljavnosti evidenčnih zapisov.

V tej nalogi uporabljam rezultate in izkušnje, ki sem jih pridobila pri delu v tehnološkem centru SETCCE v Ljubljani (www.setcce.si), kjer sem članica delovne skupine, ki je pri IETF odgovorna za pripravo standarda ERS za XML zapis. Predmet magistrske naloge je zagotavljanje dolgoročne verodostojnosti elektronskih dokumentov. Namen magistrske naloge je analiza potreb in obstoječih standardov za ustvarjanje, ohranjanje in preverjanje dokazil o dolgoročni verodostojnosti elektronskih dokumentov ter izdelava ustrezne rešitve pri zagotavljanju le-te. Cilj magistrske naloge je priprava predloga standardnega zapisa dokazil o dolgoročni verodostojnosti elektronskega dokumenta v formatu XML, ki bo skladen s standardom ERS.

Magistrsko delo se v prvem koraku opira na študij in raziskovanje teorije ter empiričnih spoznanj s področij: dolgoročne hrambe elektronskega gradiva in dokazil o verodostojnosti, ANS.1 in XML ter povezanih tehnologij, obstoječih relevantnih standardov. Posebej se ukvarjam s primerjavo med ASN.1 in XML standardoma ter vplivom razlik na iskanje rešitve. Rezultati obeh delov raziskav in analize so izhodišča za pripravo predloga standarda ERS v formatu XML in kriteriji za vrednotenje rešitve. V naslednjem koraku predlagam možne rešitve, ki ustrezajo izhodiščem in kriterijem. Na koncu analiziram predloge in izberem najustreznejšega. Magistrskemu delu prilagam osnutek standarda ERS v formatu XML, ki smo ga napisali Aljoša Jerman Blažič, Svetlana Šaljić in Tobias Gondrom ter je javno objavljen kot predlog za standard na straneh organizacije IETF (http://www.tools.ietf.org/html/draft-ietf-ltans-xmlers-02).

1. Splošne opredelitve

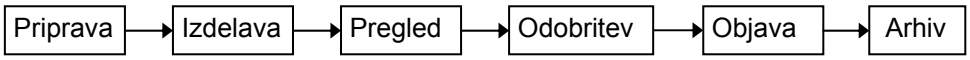
1.1. Elektronski dokumenti in dolgoročna hramba

Pojem dokumenta je z razmahom uporabe računalnikov prešel iz sinonima za papirno potrdilo v sinonim za zapis v digitalni obliki, t. i. datoteka (angl. file), ki vsebuje zapise, podatke, risbe, slike, multimedijske predstavitve idr. V letu 1997 je Mednarodni arhivski svet objavil Smernice za upravljanje elektronskih dokumentov iz arhivske perspektive, ki jih je pripravil Komite za elektronske dokumente v letih od 1993 do 1996. V njih (ICA, 1997, str. 22) opredeljujejo, da je dokument zapisana informacija, ki je oblikovana ali prejeta; je odraz postopka ali zaključek

aktivnosti posameznika ali ustanove ter zajema zadostno vsebino, kontekst in strukturo za zagotovitev dokaza o dejanju. Navadno so elektronski dokumenti predstavljeni kot logično razmejeni informacijski objekti, vendar pa vedno pogosteje najdemo dokumente v obliki porazdeljenih objektov, kot so relacijske baze podatkov. S pojmom elektronski dokument označujemo tiste dokumente, ki so bili prvotno ustvarjeni v digitalni obliki in so v tej obliki ostali skozi njihov celotni življenjski cikel (Novak, 2002, str. 3). V nadaljevanju ne bomo ločevali med dokumenti, ki so izvorno nastali v papirni obliki in so bili nato zajeti v elektronsko obliki (za tovrstne dokumente se uporablja tudi izraz digitalizirano gradivo), ter med izvorno elektronskimi dokumenti.

Za poljuben dokument lahko od trenutka nastanka spremljamo različna stanja. Pot prehajanja med temi stanji imenujemo življenjski cikel dokumenta in na sliki št. 1 prikazujem nekaj možnih stanj od priprave do hrambe dokumenta v arhivu.

Slika 1: Primer življenjskega cikla nekega dokumenta



Arhiviranje je postopek prevzemanja, hranjenja, vzdrževanja, obdelave in uporabe dokumentov ob koncu življenjskega cikla v arhivu organizacije ali posameznika. Arhivirajo se dokumenti, ki so zaključeni in niso več predmet obdelave. Dokumente arhiviramo zaradi različnih potreb in hranimo, dokler ne potečejo roki hranjenja, ki jih narekujejo predpisi in potrebe poslovanja, ali dokler del dokumentarnega gradiva, ki ima značaj arhivskega gradiva (trajni pomen za zgodovino, znanost ali kulturo), ne odberemo ali izločimo pristojnemu javnemu ali zasebnemu arhivu (Žumer, 2001, str. 48). Kadar arhiviramo dokumente v elektronski obliki, govorimo o elektronskem arhivu.

Arhiv mora zagotavljati:

- varnost in trajnost hrambe,
- verodostojnost dokumenta in podpisnikov,
- celovitost (da dokument ni bil spremenjen delno ali v celoti),
- uporabnost oziroma berljivost dokumenta in v primeru podpisanih dokumentov tudi potrditev veljavnosti podpisa v daljšem ali trajnem časovnem obdobju,
- pravno veljavnost arhiviranega dokumenta.

Problemi arhiviranja elektronskega gradiva so predvsem fizična obstojnost nosilcev, zastarevanje strojne opreme, zastarevanje formata, dokazovanje avtentičnosti in stabilnost zapisov oziroma ohranjanje celovitosti podatkov. Varno dolgoročno hrambo elektronskih dokumentov tako zagotavljamo na:

1) ravni zunanje varnosti (okolje informacijskega sistema za hrambo in njegova fizična varnost, omrežna infrastruktura, šifrirane povezave, varnostne kopije, zagotavljanje energentov idr.)
2) ravni notranje varnosti:
    - ohranjanje uporabnosti in dostopnosti gradiva in pripadajočih opisnih podatkov (varovanje podatkov pred izgubo, ohranjanje berljivosti gradiva, verodostojne pretvorbe formatov),
    - zagotavljanje informacijske varnosti (nadzor dostopa, tehnike za zaščito podatkov na osnovi kriptografije),
    - zagotavljanje avtentičnosti in celovitosti (vzdrževanje varnostnih atributov, ki dokazujejo, da je gradivo obstajalo ob določenem času in se od takrat ni spreminjalo; zagotavljanje veljavnosti digitalnih podpisov; časovno žigosanje in obnavljanje časovnih žigov).

Pojem nespremenjenosti ne pomeni nujno ohranjanja bitov elektronskega dokumenta, torej je lahko relativen in naj se tudi v skladu s prednostmi elektronskih oblik dokumentov in pretvarjanja med njimi obravnava bolj fleksibilno (Novak, 2002, str. 3).

1.1.1 Digitalni podpis in infrastruktura javnih ključev

Podpis je v splošnem metoda za potrditev avtorja dokumenta in izraz avtorjevega strinjanja s podpisano vsebino. Elektronski podpis je v skladu z ZEPEP (2. člen) niz podatkov v elektronski obliki, ki je vsebovan, dodan ali logično povezan z drugimi podatki, in je namenjen preverjanju pristnosti teh podatkov in identifikaciji podpisnika; zaporedje znakov, ki je ustvarjeno na tak način, da lahko nedvoumno preverimo, ali se je dokument po podpisu spremenil, in lahko ugotovimo identiteto podpisnika. ZEPEP opredeljuje tudi pojem varnega digitalnega podpisa, ki mora izpolnjevati še zahtevi, da je podpis povezan izključno s podpisnikom in da je ustvarjen s sredstvi za varno elektronsko podpisovanje, ki so izključno pod podpisnikovim nadzorom.

V splošnem se pojem »elektronski podpis« nanaša na vse tehnologije, vključno z biometričnimi, medtem ko se »digitalni podpis« nanaša na elektronski podpis, ustvarjen z uporabo kriptografije (MJU, str. 1). Za digitalni podpis so danes najpogosteje uporabljeni asimetrični kriptografski algoritmi, za katere je eden od parametrov vsebina dokumenta, drugi pa ključ. V nalogi uporabljam izraz digitalni podpis v skladu z zgoraj navedeno opredelitvijo varnega digitalnega podpisa.

Asimetrični kriptografski algoritmi

O asimetričnih kriptografskih algoritmih govorimo, kadar uporabljajo različen ključ za šifriranje kot za dešifriranje (Burnett, 2001, str. 95). Oba ključa sta povezana s konkretno matematično funkcijo, vendar imata lastnost, da iz enega ne moremo izračunati drugega in da, če smo z enim šifrirali, lahko samo z drugim dešifriramo. Uporabljamo jih lahko tako za ustvarjanje digitalnega podpisa kot za šifriranje sporočil.

Pri ustvarjanju digitalnega podpisa uporabimo ključ (imenujemo ga zasebni ključ), ki je znan samo lastniku, medtem ko je ključ, ki je potreben za preverjanje, lahko javno dostopen (imenujemo ga javni ključ). Podpisovanje z uporabo asimetričnih kriptografskih algoritmov poteka v dveh fazah. V prvi fazi z uporabo zgostitvenih funkcij izračunamo prstni odtis dokumenta, v drugi ta prstni odtis šifriramo z zasebnim ključem. Pomembna je varnost zgostitvenega algoritma, saj nihče ne sme znati sestaviti smiselnega dokumenta iz določenega prstnega odtisa. Pri preverjanju se z javnim ključem dešifrira podpis in se preveri, ali se dobljena vrednost prstnega odtisa ujema z izračunom prstnega odtisa dokumenta.

Digitalni podpis je zanesljiv dokaz le v primeru, kadar je zasebni ključ varovan in so uporabljeni varni algoritmi. Algoritem je obravnavan kot varen, kadar obstaja zanemarljivo majhna možnost, da ustvarimo enak podpis oziroma prstni odtis iz spremenjenega dokumenta ali pa da iz podpisa uganemo zasebni ključ. V nasprotnem primeru bi lahko vsebino dokumenta nadomestili s poljubno drugo vsebino ali pa ustvarili podpis brez vednosti lastnika zasebnega ključa. Podpisnik z digitalnim podpisom zagotovi:

- verodostojnost sporočila,
- potrjuje svojo identiteto in s tem
- sprejme tudi odgovornost za sporočilo oziroma potrjuje, da se strinja z njegovo vsebino.

Infrastruktura javnih ključev

Lastnoročni podpis je biološko povezan s podpisnikom, za digitalni podpis pa se lastništvo uporabljenega ključa ugotavlja s pomočjo potrdila o lastništvu, ki ga izda zaupanja vredna organizacija, t. i. overitelj potrdil (angl. Certificate Authority). Overitelji potrdil so med seboj povezani v drevesno strukturo; sistem v celoti se imenuje infrastruktura javnih ključev, s kratico PKI (angl. Public Key Infrastructure). Overitelj sprejema zahteve za izdajo digitalnih potrdil, izvaja ustrezno identifikacijo bodočih imetnikov, izdaja digitalna potrdila in skrbi za register izdanih potrdil, saj so informacije o izdanih potrdilih javnega značaja (razen v zaprtih sistemih). Overitelj prav tako skrbi za preklic digitalnih potrdil in informacije o preklicih osvežuje v seznamu preklicanih potrdil, s kratico CRL (angl. Certificate Revocation List), ki je prav tako javnega značaja.

Prvotna zamisel PKI je bila, da bi s povezovanjem overiteljev v drevesno strukturo postopoma gradili svetovno PKI in tako ustvarili podlago za globalni sistem za overjanje digitalnih podpisov. Zdaj takih pričakovanj ni več - PKI bodo ostale omejene na posamezna območja ali aplikacije, znotraj katerih je možno natančno določiti imetnika digitalnega potrdila in namen uporabe potrdila. Potrdilo na ime "Svetlana Šaljić" ne pove dovolj, če je oseb s tem imenom več. Potrebna je povezana povezava na običajne identifikatorje, ki se razlikujejo v odvisnosti od konteksta uporabe: EMŠO, davčna številka, številka bančnega računa ipd. Tudi ugotavljanje veljavnosti digitalnega potrdila je v omejenih področjih lažje rešljivo.

Ker je hranjenje CRL skupaj s podpisom za namene ohranjanja dokaza o nepreklicnosti lahko potratno (nekateri so tudi po nekaj MB veliki), je delovna skupina PKIX pri IETF pripravila standard RFC 2560, s kratkim imenom OCSP (angl. Online Certificate Status Protocol). Aplikacija pošlje zahtevo za preverjanje statusa digitalnega potrdila direktno »pooblaščenemu« strežniku. Protokol OCSP že uporabljajo banke, ki so vključene v sistem Identrus. V okviru IETF je nastal tudi eksperimentalen predlog RFC 3029, s kratkim naslovom DVCS (angl. Data Validation and Certification Server Protocols), ki obsega širše področje, in sicer sintakso in komunikacijski protokol za preverjanje podpisa in veljavnosti digitalnih potrdil, sam odgovor s strani zaupanja vrednega strežnika pa šteje tudi kot dokazilo o obstoju podatkov.

Vsak uporabnik sam nosi odgovornost za uporabo in varovanje zasebnega ključa, povezanega z javnim ključem, za katerega ima izdano overjeno potrdilo. Digitalno potrdilo je imetnikova osebna izkaznica v elektronskem poslovanju. Za zapis

digitalnih potrdil se uporablja standard X.509, ki opredeljuje ASN.1 strukturo in DER kodiranje za zapis (glej str. 19). V primeru št. 2 je prikazana struktura digitalnega potrdila X.509. V polju 'Subject' so razvidni podatki, s katerimi se predstavlja imetnik javnega ključa, ki ga overja digitalno potrdilo.

Primer 1: Izvleček iz digitalnega potrdila po standardu X.509 verzija 3

```
Data:
     Version: V3
     Serial Number: 3C 0B 50 58
     Signature Algorithm: sha1RSA
     Issuer: C=si, O= state-institutions,OU= sitest-ca
     Validity:
          Not Before: 16. oktober 2003 12:06:01
          Not After : 16. oktober 2008 12:36:01
      Subject: CN = Janez Novak, OU = certificates-web, OU = SITEST-CA,
               O = state-institutions, C = si
      Subject Public Key Info:
         Public Key Algorithm: rsaEncryption
         RSA Public Key: (1024 bit)
             Modulus (1024 bit):
             3081 8902 8181 00C7 08CF BB50 …
             Exponent: 65537 (0x10001)
    X509v3 extensions:
        … raba digitalnega potrdila in drugi atributi
 Certificate Signature Algorithm: …algoritem, ki ga je za podpis
                                   uporabil izdajatelj
 Certificate Signature:…podpis izdajatelja, ki ga preverimo
                        z javnim ključem izdajatelja
```

Overitelji lahko sklenejo dogovor s proizvajalci brskalnikov, da so njihova digitalna potrdila vključena v seznam zaupanja vrednih overiteljev (angl. Trusted Root Certification Authorities). Takšnemu overitelju uporabnikov brskalnik avtomatično zaupa in s tem seveda zaupa vsem digitalnim potrdilom, ki jih je overitelj izdal. Za brskalnik Internet Explorer mora overitelj vsako leto pridobiti revizijsko poročilo organizacije Webtrust; zaradi tega potrdila manjših overiteljev niso avtomatično vključena v brskalnike, kar je varnostno še bolje - uporabnik se sam odloči, ali bo nekega overitelja v svojem brskalniku vključil med zaupanja vredne. Med slovenskimi overitelji so od začetka leta 2008 na seznam dodani Sigen CA, NLB CA in Halcom CA. Bistveno je, da digitalni podpis skupaj s PKI omogoča:

- da se s preverjanjem lahko ugotovimo, ali se je podpisan dokument spremenil,
- da se podpisa ne da ponarediti, v kolikor varujemo zasebni ključ,
- da se podpisa ne da zanikati, v kolikor temelji na varovanju zasebnega ključa in zaupanja vrednem sistemu povezovanja zasebnega ključa z identiteto njegovega lastnika.

Standardi za zapis digitalnih podpisov

Digitalni podpis je lahko ločen od dokumentov, ki jih podpisuje ali pa je sestavni del podpisanega dokumenta; za zapis podpisa pa je v rabi več standardov. Med seboj se razlikujejo glede na tip zapisa (binaren, XML, PDF), glede na možnosti dodajanja več podpisnikov in glede na možnost podpisovanja posameznih delov dokumenta. Pri dodajanju več podpisov v isti dokument nastopi problem, kako ohraniti veljavnost predhodnega podpisa. Pri podpisovanju posameznih delov pa je ključna možnost nedvoumne določljivosti posameznega dela in tudi možnost širitve dokumenta, ne da bi razveljavili predhodne podpise.

V nadaljevanju predstavljam najbolj uveljavljene, ki temeljijo na infrastrukturi javnih ključev in uporabi digitalnih potrdil v formatu X.509. Pri vsakem je podan primer zapisa podpisa, pri čemer je uporabljeno testno digitalno potrdilo 'Janez Novak' iz primera št. 2, podpisan je dokument 'test.xml' iz primera št. 3, podpis pa je shranjen ločeno od dokumenta.

Primer 2: Testni dokument 'test.xml'

```
<?xml version="1.0" encoding="UTF-8"?>
<test>
    <vsebina>datoteka za testiranje digitalnega podpisa</vsebina>
</test>
```

a. Standarda PKCS#7 in CMS

De facto standard **PKCS** (angl. Public-Key Cryptography Standards) je razvil raziskovalni laboratorij RSA, ustanovila ga je skupina izumiteljev RSA algoritma, in sicer Rivest, Shamir in Adleman (RSA Laboratories), danes je last mednarodne korporacije EMC. Proces ustvarjanja podpisa in struktura zapisa je opredeljena z ASN.1 specifikacijo PKCS, številka 7 (angl. **PKCS#7**). V praksi se uporabljajo tako binarna kodiranja kot tekstovna, na primer base64 za podpisovanje elektronske pošte. Organizacija IETF je PKCS#7 specifikacijo razširila v standard RFC 3852 s skrajšanim nazivom **CMS** (angl. Cryptographic Message Syntax) in opredeljuje sintakso za kriptografska sporočila. Primer št. 3 prikazuje ASN.1 specifikacijo strukture CMS podpisa. Primer DER (glej str. 19) kodiranega podpisa vidimo v primeru št. 4, ki je zaradi binarnega kodiranja seveda za človeka nerazumljiv, zato v primeru št. 5 podajam interpretacijo ključnih delov podpisa.

Primer 3: Izvleček iz ASN.1 modula (RFC 3852) za strukturo CMS podpisa

```
ContentInfo ::= SEQUENCE {
  content-type   CMS-CONTENT-TYPE.&id({CMSContentTable}),
  pkcs7-content  [0]  CMS-CONTENT-TYPE.&Type({CMSContentTable}) {

      SignedData ::= SEQUENCE {
        version           CMSVersion,
        digestAlgorithms  DigestAlgorithmIdentifiers,
        encapContentInfo  EncapsulatedContentInfo,
        certificates      CertificateSet OPTIONAL,
        crls              CertificateRevocationLists OPTIONAL,
        signerInfos       SignerInfos
      }
  }
}
```

Primer 4: Primer DER kodiranega ločenega podpisa po standardu CMS

Podpisan je dokument 'test.xml' z digitalnim potrdilom 'Janez Novak'. Binaren niz je znakovno predstavljen z base64 kodiranjem.

```
MIIGwQYJKoZIhvcNAQcCoIIGsjCCBq4CAQExCzAJBgUrDgMCGgUAMAsGCSqGSIb3DQEHAaCCB
Z8wggWbMIIEg6ADAgECAgQ8C1BYMA0GCSqGSIb3DQEBBQUAMD4xCzAJBgNVBAYTAnNpMRswGQ
YDVQQKExJzdGF0ZS1pbnN0aXR1dGlvbnMxEjAQBgNVBAsTCXNpdGVzdC1jYTAeFw0wMzEwMTY
xMTA2MDFaFw0wODEwMTYxMTM2MDFaMIGFMQswCQYDVQQGEwJzaTEbMBkGA1UEChMSc3RhdGUt
aW5zdGl0dXRpb25zMRIwEAYDVQQLEwlTSVRFU1QtQ0ExGTAXBgNVBAsTEGNlcnRpZmljYXRlc
y13ZWIxKjASBgNVBAMTC0phbmV6IE5vdmFrMBQGA1UEBRMNMjYwOTIwMDMwMDAwMTCBnzANBg
kqhkiG9w0BAQEFAAOBjQAwgYkCgYEAxwjPu1BMZRUWOXr3nokdT5FAu/oHKAeIcJbAPvp0w05
JXhl+mJPyAarJs6wU+xqZFDHq8ZmKZ46Ed2hSk4tcVajIP2q+uW6f6Obd1H79nOEopFTvHQWl
Lzmo+5KjM2XMBKQHEym6h2p1M+amKzBurMXZ/Da0TPbzl6j/Be7kqVsCAwEAAaOCAtswggLXM
AsGA1UdDwQEAwIFoDArBgNVHRAEJDAigA8yMDAzMTAxNjExMDYwMVqBDzIwMDgxMDE2MTEzNj
AxWjARBglghkgBhvhCAQEEBAMCBaAwLwYJYIZIAYb4QgECBCIWIGh0dHBzOi8vd3d3LnNpZ2V
uLWNhLnNpL2NkYS1jZ2kvMEQGCWCGSAGG+EIBAwQ3FjVjbGllbnRjZ2k/YWN0aW9uPWNoZWNr
UmV2b2NhdGlvbiYmQ1JMPWNuPUNSTDEmc2VyaWFsPTBQBglghkgBhvhCAQ0EQxZBU3BsZXRub
yBrdmFsaWZpY2lyYW5vIGRpZ2l0YWxubyBwb3RyZGlsbyB6YSBwcmF2bmUgb3NlYmUgU0lHRU
4tQ0EwQAYDVR0gBDkwNzA1BgorBgEEAa9ZAgECMCcwJQYIKwYBBQUHAgEWGWh0dHA6Ly93d3c
uZ292LnNpL2NhL2Nwcy8wGwYDVR0RBBQwEoEQbWF0ZWpAc2V0Y2NlLm9yZzCB+QYDVR0fBIHx
MIHuMFWgU6BRpE8wTTELMAkGA1UEBhMCc2kxGzAZBgNVBAoTEnN0YXRlLWluc3RpdHV0aW9uc
zESMBAGA1UECxMJc2l0ZXN0LWNhMQ0wCwYDVQQDEwRDUkwxMF6gXKBahlhsZGFwOi8veDUwMC
5nb3Yuc2kvb3U9c2l0ZXN0LWNhLG89c3RhdGUtaW5zdGl0dXRpb25zLGM9c2k/Y2VydGlmaWN
hdGVSZXZvY2F0aW9uTGlzdD9iYXNlMDWgM6Axhi9odHRwOi8vd3d3LnNpZ2VuLWNhLnNpL2Ny
bC9zaXRlc3Qvc2l0ZXN0LWNhLmNybDAfBgNVHSMEGDAWgBRUSQdGh88diZ3KrnKnqKCgabch5
zAdBgNVHQ4EFgQUyvaaeAMX1OYp8aPAQMZxqmE/YMswCQYDVR0TBAIwADAZBgkqhkiG9n0HQQ
AEDDAKGwRWNS4wAwIDqDANBgkqhkiG9w0BAQUFAAOCAQEAKu96gGS26Th7JEpMpOWzNFmlAU2
RzQMkMy3BLV5e8icQz3uIhcTtD/7fawd2TlgfijfhlECj0SQVeLe9zxBO1q98EGQpe60ttp6A
jD56/9I6fm5HFmw1exDQq6TyBA2bnFxQYTNguXRa/1feFbnG6I8bI8iEnIch2WI9SleMZ3fhd
cn9KYVQSFNa5BC6s/zLift/MOtyornLyyonjEAnZks0LN5W3cATIchJOfojNzYvURSB3Dd8Qk
+x/80PA8iJxt5yOr5VlwrHGrPAdqvpD05Ti2DmI+ONKz7wWNm8Xw/Upi8E6DqeGM3dG0aiv3+
1DB+qBx85hRwpdW8T83aKTDGB6zCB6AIBATBGMD4xCzAJBgNVBAYTAnNpMRswGQYDVQQKExJz
dGF0ZS1pbnN0aXR1dGlvbnMxEjAQBgNVBAsTCXNpdGVzdC1jYQIEPAtQWDAJBgUrDgMCGgUAM
A0GCSqGSIb3DQEBAQUABIGAVQhmW96R1L8nWHeP05gP4eoF6Fj66aPY5zcS+zQItRNp559U4r
PkmGhgo228JNGikfSonxZwQUzVqb/uoTTARaIBB6pjReoHbUYTeJC5Z85wLZb+vNgL+oc9/Vs
hV7shWDmyKJsFnlTcPBVrjWJPmuPwZzYcUVW+YDnp+LlLP0Q=
```

Ker je PKCS#7 podmnožica CMS standarda, v mnogih virih nastopata oznaki CMS in PKCS#7 kot sinonima za iste podatkovne strukture. CMS podpis lahko vsebuje digitalna potrdila, sezname preklicanih potrdil, časovne žige, s časovnim žigom povezana digitalna potrdila in sezname preklicanih potrdil ter druge atribute.

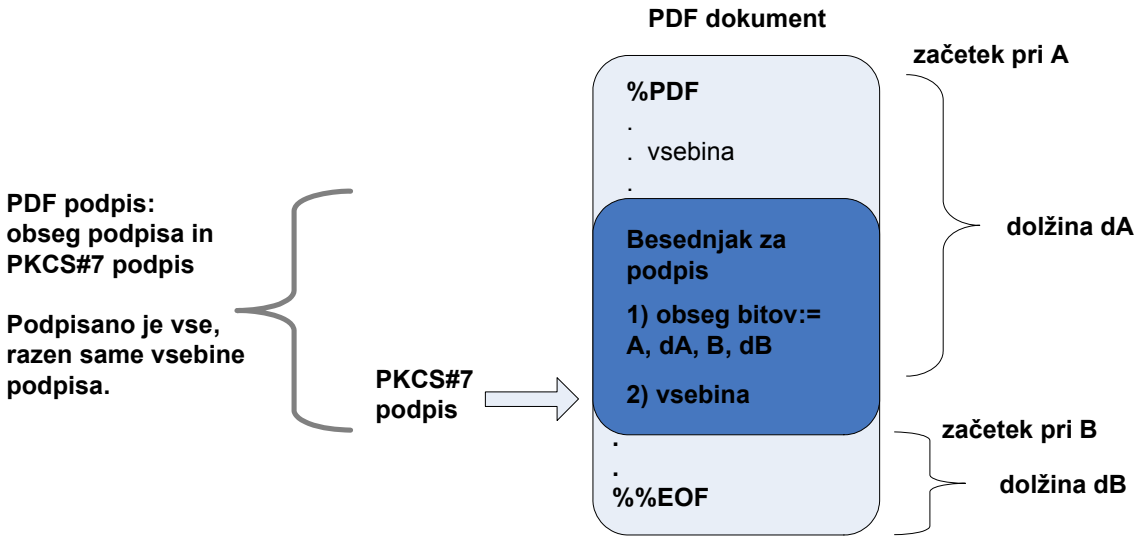
Primer 5: Interpretacija ključnih delov zgornjega podpisa glede na CMS ASN.1 modul

- Podpisani podatki
  a. Verzija CMS =1
  b. Digitalna potrdila
    i. Digitalno potrdilo št. 1
      1. Serijska številka=3C0B5058
      2. Subjekt: C=si,O=state-institutions,OU=SITEST-CA,OU=certificates-web,CN=Janez Novak …
- Podpisniki (število=1)
  c. Podpisnik št. 1
    i. Identifikatorji podpisnika
      1. Serijska številka dig. potrdila = 3C0B5058
      2. …
    ii. Kriptografski algoritem = RSA//PKCS1PADDING (OID= 1.2.840.113549.1.1.1)
    iii. Zgostitveni algoritem = SHA-1 (OID= 1.3.14.3.2.26)
    iv. Vrednost podpisa zakodirana z base64 = VQhmW96R1L8nWHeP05gP4eoF6Fj66aPY5zcS+zQItRNp559U4rPkmGhgo228JNGikfSonxZwQUzVqb/uoTTARaIBB6pjReoHbUYTeJC5Z85wLZb+vNgL+oc9/VshV7shWDmyKJsFnlTcPBVrjWJPmuPwZzYcUVW+YDnp+LlLP0Q=

b. Standard za podpis v formatu PDF

Standard za **podpis v formatu PDF** se navezuje na standard za dokumente v formatu PDF (angl. Portable Document Format); oboje so razvili v podjetju Adobe Systems in sta odprta standarda. Različica PDF/A, opredeljena z ISO 19005-1, se je uveljavila kot zanesljiv format za dolgoročno shranjevanje dokumentov. Podpis PDF temelji na specifikaciji PKCS#7 in opredeljuje načine določanja vsebine podpisa in umeščanja PKCS#7 zapisanega podpisa v PDF dokument. Način umeščanja v podpisa v PDF dokument omogoča zaporedno podpisovanje več podpisnikov in je ilustriran na sliki št. 2.

Slika 2: Umestitev PKCS#7 podpisa v PDF dokument



Vir: Adobe Systems, 2006, str. 5.

c. Standarda XMLDsig in XAdES

Standard **XMLDsig** (Bartel, 2002), je nastal v sodelovanju W3C konzorcija in IETF ter opredeljuje proces ustvarjanja podpisa in strukturo zapisa v jeziku XML. Primer št. 6 prikazuje strukturo ločenega digitalnega podpisa po standardu XMLDsig.

Primer 6: Primer ločenega podpisa po standardu XMLDsig

Podpisan je dokument 'test.xml' z digitalnim potrdilom 'Janez Novak'. Podpis je na mestih, kjer vključuje digitalna potrdila, skrajšan zaradi preglednosti.

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<ds:SignedInfo>
   <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-
    xml-c14n-20010315" />
   <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-
   sha1" />
  <ds:Reference URI="file:/// test.xml">
    <ds:DigestMethodAlgorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
    <ds:DigestValue>vPuI6WI3RYxzNwtYZ0a3BNBHNBc=</ds:DigestValue>
  </ds:Reference>
  </ds:SignedInfo>
<ds:SignatureValue>tyLaLNARGqSJmswhn6CeKo27UD+bnSWwpS9aIBQZkYzim/FrxyfVSg
ngQPZhnmdLIXcrSKIM5LnSt4nUcItFwO96d+KfrLjxNd9q21Bn3N31d2D12unnPVxpomCgohZ
+xgDYDZ/AJLp2yoNDJzjliahx jdtuBsJFWMOpssjz3Qk=</ds:SignatureValue>
<ds:KeyInfo>
   <ds:X509Data>
   <ds:X509SubjectName>C=si, O=state-institutions, OU=SITEST-CA,
      OU=certificates-web, CN=Janez Novak OID.2.5.4.5=2609200300001
   </ds:X509SubjectName>
    …
   <ds:X509Certificate>MIIFmzCCBIOgAwIBAgIEPAt… </ds:X509Certificate>
   <ds:X509CRL>MIII… </ds:X509CRL>
  </ds:X509Data>
</ds:KeyInfo>
</ds:Signature>
```

Vsebina podpisa je podobna kot pri standardih CMS/PKCS#7, pomembni razliki sta berljivost, zaradi narave kodiranja, in možnost vključevanja več referenc na podpisano vsebino. Uporaba XML jezika in referenc omogočata, da hkrati podpisujemo več dokumentov iz različnih virov; lahko podpišemo tudi del XML dokumenta, v katerem se nahaja podpis preko sklicevanja na identifikator določenega elementa. XMLDsig standard je zato uporaben za podpisovanje tako binarnih kot XML dokumentov; za slednje omogoča tudi podpisovanje posameznih delov dokumenta na način, ki omogoča širitev dokumenta, tako z novo vsebino kot z drugimi podpisi. Podpisi so lahko vsebovani v isti datoteki kot podatki, ki se podpisujejo; podpisuje pa se lahko večkrat iste dele dokumenta (vzporedni podpisi) ali pa različne dele dokumenta.

Standard **XAdES** (ETSI, 2002) je razširitev XMLDsig standarda na način, podoben kot pri standardu CMS, in sicer, da dovoljuje vključevanje podpisanih in nepodpisanih atributov v sam podpis z namenom shranjevanja dodatnih varnostnih vsebin, kot so časovni žig podpisa, veriga digitalnih potrdil ter seznami preklicanih potrdil za časovni žig in drugo. Opisane razširitve izpolnjujejo potrebe po dokazovanju dolgoročne verodostojnosti v določenih primerih, zato je ta standard podrobneje obravnavan v poglavju št. 2.7.

Primerjava med XMLDsig in CMS (PKCS#7) standardoma za podpisovanje

Razlika med XMLDsig in CMS standardoma je rezultat razlik med formatoma zapisa, torej med binarno kodiranimi ASN.1 moduli in XML dokumenti.

- XML podpis ima prednost, ker lahko enostavno in zanesljivo naslavljamo posamezne dele dokumenta, najenostavneje preko id atributov XML elementov, ki so enolični identifikatorji elementa znotraj XML dokumenta. Na ta način lahko razširimo dokument tako, da ne spreminjamo podpisanih elementov, temveč vključujemo nove tako podpise kot tudi vsebine. Te lastnosti so v podporo podpisovanju v delovnih procesih, kjer posamezne korake procesa podpisujejo različni udeleženci, skozi celoten proces pa nastaja en sam dokument, na primer pri registraciji zdravil.
- Prednost XML je tudi človeku razumljiv zapis, saj lahko z odpiranjem XML zapisa v poljubnem tekstovnem urejevalniku ugotovimo vsebovane podpise in na katere dele naj bi se nanašali. Vendar lahko z ASN.1 bralniki za CMS podpise naredimo podobno (glej poglavje 1.2).
- Slabost XML podpisa je zahtevnejše procesiranje. V primerjavi s CMS je treba XML dokument kanonizirati pred izračunom prstnega odtisa. Počasnost se bistveno poveča z velikostjo dokumenta. Palmer (2004, str. 5) v testih ugotavlja več kot 10-kratne časovne razlike pri procesu preverjanja veljavnosti podpisa.
- Slabost XML je tudi velikost zapisa, CMS binarno kodirani zapisi so veliko kompaktnejši.

1.1.2 Časovni žig

Zamisel časovnega žiga nam je znana že iz poštnih žigov na pošiljkah. Dokazilo, da je dokument nastal po določenem datumu, lahko ustvarimo tudi tako, da v vsebino dodamo rezultate nogometne tekme ali številke, izvlečene na loteriji, saj je nezmožnost zagotove napovedi dovolj velika (Open Evidence, str. 27). Za elektronske dokumente je danes najpogosteje za namene dokaza obstoja ob določenem času v uporabi digitalen časovni žig (angl. Digital Time-Stamp), ki je

digitalni podpis prstnega odtisa vsebine dokumenta, skupaj s časovno komponento. Podpisnik je zaščiten strežnik zaupanja vredne organizacije ali s kratico TSA (angl. Time-Stamp Authority) in ima vlogo nepristranske priče. TSA potrjuje, da je ob določenem datumu/času prejel prstni odtis v podpis. Tehnologija prstnega odtisa zagotavlja, da je prstni odtis lahko nastal samo iz točno določenega dokumenta. Kadar torej potrebujemo dokazilo, da je dokument obstajal ob določenem času, pošljemo TSA strežniku prstni odtis tega dokumenta. Strežnik temu dopiše čas in vse skupaj podpiše s svojim zasebnim ključem ter nam vrne na ta način ustvarjen digitalni podpis. S tem lahko dokažemo, da je elektronski dokument obstajal pred časom, navedenim v časovnem žigu, poleg tega pa je možno preveriti, da se od časa žigosanja ni spremenil (naredimo ponovni prstni odtis dokumenta,ki se mora ujemati s tistim, ki je del časovnega žiga).

Standardi za zapis časovnega žiga

Organizacija IETF je pripravila standard RFC 3161, ki uporablja binarno kodirano ASN.1 podatkovno strukturo. Strežnik za časovno žigosanje prejme zahtevek s prstnim odtisom in vrne datoteko, ki vsebuje DER zakodirano (glej str. 19) ASN.1 strukturo TimeStampToken, ki je prikazana v primeru št. 7.

Primer 7: ASN.1 specifikacija TimeStampToken za časovni žig iz standarda RFC 3161

```
TimeStampToken ::= SEQUENCE {
      tstInfo TSTInfo,
      signature BIT STRING,  -- podpis nad ASN.1 DER zakodiranim TSTInfo
}
TSTInfo ::= SEQUENCE {
      policy PolicyInformation,
      status PKIStatusInfo,
      requester [0] GeneralName OPTIONAL,
      tsa GeneralName,
      signatureAlgorithm AlgorithmIdentifier,
      certId CertId, -- digitalno potrdilo strežnika za časovno žigosanje
      certs SEQUENCE OF Certificate OPTIONAL,
      genTime GeneralizedTime,
      messageImprint MessageImprint -- prstni odtis dokumenta, ki smo ga
                                    --  poslali v zahtevku
}
```

V uporabi je tudi zapis časovnega žiga z jezikom XML, ki nima formalne podlage. V praksi se je »prijel«, ker ga uporablja podjetje Entrust, ki je eden največjih ponudnikov opreme za izvedbo storitve časovnega žigosanja. Ker je časovni žig v svoji osnovi digitalni podpis, je Entrust uporabil standard XMLDsig; bolj natančno podvrsto podpisa, pri kateri je podpisana vsebina vključena v podpis v elementu <Object> (angl. enveloped). Ta vsebuje prstni odtis dokumenta, ki smo ga poslali strežniku, skupaj s časom izdelave podpisa in še nekaj drugimi atributi. Ta postopek ilustrira primer št. 8.

Primer 8: Primer XML zapisa v skladu z Entrustovo shemo za časovni žig.

```
<dsig:Signature xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
Id="TimeStampToken">
  <dsig:SignedInfo>
     ...<dsig:Reference URI="#TimeStampInfo-3699"> ...
  </dsig:SignedInfo>
  <dsig:SignatureValue>...</dsig:SignatureValue>
  <dsig:KeyInfo Id="TimeStampAuthority">...<dsig:KeyInfo>

  <dsig:Object Id="TimeStampInfo-3699">

      <ts:TimeStampInfo
       xmlns:ts="http://www.entrust.com/timestamp-protocol-20020207"  >
      <ts:Policy id="politika-za-casovni-zig-1.pdf"/>
      <ts:Digest>
          <ds:DigestMethod  Algorithm="xmldsig#sha1"/>
          <ds:DigestValue>AJ0ZnHcsJanB4m5xwSecZLB4kb8=</ds:DigestValue>
      </ts:Digest>
      <ts:SerialNumber>879242967855930593958</ts:SerialNumber>
      <ts:CreationTime>2006-06 19T09:49:46.911Z</ts:CreationTime>
      <ts:Nonce>14194661</ts:Nonce></ts:TimeStampInfo>

  </dsig:Object>
</dsig:Signature>
```

Opomba: Primer je okrnjen z namenom povečanja preglednosti strukture XML zapisa. Zahtevek za časovni žig je izdelan z orodjem organizacije SETCCE proXSign, časovni žig pa pridobljen na testnem sistemu Ministrstva za javno upravo.

Vsebinsko sta standarda RFC3161 in EnTrust enakovredna. Za potrebe trajne hrambe veljajo zanju enake zahteve kot za digitalni podpis.

1.1.3 Standardi in zakonodaja na področju hrambe elektronskih dokumentov

Okvirne funkcionalne zahteve za arhiviranje elektronske dokumentacije podajajo mednarodno priznani dokumenti in standardi; najvplivnejši med njimi so na kratko predstavljeni v nadaljevanju.

Na pobudo Mednarodne organizacije za standardizacijo ISO je mednarodno združenje vesoljskih agencij CCSDS (angl. Consultative Committee for Space Data Systems) pripravilo predlog referenčnega modela za arhivski informacijski sistem z namenom standardizacije gradnje arhivov. Predlog se s kratico imenuje **OAIS** (angl. Open Archival Information System) in ta dokument je bil leta 2003 sprejet kot mednarodni standard **ISO 14721** za izdelavo arhivskih sistemov elektronskih virov.

Referenčni model OAIS sestavljata informacijski model in funkcionalni model. Informacijski model opredeljuje štiri vrste informacijskih objektov (CCSDS, 2002, str. 28), ki se nanašajo na:

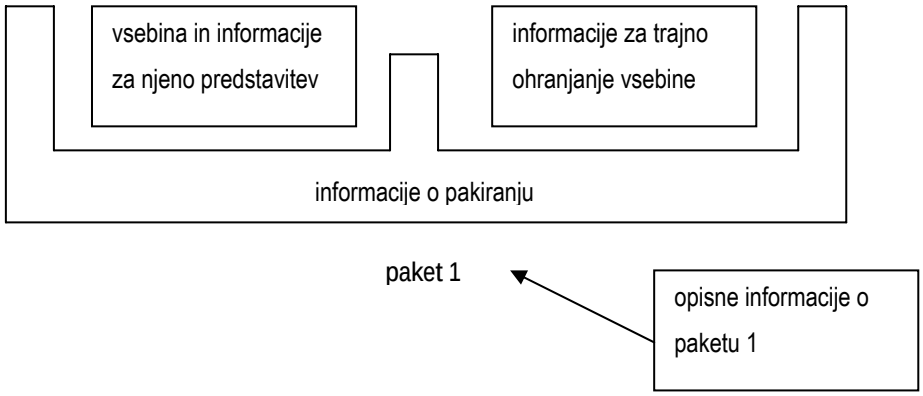
- informacije, potrebne za trajno ohranjanje vsebine:

a. referenčne informacije, ki so potrebne za nedvoumno identifikacijo vsebine elektronskega vira (npr. ISBN številka za knjigo);
b. informacije o kontekstu, ki povedo, kako je določen elektronski vir povezan z drugimi vsebinami;
c. informacije o provenienci, ki dokumentirajo zgodovino izvora, prenosa odgovornosti, dejanj pri ohranjanju in podobno,
d. informacije o pristnosti, ki dokumentirajo mehanizme za dokazovanje pristnosti informacij o vsebini in njeni nespremenljivosti;

- vsebino in informacije za njeno predstavitev,
- informacije o pakiranju,
- opisne informacije.

Informacije o pakiranju dejansko ali logično povezujejo vsebino z informacijami za trajno ohranjanje vsebine. Opisne informacije pa pomagajo pri identificiranju paketov, ki nas zanimajo. Povezave med paketi ilustrira slika št. 3.

Slika 3: Odnosi med informacijskimi objekti v OAIS modelu



Vir: CCSDS, str. 27.

Osnovni koncept modela OAIS obravnava tri vloge udeležencev v procesu (CCSDS, str. 24): ustvarjalca, upravljavca in uporabnika elektronskega vira. Osnovna entiteta je informacijski paket, ki vsebuje vse štiri vrste zgoraj opisanih informacijskih objektov. Ta paket se od vstopa v arhivski sistem do izhoda spreminja, odvisno od arhivskega procesa, v katerem se nahaja. Funkcionalni model (CCSDS, str. 38) opisuje arhivske procese, ki se odvijajo v šestih funkcionalnih entitetah: v vložišču, arhivskem skladišču, v enoti za upravljanje s podatki, v administraciji, v enoti za načrtovanje trajnega ohranjanja in enoti za dostop.

Na tem področju je tudi organizacija Cornwell Management Consultants po naročilu Evropske unije pripravila specifikacijo Model zahtev za upravljanje elektronskih dokumentov (DLM Forum, 2005), znana s krajšim imenom **MoReq**. Specifikacija določa funkcionalne zahteve za upravljanje elektronskih dokumentov, podaja model medsebojnih povezav (za na primer klasifikacijski načrt, dokumente, zapise) in model metapodatkov za upravljanje dokumentov. MoReq predvideva, da se bodo model in opisane zahteve izvajali preko sistema, imenovanega ESUD - elektronski sistem za upravljanje dokumentarnega gradiva, za katerega MoReq tudi opisuje, kaj naj bi delal. Trenutno je v pripravi druga verzija (MoReq2), ki bolj natančno opredeljuje zahteve, opremljena je tudi z opredelitvijo testnega okolja in postopkov testiranja, saj cilja na vzpostavitev sistema certificiranja programskih rešitev.

V ZDA pa obrambno ministrstvo od leta 1997 dopolnjuje standard Kriteriji za oblikovanje programske opreme za upravljanje elektronskih dokumentov **DoD 5015.2-STD**. Ta dokument vsebuje smernice za implementacijo in postopke sistemov za upravljanje z dokumenti.

Prvi zakon s področja elektronskega podpisovanja na svetu so sprejeli v ameriški zvezni državi Utah leta 1995[2]. Med naslednjimi najbolj znanimi je zakon o elektronskem poslovanju iz leta 1996, ki ga je sprejela Komisija za mednarodno in trgovinsko pravo v sklopu Organizacije združenih narodov, s kratico **UNCITRAL** (angl. United Nations Comission on International Trade Law). Ta zakon določa pravni okvir za elektronsko poslovanje in načela enakovrednosti elektronskega poslovanja papirnemu. Kasneje je bil še razširjen z zakonom o elektronskem podpisu, ki določa pogoje za pravno enačenje digitalnega in lastnoročnega podpisa.

Na ravni Evropske Unije pravila elektronskega poslovanja, pravnomočnost elektronske oblike in njeno hrambo urejajo predvsem **Direktiva o elektronskem poslovanju** 2000/31EC (angl. Directive on electronic commerce) in **Direktiva o skupnem okviru skupnosti za elektronske podpise** 1999/93/EC (angl. Community framework for electronic signatures), za države članice pa je merodajen pravni red posamezne države.

---

[2] Avtor slovenskega prevoda je Janez Toplišek, maj 1996.

[URL:http://www2.arnes.si/~rzjtopl/index.htm]

V Sloveniji področje pravnomočnosti elektronskega gradiva in njegove hrambe urejajo: **Zakon o varstvu dokumentarnega in arhivskega gradiva ter arhivih** (ZVDAGA), **Zakon o elektronskem poslovanju in elektronskem podpisu** (ZEPEP), **Uredba o pogojih za elektronsko poslovanje in elektronsko podpisovanje,** davčna zakonodaja in računovodski standardi, ki predpisujejo roke hrambe določenih poslovnih listin in dokazil. Za javni sektor so pomembni še podzakonski akti o ravnanju z dokumentarnim gradivom.

Po 12. členu ZEPEP mora hramba elektronskega gradiva izpolnjevati naslednje pogoje:

- gradivo mora biti dostopno na uporaben način in primerno za kasnejšo rabo,
- gradivo mora biti shranjeno v izvorni obliki ali obliki, ki verodostojno predstavlja izvorno obliko,
- možno je ugotoviti izvor in cilj sporočila ter čas prenosa ali prejema,
- uporabljena tehnologija in postopki morajo zanesljivo zagotavljati nespremenljivost sporočila.

Bistveni pomen ZEPEP je, da pod posebnimi pogoji digitalnemu podpisu priznava enako veljavo, kot jo ima v papirnatem svetu lastnoročni podpis. ZEPEP je usklajen tudi z določili UNCITRAL o elektronskem poslovanju. Po ZVDAGA se avtentičnost in celovitost zajetega gradiva v digitalni obliki za dolgoročno hrambo zagotavljata na tehnološki in organizacijski način na ravni posameznih enot, skupine enot ali celotnega zajetega gradiva:

- z dodajanjem varnostnih vsebin gradivu (npr. dodani metapodatki o preverjanju avtentičnosti in celovitosti, elektronski podpis, časovni žig in podobno),
- z drugimi sorodnimi tehnološkimi sredstvi,
- z zagotavljanjem dodatnih organizacijskih ukrepov.

Uredba o varstvu arhivskega in dokumentarnega gradiva (v nadaljevanju Uredba) na splošno opredeljuje postopke zajema in hrambe gradiva v digitalni obliki. Uredba elektronsko hrambo prepoznava kot temeljno storitev, ki se jo dopolnjuje s spremljevalni storitvami, z namenom zagotavljanja dolgoročnega obstoja gradiva, zaščite pred nepooblaščenim odstopom, avtentičnosti in celovitosti.

Enotne tehnološke zahteve (ETZ), ki jih je pripravil Arhiv Slovenije, pa opredeljujejo postopke zajema gradiv, obvezne metapodatke, veljavne postopke in oblike pretvorbe gradiv, obliko zapisa za dolgoročno hrambo, vrste tehnoloških sredstev za zagotavljanje avtentičnosti in celovitosti gradiva v digitalni obliki.

Infrastruktura za elektronsko arhiviranje

Slovenska zakonodaja opredeljuje infrastrukturo in storitve elektronskega arhiviranja na treh ravneh:

- strojna in programska oprema (splošni pogoji, ki jih mora izpolnjevati strojna in programska oprema),
- storitve e-hrambe in spremljevalne storitve (splošni pogoji hrambe in spremljevalnih storitev),
- ponudniki opreme in storitev (registracija in akreditacija).

Vsako infrastrukturo področje in z njim povezane tehnološke rešitve opisujejo posamezni standardi, na primer XMLDsig kot priporočilo organizacije W3C za digitalne podpise v XML obliki. Ponudniki strojne in programske opreme pa se ukvarjajo s konkretnimi implementacijami.

1.2. ASN.1

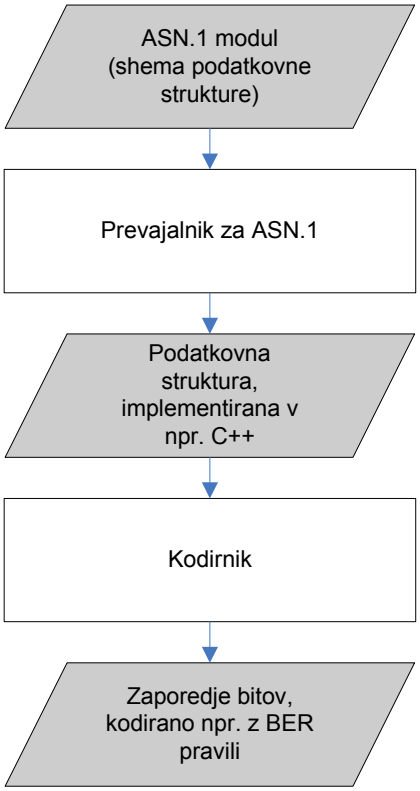
ASN.1 (angl. Abstract Syntax Notation One) je del sistema OSI (Open Systems Interconnection) za komunikacijske standarde, v uporabi je od leta 1984 in je formalni jezik za tehnološko neodvisen opis strukture podatkov. Opredeljuje ga priporočilo Mednarodne telekomunikacijske unije (s kratico ITU-T) z oznako X.680. Uporablja se na področju UMTS (3 generacija mobilnih sistemov), na področju internetne telefonije (H.323), radio-frekvenčne identifikacije, zajema in varovanja biometričnih podatkov (ANSI X9.84-2000), digitalnih potrdil (X.509), infrastrukture javnih ključev (PKCS), protokola za nadzor naprav v omrežju (SNMP) in drugih.

Priporočilo opredeljuje pravila za opis strukture podatkov (opis strukture se imenuje tudi ASN.1 shema) in navodila za zapisovanje podatkov, ki so lahko predstavljeni kot niz bitov ali kot besedilo. Za opis strukture podatkov je možno določiti enostavne tipe podatkov (BOOLEAN, INTEGER, BIT STRING, OCTET STRING, REAL, ENUMERATED…) in sestavljene (SEQUENCE, SET, CHOICE…), ki jih lahko poljubno sestavljamo ali gnezdimo v drevesno strukturo podatkov, ter na ta način tvorimo kompleksne podatkovne strukture. ASN.1 jezik omogoča, da iz obstoječih struktur izpeljujemo nove ali pa da obstoječe širimo.

Opis strukture podatkov z ASN.1 jezikom ni odvisen od dejanskih implementacij za pisanje, branje in kodiranje podatkov. Zato je za prenos podatkov poleg ASN.1 sheme potreben še način zapisa dejanskih podatkov oziroma kodiranje.

Slika št. 4 ilustrira proces kodiranja zapisa po ASN.1 shemi; shemo implementiramo s podatkovnimi strukturami v programskem okolju z generičnimi prevajalniki iz ASN.1 specifikacije v na primer C++ programsko kodo, nato napolnimo podatkovno strukturo s podatki in v naslednjem koraku prekodiramo zajete podatke v zaporedje bitov po pravilih za izbrano binarno kodiranje.

Slika 4: Proces kodiranja zapisa po ASN.1 shemi



Za kodiranje je seveda pomembno, da ga razumeta obe strani, tako pošiljatelj kot prejemnik podatkov. Po priporočilih Mednarodne telekomunikacijske unije se za ASN.1 sheme uporabljajo naslednja navodila za kodiranje podatkov:

- osnovna pravila BER (angl. Basic Encoding Rules) iz priporočila X.690 in dve podmnožici teh pravil CER (angl. Canonical Encoding Rules) ali DER (angl. Distinguished Encoding Rules),
- zgoščena pravila PER (angl. Packed Encoding Rules) iz priporočila X.691,

- kodiranje XER (angl. XML Encoding Rules) iz priporočila X.693. Rezultat XER kodiranja je XML dokument. V primeru št. 23 je prikazan primer XER kodiranja za ASN.1 shemo.

BER kodiranje na primer zapisuje podatke v trojicah: tip-dolžina-vrednost, kjer je tip številčna koda za določen tip podatka, dolžina pomeni dolžino naslednjega polja; ponavadi je izražena v številu zlogov (zaporedje osmih bitov). V primeru št. 9 sta prikazana ASN.1 modula za podatkovni strukturi za vprašanje in odgovor po namišljenem protokolu in kodiranje DER za konkreten primer vprašanja.

Primer 9: Podatkovne strukture za namišljen protokol, opredeljene z ASN.1 modulom

Primer je povzet po viru: http://en.wikipedia.org/wiki/Abstract_Syntax_Notation_One

```
FooProtocol DEFINITIONS ::= BEGIN
    FooQuestion ::= SEQUENCE {
        trackingNumber INTEGER,
        question       VisibleString
    }

    FooAnswer ::= SEQUENCE {
        questionNumber INTEGER,
        answer         BOOLEAN
    }
END
```

Zamislimo si, da bi zdaj radi poslali vprašanje:

```
myQuestion FooQuestion ::= {
    trackingNumber     5,
    question           "Anybody there?"
}
```

Da bi lahko zares poslali podatke iz tega primera, potrebujemo določeno kodiranje v niz bitov. ASN.1 opredeljuje različne algoritme kodiranja, med najbolj preprostimi je DER. DER pravila določajo, kako vsak objekt, predstavljen v notaciji ASN.1, na en sam način predstavimo kot zaporedje oktetov (8-bitov).

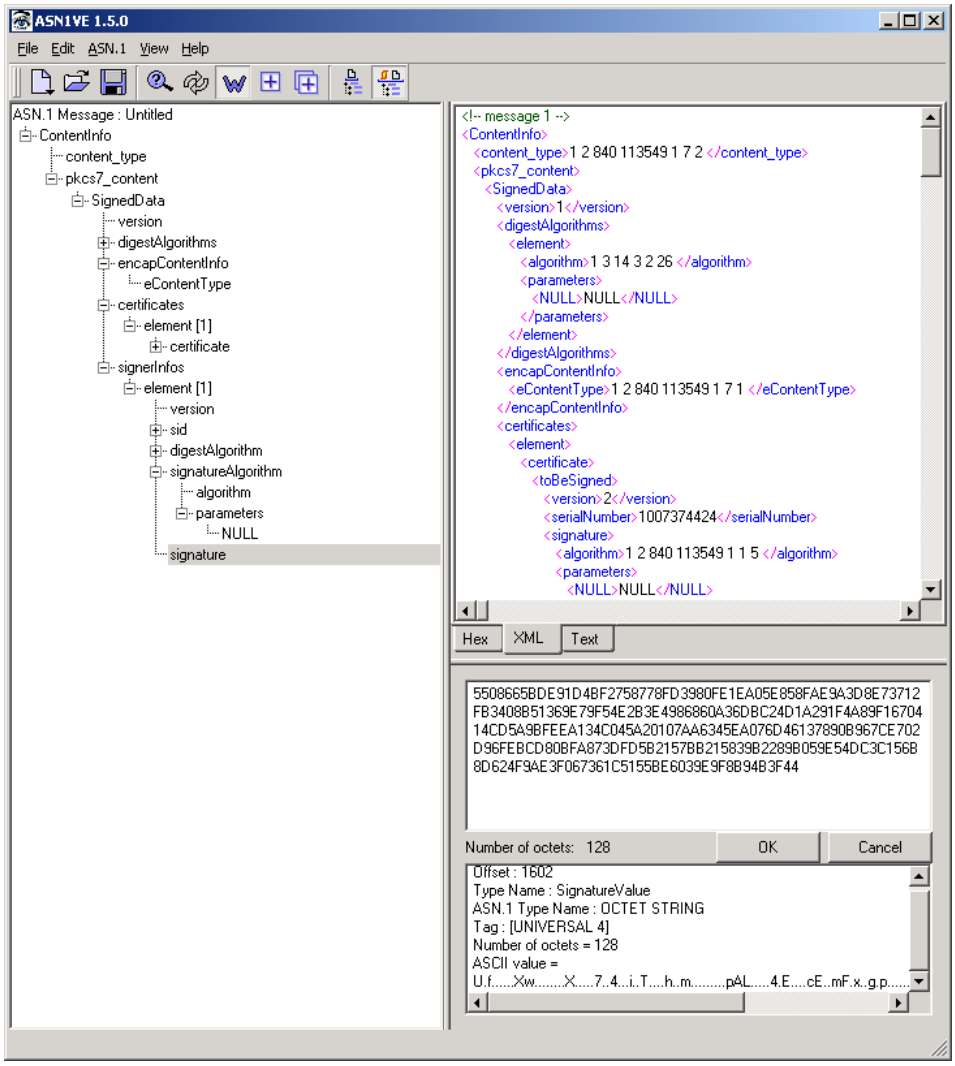
```
30 -- tag indicating SEQUENCE
13 -- length in octets

02 -- tag indicating INTEGER
01 -- length in octets
05 -- value

1a -- tag indicating VisibleString
0e -- length in octets
41 6e 79 62 6f 64 79 20 74 68 65 72 65 3f -- value ("Anybody there?" in
ASCII)
```

Poslali bomo torej 21 oktetov:
30 13 02 01 05 1a 0e 41 6e 79 62 6f 64 79 20 74 68 65 72 65 3f

Slika 5: Pregled podpisa po standardu CMS z orodjem ASN1VE



Vir: Na sliki je naložen CMS podpis iz str. 10 in CMS moduli iz spletne strani http://www.itu.int/ITU-T/asn1/database/ietf/rfc/rfc2630/index.html, 23.3.2008.

Za rabo ASN.1 shem in zapisov podatkov so na voljo orodja za razvijalce v različnih okoljih. Tipično so na voljo prevajalniki iz ASN.1 specifikacij v razrede objektov v na primer programskem jeziku C++, s katerimi lahko razvijalec dela s podatkovno strukturo ter kodirnik/dekodirniki za zapis podatkov v obliki za prenos oziroma za branje podatkov iz te oblike.

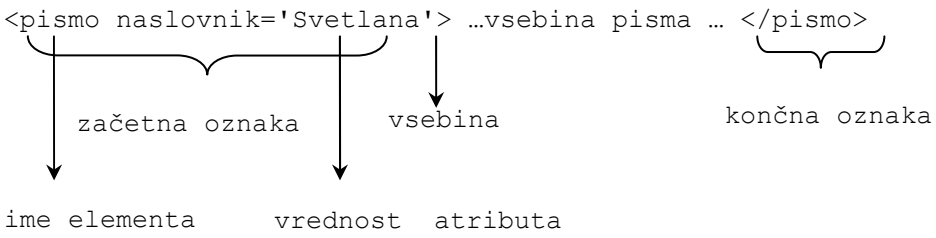
Na sliki št. 5 je prikazan posnetek orodja ASN1VE, ki ga na trgu ponuja podjetje Objective Systems. Orodju podamo binarno kodirano sporočilo in ASN.1 specifikacijo za strukturo; na posnetku na sliki je primer podpisa v skladu s standardom CMS. Tovrstna orodja omogočajo uporabniku razumljiv prikaz vsebine binarno kodiranega sporočila.

1.3. XML in povezane tehnologije

XML je metajezik, s katerim ustvarjamo druge označevalne (angl. markup) jezike. Lahko se uporablja podobno kot ASN.1 tudi za specifikacijo podatkovnih struktur, vendar je pristop drugačen. Specifikacija ASN.1 sloni na formalni sintaksi, jezik XML pa uporablja sistem označevanja vsebine in pripisovanja pomena tem oznakam (angl. tags), pri čemer oznake in njihov pomen sami določimo. Zaradi opisanega ga imenujemo označevalni jezik. Ker jezik XML podpira naknadno razširjanje strukture, ga imenujemo tudi razširljivi jezik (angl. extensible). XML je standardizirala organizacija W3C leta 1998 (http://www.w3.org/TR/xml11/).

Z oznakami v ostrih oklepajih <> opisujemo strukturo in pomen podatkov. Znotraj začetne (na primer: <ime_oznake>) in končne oznake (na primer: </ime_oznake>) zapisujemo vsebino elementa oziroma podatke. Oznaka torej nastopa v paru: začetna in končna oznaka. Vsebino med začetno in končno oznako, vključno z njima, imenujemo element. Jezik XML ne vsebuje vnaprej opredeljene množice oznak, tako kot na primer jezik za oblikovanje hiperteksta HTML, ampak omogoča določanje lastnih oznak. Primer št. 10 prikazuje primer elementa.

Primer 10: Element pismo



Elementi vsebujejo drugo besedilo, druge elemente ali oboje, lahko so prazni. Elementi so hierarhično urejeni, zato lahko dokument XML ponazorimo z drevesno strukturo. Oznake lahko vsebujejo enega ali več atributov.

Atributi so pari »ime-vrednost«, ki so ločeni z enačajem. Atributi so primerni predvsem za predstavitev metapodatkov (informacij o samih podatkih), kot je na primer podatkovni tip podatka. V tabeli št. 1 so primeri elementov z ali brez atributov ter različnih podatkovnih struktur, ki jih lahko z uporabo obojih ustvarimo, v primeru št. 11 pa je prikazan konkreten dokument XML.

Na dokument XML lahko gledamo tudi kot na urejeno drevo, z označenimi vozlišči ter z besedili v listih in elementi v vozlih, ki imajo atribute in vrednosti atributov, kot je prikazano na sliki št. 6.

Tabela 1: Primeri elementov XML

Element , ki vsebuje besedilo	`<ime_elementa atribut1='vrednost1' atribut2='vrednost2'>`
	`  Besedilo …`
	`</ime_elementa>`

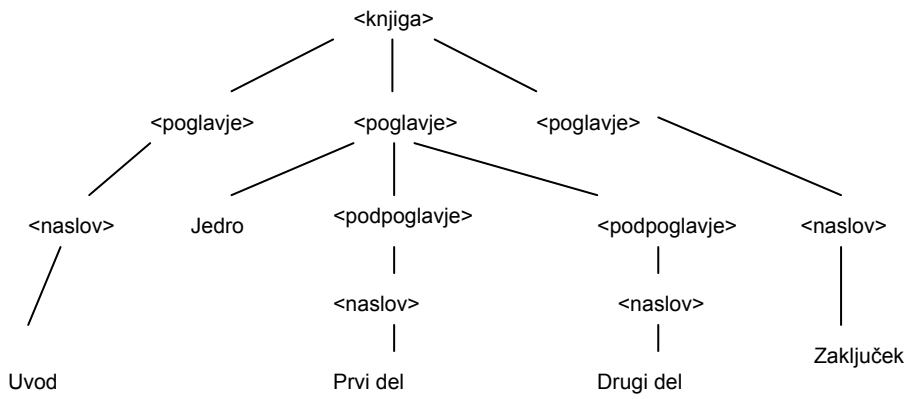
Prazen element	`<ime_elementa atribut1='vrednost1' atribut2='vrednost2' />`

Element, ki vsebuje drug element, ki vsebuje več drugih elementov	`<ime_elementa atribut1='vrednost1'> Besedilo …`
	`  <ime_elementa2>Besedilo…`
	`  <elt21></elt21>`
	`  <elt22></elt22>`
	`  </ime_elementa2>`
	`</ ime_elementa>`

Primer 11: Primer dokumenta XML

```
<knjiga>
     <poglavje>
          <naslov>Uvod</naslov>
     </poglavje>
     <poglavje> Jedro
          <podpoglavje>
               <naslov >Prvi del</naslov>
          </podpoglavje>
          <podpoglavje>
               <naslov>Drugi del</naslov>
          </podpoglavje>
     </poglavje>
     <poglavje>
          <naslov>Zaključek</naslov>
     </poglavje>
</knjiga>
```

Slika 6: XML iz primera 16 predstavljen kot drevo



Standard XML določa pravila, ki jim mora zadoščati dokument XML. :

- Dokument se naj prične z deklaracijo XML, ki definira različico specifikacije XML, s katero je dokument usklajen. To je obenem minimalna osnova za dokument.
- Dokument mora vsebovati natanko en element, ki vsebuje vse druge elemente. Ta element imenujemo korenski dokument (angl. root) ali element Dokument. V primeru št. 11 je to element <*knjiga*>.
- Pri elementih, ki vsebujejo podatke, se mora ime končne oznake ujemati z imenom začetne oznake: <knjiga …>…</knjiga>.
- Elementi, ki ne vsebujejo podatkov, se lahko zaključijo z znakom /: <knjiga/>.
- Imena elementov so občutljiva na velike in male znake. Sestavljajo jih lahko črke, številke, podčrtaji in pomišljaji. Besede na xml (ali XML, Xml …) so rezervirane za določene namene. Dvopičja se uporabljajo za določitev pripadnosti imenskemu prostoru (glej spodaj). Poimenovanje ne sme vsebovati presledkov in se ne sme začeti s številko.
- Elemente lahko gnezdimo, vendar se ne smejo prekrivati, podobno kot to velja pri algebraičnih izrazih za oklepaje. To pomeni, da kadar element vsebuje druge elemente, mora vsebovati njegovo začetno in končno oznako.
- Element lahko ima nič, enega ali več atributov. Ime atributa se z enačajem loči od vrednosti atributa, ki mora biti zapisana v narekovajih (opuščaj ' ali dvojni narekovaj ").
- Znaki <,>, & se uporabljajo za označevanje in za sklicevanje na posebne znake (kadar potrebujemo v podatkih takšen znak, uporabimo < > &).

Prednosti XML pred drugimi standardi za opis podatkovnih struktur so predvsem naslednje:

- je enostaven za razumevanje, oznake so v naravnem jeziku,
- oznake so del dokumenta in ne potrebujemo preučevati dodatnega dokumenta, da bi razumeli strukturo podatkov,
- je odprt standard, ki ga razvijalci programske opreme množično uporabljajo.

Imenski prostori

Z označevalnim jezikom lahko opredelimo besednjak za opis nekega področja. Na primer MathML (angl. Mathematical Markup Language) je besednjak za prikaz matematičnih notacij na svetovnem spletu. Posamezne besednjake lahko med seboj povezujemo. Pri hkratni uporabi več besednjakov v istem dokumentu lahko pride do prekrivanja uporabe istoimenskih elementov. Problem je razrešen z uporabo imenskih prostorov (angl. namespace). Opredelimo ga tako, da elementu dodamo atribut z naslednjo strukturo: `xmlns:prefiks="univerzalni identifikator vira"`. V primeru št. 12 je prikazano, kako s pomočjo imenskih prostorov uporabljamo v istem dokumentu istoimenske oznake iz različnih besednjakov. Opredelitev imenskega prostora dedujejo vsi potomci tega elementa, pri sklicevanju na imenski prostor pa pred posamezen naziv elementa ali atributa dodamo 'prefiks:'.

Primer 12: Primer uporabe imenskega prostora v dokumentu XML

```
<primer>
      xmlns:tekma="http://test.svetlana.org/tekma"
      xmlns:liki="http://test.svetlana.org/liki">

      <tekma:krog>
         <tekma:trajanje>15</tekma:trajanje>
         <tekma:polmer>20m<tekma:polmer>
      </tekma:krog>

      <liki:krog>
         <liki:barva>rdeča</liki:barva>
         <liki:polmer>10cm</liki:polmer>
      </liki:krog>

</primer>
```

1.3.1 Shema XML

Strukturo podatkov XML dokumenta s seznamom veljavnih elementov, atributov in povezav med elementi, oziroma t. i. slovnico ali besednjak, opredelimo v posebnem dokumentu, ki ga lahko zapišemo z različnimi jeziki:

- z jezikom **DTD** (angl. Document Type Definition), opredeljen v priporočilu XML 1.0 organizacije W3C (http://www.w3.org/TR/REC-xml/),
- z jezikom **W3C shema XML** (angl. XML Schema); za primerke shem uporabljamo kratico, ki jo imenujejo XSD (angl. XML Schema Definition); jezik je priporočilo organizacije W3C in je nastal na osnovi izkušenj z DTD in drugimi jeziki (http://www.w3.org/TR/xmlschema-1/),

- jezik **Relax NG** (angl. REgular LAnguage for XML Next Generation) je prvotno opredelila organizacija OASIS (angl. Organization for the Advancement of Structured Information Standards), danes je del ISO/IEC standarda 19757 - Document Schema Definition Languages (DSDL).

V primeru št. 13 podajam primer preprostega dokumenta XML, in sicer v vsakem od naštetih jezikov ustrezno specifikacijo za strukturo tega dokumenta.

Primer 13: Primeri shem XML dokumenta zapisani v različnih jezikih

Primer XML dokument:
```
<knjiga>
        <stran>Prva stran.</stran>
        <stran>Druga stran.</stran>
</knjiga>
```

Shema XML v jeziku DTD za zgornji primer XML dokumenta**:**
```
<!ELEMENT knjiga (stran*)>
<!ELEMENT stran (#PCDATA)>
```

Shema XML v jeziku RELAX NG za zgornji primer XML dokumenta**:**
```
<grammar xmlns="http://svetlana.org/primer">
   <start>
      <element name="knjiga">
         <oneOrMore>
            <element name="stran">
               <text/>
            </element>
         </oneOrMore>
      </element>
   </start>
</grammar>
```

Shema XML v jeziku RELAX NG – zgoščena sintaksa za zgornji primer XML dokumenta**:**
```
start = element knjiga
  {     element stran { text }+ }
```

Shema XML v jeziku XSD (W3C XML shema) za zgornji primer XML dokumenta**:**
```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="knjiga" type="Country">
      <xs:complexType name="Country">
        <xs:sequence>
            <xs:element name="stran" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
</xs:schema>
```

Dokument, ki opredeljuje določen besednjak, bomo v nadaljevanju imenovali shema XML dokumenta, nanašal pa se bo na definicijo z jezikom W3C, shema XML (XSD). Shema XML dokumenta opredeljuje, kateri elementi so dovoljeni ali

zahtevani v XML dokumentu tega tipa. Za dokument XML pravimo, da je veljaven, v kolikor je dobro oblikovan, torej zadošča specifikaciji jezika XML, in v kolikor elementi dokumenta ustrezajo zahtevam glede strukture iz njegove sheme. Slednje opišemo tudi kot potrditev skladnosti dokumenta s shemo. Pojem preverjanja vsebinske veljavnosti dokumenta XML se nanaša na:

- preverjanje strukture dokumenta,
- preverjanje vsebine posameznega elementa (skladnost s podatkovnim tipom),
- preverjanje integritete (povezave med posameznimi elementi v dokumentu ali med več dokumenti),
- preverjanje poslovnih pravil.

1.3.2 Kodiranje znakov

Znake za namene procesiranja z računalnikom zapišemo kot zaporedje ničel in enk. Preslikavi znaka v kombinacijo bitov z ustrezno dolžino pravimo kodiranje (povezuje grafično predstavitev znaka z njegovim binarnim zapisom). Na primer, z osem-bitno kombinacijo je možno zapisati 256 različnih znakov s številčnimi vrednostmi od 0 do 255 (od 00000000 do 11111111). V splošnem lahko z nizom n bitov zapišemo 2^n različnih znakov.

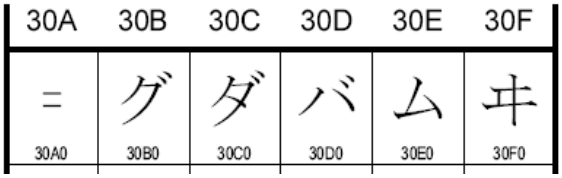
Med prvimi kodirnimi standardi je bila kodna tabela ASCII (American Standard Code for Information Interchange) oziroma ISO 646. ASCII kodna tabela je prvotno določala 7-bitne kode za 128 znakov, kot so črke angleške abecede, nekateri diakritični in krmilni znaki. Ker med njimi ni bilo šumnikov, cirilskih ali drugih posebnih znakov, saj jih v angleški abecedi ni, so se razvili dodatni 8-bitni kodni sistemi, ki ohranjajo prvotnih 128 znakov ASCII kodne tabele in preostalih 128 uporabljajo za posamezne skupine evropskih jezikov. Tovrstne razširitve niso rešile vseh težav s kodiranjem znakov; na primer, kadar je potrebno prikazovati besedila iz različnih jezikov v istem dokumentu. Zanje tudi ne obstajajo zanesljivi testi za ugotavljanje tipa kodne strani iz zapisa, kar povzroča velike težave pri prikazovanju datotek.

Nastal je standard ISO 10646[3], ki opredeljuje univerzalen nabor znakov, s kratico UCS (angl. Universal Charcter Set). V UCS postopoma vključujejo znake, simbole,

[3] Od leta 1991 je razvoj UCS sistema pod okriljem ISO in konzorcija Unicode (http://www.unicode-org); sistem UCS je del industrijskega standarda Unicode.

ideograme iz vseh svetovnih jezikov. Vsak znak je predstavljen s sliko in lokacijsko kodo (angl. code point) v sistemu vseh znakov. Na sliki št. 7 je primer grafične predstavitve nekaj znakov in lokacijskih kod iz japonske pisave katakana.

Slika 7: Nekaj znakov pisave katakana iz sistema UCS



Vir: http://www.unicode.org

Lokacijske kode so v UCS logično razdeljene na 17 ravni, med njimi je v splošni uporabi samo prva, osnovna večjezična raven, s kratico BMP (angl. Basic Multilingual Plane). Vsaka raven vsebuje po 65,536 (2^{16}) znakov, vsak znak se lahko zapiše z 32 bitno kodo. Ker je 32 bitno kodiranje za večino pisav potratno, so v uporabi optimalnejša kodiranja, najpogosteje UTF-8 in UTF-16.

Kodiranje UTF-8 zapisuje vsak znak bodisi z enim, dvema, tremi ali štirimi okteti:

- en oktet za 128 ASCII znakov (Unicode števila U+0000 do U+007F),
- dva okteta za Llatinske znake, vključno z diakritični znaki, grško pisavo, cirilico in še nekaj drugimi pisavami (Unicode števila U+0080 to U+07FF),
- trije okteti za večino ostalih pisav, kot so kitajska, japonska in hindujska,
- štirje okteti za znake iz drugih ravni, ki se zelo redko uporabljajo v praksi.

V primeru št. 14 je prikaz postopka UTF-8 kodiranja za znak aleph (א). Ilustrirano je tudi, da je UTF-8 kodiranje zasnovano tako, da za vsak oktet nedvoumno vemo, ali predstavlja znak, zapisan z enim, dvema, tremi ali štirimi okteti. Na primer, vsak oktet, ki se prične z 0, predstavlja enega izmed ASCII znakov.

Primer 14: Primer postopka UTF-8 kodiranja znaka

Na primer znak **aleph(א),** čigar Unicode število je U+05D0, bo v UTF-8 kodiranju predstavljen z naslednjim številom:

- ker pade v nabor Unicode znakov med U+0080 in U+07FF, bo kodiran z dvema zlogoma, in sicer po pravilih UTF-8 kodiranja po maski "110xxxxx 10xxxxxx"
- šestnajstiška koda 0x05D0 ustreza dvojiškemu številu 101-1101-0000 (zaradi preglednosti zapisano s pomišljaji).
- Teh 11 bitov razvrstimo v zgornjo masko (po vrsti nadomestimo znake "x") in dobimo **11010111 10010000.**

Kodiranje UTF-16 je podobno zasnovano kot UTF-8, le da zapisuje znake prve ravni vedno z dvema oktetoma. UTF-16 kodiranje je zaradi tega povezano s problemom vrstnega reda zapisovanja teh dveh oktetov (možnosti sta z leve ali z desne), kar izhaja iz razlik med mikroprocesorji različnih proizvajalcev. Eden od mehanizmov, ki rešuje problem vrstnega reda, je oznaka BOM (angl. Byte Order Mark) na začetku datoteke, ki pove, v kakšnem vrstnem redu so okteti zapisani. V idealnem svetu bi uporabljali samo eno množico pravil, vendar zaradi omenjenih razlik med mikroprocesorji to ni mogoče. UTF-16 kodiranje je primernejše kot UTF-8, kadar uporabljamo kitajsko, japonske ali hindi pisave, ker je za zapis potrebno manj prostora, v primerjavi z uporabo kodiranja UTF-8, med tem ko velja za znake z Unicode kodo manjšo U+0800 ravno obratno.

Na žalost na področju formatiranja dokumentov nimamo široko sprejetih standardov za zapisovanje besedila. Zgoraj opisani so za zapisovanje posameznih znakov. Ker posamezni operacijski sistemi uporabljajo različne znake za označevanje konca vrstice, bo pomensko enako besedilo kljub rabi standardnega kodiranja znakov na različnih sistemih zapisano različno.

Za namene digitalnega podpisovanja je preslikava iz znakov v niz bitov ključnega pomena, ker so predmet digitalnega podpisa binarni podatki. Torej, če bomo enkrat besedilo »ŠČŽ« pretvorili v niz bitov z uporabo ustreznih kodnih kombinacij iz kodne tabele »Windows-1250«, drugič pa s kodno tabelo »UTF-8«, bomo dobili dva različna niza bitov, zaradi tega bomo seveda izračunali dva različna prstna odtisa in posledično dobili dva različna digitalna podpisa za isto besedilo (opomba: predpostavljena je uporaba istega ključa za šifriranje v obeh primerih).

1.3.3 Vključevanje binarnih podatkov v XML

XML je opredeljen kot zaporedje znakov; da lahko vanj vključimo zaporedje bitov, jih moramo predhodno zapisati kot znake. Obstaja več načinov, kako lahko poljubno zaporedje bitov zapišemo kot znake, na primer šestnajstiško ali v internetnem sistemu kodiranj pogosto Base 64 kodiranje (Borenstein, 1996, str. 24). Kod dodatno možnost omenimo, da lahko v XML dokumentu navedemo referenco na zunanji vir podatkov. Kodiranje binarnih podatkov ni uporabno zgolj za namene dela z XML dokumenti. S tem omogočimo, da lahko v poljubna sporočila, ki jih prenašamo med sistemi ali omrežji, vključujemo binarne podatke brez nevarnosti, da bi vsebovali zaporedje bitov, ki bi imeli v trenutnem okolju poseben pomen, na primer zaključek prenosa.

Šestnajstiško kodiranje (angl. hex encoded binary data) vsak oktet pretvori iz dvojiškega v šestnajstiško število in ga zapiše s ciframi šestnajstiškega številskega sistema (števke ali črke abecede od A do F). Na primer: "0FB7" je šestnajstiško kodiranje za naslednje zaporedje dveh oktetov 0000111110110111.

Z **Base64 kodiranjem** poljubno zaporedje bitov pretvorimo v zaporedje natisljivih ASCII znakov tako, da ga lahko kadarkoli z obratnim postopkom pretvorimo v izvorno zaporedje. Zaporedje 3 oktetov (24 bitov) razdelimo na 4 sklope po 6 bitov. Dobimo 4 števila v vrednostih 0 - 63, te pa pretvorimo v znake, ki jih vse naprave interpretirajo enako in se jih da izpisati:

0 - 25 : A - Z (angleška abeceda)
26 - 51 : a - z
52 - 61 : 0 - 9
62 : +
63 : /

Nepopolne bloke na koncu zaporedja zapolnimo z znaki =.

1.3.4 Kanonična oblika XML

V skladu s priporočili W3C rečemo, da imata dva XML dokumenta enako kanonično obliko, kadar sta logično ekvivalentna znotraj danega konteksta. To je še posebej pomembno pri izračunu prstnega odtisa dokumenta. XML je tekstovni dokument, prstni odtis pa se računa iz njegove binarne predstavitve. Kadar procesiramo XML, ga večinoma naložimo v ustrezne drevesne strukture elementov in atributov. Postopek, pri katerem podatke pretvarjamo iz objektnih oblik v zaporedja zlogov, imenujemo »serializacija«. Ime izhaja iz tega, ker se za nek objekt shrani v datoteko »zaporedno« vsebina vseh zlogov pomnilnika. Zaradi lastnosti XML, da dokument ohrani svojo celovitost, če na primer zamenjamo vrstni red atributov, je potrebno dokument pred računanjem prstnega odtisa spraviti v njegovo kanonično obliko. Primer št. 15 kaže tri primere elementov XML, ki so logično ekvivalenti, njihovi binarni zapisi pa se med seboj razlikujejo.

Primer 15: Primeri različnih XML elementov, ki so logično ekvivalentni

Naslednji primeri XML elementov so po standardu XML logično ekvivalenti:

- <oseba ime=”Janez” priimek=”Novak” ></oseba>
- <oseba priimek=’Novak’ ime=’Janez’ ></oseba>
- <oseba ime=”Janez” priimek=”Novak”/>

Organizacija W3C je opredelila več standardnih algoritmov kanonizacije (Boyer, 2001). Tukaj povzemam v ilustracijo nekaj pravil kanonizacije:

- znaki dokumenta XML se kodirajo po UTF-8,
- prelomi vrstic se zamenjajo z znakom #xA,
- prazni elementi (< />) se pretvorijo v začetne in zaključne oznake (<></>),
- vrednosti atributov so zapisane v dvojnih narekovajih,
- presledki izven elementa se normalizirajo,
- atributi in imenski prostori se v okviru elementa uredijo po abecednem vrstnem redu,
- odstrani se odvečne deklaracije imenskih prostorov iz elementov.

Posebno pozornost je potrebno nameniti imenskim prostorom (glej str. 26). Primer št. 16 prikazuje dva logično ekvivalentna dokumenta XML, ki pa se zaradi uporabe različnih prefiksov za označevanje pripadnosti oznake imenskemu prostoru razlikujeta v binarnem zapisu.

Primer 16: Kanonizacija in imenski prostori

```
<?xml version='1.0' encoding='UTF-8' ?>
<root  xmlns ='a1.xsd' xmlns:b1='b1.xsd' >
      <b1:b />
</root>

<?xml version='1.0' ?>
<a1:root xmlns:a1 =a1.xsd' xmlns:b2='http://test. org/b1.xsd'>
      <b2:b />
</a1:root>
```

S temi pravili smo končali s predstavitvijo vseh elementov, potrebnih za razumevanje potreb, zahtev in kriterijev za pripravo in vzdrževanje dokazov o verodostojnosti elektronskih dokumentov v skladu s standardom ERS. Prvi sklop elementov se nanaša na tehnologije in mehanizme zaupanja za ustvarjanje dokazov o obstoju dokumenta in njegovi nespremenjenosti od časa nastanka, in sicer sta to digitalni podpis in časovni žig. Drugi sklop elementov se nanaša na priporočila, smernice, standarde in zakonodajo s področja hrambe elektronskih dokumentov. Tretji sklop elementov se nanaša na obliko zapisa dokazil o verodostojnosti elektronskih dokumentov; obravnavana sta specifikacijska jezika ASN.1 in XML, s poudarkom na razlikah oziroma podobnostih med njima in tistih značilnostih, ki vplivajo na postopke priprave dokazil.

2. Verodostojnost elektronskih dokumentov

2.1. Pojem verodostojnosti

Pojem verodostojen lahko opišemo z avtentičen, originalen, neponarejen, ki izvira od avtorja, pristen, izviren, ki mu je verjeti; tudi: ki se ujema z originalom, natančen, točen, v skladu z resničnostjo (SSKJ, 2008). ZVDAGA v 27. členu in Uredba o varstvu dokumentarnega in arhivskega gradiva v 16. členu opredeljujeta avtentičnost digitalnega gradiva kot dokazljivost povezanosti reproducirane vsebine z vsebino izvirnega gradiva oziroma izvorom tega gradiva.

V Smernicah za upravljanje elektronskih dokumentov iz arhivske perspektive (ICA, 1997, str. 24) in kasneje v specifikaciji MoReq (DLM Forum, 2005, str. 28) je verodostojnost opredeljena kot »ohranjanje izvirnih lastnosti dokumenta v daljšem časovnem obdobju glede na kontekst, strukturo in vsebino«, to pomeni, da je dokument tisto, kar naj bi bil. MoReq poudarja, da je bistveno, da se od takrat, ko je dokument zajet, vsi njegovi deli, struktura in metapodatki, potrebni za zagotovitev avtentičnosti dokumenta, ne spreminjajo več. Da bi dokumenti ohranili avtentičnost, moramo zajete dokumente ohraniti v nespremenljivi obliki in jih v celotnem življenjskem ciklusu zavarovati pred namernimi ali naključnimi spremembami vsebine, konteksta, strukture in videza. Tako MoReq zahteva, da mora sistem za zajem arhivskega gradiva opozoriti na nepopolnost in neskladnost dokumenta ob zajemu, ki bi lahko vplival na kasnejše dokazovanje avtentičnosti (na primer dokument vsebuje podpis s preklicanim potrdilom). V Smernicah (ICA, 1997, str. 24) je verodostojnost opisana tudi kot ohranjanje izvorne zanesljivosti dokumenta, pri čemer je za izvorno zanesljivost seveda odgovoren avtor dokumenta.

V priporočilih DoD (Department of Defense, 1997, str. 10) je verodostojnost opisana kot okoliščine, ki dokazujejo, da je dokument izvoren na podlagi načina, s katerim je dokument bil prenesen skozi prostor in čas, na podlagi oblike (formata, v katerem se nahaja od sprejema), na podlagi stanja prenosa (preprostost, celovitost in učinkovitost dokumenta ob začetku hrambe) ter na podlagi samega načina varovanja in hrambe.

Priročnik Elektronski dokumenti (Mednarodni arhivski svet, 2006, str. 35) podrobneje pojasnjuje, da je za ugotavljanje verodostojnosti pri elektronskem dokumentu treba pokazati, da je dokument obstajal takrat, ko trdimo, da je, in da njegove vsebine niso bile spremenjene, odkar je vstopil v arhiv oziroma moramo

dokazati, da smo ohranili to, kar smo vedeli o njem, ko smo ga prejeli. Verodostojnost dokumenta lahko po navadi dokažemo brez poznavanja njegovih vsebin (ali celo brez vsake možnosti dostopa do njih).

Verodostojnost dokumenta ne more temeljiti samo na ohranjanju bitov, saj je pomemben celoten kontekst dokumenta (Blanchette, 2006, str. 15). Ohranjati je potrebno kontekstualno verodostojnost dokumenta, torej vse nadzorne, administrativne procese, skozi katere je dokument šel v svojem življenjskem ciklu. Blanchette primerja digitalni podpis z uporabo DNA kot dokazom o prisotnosti osumljenca na prizorišču kriminala. DNA je verodostojen kot dokazno gradivo samo, v kolikor je bil verodostojen celoten proces od zajema DNA do prenosa v laboratorij in v nadaljnjo hrambo.

Ugotovitve glede verodostojnosti za storitev dolgoročnega arhiviranja lahko strnemo v naslednja ključna tveganja:

- izguba integritete podatkov,
- preklic digitalnega potrdila retrogradno razveljavi podpise v primerih, kadar digitalni podpis temelji na infrastrukturi javnih ključev,
- prenehanje veljavnosti mehanizmov za avtorizacijo in s tem tudi od njih odvisen dostop do podatkov,
- aplikacije za prikazovanje določenega formata lahko dolgoročno izginejo,
- uporabljeni kriptografski algoritmi lahko postanejo nezanesljivi,
- izguba ustreznih aplikacij za preverjanje digitalnih podpisov.

2.2. Slovenska zakonodaja na področju zagotavljanja verodostojnosti arhivov

Slovenija je ena redkih držav, ki ima sistemski zakon o elektronskem arhiviranju (Berčič, 2008, str. 13). Druge države od podjetij v zvezi z elektronskim arhiviranjem ne zahtevajo dodatnih aktivnosti, kot so pisanje notranjih pravil za elektronsko hrambo pravno veljavnih dokumentov ter potrjevanje pravil pred uradnimi organi.

Zagotavljanja in dokazovanje celovitosti in avtentičnosti elektronskih dokumentov obravnavajo v Sloveniji naslednji zakoni in podzakonski akti:

- z ZVDAGA so opredeljene zahteve, ki jim mora zadostiti arhiviranje v elektronski obliki, če gre za dokumente, pri katerih bo potrebna formalno-pravna veljava. ZVDAGA govori na splošno o tem, da mora biti digitalno

gradivo ves čas avtentično in celovito, hkrati zakon ne predpisuje konkretnih tehnologij.

- Uredba o varstvu dokumentarnega in arhivskega gradiva v 17. in 18. členu opredeljuje zahteve glede ohranjanja avtentičnost in celovitost gradiva v digitalni obliki na naslednji način:
    - Avtentičnost in celovitost gradiva se zagotavljata z dodajanjem varnostnih vsebin gradivu (npr. digitalni podpis, časovni žig in druga tehnološka sredstva) ali z zagotavljanjem dodatnih organizacijskih ukrepov.
    - Če se sčasoma izgubi vrednost dokazov, se v strogo nadzorovanih postopkih dodajajo vsebine, ki ohranjajo vrednost dokazov (npr. časovni žigi ali drugi metapodatki).
- Glede podaljševanja dokazil se v 33. členu Uredbe o pogojih za elektronsko poslovanje zahteva, da je treba za elektronsko podpisane podatke najkasneje mesec pred iztekom roka digitalnega potrdila zagotoviti ponoven digitalni podpis s strani vseh podpisnikov ali potrditev podatkov z varnim časovnim žigom.
- ETZ v poglavju 3.9.6. opredeljuje zahteve, ki jih mora izpolnjevati informacijski sistem za upravljanje z dokumenti, s kratico ISUD, glede hrambe elektronskih podpisov, časovnih žigov, varnostnega šifriranja in elektronskih vodnih znakov za slikovni, avdio ali video material.

2.3. Koncept zaupanja vredne storitve arhiviranja (TAS) v mednarodni praksi

Koncept zaupanja vredne storitve arhiviranja, s kratico TAS (angl. Trusted Archival Service), zasledimo v zaključnem poročilu konzorcija Evropske iniciative za standardizacijo digitalnega podpisa, s kratico EESSI (angl. European Electronic Signature Standardization Initiative), ki je bil ustanovljen z namenom, da pripravi celovito poročilo o tem, kako ustrezno podpreti s standardi zahteve Evropske direktive o elektronskem podpisu (EESSI, 1999, str. 52). Storitev TAS je namenjena vzdrževanju zapisov o obstoju in veljavnosti elektronskih podpisov čim bližje času njihovega nastanka in za potrebe dolgoročnih dokazil. Koncept torej opisuje novo vrsto komercialne storitve za zagotavljanje dolgoročne verodostojnosti digitalno podpisanih dokumentov. Podrobneje je ta storitev opisana v posebnem poročilu z naslovom TAS (Libon, 2000). Med zahtevami je tudi ta, da podpisnik dokumenta vidi natanko to, kar podpiše. To načelo

imenujemo načelo WYSIWYS (angl. What You Sign Is What You See). Med tehničnimi zahtevami sta v dokumentu TAS poudarjena podaljševanje veljavnosti digitalnega podpisa s časovnim žigom ter kompatibilnost programske in strojne opreme za nazaj, bodisi z neposredno ohranitvijo ali z razvojem orodij za emulacijo, z namenom zagotavljanja možnosti ogleda dokumentov in preverjanja različnih vrst podpisov (Libon, 2000, str. 37). Slednje je poudarjeno predvsem zato, ker je najpogostejša praksa v svetu elektronskih dokumentov migracija dokumentov iz ene obliko v drugo, na primer iz Wordovega dokumenta v PDF, pri čemer seveda ni možna ohranitev veljavnosti digitalnega podpisa brez dodatnih postopkov. Projektni konzorcij Advanced eGovernment Information Service Bus v sklopu šestega okvirnega programa EU je med svojimi izročki izdelal tudi demonstracijo storitve za transformacijo digitalnega podpisa iz ene oblike v drugo, na primer iz CMS v XAdES podpis. Demonstrirana storitev uporabniku vrne kot rezultat originalen podpis, podpis potrdila o veljavnosti originalnega podpisa in nov podpis.

Storitev TAS je mogoče zagotoviti kot skupek standardiziranih tehnologij na petih ravneh, katerih kompleksnost je uporabniku dostopna preko enotnega vmesnika (Jerman Blažič, 2005, str. 5):

- fizična raven - pisanje in branje na medije, fizična obstojnost;
- raven upravljanja - dostop do podatkov, povezave z metapodatki, določanje časa hrambe dokumenta ipd.;
- raven zagotavljanja stabilnosti - zagotavljanje integritete in verodostojnosti;
- predstavitvena raven - berljivost in interpretacija dokumentov;
- aplikacijska raven - uporabniški vmesniki;

Internetni standard RFC 4810 opredeljuje zahteve za vzpostavitev dolgoročnega arhiva in storitev arhiviranja. Storitev je zamišljena kot sestavni del dokumentnega sistema in dokazila, ki jih ustvarja, so lastnosti dokumentov, za katere sicer skrbi dokumentni sistem (Brandner, 2007b, str. 5). Zahteve za storitev se nanašajo predvsem na dokaze, ki se lahko uporabijo za dokazovanje, da je arhivirano gradivo obstajalo v določenem trenutku v preteklosti in da se od takrat ni spreminjalo. Če je gradivo digitalno podpisano, storitev omogoča razpoznavanje podpisnika. Ključne zahteve, ki jim tovrstna storitev mora zadostiti, so (Brandner, 2007b, str. 7−13):

- sprejem gradiva za različna obdobja arhiviranja, vključno z njihovimi metapodatki, posredovanje informacij o stanju, izvoz dokazil in brisanje podatkov,
- delovanje v skladu z vnaprej opredeljeno in uporabniku poznano politiko, ki opisuje dejavnike in parametre delovanja,
- zagotavljanje dokazil o času sprejema gradiva in o celovitosti od takrat tako, da so razvidne morebitne spremembe s strani vzdrževalcev storitve,
- podpirati zaupnosti podatkov, torej zagotavljati dokazila tudi za šifrirane podatke, vendar tako, da se dokazila o celovitosti nanašajo na izvorne podatke,
- omogočiti, da se dokazila prenesejo k drugemu ponudniku storitve,
- omogočati skupinsko obdelavo vhodnega gradiva in povezovanje več datotek v eno logično enoto.

V pripravi je IETF **standard za komunikacijski protokol z dolgoročnim arhivom**, s kratico **LTAP** (angl. Long-term Archive Protocol), ki opredeljuje komunikacijska sporočila za pet osnovnih funkcij: vlaganje gradiva v arhiv, ugotavljanje statusa gradiva, preverjanje veljavnosti dokazil, izvoz dokazil in brisanje podatkov, povezanih z gradivom iz arhiva (Jerman Blažič, 2008).

Potrebno **infrastrukturo za izvajanje storitve TAS** opišemo z naslednjimi sestavnimi deli (Jerman Blažič, 2007, str. 6):

- sistem za interpretacijo gradiva,
- sistem za validacijo gradiva (preverjanje podpisov ipd.),
- sistem za upravljanje s podatkovnim skladiščem (dokumenti sistemi),
- sistem za dolgoročna dokazila; t. i. konzervacijski atributi s skladiščem za evidenčne zapise.

Nastavitve in parametre, ki vplivajo na proces vlaganja gradiva v sistem TAS in kasnejše vzdrževanje, obravnavamo kot **politiko arhiviranja**, ki jo v splošnem razdelimo na naslednje kategorije (Jerman Blažič, 2007, str. 4−5):

- splošna politika delovanja, ki opredeljuje vrste storitev, ki jih TAS zagotavlja, vključno z vrstami podatkovnih objektov, časovnimi okviri za procesiranje, uničevanje podatkov, postopki za obnovo podatkov ipd.,

- vzdrževalna politike, ki opredeljujejo specifične lastnosti, kot so uporabljeni algoritmi, mehanizmi redundantnega vzdrževanja, pravila za skupinske obdelave ipd.,
- avtorizacijska politika, ki opredeljuje pravice do uporabe storitev in pravice nad objekti.

2.4. Sistem za generiranje dolgoročnih dokazil

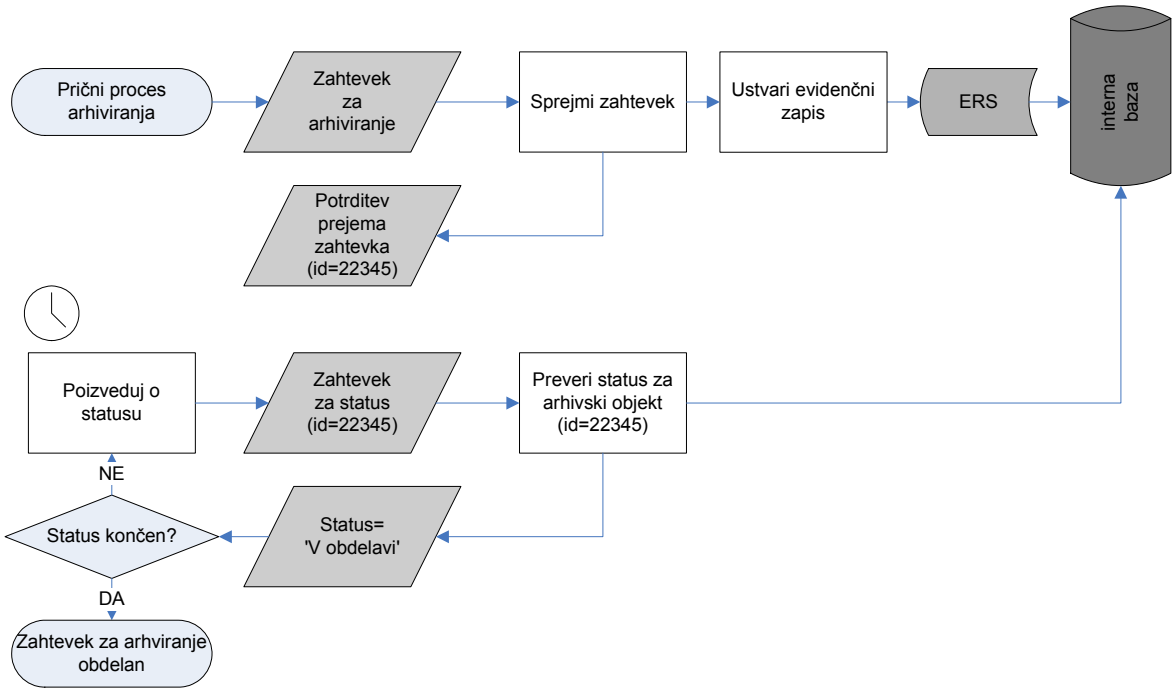
Načeloma s sistemom za dolgoročna dokazila, v nadaljevanju s kratico SDD, komunicira storitev TAS; uporabnik naj ne bi imel neposrednega dostopa. Standard ERS ne opredeljuje posebnega komunikacijskega protokola, zato bomo v nadaljevanju predpostavili, da poteka interakcija med sistemom TAS in njegovim podsistemom SDD po protokolu LTAP.

Na izvedbo procesa ustvarjanja dokazil vplivajo določene nastavitve, pod katerimi SDD deluje, in parametri, ki jih je uporabnik izbral z določeno arhivsko politiko. Naštejmo nekatere najbolj pomembne:

- izbira individualne ali skupinske obdelave,
- izdelava redundantnih evidenčnih zapisov,
- čas hrambe dokazil,
- algoritmi, s katerimi zna/lahko sistem dela,
- izbira ponudnikov storitve časovnega žigosanja idr.

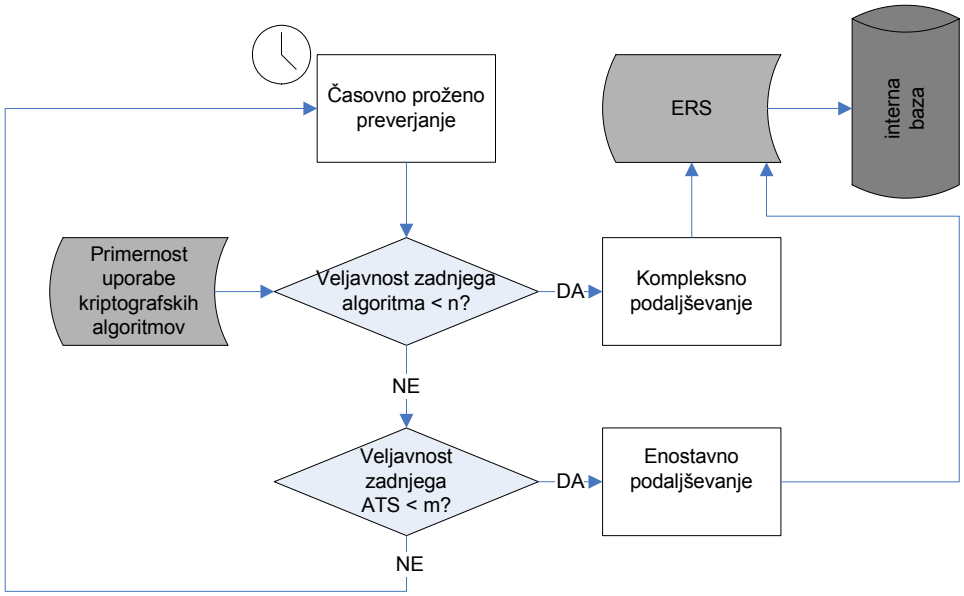
Za SDD sta ključnega pomena proces ustvarjanja dokazil in proces vzdrževanja dokazil. Proces ustvarjanja dokazil sproži storitev TAS ob prejemu elektronskega dokumenta v arhiv. V skladu s protokolom LTAP storitev TAS pošlje sistemu SDD zahtevek za izvedbo funkcije vlaganje gradiva v arhiv. Ker je narava procesa pridobivanja časovnega žiga časovno zahtevna in predvsem zaradi optimizacij delovanja, sistem SDD na tak zahtevek vrne zgolj potrditev, ali je zahtevek sprejet. V kolikor je zahtevek za funkcijo vlaganja gradiva v arhiv sprejet v obdelavo, lahko uporabnik po rezultatu te funkcije poizveduje s funkcijo status tako dolgo, dokler ne dobi odgovora, ki pomeni uspešnost procesa ustvarjanja dokazil ali pa neuspešnost. Oba delovna procesa, arhiviranje in preverjanje statusa, sta prikazana na sliki št. 8.

Slika 8: Delovni proces za metodi arhiviranje in preverjanje statusa



Po sprejemu gradiva sistem ustvari evidenčni zapis. Ta točka hkrati pomeni začetek arhivskega procesa, ki mu mora slediti avtomatiziran proces vzdrževanja veljavnosti evidenčnega zapisa. Na sliki št. 9 je prikazano, da se ob določenih časovnih intervalih proži preverjanje veljavnosti zadnjih uporabljenih kriptografskih algoritmov in preverjanje veljavnosti zadnjega časocnega žiga.

Slika 9: Avtomatiziran proces vzdrževanja evidenčnih zapisov



2.4.1 Rešitev eKeeper

Implementacijo opisanih postopkov smo izvedli v tehnološkem centru SETCCE, v procesu sodelovanja z IETF ter drugimi organizacijami pri pripravi standardov na področju arhiviranja. SETCCE danes trži rešitev z imenom eKeeper. Sistem je na primer v uporabi pri večjih slovenskih izdajateljih elektronskih računov, poteka tudi postavitev sistema pri Belgijski notarski zbornici, kjer je pričakovana doba hrambe prejetega gradiva 100 let. EKeeper je sistem za dolgoročna dokazila in hkrati za validacijo dokumentov. V povezavi z dokumentnim sistemom izpolnjuje zahteve zakonsko usklajenega sistema za dolgoročno arhiviranje.

Infrastrukturo sistema eKeeper sestavljajo naslednji elementi:

- vmesniki za komunikacijo v skladu s specifikacijami protokola LTAP,
- moduli za validacijo prejetega gradiva in zbiranje komplementarnih dokazil, kot so seznami preklicanih potrdil ter podobno,
- sistem za ustvarjanje in samodejno podaljševanje veljavnosti dokazil o verodostojnosti.

Avtorica je sodelovala kot projektni vodja in razvijalec pri razvoju različice sistema eKeeper 3.0, ki podpira poskusno verzijo standarda ERS z zapisi v XML obliki.

2.5. Preverjanje verodostojnosti

Verodostojnost papirnih dokumentov temelji na lastnoročnem podpisu in obsega dokazovanje, da podpis ni bil ponarejen, in da se dokument od takrat, ko je bil podpisan, ni spremenil. Kadar podpišemo dokument, potrdimo tudi, da se strinjamo z njegovo vsebino. Za zagotavljanje verodostojnosti podpisov lahko uporabimo tudi zunanje storitve, ki delujejo na osnovi javnega zaupanja v njihovo verodostojnost. Lastnoročni podpis in dokument lahko overi notar, ki pri sebi tudi shrani podpisan in overjen dokument.

Za zagotavljanje verodostojnosti elektronskih dokumentov uporabljamo vsebinsko podobne mehanizme kot za papirne dokumente. Lastnoročni podpis nadomestimo z digitalnim podpisom in notarske storitve s storitvami pooblaščenih izdajateljev potrdil za podpisnike ter s storitvami pooblaščenih časovnih overiteljev obstoja dokumenta.

Zagotavljanje avtentičnosti in celovitosti tako obsega:

- preverjanje digitalnih podpisov,
- ohranjanje veljavnosti digitalnih potrdil,
- možnost dokazovanja časa prejema gradiva v arhiv in časa preverjanja veljavnosti podpisov,
- obnavljanje dokazov o avtentičnosti in celovitosti, ki sčasoma izgubijo vrednost.

Dokazila o verodostojnosti dokumenta naj bi omogočala, da lahko s pomočjo tehnoloških rešitev kadarkoli neizpodbitno dokažemo obstoj in veljavnost digitalnih vsebin ob določenem času v preteklosti in nespremenljivost digitalnega gradiva za celotno obdobje arhiviranja.

2.5.1 Preverjanje veljavnosti digitalnega podpisa

Digitalni podpis zaradi svoje tehnološke vsebine ni tako trajen kot lastnoročni podpis. Temelji namreč na kriptografskih metodah, ki v določenem trenutku, glede na tehniko (zmogljivost računalnikov) zagotavljajo, da v razumnem času ni mogoče iz šifriranega sporočila dobiti izvornega sporočila in s tem tudi ključa za šifriranje ali podpisovanje.
Digitalni podpis generiramo s pomočjo naslednjih kriptografskih funkcij:

- zgostitvene funkcije, ki iz zaporedja bitov dokumenta·izračunajo zaporedje točno določene dolžine glede na uporabljeno funkcijo, na primer 160 bitov, ki jih potem podpišemo (prstni odtis),
- algoritma za podpisovanje in preverjanje, ki uporabljata par ključev za ustvarjanje in preverjanje podpisa,
- proces ustvarjanja para ključev.

V takšnem modelu je podpis zanesljiv, dokler je imetnik ključa za podpis edini sposoben ustvariti podpis, katerega bo potrdil algoritem za preverjanje podpisa. Zaradi tega mora za zgostitveno funkcijo veljati, da je praktično nemogoče, da bi za prstni odtis poiskali poljuben izvoren dokument; za par ključev mora veljati, da iz javnega dela ne moremo izračunati njegovega zasebnega ključa, in da v procesu ustvarjanja ključa nastaneta naključno. Tako na ravni tehnologije govorimo o **kriptografski veljavnosti** digitalnega podpisa, ki jo preverjamo tako, da podpis dešifriramo s podpisnikovim javnim ključem in dešifrirano vrednost primerjamo s prstnim odtisom podpisane vsebine.

Na ravni infrastrukture javnih ključev govorimo o **formalni veljavnosti** digitalnega podpisa. To naredimo tako, da preverjamo:

1. ali je bilo digitalno potrdilo v času podpisa časovno veljavno (vsako potrdilo se izda za omejen čas veljavnosti, večinoma od 2 do 5-ih let). Praksa glede tega kriterija je med državami različna. V Nemčiji je digitalen podpis veljaven, če je bilo kvalificirano potrdilo veljavno v času podpisa; torej je dokazovanje časovne veljavnosti digitalnega potrdila v času podpisa še posebej pomembno, ker je lahko podpisnik pozneje s preklicem potrdila poizkušal razveljaviti podpis. V Avstriji in na Poljskem pa le, če ga je mogoče preveriti z veljavnim kvalificiranim potrdilom (Adamski, 2006, str. 2). Slednje skorajda onemogoča arhiviranje podpisov, saj izdajatelj ne bo izdal novega potrdila za stari ključ; podpisnik sicer lahko ponovno podpiše z drugim ključem vsebino in prejšnji podpis.

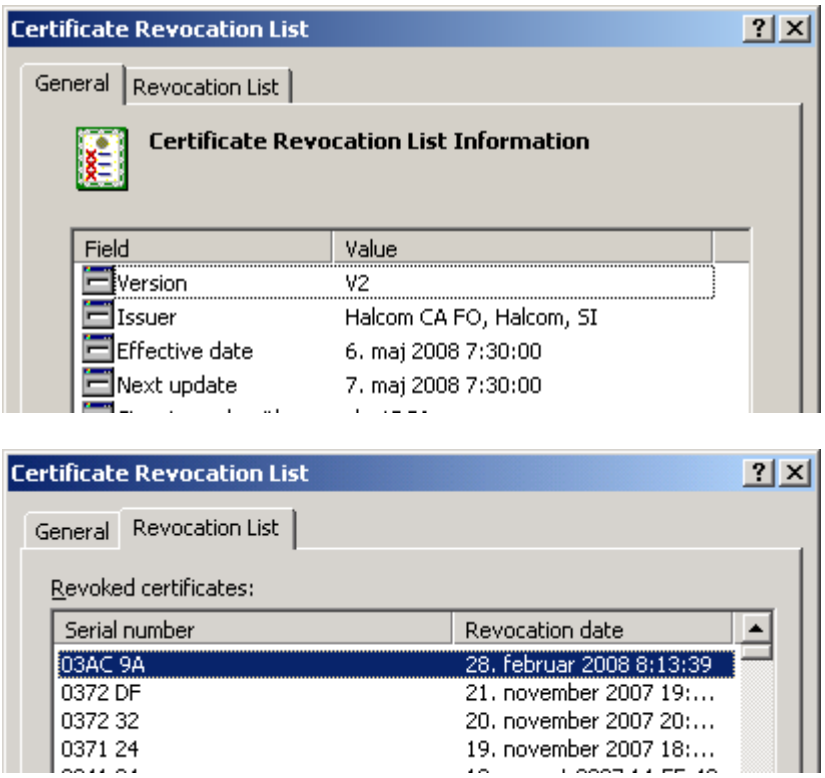
2. da digitalno potrdilo ni preklicano, torej da se njegova serijska številka ne nahaja na seznamu preklicanih potrdil izdajatelja potrdila. Slovenska overitelja Sigen CA in Halcom CA bosta na podlagi pravilno izpolnjenega zahtevka za preklic potrdilo preklicala najkasneje v štirih urah po prejemu zahtevka. Ilustracija na sliki št. 10 prikazuje izsek seznama preklicanih potrdil overitelja Halcom CA, izdanega 6. maja 2008, ob 7:30 uri.

   Priporočila Ministrstva za javno upravo (CVI, 2003, str. 30) svetujejo preverjanje preklicanosti digitalnega potrdila z uporabo prehodnega obdobja:

   - v času 1 podpisnik podpiše dokument,
   - podpisu je dodan časovni žig v času 2, ki je za časom 1,
   - času 2 dodamo prehodno obdobje, ki se konča ob času 3,
   - potrdilo je preklicano ob času 4, ki je lahko pred ali po času 3,
   - podpis je prvič overjen ob času 5, ki ne sme biti pred časom 3.

   Če je potrdilo preklicano pred časom 3, potem podpisa ne moremo overiti. Če je potrdilo preklicano po času 3, potem podpis lahko overimo.

Slika 10: Seznam preklicanih digitalnih potrdil



Vir: Spletne strani overitelja Halcom CA, http://www.halcom-ca.si/index.php?section=43

3. da je veriga digitalnih potrdil popolna. Veriga digitalnih potrdil je zaporedje potrdil od korenskega izdajatelja do izdajatelja potrdila podpisnika (korenski izdajatelj je lahko pooblastil izdajatelja, ki je lahko pooblastil izdajatelja…, ki je izdal uporabljeno potrdilo). Popolnost verige pomeni veljavnost vseh potrdil. Dodatno lahko glede na kontekst uporabe razločujemo tudi v stopnji zaupanja izdajateljem in dovolimo oziroma priznavamo samo tiste, ki jih označimo kot zaupanja vredne (na primer na ravni politike podpisovanja ali osebne odločitve).

Ob digitalnem podpisu, ki temelji na infrastrukturi javnih ključev, je torej treba za zagotavljanje dolgoročne verodostojnosti ohraniti:

- digitalno potrdilo podpisnika,
- verigo digitalnih potrdil od podpisnika do krovnega izdajatelja,
- seznam preklicanih potrdil izdajatelja.

V tabeli št. 2 podajam pregled možnosti za vključevanje dokazil o verodostojnosti v sam digitalni podpis glede na izbran standard za zapis podpisa.

Tabela 2: Vključevanje dokazil v podpis

	PKCS#7	CMS	PDF	XMLDsig	XAdES
digitalno potrdilo podpisnika	da	da	da	da	da
veriga digitalnih potrdil	da	da	da	da	da
crl	da	da	da	da	da
časovni žig	ne	da	da	ne	da

2.5.2 Preverjanje veljavnosti časovnega žiga

Časovna veljavnost arhivskega časovnega žiga je določena z veljavnostjo digitalnega potrdila za zasebni ključ, s katerim je TSA podpisal zaščitene vsebine. V praksi je veljavnost takega potrdila od dveh do petih let. Veljavnost je omejena tudi z veljavnostjo uporabljenih zgostitvenih algoritmov za izračun prstnih odtisov. Še pred iztekom veljavnosti je treba ponovno zaščititi vsebine z dodatnim časovnim žigom. Časovni žig ne bi bil več veljaven tudi v primeru, ko bi TSA preklical svoje potrdilo, na primer zaradi zlorabe. Takšnim nepredvidljivim težavam se je mogoče izogniti s sočasno uporabo časovnih žigov različnih izdajateljev oziroma z ustvarjanjem redundantnih evidenčnih podatkov. Postopek obnove veljavnosti dokumenta, ki ga ščiti iztekajoči se časovni žig, je lahko zaradi vsega tega zelo kompleksen in predstavlja jedro standarda ERS.

2.6. Vpliv oblike e. dokumentov na zagotavljanje verodostojnosti

ZVDAGA opredeljuje naslednje lastnosti formata dokumenta za dolgoročno hrambo:

- zagotavlja ohranitev izvirne vsebine gradiva,
- je široko priznan, uveljavljen oziroma uporabljan,
- je neposredno uporaben za reprodukcijo vsebine,
- omogoča samodejno pretvorbo zapisa,
- je neodvisen od posamezne programske ali strojne opreme oziroma okolja,
- zagotavlja varno hrambo več kot pet let in omogoča po tem obdobju pretvorbo v novo, takrat določeno obliko zapisa za dolgoročno hrambo,
- temelji na mednarodnem, državnem ali splošno priznanem in praviloma odprtem standardu.

Enotne tehnološke zahteve pa podrobno opredeljujejo ustrezne oblike za posamezno vrsto dokumenta, kar je predstavljeno v tabeli št. 3.

Tabela 3: Seznam ustreznih oblik dokumentov za dolgoročno hrambo

Vrsta dokumenta	**Oblika dokumenta za dolgoročno hrambo**
Tekstovni in mešani dokumenti	ISO Latin -1 8859 -1, PDF/A - ISO 19005 -1, XML - SGML - ISO 8879, ODF - ISO/IEC DIS 26300
Grafični dokumenti	TIFF - v6.0 - ISO 12639, SVG - v1.1 - W3C
Film/Video/Audio	ANSI/SMPTE 268M, MPEG - 2 - ISO/IEC, MPEG - 4 - ISO/IEC 14496
Kompresirani dokumenti	LZW - barvni dokumenti, 13818CCIT group4 - č/bdok.

Vir: Enotne tehnološke zahteve (Arhiv RS, 2006 str. 108).

Zaradi ohranjanja z dokumentom povezanih dokazil, ki temeljijo na kriptografskih metodah, je potrebno ohraniti dokument kot točno določeno zaporedje bitov. Dosedanje izkušnje z razvojem strojne in programske opreme kažejo, da je povsem verjetno, da določena računalniška oprema lahko preprosto preneha obstajati.

Sprejeti sta dve načelno različni rešitvi: ohranjanje uporabljene tehnologije za interpretacijo arhiviranega gradiva (vključno s tehnologijami za preverjanje verodostojnosti), bodisi neposredno bodisi z razvojem emulacij ali pa pretvarjanje formata gradiva in ohranjanje veljavnosti povezanih dokazil o verodostojnosti. Za slednje je na voljo nekaj rešitev. Podpisnik bi lahko gradivo ponovno podpisal, kar je sicer zelo neverjetno; morda podpisnik ne bo več dosegljiv, morda pa tudi ne bo mogel presoditi ustreznosti novega formata. Bolj verjetna je možnost, da verodostojen arhiv spremeni format, podpiše prejšnja dokazila in ustvari nova. Lynch (Lynch, 1999, str. 4) predlaga opredelitev kanoničnih formatov za besedila in ustvarjanje dokazil o verodostojnosti nad temi formati, hkrati pa ugotavlja, da en sam kanoničen format za zapis arhivskega gradiva ne bi rešil vseh potreb po ohranitvi informacije; v nekaterih primerih je vsebina slika ali glasba, v drugih je spet pomemben slog oblikovanja.

Izbira formata je torej pomemben dejavnik pri zagotavljanju dolgoročne verodostojnosti elektronskih dokumentov, saj sta si zahteva po ohranjanju berljivosti in zahteva po ohranjanju podpisa lahko konfliktni.

2.7. Alternativni pristopi ERS

Standard napredni XML podpis XAdES (angl. XML Advanced signature) opredeljuje pravila in strukture, ki nadgrajujejo standard XMLDsig tako, da ustreza zahtevam za varen digitalni podpis iz Direktive 1999/93/EC, podobno kot standard IETF za formate elektronskih podpisov RFC 3126 (ETSI, 2002). S temi dopolnitvami lahko digitalen podpis nosi dovolj informacij (sezname preklicanih potrdil, korenska digitalna potrdila, časovne žige in njihova podaljšanja, validacijska potrdila ipd.), da lahko dokažemo, da je bil podpis veljaven v preteklosti. XAdES opredeljuje šest profilov (ETSI, 2002, str. 13), ki se razlikujejo glede na zagotovljeno stopnjo varnosti, vsak naslednji pa vključuje prejšnjega:

- XAdES − osnova je XMLDsig podpis in dodane strukture za podpisane in nepodpisane atribute,
- XAdES-T − dodan je časovni žig za podpis,
- XAdES-C − dodane so reference na dopolnilne podatke za dokazovanje veljavnosti podpisa (na dig. potrdila, sezname preklicanih potrdil ipd.),
- XAdES-X − dodani so časovni žigi na reference iz prejšnjega profila,
- XAdES-X-L − dodana so dejanska digitalna potrdila in seznami preklicanih potrdil tako, da je omogočeno preverjanje tudi, če referenčni viri niso dostopni,
- XAdES-A − dodani periodični časovni žigi, ki ohranjujejo veljavnost podatkov, tudi ob spremembah veljavnosti kriptografskih algoritmov.

Za celovito rešitev verodostojne dolgoročne hrambe standard XAdES ne zadostuje iz več razlogov. Načeloma želimo hraniti poljubne dokumente, na primer tudi nepodpisane. Za ta primer bi sicer lahko TAS uporabljal strukture XAdES standarda in bi vanje vključeval prejete podatke ter jih podpisoval, vendar bi nastopile težave, kadar bi sprejeli dokument, podpisan po kakšnem drugem standardu, kot je XMLDsig. Za potrebe delovanja TAS, ki mora naenkrat časovno podaljšati vse dokumente, bi to v skladu z XAdES pomenilo pridobivanje časovnega žiga za vsak dokument posebej.

Zaradi opisanega so pri IETF nadaljevali delo za razvoj novega standarda za ustvarjanje in zapis dokazil o verodostojnosti poljubnega dokumenta s tem, da so tako strukture kot tudi procesi prilagojeni za masovno obdelavo dokumentov. Nastal je standard ERS.

3. Standard ERS − Sintaksa za evidenco dokazil

Standard ERS, definiran na strani št. 2, opredeljuje strukturo in postopke ustvarjanja, ohranjanja in preverjanja zapisov z dokazili o zagotavljanju dolgoročne verodostojnosti digitalnega gradiva, krajše ga imenujem v nadaljevanju evidenčni zapis. Evidenčni zapis tvorijo podatki, ki dokazujejo, da je arhivski objekt v določenem trenutku obstajal in se od takrat ni spreminjal. Pomembna motivacija za standard ERS je možnost, da lahko zaradi uporabe standardnih postopkov in podatkovnih struktur neodvisna entiteta preveri verodostojnost dokumenta; pa tudi uporabnik, če želi, lahko zamenja ponudnika storitve TAS ter ohrani dokazila.

Osnovna tehnologija, ki se uporablja za ustvarjanje evidenčnega zapisa, je digitalni časovni žig. Za dokaz, da je digitalno gradivo obstajalo ob določenem času, zadošča veljaven digitalen časovni žig, ki ščiti: dokumentarno gradivo, opisne podatke in pripadajoča dokazila o avtentičnost dokumentarnega gradiva. V praksi so slednje najpogosteje digitalni podpisi z verigami potrdil, uporabljenih v digitalnih podpisih in v času preverjanja veljavni seznami preklicanih potrdil. Takšen časovni žig se v skladu s sintakso ERS imenuje arhivski časovni žig. Evidenčni zapis vsebuje informacije o arhivskih časovnih žigih, njihovem vrstnem redu in podatke za verifikacijo arhiviranih elektronskih dokumentov.

3.1. Ustvarjanje evidenčnega zapisa

Evidenčni zapis je lahko ločen od arhiviranega digitalnega gradiva ali pa je lahko tudi njegov sestavni del. Za povezavo med arhiviranim digitalnim gradivom in njegovim evidenčnim zapisom skrbi zunanji sistem, na primer sistem za upravljanje z dokumenti. Verodostojnost povezave zagotavljajo prstni odtisi dokumentov, ki so časovno žigosani. Arhivirano digitalno gradivo imenujemo v kontekstu standarda ERS arhiviran objekt, ki je lahko sestavljen iz enega ali več elektronskih dokumentov. Temeljni element evidenčnega zapisa je arhivski časovni žig (angl. Archive Time-Stamp), v nadaljevanju ATS.

3.1.1 Arhivski časovni žig (ATS)

ATS tvorijo:

- digitalni časovni žig,
- identifikator algoritma, s katerim so bili izračunani prstni odtisi v postopku ustvarjanja ATS,

- dodatni varnostni atributi (na primer digitalna potrdila, seznami preklicanih potrdil za časovni žig in podobno),
- seznami prstnih odtisov, v kolikor smo z enim časovnim žigom povezali več digitalnih vsebin (na primer pri skupinski obdelavi, kot je opisano v nadaljevanju).

ATS ščiti tiste digitalne vsebine, za katere lahko dokažemo, da so na enoličen način povezane z vrednostjo, ki jo je podpisal overitelj časovnega žiga. Overitelj časovnega žiga bi lahko podpisal celotne digitalne vsebine, vendar zaradi večje učinkovitosti podpiše samo prstni odtis dokumenta oziroma enolično ustvarjeno vrednost iz več prstnih odtisov, kadar se digitalna vsebine nahajajo v več datotekah.

Overitelj časovnih žigov (TSA strežnik) prstnemu odtisu dokumenta dopiše čas in vse skupaj podpiše s svojim zasebnim ključem - to je časovni žig. S tem je dokazano, da je digitalna vsebina obstajala pred časom, navedenim v časovnem žigu, poleg tega pa lahko preverimo, da se od časa žigosanja ni spremenila. Naredimo ponovni povzetek digitalne vsebine, ki se mora ujemati s tistim povzetkom, ki je del časovnega žiga.

3.1.2 Struktura evidenčnega zapisa

Evidenčni zapis sestavlja zaporedje verig arhivskih časovnih žigov (glej primer št. 17):

- Začetni časovni žig (na skici ima oznako ATS1[1]) žigosa prstni odtis arhivskega objekta, ki ga lahko tvori ena ali več datotek: na primer dokument, meta podatki in različni komplementarni podatki, potrebni za dodatno preverjanje verodostojnosti dokumenta, kot so seznami preklicanih potrdil ali digitalna potrdila podpisnikov dokumenta.
- Znotraj posamezne verige vsak naslednji arhivski časovni žig žigosa prstni odtis predhodnega. Vsi ATS, razen prvega v verigi, nastanejo v postopku podaljševanja veljavnosti predhodnega ATS pred iztekom veljavnosti digitalnega potrdila overitelja, ki je ustvaril časovni žig. V sklopu iste verige je vedno uporabljen enak algoritem za izračun prstnih odtisov.
- Vsak prvi ATS v verigi, razen začetnega ATS, žigosa vrednost, izračunano iz: prstnega odtisa arhiviranega podatkovnega objekta in prstnega odtisa celotnega, do takrat ustvarjenega evidenčnega zapisa. Pri tem je uporabljen močnejši zgostitveni algoritem kot predhodno. Prvi ATS v verigi,

razen začetnega, nastanejo v postopku podaljševanja v primeru nevarnosti, da bi uporabljeni zgostitveni algoritmi postali prešibki.

Primer 17: Skica strukture evidenčnega zapisa:

Zaporedje verig arhivskih časovnih žigov::=

Veriga 1 (sha-1)::= $ATS1^1 \rightarrow ATS2^1 \rightarrow \ldots \rightarrow ATSi^1$

Veriga 2 (sha-256)::= $ATS1^2 \rightarrow \ldots \rightarrow ATSj^2$

Veriga 3 (nov H)::= $ATS1^3 \rightarrow \ldots \rightarrow ATSk^3$

…

Veriga n (najnovejši H)::= $ATS1^n \rightarrow \ldots \rightarrow \mathbf{ATSm^n}$

ASN.1 modul za ERS

Standard ERS opredeljuje ASN.1 shemo v dveh različicah, in sicer za sintakso iz leta 1988 in sintakso iz leta 1997. Za potrebe priprave različice ERS standarda za XML zapis zadostuje analiza ASN.1 sintakse iz leta 1997; ustrezen modul je predstavljen v 2. prilogi tega dela.

Poenostavljen prikaz ASN.1 modula za strukturo evidenčnega zapisa:

```
EvidenceRecord {
   version,
   digestAlgorithms,
   cryptoInfos               OPTIONAL,
   encryptionInfo            OPTIONAL,
   archiveTimeStampSequence  {
         ArchiveTimeStampChain {
                  ArchiveTimeStamp {
                     digestAlgorithm   OPTIONAL,
                     attributes        OPTIONAL,
                     reducedHashtree   OPTIONAL,
                     timeStamp }

         }
   }
}
```

3.1.3 Proces ustvarjanja evidenčnega zapisa

Proces ustvarjanja evidenčnega zapisa opišemo z naslednjimi koraki:

1. zberemo datoteke arhivskega objekta,
2. ustvarimo začetni ATS,
3. po potrebi podaljšujemo ATS.

Za začetni ATS zberemo vse datoteke arhivskega objekta in z izbranim zgostitvenim algoritmom izračunamo prstni odtis za vsako datoteko. Kadar ima

arhivski objekt več kot eno datoteko, izračunamo skupni prstni odtis tako, da sestavimo prstne odtise posameznih datotek, urejene naraščajoče po binarnem vrstnem redu in iz dobljenega niza izračunamo prstni odtis. Na opisan postopek se v nadaljevanju sklicujem kot **postopek lepljenja**. Skupni prstni odtis bomo časovno žigosali, v ATS pa dodali seznam teh vhodnih prstnih odtisov v strukturo, ki se imenuje 'Delno drevo prstnih odtisov'. V primeru št. 18 je prikazana struktura ATS za arhivski objekt, ki ima več datotek. Kadar ima arhivski objekt en sam dokument, potem časovno žigosamo njegov prstni odtis in v tem primeru lahko drevo delnih prstnih odtisov izpustimo.

Primer 18: Drevo prstnih odtisov v začetnem arhivskem časovnem žigu

Prikazan je arhivski objekt, ki ima več datotek (d1,d2,d3), ATS pa smo ustvarili v individualnem procesu

```
ArchiveObject := {d1, d2, d3}
ATS1:= {
      DigestAlgorithm:= {…identifikator algoritma, npr. SHA1}
      TimeStamp:= {…digitalni časovni žig podpisuje skupni prstni odtis
                  iz zlepljenega niza binarno urejenih posameznih
                  prstnih odtisov iz vhodnih dokumentov: h3+h1+h2 }
      ReducedHashTree:= {
            {h1, h2, h3}
      }
}
```

Pri preverjanju najprej potrdimo, da je časovno žigosan prstni odtis ustvarjen natanko iz vrednosti, ki so del strukture delnega drevesa prstnih odtisov, nato pa ponovno preverimo povzetke dokumentov, da se ujemajo s temi iz delnega drevesa prstnih odtisov. V kolikor delnega drevesa prstnih odtisov ni, potem mora imeti arhivski objekt samo en dokument, ob tem pa se morata ujemati časovno žigosana vrednost in prstni odtis dokumenta.

3.1.4 Skupinska obdelava arhivskih objektov

Storitev časovnega žigosanja je časovno zahtevna in povezana s stroški (povprečna cena slovenskih ponudnikov leta 2007 je bila 0,02 EUR na časovni žig). Spodaj opisan postopek omogoča časovno žigosanje skupine arhiviranih podatkovnih objektov z enim samim časovnim žigom, kar ni zgolj cenejše, je predvsem časovno učinkovitejše.

Za časovno žigosanje skupine arhivskih objektov hkrati potrebujemo niz bitov, za katerega bomo lahko nedvoumno dokazali, da je nastal natanko iz vseh arhivskih objektov; in sicer tako, da ob času preverjanja posameznega arhivskega objekta ne bo potrebno preverjanje celotne skupine.

Za ustvarjanje takšnega niza bitov lahko uporabimo Merklejevo drevo (Merkle, 1980, str. 122). Drevo je zgrajeno tako, da ima v listih prstne odtise posameznih arhivskih objektov, v vozlih pa s postopkom lepljenja izračunane prstne odtise iz potomcev vozla. Časovni žig pridobimo za vrednost v korenu drevesa.

Za dokazilo verodostojnosti posameznega podatkovnega objekta ne potrebujemo celotnega drevesa, temveč le poddrevo podatkov (angl. Reduced Hash Tree), ki je neposredno povezano z njim. Pri tem ostanejo dokazila za posamezen podatkovni objekt veljavna, tudi če kakšnega iz te skupine izbrišemo iz arhiva. Ta postopek uporabimo, kadar izvajamo skupinsko obdelavo in želimo ustvariti dokazila o verodostojnosti za več podatkovnih objektov z uporabo samo enega časovnega žiga.

Ustvarjanje Merklejevega drevesa prstnih odtisov

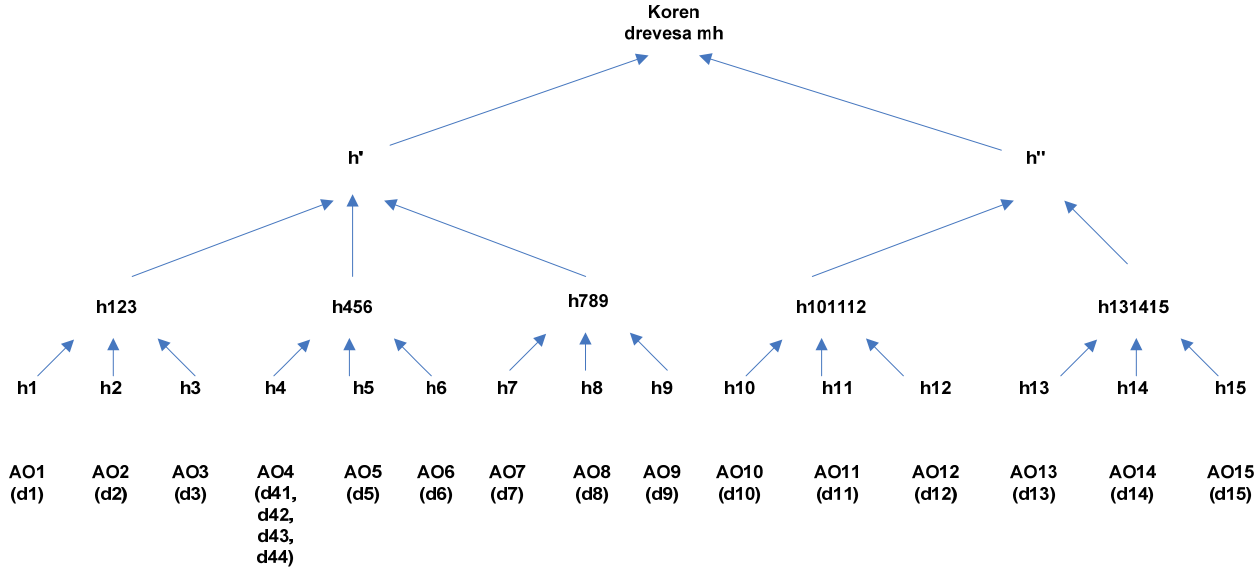
Algoritem za ustvarjanje drevesa prstnih odtisov poteka ob uporabi petih korakov:

1. korak: Zberemo arhivske objekte in za vsak arhivski objekt vse dokumente d_x, ki ga sestavljajo.
2. korak: Z izbranim zgostitvenim algoritmom H izračunamo prstni odtis za vsak dokument ($h_{x=} H(d_x)=$).
3. korak: Za vsak arhivski objekt, ki ima več dokumentov, izračunamo skupni prstni odtis arhivskega objekta z metodo lepljenja $H(ao_i)=H(d_1 + d_2+ \ldots + d_k)$; k je število dokumentov. Prstni odtisi arhivskih objektov so listi Merklejevega drevesa. V nadaljevanju zlivamo liste v poddrevesa tako dolgo, dokler ne sestavimo celotnega drevesa. Liste damo na seznam.
4. korak: Če je na seznamu več kot en prstni odtis, jih potem po vrsti dajemo v skupine in za vsako skupino s postopkom lepljenja izračunamo prstni odtis. V tem koraku ustvarjene prstne odtise damo na nov seznam.
5. korak: Ponavljamo 4. korak, dokler na seznamu ni samo en prstni odtis - koren Merklejevega drevesa.

Na sliki št. 11 je prikazan primer Merklejevega drevesa za skupino petnajstih arhivskih objektov, od katerih ima četrti štiri dokumente, ostali pa po enega. V 4. koraku smo v primeru na sliki št. 11 ustvarjali skupine s tremi elementi. Načeloma so skupine lahko poljubnih velikosti in jih lahko tudi razširimo s slepimi vrednostmi, da dobimo polno n-narno drevo. Najbolj učinkovito glede na velikost delnega poddrevesa za posamezen arhivski objekt je dvojiško drevo.

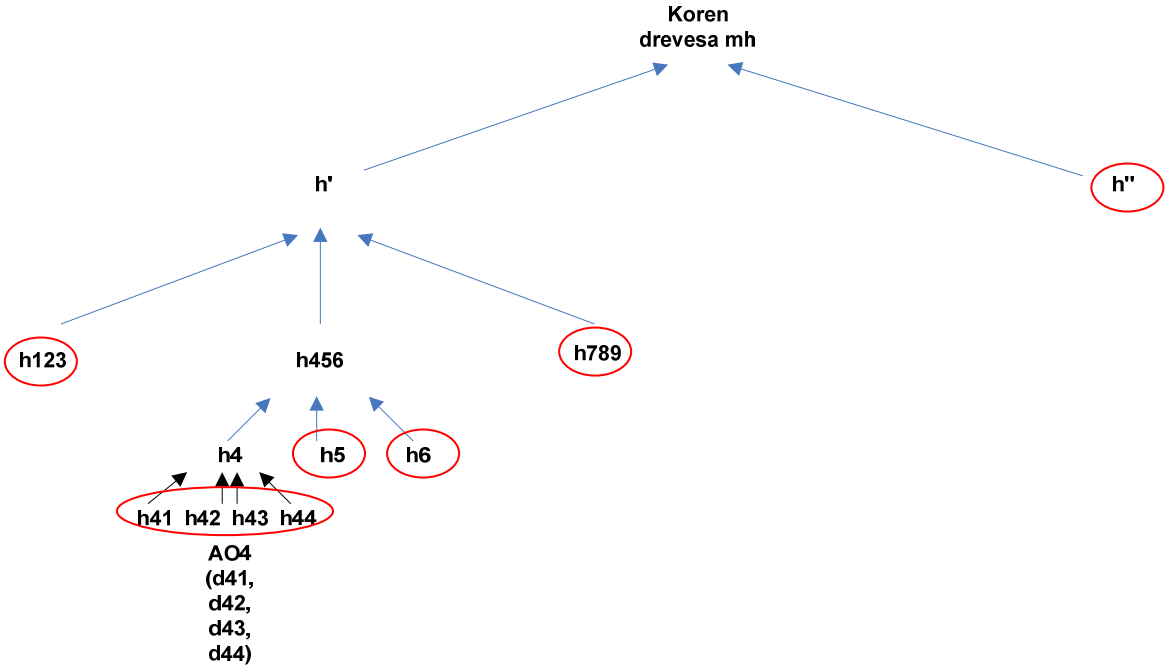
Slika 11: Merklejevo drevo

Primer je za 15 arhivskih objektov, pri čemer je 4. arhivski objekt sestavljen iz štirih dokumentov.



Primer 19: Delno drevo prstnih odtisov za primer iz zgornje slike za arhivski objekt AO4.

Delno drevo prstnih odtisov:= { {h41,h42,h43,h44}, {h5,h6}, {h123,h789}, {h''} }



Ustvarjanje delnega M. drevesa za posamezen arhivski objekt

Algoritem za ustvarjanje delnega drevesa za posamezne arhivski objekt poteka ob uporabi štirih korakov:

1. korak: Ustvarimo prvi seznam prstnih odtisov: če ima arhivski objekt več dokumentov, potem damo na prvi seznam prstne odtise posameznih dokumentov, sicer je v njem samo prstni odtis tistega enega dokumenta.
2. korak: Za izbran arhivski objekt, ki ga zastopa kot prstni odtis list v M. drevesu, poiščemo vse brate tega lista (otroke njegovega starša) in jih damo na dodatnem seznam.
3. korak: Premaknemo se na vozel starša. Na dodaten seznam dodamo vse brate.
4. korak: 3 ponavljamo, dokler ne dosežemo korena drevesa.

V primeru št. 19 je ilustriran ta postopek. V delno drevo shranimo samo vrednosti, ki jih ne moremo izračunati, na primer vrednosti h456 nam ni potrebno hraniti, saj jo lahko izračunamo iz h4, h5 in h6 ali pa vrednosti korena drevesa. V primeru so vrednosti, ki jih shranimo v delno drevo prstnih odtisov, obkrožene z rdečo.

Izračun vrednosti korena M. drevesa iz delnega drevesa

Iz delnega drevesa z naslednjimi korak izračunamo korenski prstni odtis (ta, ki je digitalno časovno žigosan):

1. korak: Vzamemo prvo zaporedje iz drevesa delnih prstnih odtisov. Binarno naraščajoče uredimo prstne odtise, jih zlepimo in izračunamo prstni odtis.
2. korak: Dobljen prstni odtis dodamo na naslednji seznam. Binarno naraščajoče uredimo prstne odtise, jih zlepimo in izračunamo prstni odtis.
3. korak: Ponavljamo postopek v drugem koraku, dokler ne obdelamo vseh seznamov. Zadnji prstni odtis, ki smo ga izračunali, je vrednost korena M. drevesa.

3.2. Podaljševanje veljavnosti evidenčnega zapisa

Evidenčne zapise je potrebno podaljševati, ker imajo časovni žigi (angl. Time-Stamp Renewal) omejeno veljavnost zaradi več razlogov:

1) Digitalno potrdilo zasebnega ključa, s katerim se časovno žigosa, ima tako kot vsa ostala digitalna potrdila omejen rok veljavnosti (na primer: slovenski overitelji izdajo potrdilo za največ 5 let. Vir: 32. člen Uredbe o pogojih za elektronsko poslovanje).
2) Digitalno potrdilo strežnika je lahko pred iztekom veljavnosti preklicano.

3) Kriptografski algoritmi, s katerim je bil narejen časovni žig, lahko postanejo prešibki.
4) Algoritem, s katerim je bil izračunan prstni odtis dokumenta, ni več varen (na primer tehnologija omogoči, da lahko ustvarimo poljuben dokument, katerega prstni odtis bo enak žigosanemu prstnemu odtisu).

3.2.1 Enostavno podaljševanje

Preden postanejo kriptografski algoritmi, ki so uporabljeni v časovnem žigu šibki, in/ali preden poteče veljavnost digitalnemu potrdilu, s katerim preverjamo časovni žig, je potrebno podaljšati časovni žig. V tem primeru zadostuje, da časovno žigosamo predhodni arhivski časovni žig, torej izračunamo prstni odtis predhodnega arhivskega časovnega žiga in ga časovno žigosamo. V primeru št. 20 je prikazana strukture evidenčnega zapisa po prvem enostavnem podaljšanju. Ustvarjen ATS v postopku enostavnega podaljševanja dodamo v obstoječo verigo.

Primer 20: Struktura evidenčnega zapisa po prvem enostavnem podaljšanju

Zaporedje verig arhivskih časovnih žigov::=

Veriga 1 (sha-1)::= $ATS1^1 \rightarrow ATS2^1$

3.2.2 Kompleksno podaljševanje

V kolikor se napove zmanjšanje varnosti algoritma, s katerim je bil izračunan prstni odtis dokumenta, je potrebno, preden postane ta šibek, podaljšanje časovnega žiga izvesti na bolj kompleksen način. Izberemo močnejši zgostitveni algoritem. Nato izračunamo prstne odtise z izbranim močnejšim algoritmom iz vseh dokumentov arhivskega objekta in iz vseh predhodnih arhivskih časovnih žigov (in pripadajočih podatkov) oziroma iz celotnega, do takrat ustvarjenega evidenčnega zapisa. Nazadnje iz vseh zbranih prstnih odtisov z metodo lepljenja in z uporabo izbranega močnejšega algoritma izračunamo skupni prstni odtis. Tega časovno žigosamo. V primeru št. 21 je prikazana struktura edvidenčnega zapisa po prvem kompleksnem podaljšanju. Za ustvarjen ATS v postopku kompleksnega podaljševanja ustvarimo novo verigo in na prvo mesto damo ustvarjen ATS.

Primer 21: Struktura evidenčnega zapisa po prvem kompleksnem podaljšanju

Zaporedje verig arhivskih časovnih žigov::=

Veriga 1 (sha-1)::= $ATS1^1 \rightarrow ATS2^1$

Veriga 2 (sha-256)::= $ATS1^2$

Bolj podrobno je postopek kompleksnega podaljševanja (kadar ima arhivski objekt samo en elektronski dokument) sestavljen iz naslednjih korakov:

- za vsak arhivski objekt izračunamo nov prstni odtis,
- izračunamo prstni odtis iz vseh preteklih verig arhivskih časovnih žigov,
- oba prstna odtisa združimo in izračunamo nov prstni odtis,
- tega damo v prvi seznam drevesa prstnih odtisov,
- časovno ga žigosamo.

Kadar imamo več datotek v arhiviranem podatkovnem objektu, se postopek razlikuje v naslednjih korakih:

- za vsako datoteko združimo oba prstna odtisa (kot opisano zgoraj) in izračunamo nov prstni odtis,
- te prstne odtise damo v prvi seznam drevesa prstnih odtisov,
- iz njih izračunamo prstni odtis, ki ga žigosamo.

3.2.3 Proženje podaljševanja

Proženje postopkov enostavnega podaljševanja lahko sistem za zagotavljanje dolgoročne verodostojnosti podatkov izvaja samodejno. Za spremljanje varnosti in zanesljivosti uporabljenih algoritmov je odgovoren ponudnik storitve dolgoročnega arhiviranja ali pa lastnik podatkov. V zvezi s tem mora vzdrževalec sistema za dokazila zagotoviti, da lahko nek zunanji sistem proži postopke kompleksnega podaljševanja.

Tabela 4: Priporočene zgostitvene funkcije

Zgostitvena funkcija	2008	2010	2015	2025 let (spekulativno)
sha1	uporabna	neznano	neuporabna	neuporabna
ripemd160	uporabna	uporabna	neuporabna	neuporabna
sha224	uporabna	uporabna	uporabna	neznano
sha256	uporabna	uporabna	uporabna	neznano
sha384	uporabna	uporabna	uporabna	uporabna
sha512	uporabna	uporabna	uporabna	uporabna
Whirlpool	uporabna	uporabna	uporabna	neznano

Vir: ETSI: Algorithms and Parameters for Secure Electronic Signatures, TS 102 176-1 , str. 44

V skladu z Evropskimi direktivami priporočilo organizacije ETSI usmerja k uporabi zgostitvenih funkcij, pri katerih je izračunan prstni odtis daljši kot 160 bitov, kljub temu pa še ne izključuje uporabe SHA-1 in RIPMED-160 algoritmov. Za zelo

dolgoročno veljavne podpise priporoča uporabo algoritma SHA-512, sicer pa algoritma SHA-256 (ETSI, 2007). Za ameriške vladne organizacije pa je merodajno poročilo FIPS 180-2 o varnih zgostitvenih algoritmih; v njem so trenutno odobreni naslednji algoritmi: SHA-1, SHA-224, SHA-256, SHA-384 in SHA-512 (NIST). Oceno o primernosti rabe naštetih algoritmov kaže tabela št. 4.

Navedene organizacije, ki se ukvarjajo z analizo in ovrednotenjem dolgoročne zanesljivosti kriptografskih algoritmov, objavljajo izsledke v svojih poročilih, ki niso v obliki, primerni za strojno obdelavo. V sklopu IETF nastaja predlog za standard, ki bo opredeljeval strukturo za zapis politike o primernosti uporabe kriptografskih algoritmov, s kratico **DSSC** (angl. Data Structure for Security Suitabilities of Cryptogtaphic Algorithms). Predlagana struktura, ki je prikazana v primeru št. 22, bo omogočala, da lahko na avtomatiziran način preverimo, ali je določen algoritem trenutno veljaven, ali je bil veljaven na določen dan v preteklosti, do kdaj naj bi bil veljaven in kdaj je prenehal veljati.

Primer 22: Primer zapisa o primernih kripotgrafskih algoritmih po strukturi DSSC

```
<SecuritySuitabilityPolicy xmlns="http://www.sit.fraunhofer.de/dssc"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
     <PolicyName>
       <name>Evaluation of suitable signature algorithms 2008</Name>
     </PolicyName>
     <Publisher>
       <Name>Federal Network Agency</Name>
     </Publisher>
     <PolicyIssueDate>2007-12-17T00:00:00</PolicyIssueDate>
     <Usage>Qualified electronic signatures</Usage>
     <Algorithm>
       <AlgorithmIdentifier>
         <Name>SHA-1</Name>
         <ObjectIdentifier>1.3.14.3.2.26</ObjectIdentifier>
       </AlgorithmIdentifier>
       <Validity>
         <End>2008-06-31</End>
       </Validity>
     </Algorithm>
</SecuritySuitabilityPolicy>
```

Vir: Kunz et al: DSSC, draft-ietf-ltans-dssc-02.txt, str. 21

Opomba: Za ilustracijo je naveden samo eden algoritem. Celotna politika vsebuje ponovitve elementov <Algorithm> za vse znane algoritme in njihove kombinacije.

Uporaba takšne strukture bo omogočila, da lahko sistem na osnovi spremembe politike o veljavnosti algoritmov sproži avtomatsko kompleksno podaljševanje evidenčni zapisov. Pri tem je seveda treba upoštevati dodatne mehanizme zaupanja in varnosti glede tega, da je sama politika podpisana s strani zaupanja vredne organizacije in s pravili na organizacijski ravni glede tega, čigavo politiko veljavnosti upoštevamo.

3.3. Preverjanje evidenčnega zapisa

Proces preverjanja veljavnosti evidenčnega zapisa opišemo z naslednjimi koraki:

1) izberemo arhiviran podatkovni objekt,
2) v kolikor je uporabljeno polje za enkripcijo, preverimo, ali se prstni odtis ponovno kriptiranega podatkovnega objekta, ki ga želimo preveriti, ujema s shranjenim prstnim odtisom
3) preverimo zaporedje arhivskih časovnih žigov.

Algoritem za preverjanje zaporedja arhivskih časovnih žigov poteka v treh korakih:

1) prvi ATS v prvi verigi mora vsebovati prstni odtis podatkovnega objekta,
2) za vsako verigo je potrebno preveriti, da:
    a. prvi ATS vsebuje prstni odtis izračunan iz celotnega predhodnega zaporedja arhivskih časovnih žigov in podatkovnega objekta,
    b. da vsak naslednji ATS vsebuje prstni odtis predhodnega ATS,
    c. je vsak ATS v verigi veljaven v času nastanka njegovega naslednika,
    d. da je v vseh ATS uporabljen enak zgostitveni algoritem in je ta bil varen ob času nastanka naslednje verige.
3) zadnji ATS mora biti veljaven v času preverjanja.

3.3.1 Preverjanje veljavnosti ATS v času nastanka njegovega naslednika

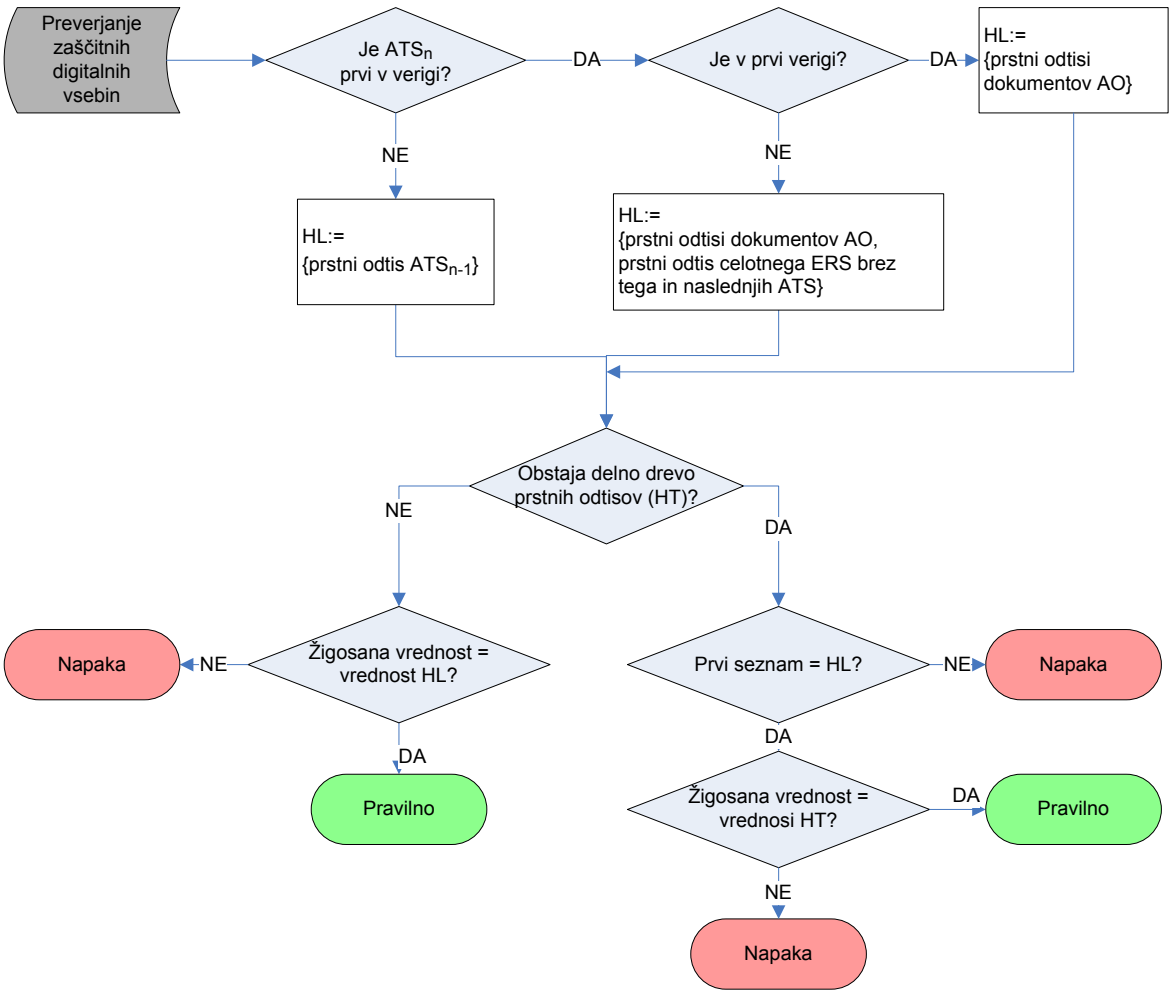
Preverjanje veljavnosti ATS je sestavljeno iz naslednjih sklopov:

a) <u>preverjanje, ali ATS ščiti prave vsebine in ali se le-te niso spremenile</u>

   V tem sklopu je potrebno pokazati, da ATS ščiti vsebine, ki jih mora, glede na to, kje v sklopu verige in zaporedja verig se ATS nahaja. Preverjanje poteka tako, da izračunamo prstni odtis, ki bi ga ATS moral ščititi in ga primerjamo z vrednostjo, ki jo je overitelj časovnega žiga podpisal. V kolikor se vrednosti ujemata, ATS ščiti prave digitalne vsebine in le-te se niso spremenile. Postopek je prikazan na sliki št. 12.

b) <u>preverjanje, ali je digitalni časovni žig kriptografsko pravilen</u> (opisano v poglavju o preverjanju digitalnega časovnega žiga 2.5.2)

Slika 12: Preverjanje veljavnosti arhivskega časovnega žiga



c) <u>preverjanje, ali je TS bil veljaven v času naslednjega ATS</u> izvedemo tako, da preverimo priložena dokazila o veljavnosti digitalnega potrdila overitelja časovnega žiga, najpogosteje je to časovna veljavnost digitalnega potrdila in neprisotnost na seznamu preklicanih potrdil izdajatelja digitalnega potrdila.

V nadaljevanju bo med ključnimi kriteriji za izbiro rešitve za standard ERS v XML zapisu zagotavljanje enakovrednih učinkov, kot jih imajo v tem poglavju predstavljeni postopki za ustvarjanje in ohranjanje dokazil o dolgoročni verodostojnosti:

- ustvarjanje arhviskih časovnih žigov,
- enostavno in kompleksno podaljševanje časovnih žigov ter možnost za samodejno proženje obeh procesv,
- preverjanje veljavnosti,
- skupinska obdelava arhivskih objektov.

4. ERS v XML zapisu

4.1. Ključna izhodišča za prehod na XML strukture

4.1.1 Ponovljivost zaporedja bitov za izračun prstnega odtisa

Za preverjanje veljavnosti evidenčnega zapisa je ključnega pomena **ponovljivosti ustvarjanja zaporedja bitov** vhodnega digitalnega gradiva in predhodnega časovnega žiga, iz katerih se računa prstne odtise. Za zagotavljanje ponovljivosti izračuna prstnega odtisa iz XML dokumenta je treba zmeraj pred izračunom zapisati **XML dokument v kanonični obliki** (podrobnosti podaja poglavje 1.3.4).

Najprej sem raziskala možnost, da bi postopke in podatkovne strukture za XML zapis ERS brez težav pretvarjali iz enega od kodiranj po ASN.1 shemi v enakovreden XML dokument, odvisno od procesnih potreb, na primer glede na to, ali nam bo pomembnejša berljivost od kompaktnosti zapisa.

Usmerimo pozornost na postopek kompleksnega podaljševanja, kot je opredeljen v standardu ERS. V postopku nastopa izračun prstnega odtisa iz vseh preteklih verig arhivskih časovnih žigov. Ta odtis potem zlepimo s prstnim odtisom vhodnih podatkov in tvorimo skupni prstni odtis, ki ga časovno žigosamo. V tem postopku je potrebno kronološko zaporedno zapisati vse verige (in znotraj posamezne verige vse arhivske časovne žige, kot to opredeljuje struktura) in iz njih izračunati prstni odtis. Pri tem morajo biti v skladu z ERS standardom verige DER kodirane, torej ob vsebini verige zapis vsebuje tudi oznake za tipe ASN.1 podatkovnih struktur in dolžine za posamezne elemente. Izračun torej temelji na zaporedju bitov po ASN.1 strukturi in DER kodiranju podatkov. Prav to pa je razlog, da **ne moremo že ustvarjen ASN.1 evidenčni zapis pretvoriti v ekvivalenten XML zapis**, saj brez ne bi mogli izračunati enakih vrednosti prstnih odtisov iz preteklih verig arhivskih časovnih žigov.

Za ponovljivost zaporedja bitov je ob naštetem potrebno ohranjati tudi vrstni red primerkov elementov istega tipa, na primer vrstni red elementov tipa 'arhivski časovni žig'. XML specifikacija ne določa eksplicitno, da je vrstni red primerkov elementov istega tipa določen z vrstnim redom zapisa primerkov v dokumentu. Načeloma orodja za delo z XML privzemajo vrstni red zapisa elementov, kot ga preberejo iz dokumenta.

4.1.2 Uporaba XER kodiranja

Do primera XML zapisa za ERS lahko pridemo tudi s pomočjo uporabe XER[4] kodiranja ASN.1 sheme za ERS.

Toda ERS standard temelji na postopkih, ki so, kot je v prejšnjem podnaslovu predstavljeno, vezani na DER kodiranje. Zato je treba, ob nadomestitvi DER kodiranja s XER kodiranjem, v ERS standardu dodatno opredeliti vsaj še kanonizacijo in ustrezno XSD shemo. Brez XSD sheme ne bi dosegli namena nadgradnje obstoječega standarda, ki je zapis v obliki, ki se jo lahko ustvarja, prikazuje in preverja s pomočjo orodij za delo z XML dokumenti.

V nadaljevanju bomo pokazali, da je mogoče ustvariti XSD shemo, za katero bi bili ustrezni XML dokumenti ekvivalenti kot ASN.1 za ERS in XER kodiranje. Vendar sama opredelitev prevedljive podatkovne strukture še ne zadošča, saj na vsebino, kot je na primer arhivski časovni žig, vplivajo s strukturo povezani postopki ustvarjanja in preverjanja evidenčnih zapisov. V kolikor je mogoče, je potrebno opredeliti take postopke, ki bodo neodvisni od kodiranja podatkovnih struktur.

4.1.3 Priporočila za rabo XML v IETF standardih

V nadaljevanju so povzete ključne točke iz priporočil za rabo XML v IETF standardih (Hollenbeck, 2003, str. 3−12):

- XML mora biti dobro oblikovan;
- priporočena je raba kanoničnih oblik XML dokumentov;
- odsvetuje se raba komentarjev za podajanje navodil za procesiranje podatkov in se priporoča, da komunikacijski protokoli ne upoštevajo komentarjev;
- protokoli naj slonijo na formalnih metodah za preverjanje veljavnosti strukture XML dokumentov;
- odsvetuje se raba prednastavljenih vrednosti atributov;
- za protokole, kjer bi XML nastopil kot ovojnica za velike količine binarnih podatkov, je potrebno razmisliti, ali je XML sploh primeren format, saj so Base64 ali druga kodiranja binarnih nizov v znake potratna; kadar pa je potrebna takšna pretvorba, se svetuje raba Base64 kodiranja;
- pripraviti je potrebno navodila za obravnavo presledkov v XML dokumentu;

[4] XER (angl. Xml Encoding Rules) kodiranje je definirano na strani 21 in ilustrirano v primeru št. 23.

- priporočena je raba XML deklaracije (ki jo specifikacija XML 1.1 tudi zahteva, medtem ko je verzija 1.0 ni), še posebej kadar nista uporabljena sistema kodiranja UTF-8 ali UTF-16;
- XML deklaracija vsebuje verzijo specifikacije XML, s katero je dokument skladen in priporočen sistem kodiranja;

  Primer: `<?xml version="1.0" encoding="UTF-8"?>`
- priporočena je raba sistemov kodiranja UTF-8 in UTF-16 z znakom BOM za vrstni red bitov na začetku (angl. ByteOrderMark);
- protokol naj opredeli mehanizme za morebitne razširitve struktur in kriterije, kako jih prepoznamo ter obravnavamo;
- priporočljiva je raba obstoječih imenskih prostorov (še posebej kadar je protokol iz zbirke IETF standardov), atribut URI pa naj kaže na opis imenskega prostora;

4.1.4 Primerjava med XML in ASN.1

Berljivost proti zgoščenosti zapisa

V osnovi je ASN.1 (ne glede na izbrana pravila za kodiranje) usmerjen na učinkovitost prenosa podatkov - čim bolj zgoščeni podatki, medtem ko je XML usmerjen na učinkovitost razvoja programskih rešitev - za človeka čim bolj berljiv zapis podatkov; na primer: človek lahko s hitrim preletom poišče v XML dokumentu zanj pomembne dele dokumenta. Kadar je najpomembnejši odločitveni kriterij zmogljivost sistema, v smislu hitrosti prenosa podatkov ali omejena količina in podobno, je ASN.1 veliko primernejša izbira. Med zadnjimi pravili za kodiranje so k ASN.1 specifikaciji dodali XER pravila za kodiranje ASN.1 sheme z XML zapisom, za namene, kjer je pomembnejša tekstovna berljivost od zgoščenosti zapisa.

Na svetovnem spletu lahko najdemo nekaj prizadevanj za pripravo specifikacije binarnega formata XML[5] (angl. Binary XML), ki definirajo zgoščen zapis XML dokumenta v binarnem formatu. V okviru organizacije W3C si prizadevajo predvsem za ohranitev kompatibilnosti z XML formatom. Bistvo zamisli je možnost, da se za XML dokument določijo pravila, ki ob poznavanju njegove sheme enoumno določajo zgoščen binaren zapis in obratno, iz zgoščenega binarnega zapisa in s pomočjo sheme, bi radi spet ustvarili XML dokument. Na ta

[5] Več o binarnem formatu XML je na straneh http://www.w3.org/TR/xbc-characterization/.

način bi lahko za potrebe protokolov, kjer je prostorska učinkovitost ključnega pomena, uporabljali zgoščene podatke, ki bi jih še zmeraj lahko za potrebe berljivosti prikazali kot XML.

Razširjenost

Med uporabniki interneta se ASN.1 ni prejel, predvsem zaradi nedostopnosti dokumentacije in posledičnega pomanjkanja prosto dostopnih programskih orodij (OpenEvidence, 2002, str. 34). Drugi razlog so tudi spremembe specifikacij za ASN.1 sheme, katerih rezultat je več ANS.1 jezikov, ki so medsebojno nezamenljivi.

Zahtevnost dela s programskimi orodji

Razlika med obema je tudi v orodjih za kodiranje in dekodiranje. Za ASN.1 specifikacije potrebujemo za učinkovito delo orodja, ki so prilagojena na specifično strukturo in vrsto kodiranja, medtem ko za procesiranje XML dokumentov uporabljamo generične razčlenjevalnike (angl. parsers).

Primerljivost opisljivih podatkovnih struktur

Glede možnosti za specifikacijo podatkovnih struktur sem primerjala ASN.1 sintakso in XSD shemo za XML (predstavljeno v naslednjem poglavju). Obe podpirata drevesne podatkovne strukture z zaporedji ali izbirami elementov na isti ravni, z določanjem tipov podatkov in omejevanjem njihove zaloge vrednosti. Med očitnimi razlikami ugotovimo, da ASN.1 sintaksa ne pozna pojma 'atribut' k elementu drevesa, kot ga jezik XML. V nadaljevanju analiziram možnosti za pretvorbo definicij podatkovnih struktur med ASN.1 in XSD shemo.

4.1.5 Pretvarjanje med XML shemo in ASN.1 shemo

Priporočila za XML kodiranje podatkov za ASN.1 sheme (ITU-T, X.693) omogočajo, da lahko kodiramo ASN.1 podatke kot XML dokumente. Tovrstno kodiranje se s kratico imenuje XER.

Priporočilo Mednarodne telekomunikacijske unije (ITU-T, X.694) opredeljuje pravila za pretvorbo XML sheme v ASN.1 shemo, s katero lahko validiramo XML shemi pripadajočo množico XML dokumentov.

V primeru št. 23 je ilustrirano, kako s XER kodiranjem zapišemo XML dokument na podlagi ASN.1 sheme.

Primer 23: Primer XER kodiranja ASN.1 sheme (ITU-T, X.693, str. 14)

ASN.1 shema za podatkovno strukturo PersonnelRecord:

```
PersonnelRecord ::= [APPLICATION 0] IMPLICIT SET {
      name Name,
      title [0] VisibleString,
      number EmployeeNumber,
      dateOfHire [1] Date,
      nameOfSpouse [2] Name,
      children [3] IMPLICIT
      SEQUENCE OF ChildInformation DEFAULT {} }

ChildInformation ::= SET{
      name Name,
      dateOfBirth [0] Date }

Name ::= [APPLICATION 1] IMPLICIT SEQUENCE
{     givenName VisibleString,
      initial VisibleString,
      familyName VisibleString }

EmployeeNumber ::= [APPLICATION 2] IMPLICIT INTEGER
Date ::= [APPLICATION 3] IMPLICIT VisibleString -- YYYYMMDD
```

Primer XML kodiranja za zgornje podatke:

```
<PersonnelRecord>
      <name>
            <givenName>John</givenName>
            <initial>P</initial>
            <familyName>Smith</familyName>
      </name>
      <title>Director</title>
      <number>51</number>
      <dateOfHire>19710917</dateOfHire>
      <nameOfSpouse>
            <givenName>Mary</givenName>
            <initial>T</initial>
            <familyName>Smith</familyName>
      </nameOfSpouse>
      <children>
            <ChildInformation>
                  <name>
                        <givenName>Ralph</givenName>
                        <initial>T</initial>
                        <familyName>Smith</familyName>
                  </name>
                  <dateOfBirth>19571111</dateOfBirth>
            </ChildInformation>
            <ChildInformation>
                  <name>
                        <givenName>Susan</givenName>
                        <initial>B</initial>
                        <familyName>Jones</familyName>
                  </name>
                  <dateOfBirth>19590717</dateOfBirth>
            </ChildInformation>
      </children>
</PersonnelRecord>
```

XML in ASN.1 nista popolnoma skladna kot jezika za podatkovne strukture; na primer ASN.1 ne pozna pojma, ki bi ustrezal pojmu atribut. V primeru št. 24 shema

opredeljuje, da ima XML element z imenom 'xyz-elem' tudi atribut z imenom 'xyz-attr'. Za kodiranje podatkov v tako XML obliko je ASN.1 shemi potrebno dodaten napotek, in sicer v obravnavanem primeru [ATTRIBUTE]. Napotek določa, da se pri XML kodiranju ASN.1 sheme tip 'xyz-atr' zapiše kot atribut starša.

Primer 24: Primer pretvorbe XSD sheme v ASN.1 shemo [ITUT, X.694, str. 45]

```
<!-- file "http://example.com/xyz/schema.xsd" -->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xyz="http://example.com/xyz"
targetNamespace="http://example.com/xyz">
      <xsd:element name="xyz-elem" type="xsd:string"/>
      <xsd:complexType name="Xyz-type">
            <xsd:attribute name="xyz-attr" type="xsd:boolean"/>
      </xsd:complexType>
</xsd:schema>

XYZ -- The module reference is not standardized
DEFINITIONS AUTOMATIC TAGS ::=
BEGIN
Xyz-elem ::= [NAME AS UNCAPITALIZED] XSD.String
Xyz-type ::= SEQUENCE {
      xyz-attr [ATTRIBUTE] BOOLEAN OPTIONAL
}
ENCODING-CONTROL XER
GLOBAL-DEFAULTS MODIFIED-ENCODINGS
GLOBAL-DEFAULTS CONTROL-NAMESPACE
"http://www.w3.org/2001/XMLSchema-instance"
END
```

Pretvarjanje XML shem v ASN.1 sheme je smiselno, kadar že imamo orodja, ki uporabljajo ASN.1 sheme in kodiranja. Še posebej, kadar je pomembna hitrost prenosa podatkov, ali pa je pasovna širina omejena, na primer pri brezžičnih povezavah. Takrat lahko za potrebe prenosa uporabimo v primerjavi z ustreznim XML zapisom do 100-krat manjše binarne zapise (Chadwick, 2004, str. 36), za na primer prikazovanje v brskalniku pa XML kodiranje.

S prevajanjem shem v drugo smer, torej iz ASN.1 sheme v XML, so se ukvarjali v okviru raziskovalnega laboratorija podjetja IBM v Tokiju (Hiroshi, 2000) in v podjetju Objective Systems, ki na svojih spletnih straneh ponuja javno dostopni prevajalnik iz ASN.1 sheme v XML shemo (Objective Systems, 2008b). Ime modula v ASN.1 se preslika v `xsd:namespace`, ASN.1 stavek IMPORT v `xsd:import`, za ASN.1 ukazni stavek EXPORT ni ustrezne XSD pretvorbe. Za vsak ASN.1 podatkovni tip lahko najdemo ustrezen XSD podatkovni tip (Objective Systems, 2008a, str. 4). V tabeli št. 5 so primeri za nekaj osnovnih podatkovnih tipov.

Tabela 5: Pretvorba med ASN.1 shemo in XML shemo

ASN.1 shema	XSD (XML shema)
TypeName ::= BOOLEAN	<xsd:simpleType name="*TypeName*"> <xsd:restriction base="xsd:boolean"/> </xsd:simpleType>
TypeName ::= INTEGER	<xsd:simpleType name="*TypeName*"> <xsd:restriction base="xsd:integer"/> </xsd:simpleType>
TypeName ::= BIT STRING	<xsd:simpleType name="BitString"> <xsd:restriction base="xsd:token"> <xsd:pattern value="[0-1]{0,}"/> </xsd:restriction> </xsd:simpleType> <xsd:simpleType name="*TypeName*"> <xsd:restriction base="asn1:BitString"/> </xsd:simpleType>

Pretvarjanje podatkovnih struktur

Sestavljeno podatkovno strukturo SEQUENCE v ASN.1 lahko pretvorimo v `xsd:sequence`. V nadaljevanju je primer:

```
ASN.1:
      Primer ::= SEQUENCE {
            prvi tip1,
            drugi tip2,
            ...
      }

XSD:
      <xsd:complexType name="Primer">
            <xsd:sequence>
                  <xsd:element name="prvi" type="tip1"/>
                  <xsd:element name="drugi" type="tip2"/>
            …
            </xsd:sequence>
      </xsd:complexType>
```

Za ASN.1 'SEQUENCE OF' podatkovni tip, ki se uporablja za zaporedje elementov istega podatkovnega tipa elementov, je od podatkovnega tipa tega elementa odvisno, kako ga bomo preslikali. Če gre za enega izmed vgrajenih tipov, bomo za preslikavo v XSD uporabili `xsd:list` (s presledkom ločene vrednosti), v splošnem pa `xsd:sequence`. V nadaljevanju je primer:

```
ASN.1:
TypeName ::= SEQUENCE OF ElementType

XSD:
<xsd:complexType name="TypeName">
      <xsd:sequence>
            <xsd:element name="ElementType" type="ElementType"
            maxOccurs=”unbounded”/>
      </xsd:sequence>
</xsd:complexType>
```

4.2. Predlogi rešitev

4.2.1 Prvi predlog: Uporaba XER kodiranja

Glede na obstoječ standard ERS, ki temelji na ASN.1 strukturi podatkov, je najbolj naraven predlog, da uporabimo XER kodiranje podatkov po obstoječem ASN.1 modulu. To bi omogočilo, da uporabljamo enak standard in popolnoma enako shemo za XML in DER kodiran zapis dokazil. Med formatoma zapisa bi lahko tudi prehajali glede na prednosti enega ali drugega; za večjo berljivost bi uporabili XER kodiranje, za hitrejši prenos pa DER.

Vendar proces podaljševanja veljavnosti dokazil vključuje izračun prstnega odtisa iz zapisa dokazil in potem časovni žig tega prstnega odtisa. Tako je zaradi izračuna prstnega odtisa potrebno standard prilagoditi formatu zapisa dokazil, ravno tako tudi ne bo možno pretvarjanje med DER kodiranimi dokazili in XML kodiranimi dokazili.

Na podlagi zgornje utemeljitve uporabo XER kodiranja zavračam kot predlog za nadaljnjo obravnavo.

4.2.2 Drugi predlog: Analogna pretvorba ASN.1 modula v XSD shemo

Najprej bom nadomestila podatkovne strukture opredeljene z ASN.1 z enakovrednimi XML podatkovnimi strukturami. Potem bom analizirala vpliv XML struktur na postopke izdelave ERS, podaljševanja veljavnosti in preverjanja veljavnosti. V kolikor bo potrebno, bom ustrezno nadgradila predlagane XML strukture.

V standardu ERS nastopajo v modulu ASN.1 sheme (glej prilogo št. 2) naslednji elementi:

- oznaka modula,
- izvoz vseh elementov,

- uvoz iz drugih shem,
- tipi podatkovnih struktur: SEQUENCE, SEQUENCE OF, SEQUENCE SIZE (1..MAX) OF in SET OF,
- podatkovni tipi: OBJECT IDENTIFIER, ANY, OCTET STRING, INTEGER,
- označevalec OPTIONAL,
- procesne oznake [0], [1], [2], EXPLICIT.

Uvoz iz drugih shem sem izvedla tako, da sem deklaracijo za uvoz naštetih elementov iz uvoženih shem (glej tabelo št. 6) nadomestila z definicijami struktur elementov. Kot je opisano v 4.1.5, lahko ASN.1 shemo, sestavljeno iz zgoraj naštetih elementov, pretvorimo v ustrezno XML shemo. Pretvorbo lahko izvedemo ročno ali pa z uporabo katerega od orodij za avtomatsko pretvorbo med ASN.1 in XML.

Tabela 6: Uvoz referenciranih elementov v ASN.1 modulu za ERS

ContentInfo iz PKCS7

`ContentInfo ::= SEQUENCE {`
`  contentType ContentType,`
`  content [0] EXPLICIT ANY DEFINED BY contentType }`
 	
`ContentType ::= OBJECT IDENTIFIER`

AlgorithmIdentifier iz AuthenticationFramework

`AlgorithmIdentifier ::= SEQUENCE {`
`  algorithm OBJECT IDENTIFIER,`
`  parameters ANY DEFINED BY algorithm OPTIONAL }`
`  -- contains a value of the type`
`  -- registered for use with the`
`  -- algorithm object identifier value`

Attribute iz InformationFramework

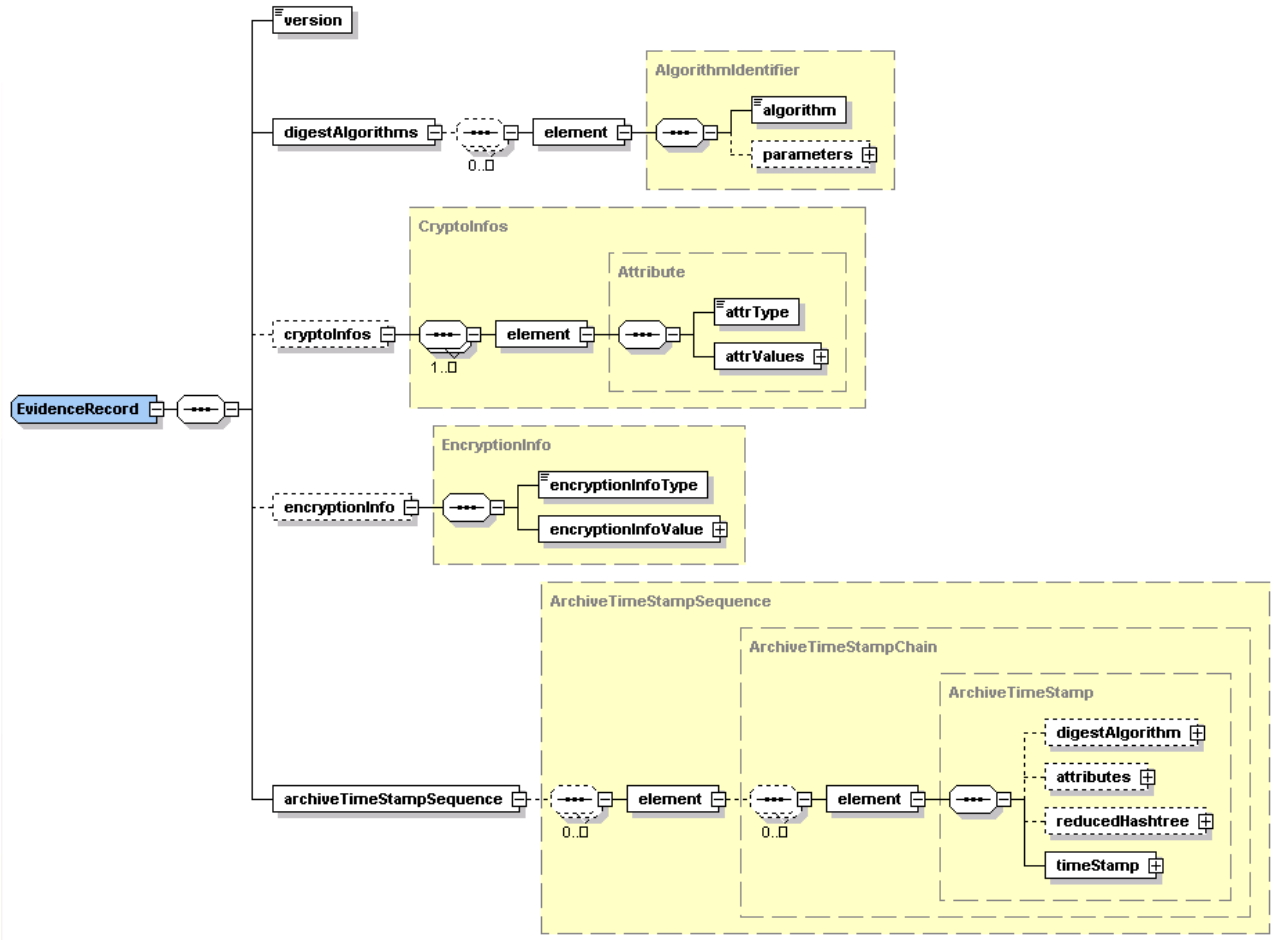
`Attribute ::= SEQUENCE {`
`  attrType OBJECT IDENTIFIER,`
`  attrValues SET OF AttributeValue }`
 	
`AttributeValue ::= ANY`

Viri: RFC 3280 za strukturo ‘AlgortihmIdentifier’, RFC 3852 za strukturi ‘ContentInfo’ in ‘Attribute’. Reference na uvoz elementov v ERS ASN.1 modulu so nadomeščene z neposrednimi opredelitvami iz naštetih virov.

Uporaba orodja za pretvorbo

Prosto dostopno orodje podjetja Objective Systems na spletnem naslovu https://www.obj-sys.com/asn2xsd.php za pretvarjanje ASN.1 modulov v XSD sheme vrne rezultat, ki je skiciran na sliki št. 13 (izvorna XSD kodo je v prilog št. 3). Orodju sem podala kot vhodne podatke definicijo ASN.1 modula v tekstovni obliki iz priloge št. 2; orodje je vrnilo kot rezuktat XSD shemo enakovrednih podatkovnih struktur.

Slika 13: Avtomatizirana pretvorba iz ASN.1 v XSD shemo



Vir: Objective Systems. https://www.obj-sys.com/asn2xsd.php. Na spletni strani vnesemo ASN.1 modul in kot rezultat dobimo prevod v XSD shemo.

Analiza potreb po spremembi podatkovnih struktur in procesnih pravil

Vrstni red istoimenskih elementov v specifikaciji XML ni opredeljen z vrstnim redom pojavitve v zapisu XML dokumenta. Čeprav bodo razčlenjevalniki pri branju načeloma vrstni red ohranili, je potrebno dodati elementu <ArchiveTimeStampChain> atribut 'Order', ki bo vseboval informacijo o vrstnem

redu. Prva veriga bo imela vrednost atributa 'Order' enako 1, vsaka naslednja pa za ena večjo vrednost (2,3,..). Tako bo veljalo, da je veriga z najvišjo vrednostjo atributa 'Order' zadnja veriga. Ravno tako dodamo atribut 'Order' v element <ArchiveTimeStamp>. Prvi ima vrednost atributa 'Order' enako 1, vsak naslednji pa za ena večjo vrednost. Zadnji arhivski žig bo tisti z najvišjo vrednostjo atributa 'Order' v zadnji verigi.

Ker ASN.1 format ne potrebuje kanonizacije, nima predvidenega polja za zapis informacije o uporabljeni kanonizaciji. Zaradi tega moramo dodati polje, v katero bomo shranili informacijo o uporabljeni metodi kanonizacije. Kot predlog strukture za shranjevanje informacije o uporabljeni kanonizaciji podajam enako pomensko strukturo iz standarda XMLDsig:

```
<element name="CanonicalizationMethod"
type="ds:CanonicalizationMethodType"/>
   <complexType name="CanonicalizationMethodType" mixed="true">
     <sequence>
       <any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
       <!-- (0,unbounded) elements from (1,1) namespace -->
     </sequence>
     <attribute name="Algorithm" type="anyURI" use="required"/>
   </complexType>
```

Primer konkretnega elementa s kanonizacijo brez komentarjev (Boyer, 2001):

```
<CanonicalizationMethod
            Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
```

Zaradi narave jezika XML ni potrebno podvajanje informacije o uporabljenih algoritmih za izračun prstnih odtisov tako kot v ASN.1 shemi standarda ERS, kjer se nahajajo na dveh mestih. Tako se umakne ustrezna struktura neposredno iz elementa <EvidenceRecord>, saj lahko z enostavnimi poizvedbami ali pregledom XML dokumenta pridobimo ustrezno informacijo o tem, kateri algoritem je zadnji akutalno uporabljen. Element <digestAlgorithm> damo v strukturo <ArchiveTimeStampChain> in ne več v strukturo <ArchiveTimeStamp>, saj velja za vse arhivske časovne žige znotraj iste verige, da so ustvarjeni z uporabo istega algoritma za izračun prstnih odtisov.

Na sliki št. 14 so razvidne opisane spremembe v podatkovni strukturi evidenčnega zapisa, proces ustvarjanja evidenčnega zapisa pa se od opredelitev v standardu ERS razlikuje v naslednjem:

1. vsebino za začetni arhivski žig ustvarimo na enak način, le da verigo in ATS zapišemo kot ustrezna XML elementa, dopolnjena z atributoma za vrstni red v verigi in v ATS,

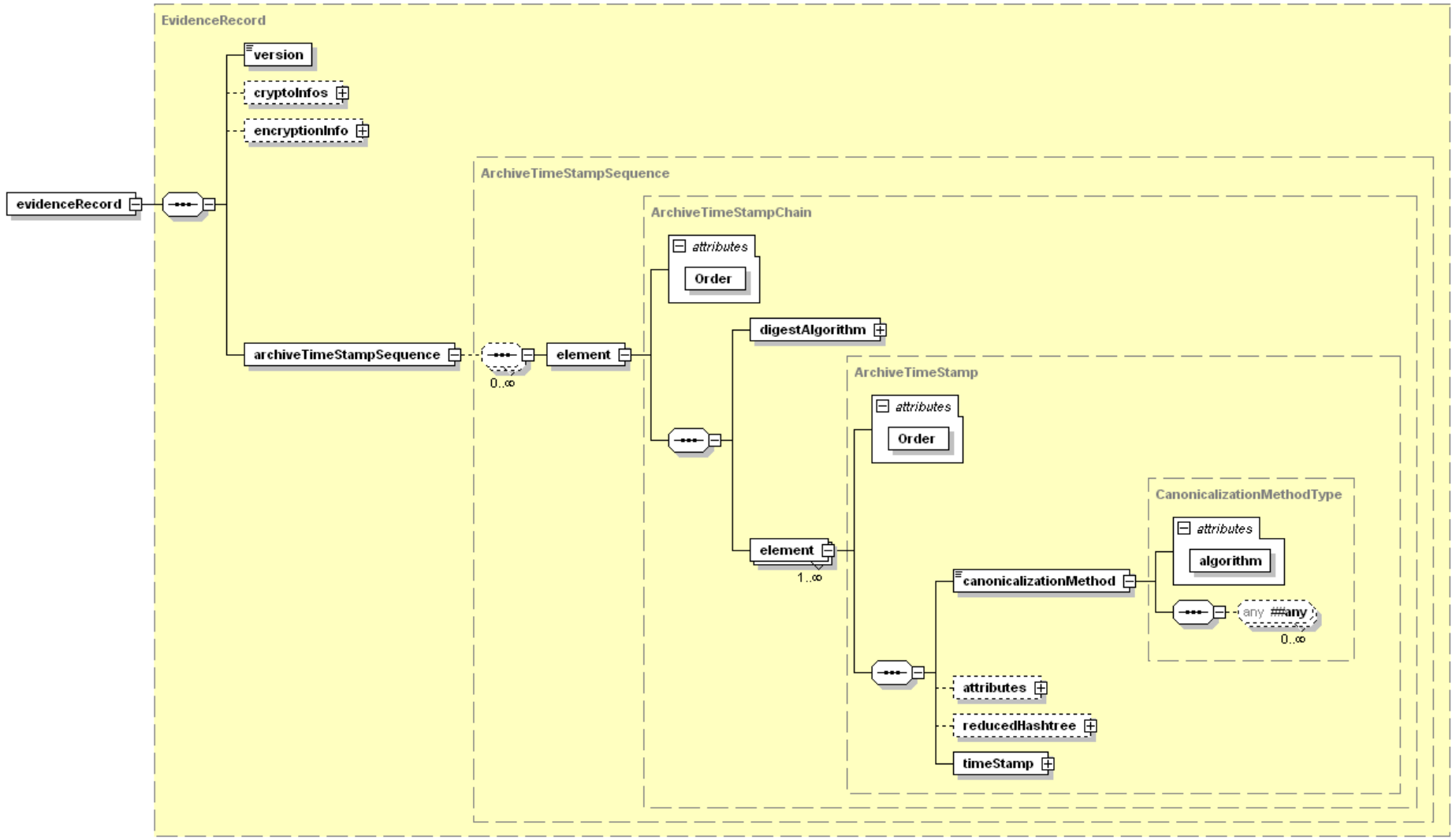
2. pri podaljševanju ustvarjamo prstne odtise XML elementov tako, da jih kanoniziramo in izračunamo prstni odtis z vključno začetno in končno značko elementa.

Pri preverjanju ERS moramo upoštevati uporabljeno kanonizacijo pri izračunu prstnih odtisov iz XML elementov. Zaradi tega se nam spremeni proces preverjanja evidenčnega zapisa pri izračunu prstnega odtisa predhodnega arhivskega časovnega žiga in pri izračunu prstnega odtisa celotnega predhodnega zaporedja arhivskih časovnih žigov.

Izračun prstnega odtisa predhodnega arhivskega časovnega žiga poteka tako, da vzamemo celoten element <ArchiveTimeStamp>, vključno z njegovo začetno in končno značko. Nad izbranim elementom izvedemo kanonizacijo z algoritmom, ki je naveden v njegovem pod elementu <CanonicalizationMethod>. Dobljen binarni niz podamo kot vhod algoritmu za izračun prstnega odtisa, ki je naveden v elementu <digestAlgorithm>.

Pred izračunom prstnega odtisa iz celotnega predhodnega zaporedja arhivskih časovnih žigov iz elementa <ArchiveTimeStampSequence>, je potrebno odstraniti iz tega elementa vse verige z višjim vrstnim redom od verige, kjer se nahaja ATS, za potrebe katerega računamo prstni odtis, in iz te verige tudi vse ATS z višjim vrstnim redom.

Slika 14: XSD shema za drugi predlog rešitve



4.2.3 Tretji predlog: XML kot ovojnica za podatke

V tem predlogu gledamo na evidenčni zapis tako, da razlikujemo med:

- dokazili oziroma podatki, ki jih ščitimo ,
- med strukturo evidenčnega zapisa, v katero umestimo dokazila.

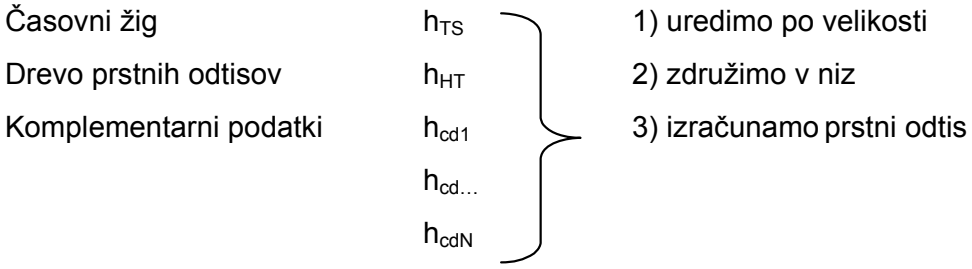
V osnovi uporabimo enako strukturo kot v drugem predlogu. Namesto izračuna prstnega odtisa iz dokazil, vključno s strukturo evidenčnega zapisa, računajmo prstne odtise samo neposredno iz dokazil: podatkovnih objektov, časovnih žigov, dreves prstnih odtisov in atributov, ki se jih uporablja pri ugotavljanju verodostojnosti podatkovnega objekta. Časovno žigosajmo prstni odtis, ki je nedvoumno povezljiv z zbranimi prstnimi odtisi posamičnih dokazil. Na primer: prstne odtise uredimo po binarnem vrstnem redu, jih zlepimo in iz tako sestavljenega niza izračunamo prstni odtis.

Opisan pristop omogoča, da postopek podaljševanja poenotimo, in sicer, ni več potrebe, da bi razlikovali med enostavnim in kompleksnim podaljševanjem. S tem poenostavimo strukturo evidenčnega zapisa ter postopka podaljševanja in preverjanja, ker ni potrebno razlikovati med arhivskimi časovnimi žigi glede na to, ali so nastali v procesu enostavnega ali kompleksnega podaljševanja. Hkrati omogočimo, da se dokazila shranjujejo neodvisno od XML strukture evidenčnega zapisa, ki jo sestavimo le po potrebi, na primer za izvoz dokazil. S tem se bo lahko XML shema za evidenčni zapis v prihodnosti razširila, ne da bi vplivali na evidenčne zapise ustvarjene s predhodno shemo.

Pri enostavnem podaljševanja časovno žigosamo predhodni arhivski časovni žig, torej izračunamo prstni odtis arhivskega časovnega žiga. Po analogiji s standardom ERS bi torej vzeli XML element, ki predstavlja arhivski časovni žig, ga kanonizirali in iz dobljenega niza izračunali prstni odtis. Zatem bi vzeli še sestavne entitete arhivskega časovnega žiga: časovni žig (ena entiteta), drevo prstnih odtisov (ena entiteta) in komplementarne podatke k časovnemu žigu (vsebuje več entitet). Iz vsake entitete bi izračunali prstni odtis, in sicer v odvisnosti od tipa entitete: XML ali binaren. Če je XML, ga je potrebno pred izračunom prstnega odtisa kanonizirati. Dobljene prstne odtise entitet, ki jih vsebuje arhivski časovni žig, uredimo po naraščajočem binarnem vrstnem redu, jih sestavimo v en niz in iz njega izračunamo prstni odtis, kot je prikazano v primeru št. 25. Novo ustvarjen arhivski časovni žig bo vseboval drevo prstnih odtisov, ki bo na prvem seznamu imelo prstne odtise posameznih entitet.

Primer 25: Postopka izračuna prstnega odtisa pri enostavnem podaljševanju

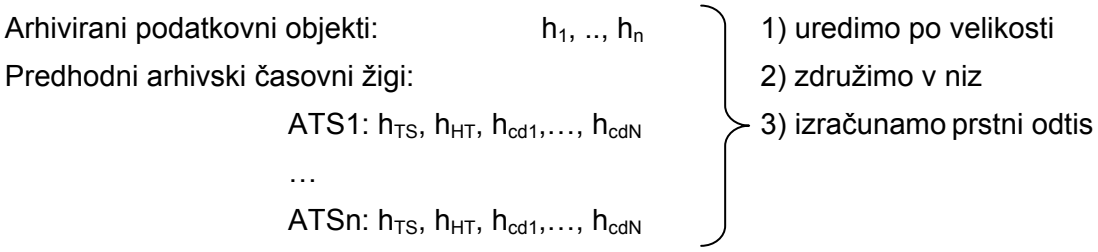
Arhivski časovni žig ::=



Pri kompleksnem podaljševanju, kot je prikazano s primerom št. 26, izračunamo skupni prstni odtis iz:

- prstnega odtisa arhiviranega podatkovnega objekta (objektov),
- prstnih odtisov vseh predhodnih časovnih žigov (in pripadajočih podatkov) oziroma iz celotnega evidenčnega zapisa.

Primer 26: Postopek izračuna skupnega prstnega odtisa pri kompleksnem podaljševanju



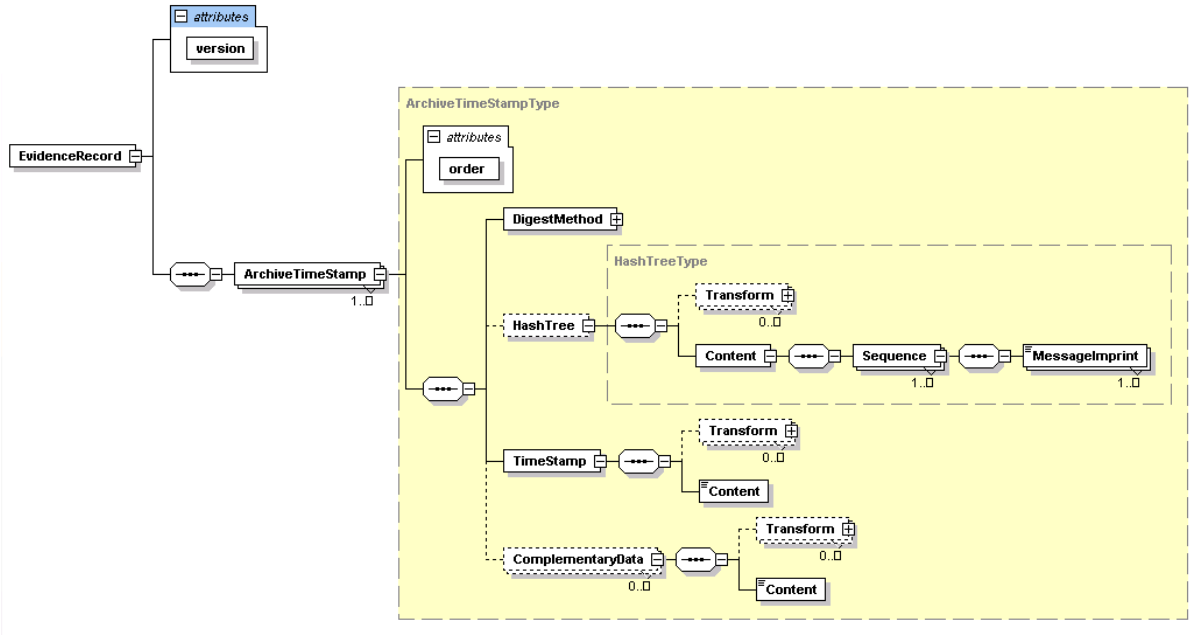
Vse postopke lahko poenostavimo tako, da zmeraj izvajamo kompleksno podaljševanje. S tem smo na varni strani. Pri ustvarjanju arhivskega časovnega žiga zbiramo več vhodnih podatkov in tudi drevo prstnih odtisov ima sčasoma na svojem prvem seznamu vse več prstnih odtisov osnovnih entitet. Pridobimo pa poenostavitev dveh ključnih postopkov (ustvarjanje arhivskega časovnega žiga in preverjanje veljavnosti evidenčnega zapisa) ter neodvisnost od uporabljene XML strukture za evidenčni zapis. Zadnje je še zlasti pomembno, saj bomo lahko v prihodnosti hranili dokazila v drugačnem formatu. Ob upoštevanju opisanega sem ustrezno prilagodila XSD shemo, kot je prikazano v primeru št. 27 in na sliki št. 15.

V bazi podatkov lahko hranimo posamezne entitete neodvisno od XML strukture evidenčnega zapisa. Glede prostora in upravljanja podatkov pridobimo za entitete, ki se ponavljajo pri večjem številu arhivskih objektov. Tovrstni sta predvsem dve: arhivski časovni žig, kadar smo izvršili skupinsko obdelavo, in komplementarni podatki k arhivskemu časovnemu žigu. Med njimi je predvsem seznam preklicanih digitalnih potrdil obsežnejši, reda nekaj MB.

Primer 27: Skica strukture evidenčnega zapisa za drugi predlog rešitve

```
<EvidenceRecord>
      <Version />
      <ArchiveTimeStamp Order>
         <DigestMethod />
         <HashTree> OPTIONAL
            (<Transform />)*
            <Content />
         </HashTree>
         <TimeStamp>
            (<Transform />)*
            <Content />
         </TimeStamp>
         <ComplementaryData> OPTIONAL
            (<Transform />)*
            <Content />
         </ComplementaryData>)*
      </ArchiveTimeStamp>) +
   </EvidenceRecord>
```

Slika 15: XSD shema za tretji predlog rešitve



Pri sestavljanju XML dokumenta je treba paziti na podatkovne tipe entitet: XML ali binarni. Entiteto binarnega tipa v XML dokument zapišemo v obliki base64 kodiranega niza (glej 1.3.3). Uporabljeno transformacijo vpišemo v element <Transform>. V primeru št. 28 je prikazana konkretna vsebina tega elementa za binarni in XML podatkovni tip entitete.

Primer 28: Vsebina element <Transform> za binarni in XML podatkovni tipa entitete

```
<Transform>http://www.w3.org/2000/09/xmldsig#base64</Transform>
<Transform>http://www.w3.org/TR/2001/REC-xml-c14n-20010315</Transform>
```

V postopke preverjanja dodamo pravilo, da se pred izračunom prstnega odtisa entitete, nad entiteto izvede navedena transformacija. V primeru XML podatkovnega tipa bo to ena izmed standardnih kanonizacij, v primeru binarnega tipa pa base64 dekodiranje.

4.3. Ovrednotenje predlogov in izbira rešitve

Prvi predlog sem opustila v toku analize (glej 4.2.1). V tabeli št. 7 podajam primerjavo drugih dveh predlogov.

Tabela 7: Primerjava 2. in 3. predloga za ERS v XML zapisu

Kriterij za primerjavo	**2. predlog: Analogna pretvorba**	**3. predlog: XML kot ovojnica**
Velikost zapisa	Ni bistvenih razlik glede samega zapisa. Glede baze podatkov tudi ne, v kolikor tudi pri drugem predlogu hranimo podatke v bazi podatkov, razdeljene po tabelah glede na sestavne dele, torej hranimo seznam preklicanih potrdil samo enkrat in sestavimo končni XML po potrebi, kadar ga potrebujemo pri podaljševanju ali za izvoz. Če hranimo celoten XML, potem se velikost baze pri 2. predlogu poveča na račun ponovitev: crl, digitalnih potrdil, časovnih žigov.	
Berljivost	Pri drugem predlogu se na prvi pogled loči enostavno podaljšane ATS od kompleksno podaljšanih.	
Ustvarjanje ERS	Ni bistvenih razlik.	
Podaljševanje	Za podaljševanje je nujno potrebno sestaviti XML zapis.	
Preverjanje	Za preverjanje je nujno potrebno sestaviti XML zapis.	Preverjanje nad podatki v bazi in nad izvozom v XML strukturo da iste rezultate.
Razširljivost XML strukture	Odvisen od strukture XML, zato strukture ni mogoče razširiti.	Neodvisen od strukture XML, lahko jo razširimo z dodatnimi elementi/atributi, specifičnimi za posamezno implementacijo.
Skladnost z ERS	Formalno in vsebinsko skladen.	Vsebinsko skladen. Formalna neskladnost v načinu izračuna prstnega odtisa iz preteklih dokazil.

Med predlogoma je značilna razlika v:

- skladnosti z obstoječim standardom ERS, in sicer je 2. predlog popolnoma skladen, medtem ko se 3. predlog razlikuje v postopkih podaljševanja,
- odvisnosti od XML strukture, in sicer je 3. predlog popolnoma neodvisen od XML strukture za shranjevanje dokazil, medtem ko v 2. predlogu pri podaljševanju računamo prstne odtise iz dokazil, vključno z XML strukturo evidenčnega zapisa.

Neodvisnost dokazil od XML strukture je pomembna z vidika neodvisnosti dokazil od oblike zapisa in z vidika razširljivosti XML strukture, saj je glede na zahtevano dolgoročnost 100 in več let, zelo verjetno, da bo potrebno razširiti strukturo za dokazila. V kolikor bi ERS standard omogočal neodvisnost dokazil od oblike za zapis, kar sedaj ne omogoča, bi ta kriterij prevladal. S 3. predlogom bi namreč uporabili procesna pravila za ustvarjanje, podaljševanje in preverjanje dokazil, ki bi bila popolnoma neodvisna od oblike zapisa. Tako bi podprli možnost za večjo medsebojno izmenjavo dokazil in za uporabo tisto obliko zapisa dokazil, ki bi bolje ustrezala namenu uporabe.

S tem je zaključena analiza treh možnih rešitev, in sicer najprej razširitev obstoječega standarda s XER kodiranjem, kot druga analogna preslikava procesov za ustvarjanje dokazil in njihov zapis ter kot tretja obravnava dokazil, neodvisno od strukture. Prvi predlog je opuščen zaradi neizpolnjevanja zahtev, druga dva sta oba ustrezna predloga za standard ERS v XML zapisu.

Skladnost z obstoječim standardom je pomembna z vidika boljše sprejetosti predloga za XML zapis v skupnosti uporabnikov in z vidika manjših razlik v procesnih pravilih, s tem pa je povezano enostavnejše utemeljevanje ustreznosti predloga. Sledenji argument je prevladal v procesu izbire rešitve znotraj odgovorne delovne skupine pri IETF; tako je predlagan predlog rešitve št. 2 kot predlog za standard ERS v XML zapisu (glej prilogo št. 4). Predstavljeni ter sprejeti predlog omogoča, da je XML podatkovna struktura skladno preslikana iz ASN.1 specifikacije in nadgrajena v skladu s specifičnimi lastnostmi XML jezika. Na primer: za potrebe izračuna prstnega odtisa je dodana v vse postopke kanonizacija XML dokumentov.

Sklep

Dokazovanje verodostojnosti elektronskih dokumentov je ključna zahteva v procesih brezpapirnega poslovanja. Elektronski dokumenti in podpisi so priznani kot enakovredni papirnim dokumentom oziroma ročnim podpisom. Slovenija pri sprejemanju podpornih pravnih aktov prav v ničemer ne zaostaja za Evropsko unijo, področje elektronskega arhiviranja je zakonodajno celo bolj podrobno opredeljeno kot v drugih evropskih državah. Na primer, v praksi nekaj slovenskih podjetji že izdaja del računov izključno v digitalni obliki.

Standardizacija na področju ustvarjanja, ohranjanja in zapisovanja dokazov o verodostojnosti elektronskih dokumentov je v začetnih korakih. V letu 2007 je pri IETF izšel standard ERS z oznako RFC 4998, drugih standardov pa še ni. Za omejen obseg scenarijev rabe se uporablja tudi standard XAdES organizacije ETSI, ki je nadgradnja digitalnega podpisa v XML obliki po standardu XMLDsig in je nastal z namenom, da bi digitalnemu podpisu dodajali atribute, ki pričajo o njegovi verodostojnosti.

Standard ERS sloni na zapisu dokazil o verodostojnosti v ASN.1 opredeljeni strukturi in DER kodiranju podatkov. Zaradi potreb po povezljivosti in izmenjavi v sistemih, ki uporabljajo XML tehnologije, je v sklopu IETF oblikovana delovna skupina, ki pripravlja predlog ERS standard v XML zapisu (trenutno je objavljena tretja revizija predloga RFC ERS-XML). Kot članica te delovne skupine sem uporabila v magistrskem delu izkušnje in rezultate, pridobljene s sodelovanjem v njej, ter svoje izvirne zamisli in predloge.

Pri obstoječem standardu ERS so dokazila odvisna od oblike zapisa, saj v procesu ohranjanja nastajajo dodatna dokazila, ki se izračunavajo iz obstoječega zapisa. Razlog je optimalnejše procesiranje dokazil, ki v največji možni meri uporablja prednosti sheme ASN.1 in kodiranja DER. V kolikor bi standard ERS predvidel, da je dokaze mogoče prenesti iz ene oblike v drugo obliko zapisa, bi procesiranje izgubilo del učinkovitosti, vendar bi lahko isti standard uporabili za drugačne oblike zapisa dokazil. Zaradi razlik v procesnih pravilih za delo z XML in drugih lastnostih XML v primerjavi z ASN.1, je pri prilagoditvi standarda ERS potrebno ob nadomestitvi ASN.1 podatkovnih struktur z ustreznimi XML strukturami tudi prilagoditi procese ustvarjanja in preverjanja veljavnosti evidenčnih zapisov. Pri iskanju ustrezne rešitve je bilo tako eno od vodil neodvisnost procesov od

podatkovnih struktur - nosilcev dokazov. S tem bi odprli vrata za zamenljivost z drugimi oblikami zapisov v prihodnosti.

V magistrskem delu sem analizirala zahteve in možnosti za zapisovanje ERS v XML obliki ter predstavila dva predloga, od katerih je prvi usmerjen na čim večjo podobnost z obstoječim standardom, drugi pa na čim večjo neodvisnost procesov ustvarjanja, podaljševanja in zapisovanja dokazil od uporabljene podatkovne strukture. Delovna skupina pri IETF za pripravo predloga standarda ERS z zapisom v XML obliki se je odločila za čim večjo skladnost s standardom ERS v ASN.1 zapisu.

Za dokazovanje dolgoročne verodostojnosti elektronskih dokumentov ne poznamo tehnologij, ki bi same zase nesporno dokazovale celovitost, čas nastanka in verodostojnost vira dokumenta. V postopkih ustvarjanja dokazil se naslanjamo na ureditev družbe in mehanizme zaupanja v izbrane organizacije, konkretno na sistem infrastrukture javnih ključev in overiteljev časovnih žigov.

Tehnologije, ki jih uporabljamo, zagotavljajo dokaze na ravni ohranjanja zaporedja bitov. Slednje se postavlja kot ovira z vidika potrebe po ohranjanju tehnologij za ustvarjanje in interpretacijo elektronskih dokumentov. Na primer, ali video posnetek zgubi svojo vsebinsko celovitost, če ga zapišemo z drugačnim kodiranjem brez izgube informacij? Na ravni sedanje stopnje razvoja tehnologije zagotavljanja verodostojnosti seveda izgubi, saj se je zaporedje bitov spremenilo, na ravni vsebine pa je celovitost ohranjena. Med izzivi prihodnosti na področju ohranjanja dolgoročne veljavnosti je zagotovo ta, kako oblikovati dokaze, ki ne bodo odvisni od fizičnega zaporedja bitov.

Literatura

1. Adamski Dariusz et al.: Why *Digital Signatures Fail - Legal Concepts for Long Term Validity in Austria, Germany and Poland. Wrocław, Poland* : The Research Centre for Legal and Economic Issues of Electronic Communication, University of Wroclaw. [URL: http://www.dzi.tu-darmstadt.de/fileadmin/content/veranstaltungen/20060606-09_etrics/adamski_kutylowski_lauks.pdf], 6. 6. 2006. 2 str.
2. Adobe Systems: *Digital Signatures in the PDF Language*. Adobe Systems Inc., 345 Park Avenue, San Jose, CA 95110-2704 USA. 28. 3. 2006.
3. Arhiv republike Slovenije: *Enotne tehnološke zahteve*. Ljubljana : Arhiv RS, 1. 12. 2006. 113 str.
4. Bartel Mark et al.: *XML-Signature Syntax and Processing*. The Internet Society and W3C. [URL:http://www.w3.org/TR/xmldsig-core/], 12.2.2002.
5. Berčič Boštjan: *12 zmot o elektronskem arhviranju*. Ljubljana: Mladina. Sistem, januar 2008. str. 12−13.
6. Blanchette Jean-Francois: *The digital signature dilemma*. Annales des télécommunications, Lavosier, 61(2006), 7−8, str. 908−923.
7. Borenstein Nathaniel et al.: Multipurpose *Internet Mail Extensions (MIME) Part One*. IETF, RFC 2045. [URL:http://www.ietf.org/rfc/rfc2045.txt], 1996.
8. Boyer John: *Canonical XML Version 1.0*. The Internet Society, W3C and IETF, RFC 3076. [URL:http://www.ietf.org/rfc/rfc3076.txt] in [URL:http://www.w3.org/TR/xml-c14n], marec, 2001.
9. Brandner Ralph et al.: *Evidence Record Syntax (ERS)*. IETF, RFC 4998. [URL:http://www.ietf.org/rfc/rfc4998.txt], avgust, 2007b.
10. Brandner Ralph et al.: *Long-Term Archive Service Requirements*. IETF, RFC 4810. [URL:http://www.ietf.org/rfc/rfc4810.txt], marec, 2007a.
11. Burnett Steve, Paine Stephen: *RSA Security's Official Guide to Cryptography*. Berkely : McGraw-Hill, 2001. 419 str.
12. CCSDS-Consultative Committee for Space Data System: *Reference Model for an Open Archival Information System (OAIS)*. Washington : CCSDS, Blue Book, Issue 1, 2002. 148 str.
13. Chadwick David, Mundy Darren: *An XML alternative for performance and security: ASN.1*. B.k. : IEEE Computer Society, IEEE IT Professional, volume 6, issue 1, februar, 2004, str. 30–36.
14. CVI: Varnostni *vidiki aplikacij z uporabo digitalnih potrdil Sigen-CA in Sigov-CA*. Ljubljana : Ministrstvo za javno upravo, 2003. 34 str.
15. Department of Defense: Design *Criteria Standard for Electronic Record Managment Software Application, standard DoD 5015.2*. Washington : Department of Defense USA, 1997. 118 str.
16. DLM Forum, Evropska komisija: *Model zahtev za upravljanje elektronskih dokumentov - specifikacija MoReq*. Ljubljana : Arhiv Republike Slovenije, 2005. 96 str.

17. EESSI Final Report. European Comission : *European Electronic Signature Standardization Initiative (EESSI)*. [URL: www.ictsb.org/EESSI/Documents/Final-Report.pdf], 20.7.1999.
18. ETSI-European Telecommunications Standards Institut: *Electronic Signatures and Infrastructures (ESI): Algorithms and Parameters for Secure Electronic Signatures*. Cedex, France : ETSI, TS 102 176-1 V2.0.0 (2007-11), november, 2007.
19. ETSI-European Telecommunications Standards Institut: *XML Advanced Electronic Signatures (XAdES)*. Cedex, Francija : ETSI, 2002.
20. Gondrom Tobias, Jerman Blažič Aljoša, Šaljić Svetlana: *Extensible Markup Language Evidence Record Syntax*. IETF, draft. [URL: http://www.ietf.org/internet-drafts/draft-ietf-ltans-xmlers-01.txt], 27. 5. 2008.
21. Graham Peter: *Long term intellectual preservation*. Proceedings from an RLG Symposium. Mountain View, Cornell University : Digital Imaging Technology for Preservation, 1994. str. 41−58.
22. Hiroshi Maruyama, Takeshi Imamura: *Mapping Between ASN.1 and XML*. B.k : IBM Research, RT0362. [URL:http://www.trl.ibm.com/projects/xml/xss4j/docs/axt-readme.html], 2000.
23. Hollenbeck Scott, et al.: *Guidelines for the Use of Extensible Markup Language (XML)*. IETF, RFC 3470. [URL:http://tools.ietf.org/rfc/rfc3470.txt], 2003.
24. ICA - International Committee On Archives, Committee on Electronic Records: *Guide for Managing Electronic Records from an Archival Perspective*. B.k : ICA Study. [URL:http://www.ica.org /en/node/30019], 8. 2. 1997. 58 str.
25. Jerman Blažič Aleksej, Džonova Jerman Borka, Klobučar Tomaž: *Long-term trusted preservation service using service interaction protocol and evidence records.* Amsterdam : Elsevier Science Publishers. Computer Standards & Interfaces, Volume 28, Issue 3, marec, 2007.
26. Jerman Blažič Aleksej, et al.: *Long-term Archive Protocol (LTAP)*. IETF, draft. [URL:http://www.ietf.org/internet-drafts/draft-ietf-ltans-ltap-06.txt], 25. 2. 2008.
27. Jerman Blažič Aleksej, Peter Sylvester: *Provision of Long-Term Archiving Service for Digitally Signed Documents Using an Archive Interaction Protocol*. Berlin, Springer : Public Key Infrastructure, 2005. 240−254 str.
28. Libon Olivier et al: *Trusted Archival Services*. European Comission : EESSI, 28. 8. 2000. 66 str.
29. Lynch Clifford: *Accessibility and Integrity of Networked Information Collections*. B.k. : Office of Technology Assessment, Congress of the United States, July 5, 1993.
30. Lynch Clifford: *Canonicalization: A Fundamental Tool to Facilitate Preservation and Management of Digital Information*. D-Lib Magazine, 5(9), [URL: http://www.dlib.org/dlib/september99/09lynch.html], 1999.
31. Mednarodni arhivski svet: *Elektronski dokumenti: priročnik za arhiviste*. Ljubljana : Arhiv Republike Slovenije, 2006. 99 str.
32. Merkle Ralph: *Protocols for Public Key Cryptosystems*. IEEE Symposium on Security and Privacy, april, 1980. str. 122−134.
33. Novak Miroslav: *Arhivska stroka in e-dokumenti*. Zbornik DOK_SIS 2002: III - Sistemi za upravljanje z dokumenti. Ljubljana : Media.doc, 2002. str. 1−10.

34. Objective Systems, Inc.: *Guidelines for ASN.1 to XML Schema Conversion*. [URL:http://www.objsys.com/docs/acv58/XSDUsersGuide/XSDUsersGuide4.html], 18. 2. 2008a.
35. OpenEvidence project: *State of the Art Report: D2.1.IST-2001-35174*. [URL: http://www.openevidence.org], 30. 6. 2002. 67 str.
36. Palmer Toni: *Comparing the Processor Load for XML Signature and PKCS7 Digital Signature*. Intel Software Network. [URL: http://softwarecommunity.intel.com/articles/eng/2385.htm]. 22. 4. 2004.
37. Žumer Vladimir: *Arhiviranje zapisov: priročnik za ravnanje z dokumentarnim in arhivskim gradivom*. Ljubljana : GV Založba, 2001. 479 str.

Viri

1. ASN.1 information site. [URL:http://asn1.elibel.tm.fr/en/index.htm], 18. 2. 2008.
2. BinaryXML. [URL:http://en.wikipedia.org/wiki/Binary_xml], 18.2.2008.
3. Cover Robin: Technology Report - ASN.1 Markup Language (AML). [URL:http://xml.coverpages.org/asn1-aml.html], 24. 11. 2003.
4. DVCS - Adams Carlisle et al.: Data Validation and Certification Server Protocols. IETF : RFC 3029. [URL: http://www.ietf.org/rfc/rfc3029.txt], Februar, 2001.
5. eGov-Bus project team: Advanced eGovernment Information Service Bus. EU : FP6-IST-4-026727-STP. [URL:http://demo.a-sit.at/e_government/egov_bus/signature_transformation.html], 17. 7. 2007.
6. European Parliament and Council: Directive on electronic commerce. DIRECTIVE 2000/31EC.
7. European Parliament and Council:Directive on a Community Framework or Electronic Signatures. DIRECTIVE 1999/93/EC.
8. FreePDFSign: Orodje za ustvarjanje pdf podpisa. Temelji na Javanski knjižnici iText. [URL: http://www.yan.cz/freepdfsign/download.php?lang=cs]. 23. 3. 2008.
9. ITU-T Study Group 17: Abstract Syntax Notation One (ASN.1). B.k. : ITU-T, Rec. X.680, 2002. 146 str.
10. ITU-T Study Group 17: ASN.1 encoding rules: Mapping W3C XML schema definitions into ASN.1. B.k. : ITU-T, Rec. X.694, 2004. 70 str.
11. ITU-T Study Group 17: ASN.1 encoding rules: XML Encoding Rules (XER). B.k. : ITU-T, Recommendation X.693, 2001. 18 str.
12. MJU: Uporaba kriptografije v internetu. [URL:http://www.ca.gov.si/kripto/index.htm], januar 2007.
13. MoReq2 - Nadgradnja modela zahtev za uporabljanje z dokumenti. [URL: http://www.moreq2.eu/]. 4. 5. 2008.
14. NIST, Barker Elain, et al.: Computer Security. B.k. : National Institute of Standards and Technology, NIST Special Publication 800-57.Part1, marec, 2007. [URL:csrc.nist.gov/publications/nistpubs/80057/SP800-57-Part1.pdf].142 str.

15. Objective Systems, Inc,: Online tool for ANS to XSD conversion. [URL:http://www.obj-sys.com/asn2xsdform.shtml], 18. 2. 2008b.
16. Overitelj digitalnih potrdil na Ministrstvu za javno upravo. [URL:http://www.ca.gov.si], 4. 3. 2008.
17. RSA Laboratories: PKCS #7: Cryptographic Message Syntax Standard. [URL: ftp://ftp.rsasecurity.com/pub/pkcs/doc/pkcs-7.doc]. November, 1993.
18. SETCCE: eKeeper – verodostojen elektronski arhiv. [URL:http://www.setcce.si/slo/index42d.php], 18. 2. 2008.
19. SSKJ - Slovar Slovenskega knjižnega jezika. [URL:http://bos.zrc-sazu.si/sskj.html], 28. 2. 2008.
20. The Unicode Consortium: Unicode Standard, Version 5.0. Addison-Wesley Professional, 5 edition. November, 2006. 1472 strani.
21. Uredba o pogojih za elektronsko poslovanje in elektronsko podpisovanje (Uradni list RS, št. 77/2000).
22. Uredba o varstvu arhivskega in dokumentarnega gradiva (Uradni list RS, št. 86/2006).
23. W3C : XML Schema recommendation. [URL:http://www.w3.org/XML/Schema], W3C , 2004.
24. Zakon o arhivskem gradivu in arhivih (Uradni list RS, št. 20/1999).
25. Zakon o elektronskem poslovanju in elektronskem podpisu – ZEPEP (Uradni list RS, št. 57/2000).
26. Zakon o spremembah in dopolnitvah zakona o elektronskem poslovanju in elektronskem podpisu – ZEPEP-A (Uradni list RS, št. 25/2004).

Priloge

Priloga 1: Slovarček uporabljenih tujih strokovnih izrazov in kratic

Abstract Syntax Notation One	ASN.1	Jezik za opis podatkovnih struktur ASN.1
American Standard Code for Information Interchange	ASCII	Ameriška kodna stran za zapis 128 znakov
Archive Time Stamp	ATS	Arhivski časovni žig
Archive Time Stamp Chain		Veriga arhivskih časovnih žigov
Archive Time Stamp Sequence		Zaporedje verig arhivskih časovnih žigov
Binary XML		Zgoščen zapis XML dokumenta
Canonicalization		Kanonizacija
Certificate Authority	CA	Overitelj digitalnih potrdil
Certificate Revocation List	CRL	Seznam preklicanih potrdil
Code page		Kodna stran (nabor znakov, pri katerem je vsakemu znaku dodeljena številka)
Cryptographic Message Syntax	CMS	Slovnica za kriptografska sporočila (opredeljeno v standardih IETF)
Data Structure for Security Suitabilities of Cryptographic Algorithms	DSSC	Podatkovna struktura za zapis o primernosti uporabe kriptografskih algoritmov
Data Validation and Certification Server Protocols	DVCS	Protokol za preverjanje veljavnosti digitalnega podpisa
Digital Signature		Digitalni podpis
Electronic Records Management System	ERMS	Informacijski sistem za upravljanje z dokumenti - ISUD (v ETZ) ali tudi elektronski sistem za upravljanje dokumentarnega gradiva - ESUD (v slovenskem prevodu MoReq)
Electronic Signature		Digitalni podpis
European Electronic Signature Standardization Initiative	EESSI	Evropska iniciativa za standardizacijo digitalnega podpisa
European Telecommunications Standards Institute	ETSI	Evropski inštitut za telekomunikacijske standarde
Evidence Record Syntax	ERS	Sintaksa za zapis dokazil o dolgoročni verodostojnosti elektronskih dokumentov
Extensible Markup Language	XML	Razširljivi označevalni jezik

eXtensible Markup Language Digital SIGnature	XMLDsig	Digitalni podpis v XML zapisu
File		Datoteka
Hash Function		Zgostitvena funkcija
Hash Tree		Drevo prstnih odtisov
International Council on Archives	ICA	Mednarodni arhivski svet
International Electrotechnical Commission	IEC	Mednarodna elektrotehniška komisija
International Organization for Standardization	ISO	Mednarodna organizacija za standardizacijo
International Telecommunication Union -Telecommunication Standardization Sector	ITU-T	Mednarodna zveza za telekomunikacije - sektor za standardizacijo
Internet Engineering Task Force	IETF	Mednarodna organizacija za standardizacijo tehnologij Interneta
Internet Society	ISOC	Mednarodna organizacija za koordinacijo interneta
Long-term Archive Protocol	LTAP	Protokol za komunikacijo z dolgoročnim arhivom
Namespace		Imenski prostor
Online Certificate Status Protocol,	OCSP	Protokol za sprotno preverjanje statusa digitalnega potrdila
Open Archival Information System	OAIS	Odprt referenčni model za arhiviranje informacijskih objektov
Organization for the Advancement of Structured Information Standards	OASIS	Organizacija za napredek strukturiranih informacijskih standardov
Public Key Infrastructure	PKI	Infrastruktura javnih ključev
Public Key, Private Key		Javni ključ, Zasebni ključ
Public-Key Cryptography Standards	PKCS	Kriptografski standard PKCS
Reduced Hash Tree		Zmanjšano drevo prstnih odtisov
Request for Comments	RFC	Priporočilo ali standard organizacije IETF
Secure Multi-Purpose Internet Mail Extensions	S/MIME	Standard za varno izmenjavo elektronske pošte
Structured Query Language	SQL	Strukturirani poizvedovalni jezik za delo z

		relacijskimi bazami podatkov
Time Stamp	TS	Časovni žig
Time Stamp Authority	TSA	Izdajatelj časovnih žigov
Time Stamp renewal		Podaljšanje časovnega žiga
Trusted Archival Service	TAS	Zaupanja vredna storitev arhiviranja
United Nations Commission on International Trade Law	UNCITRAL	Komisija Združenih narodov za mednarodno in trgovinsko pravo
What You See Is What You Sign	WYSIWYS	Koncept »podpisal si to, kar si videl«
World Wide Web Consortium	W3C	Združenje za razvoj povezljivih tehnologij svetovnega spleta

Priloga 2: ASN.1 modul

```
   ERS {iso(1) identified-organization(3) dod(6)
         internet(1) security(5) mechanisms(5)
         ltans(11) id-mod(0) id-mod-ers88(2) id-mod-ers88-v1(1) }
   DEFINITIONS IMPLICIT TAGS ::=
   BEGIN

   -- EXPORTS ALL --

   ltans OBJECT IDENTIFIER ::=
            { iso(1) identified-organization(3) dod(6) internet(1)
              security(5) mechanisms(5) ltans(11) }

  ContentInfo ::= SEQUENCE {
        contentType ContentType,
        content [0] EXPLICIT ANY DEFINED BY contentType }

      ContentType ::= OBJECT IDENTIFIER

      Attribute ::= SEQUENCE {
        attrType OBJECT IDENTIFIER,
        attrValues SET OF AttributeValue }

      AttributeValue ::= ANY

AlgorithmIdentifier  ::=  SEQUENCE  {
     algorithm               OBJECT IDENTIFIER,
     parameters              ANY DEFINED BY algorithm OPTIONAL  }
                                -- contains a value of the type
                                -- registered for use with the
                                -- algorithm object identifier value


   EvidenceRecord ::= SEQUENCE {
      version                   INTEGER { v1(1) } ,
      digestAlgorithms          SEQUENCE OF AlgorithmIdentifier,
      cryptoInfos               [0] CryptoInfos OPTIONAL,
      encryptionInfo            [1] EncryptionInfo OPTIONAL,
      archiveTimeStampSequence  ArchiveTimeStampSequence
      }

   CryptoInfos ::= SEQUENCE SIZE (1..MAX) OF Attribute

   ArchiveTimeStamp ::= SEQUENCE {
     digestAlgorithm [0] AlgorithmIdentifier OPTIONAL,
     attributes      [1] Attributes OPTIONAL,
     reducedHashtree [2] SEQUENCE OF PartialHashtree OPTIONAL,
     timeStamp       ContentInfo}

   PartialHashtree ::= SEQUENCE OF OCTET STRING

   Attributes ::= SET SIZE (1..MAX) OF Attribute

   ArchiveTimeStampChain    ::= SEQUENCE OF ArchiveTimeStamp

   ArchiveTimeStampSequence ::= SEQUENCE OF
                                ArchiveTimeStampChain
```

```
   EncryptionInfo       ::=     SEQUENCE {

       encryptionInfoType     OBJECT IDENTIFIER,
       encryptionInfoValue    ANY DEFINED BY encryptionInfoType}

   id-aa-er-internal  OBJECT IDENTIFIER ::= { iso(1) member-body(2)
      us(840) rsadsi(113549) pkcs(1) pkcs9(9) smime(16) id-aa(2) 49 }

   id-aa-er-external  OBJECT IDENTIFIER ::= { iso(1) member-body(2)
      us(840) rsadsi(113549) pkcs(1) pkcs9(9) smime(16) id-aa(2) 50 }

   END
```

Priloga 3: XML sheme za predlog ERS v XML zapisu

```
<?xml version="1.0" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns="http://www.setcce.org/ERS"
    targetNamespace="http://www.setcce.org/ERS"
    xmlns:asn1="http://www.obj-sys.com/v1.0/XMLSchema"
    elementFormDefault="qualified">
<xsd:import namespace="http://www.obj-sys.com/v1.0/XMLSchema"
    schemaLocation="asn1.xsd" />

<xsd:element name="evidenceRecord" type="EvidenceRecord" />

<xsd:complexType name="EvidenceRecord">
  <xsd:sequence>
    <xsd:element name="version">
      <xsd:union>
        <xsd:simpleType><xsd:restriction base="xsd:token">
          <xsd:enumeration value="v1" />
        </xsd:restriction></xsd:simpleType>
        <xsd:simpleType>
          <xsd:restriction base="xsd:integer" />
        </xsd:simpleType>
      </xsd:union>
    </xsd:element>
    <xsd:element name="digestAlgorithms">
      <xsd:complexType>
        <xsd:sequence minOccurs="0" maxOccurs="unbounded">
          <xsd:element name="element"
             type="AlgorithmIdentifier" />
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="cryptoInfos" minOccurs="0"
                            type="CryptoInfos" />
    <xsd:element name="encryptionInfo" minOccurs="0"
                            type="EncryptionInfo" />
    <xsd:element name="archiveTimeStampSequence"
                            type="ArchiveTimeStampSequence"/>
  </xsd:sequence>
</xsd:complexType>
```

```
<xsd:simpleType name="ContentType">
  <xsd:restriction base="asn1:ObjectIdentifier" />
</xsd:simpleType>

<xsd:complexType name="AttributeValue">
  <xsd:sequence>
    <xsd:any processContents="lax" minOccurs="1" />
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="ContentInfo">
  <xsd:sequence>
    <xsd:element name="contentType" type="ContentType" />
    <xsd:element name="content" type="asn1:OpenType" />
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Attribute">
  <xsd:sequence>
  <xsd:element name="attrType" type="asn1:ObjectIdentifier" />
  <xsd:element name="attrValues">
    <xsd:complexType>
      <xsd:sequence minOccurs="0" maxOccurs="unbounded">
      <xsd:element name="element"
                  type="AttributeValue" />
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="AlgorithmIdentifier">
    <xsd:sequence>
      <xsd:element name="algorithm"
                  type="asn1:ObjectIdentifier" />
      <xsd:element name="parameters" minOccurs="0"
                  type="asn1:OpenType" />
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="CryptoInfos">
    <xsd:sequence minOccurs="1" maxOccurs="2147483647">
      <xsd:element name="element" type="Attribute" />
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="EncryptionInfo">
    <xsd:sequence>
      <xsd:element name="encryptionInfoType"
            type="asn1:ObjectIdentifier" />
      <xsd:element name="encryptionInfoValue"
            type="asn1:OpenType" />
    </xsd:sequence>
  </xsd:complexType>
```

```
  <xsd:complexType name="Attributes">
    <xsd:sequence minOccurs="1" maxOccurs="2147483647">
      <xsd:element name="element" type="Attribute" />
    </xsd:sequence>
  </xsd:complexType>

  <xsd:simpleType name="PartialHashtree">
    <xsd:list itemType="xsd:hexBinary" />
  </xsd:simpleType>

  <!--  ArchiveTimeStampType -->
  <xsd:complexType name="ArchiveTimeStamp">
    <xsd:sequence>
      <xsd:element name="digestAlgorithm" minOccurs="0"
                   type="AlgorithmIdentifier" />

      <xsd:element name="attributes" minOccurs="0"
                   type="Attributes" />

      <xsd:element name="reducedHashtree" minOccurs="0">
        <xsd:complexType>
          <xsd:sequence minOccurs="0"
                maxOccurs="unbounded">
             <xsd:element name="element"
                type="PartialHashtree" />
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>

      <xsd:element name="timeStamp" type="ContentInfo" />

    </xsd:sequence>
  </xsd:complexType>

  <!--  ArchiveTimeStampChainType -->
  <xsd:complexType name="ArchiveTimeStampChain">
    <xsd:sequence minOccurs="0" maxOccurs="unbounded">
      <xsd:element name="element" type="ArchiveTimeStamp" />
      </xsd:sequence>
    </xsd:complexType>

  <!--  ArchiveTimeStampSequenceType -->
  <xsd:complexType name="ArchiveTimeStampSequence">
    <xsd:sequence minOccurs="0" maxOccurs="unbounded">
      <xsd:element name="element" type="ArchiveTimeStampChain"
        />
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

Priloga 4: ERS v XML zapisu – predlog standarda

Predlog standarda je javno dostopen na spletnih straneh organizacije IETF, v ilustracijo podajam prvo stran predloga:

Long-term Archive And Notary Services (LTANS)
Internet Draft
Intended status: Standards Track
Expires: November 27, 2008

A. Jerman Blazic
SETCCE
S. Saljic
SETCCE
T. Gondrom
Open Text Corporation
May 27, 2008

Extensible Markup Language Evidence Record Syntax
draft-ietf-ltans-xmlers-02.txt

Status of this Memo

Abstract

In many scenarios, users must be able to demonstrate the (time) existence, integrity and validity of data including signed data for long or undetermined period of time. This document specifies XML syntax and processing rules for creating evidence for long-term non-repudiation of existence of data. ERS-XML incorporates alternative syntax and processing rules to ASN.1 ERS syntax by using XML language.