XML a bezpečnost

Dagmar BRECHLEROVÁ

KIT PEF ČZU a EUROMISE, Praha
Dagmar.Brechlerova@seznam.cz

INFORUM 2006: 12. konference o profesionálních informačních zdrojích
Praha, 23. - 25.5. 2006

Abstrakt.

Webové služby jsou založeny na standardech W3C, jsou technologicky nezávislé. Jedním z jejich základních problémů je ale neřešení bezpečnosti. Pro komerční, ale i jiné aplikace, kde se jedná o přenos citlivých či osobních informací, je tato vlastnost (neřešení bezpečnosti) nepřijatelná.

Během přenosu informací je nutno příslušné informace ochránit před úmyslným i neúmyslným prozrazením a/nebo modifikací, je nutno ověřit identitu podepsaného pracovníka. Je nutno zajistit základní bezpečnostní požadavky (autenticita, integrita dat, nepopíratelnost a utajení, a dostupnost dat). V prostředí Internetu je nutno zjistit identitu původce informace a určit, která data smí který uživatel vidět.

Kromě dostupnosti informací, která se zajišťuje jinými způsoby, se veškeré výše uvedené požadavky dají zajistit přímo přidáním určitých částí do XML. Samotné webové služby v sobě základní bezpečnostní služby nemají, bylo nutno XML rozšířit o část zvanou XML Security. Velmi rychle se vyvíjí standardizace pro bezpečnost webu. V příspěvku bude podán přehled v současné době vyvíjených technologií. Je to XML Signature, XML Encryption, XML Key Management Specification , Secure Assertion Markup Language , XML Access Control Markup Language , WS - security (Web Services Security), EbXML Message Services.

Úvod

Webové služby jsou služby použité v distribuovaných systémech pro volání vzdálených funkcí, které vytvářejí rámec na zabezpečení spolupráce mezi jednotlivými odděleními organizace (např. nemocnice, vysoké školy), prodejcem a zákazníkem, studentem a školou atd. Tvoří technologický základ pro integraci např. obchodních či jiných procesů v rámci organizace a/ nebo s dodavateli, zákazníky atd. Obvykle používají internetu příp. jiný druh sítě. Základem je jednak použití protokolu TCP / IP a jednak jazyka XML.

Webové služby jsou založeny na standardech **W3C**, jsou **technologicky nezávislé**. Jedním z jejich základních problémů je ale **neřešení bezpečnosti**. Protokol SOAP používaný ve webových službách je ve své podstatě XML dokument, který je přenášen obvykle protokolem HTTP. Pro řadu aplikací je tato vlastnost tj**. neřešení bezpečnosti nepřijatelná**.

XML je sice velmi silný nástroj k použití v tzv. „bussines procesech“ vzhledem k možnosti sdílení informací např. mezi jednotlivými odděleními atd., ale na druhou stranu tyto možnosti zvyšují **nebezpečí zneužití informací**. Pokud se sdílejí informace o kreditních kartách, citlivé informace o pacientech, osobní údaje např. studentů, informace o fakturách atd., jsou zde vždy obsažena data, která mohou být zneužita, ať již jde o obchodní informace nebo informace osobní (osobní údaje) nebo případně i citlivé informace (ve zdravotnictví). Navíc některé z těchto informací je nutno chránit v důsledku platných zákonů (Zákon o ochraně osobních údajů aj.) a situace v této oblasti se stále zpřísňuje. Např. při přenosu informací o pacientovi je nutno během přenosu příslušné informace **ochránit před úmyslným či neúmyslným prozrazením** či **modifikací**, je nutno ověřit **identitu** podepsaného pracovníka atd. Je tedy nutno zajistit **základní bezpečnostní požadavky** (autenticita, integrita dat, nepopiratelnost a utajení a dostupnost dat). V prostředí Internetu je nutno určit **identitu původce informace**. Je nutno určit, které informace smí který uživatel vidět. Při on-line transakcích musíme určit, **které transakce jsou platné**. Musíme zajistit **utajení citlivých dat** při transferu informací. Dalším požadavkem může např. být možnost, jak případně u soudu dokázat, že někdo měl přístup k určitým informacím. Navíc bezpečnostní požadavky je nutno splnit v každém bodě komunikace. **Kromě dostupnosti informací, která se zajišťuje jinými způsoby, se veškeré výše uvedené požadavky dají zajistit přímo přidáním určitých částí do XML.**
Vzhledem k tomu, že samotné webové služby v sobě základní bezpečnostní služby nemají, bylo nutno XML rozšířit o část zvanou **XML Security**. V následující části je podán stručný přehled v současné době vyvíjených technologií.

XML Security

V posledních letech byla vyvinuta celá řada XML bezpečnostních technologií, které jsou dnes v různé fázi normalizace a vývoje. Je to XML Signature (XML-DSIG nebo také XML-SIG nebo XS), XML Encryption (XE), XML Key Management Specification (XKMS), Secure Assertion Markup Language (SAML), XML Access Control Markup Language (XACML), WS- security (Qweb Services Security), EbXML Messege Services. Dále se jedná o Liberty Alliance Project. Tyto technologie jsou nyní v různé fázi vývoje.

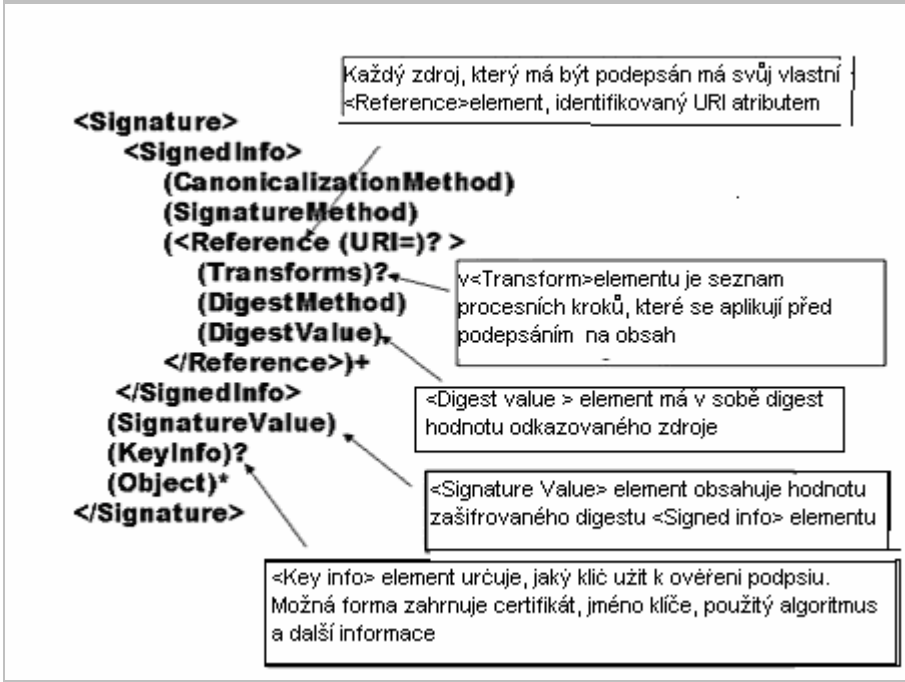
XML Digitální podpis (XS)

XML podpisy jsou digitální podpisy vyvinuté **pro užití v XML transakcích**. XML Signature je rozvíjející se standard pro digitální podpisy, který zahrnuje speciální požadavky a problémy, které XML prezentují pro podepisovací operace a užívá XML syntaxi pro vyjádření výsledku (což zjednodušuje integraci do XML aplikace.)
Standard [2] definuje schéma pro uložení výsledku operace digitálního podpisu aplikované na libovolná (ale nejčastěji XML) data. Stejně tak jako digitální podpisy, které neodpovídají specifikaci XML (např. PKCS podpis), tak i XML podpis poskytuje autentizaci, kontrolu integrity dat a podporu pro neopakování (non repundation). Navíc na rozdíl od „ne XML“ digitálního podpisu je XML podpis technologie, která se snaží **spojit výhody Internetu i XML**. Základním rysem XML podpisu je **schopnost podepsat pouze část XML stromu** (tree) spíše než celý kompletní dokument. To je velmi významné, když máme dokument, který má dlouhou historii, ve které jsou různé části dokumentu vytvořeny v různých časech různými autory, a každá má být podepsána pouze tím, kdo je pro danou část relevantní. Tato flexibilita je důležitá, když je nutné zajistit integritu určité části XML dokumentu, zatímco jiné části dokumentu je nutné nechat otevřené pro možnost změn. Předpokládejme například, že je uživatelovi doručen určitý formulář k dovyplnění.

(zdravotní karta, informace o studentovi atd.). Jestliže by podpis byl podpisem celého XML dokumentu, každá změna uživatele v původních hodnotách dokumentu by vedla k tomu, že by původní podpis porušila. **Tato možnost podepisovat pouze části dokumentu činí XML podpis mimořádně silným prostředkem.**
XML podpisem je možno podepisovat více druhů zdrojů**. Např. jeden XML podpis může pokrýt HTML data, JPG data, XML dat a specifickou části XML dat.**
Ověření podpisu vyžaduje, aby datový objekt, který byl podepsán, byl přístupný.
Samotný XML podpis obecně určuje umístění originálního podepisovaného objektu. Tento odkaz může být:

1. odkazován pomocí URI uvnitř XML podpisu
2. být umístěn ve stejném zdroji jako XML podpis (podpis je sourozenec)
3. být zakotven v XML podpisu (podpis je rodič)
4. mít podpis zakotven v sobě (podpis je dítě)

Následující obrázek ukazuje, jak se podpis postupně skládá z komponent. Jednotlivé kroky jsou dále zhruba popsány**..**



Jak vytvořit XML podpis

Následuje krátký úvod do vytvoření XML podpisu, úplná specifikace je XML Signature specification. [2]

1. Určíme, jakého druhu bude zdroj, který má být podepsán
Zdroj je identifikován svým URI (Uniform Resource Identifier).

"http://www.abccompany.com/index.html" odkazuje na HTML stránku na Webu
"http://www.abccompany.com/logo.gif" odkazuje na Gif obrázek na Webu
„http://www.abccompany.com/xml/po.xml „ odkazuje na XML soubor na Webu

"http://www.abccompany.com/xml/po.xml#sender1" odkazuje na specifický element XML souboru na Webu

2. Z každého zdroje je spočítán digest V XML podpisu je každý odkazovaný element specifikován skrze <Reference> element a každý jeho digest (spočítaný ze specifikovaného zdroje, nikoliv z elementu samého!!!!!) je umístěn v <DigestValue> dětském (child) elementu jako např.

```
<Reference URI="http://www.abccompany.com/news/2000/03_27_00.htm">
 <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
 <DigestValue>j6lwx3rvEPO0vKtMup4NbeVu8nk=</DigestValue>
</Reference>
<Reference
 URI="http://www.w3.org/TR/2000/WD-xmldsig-core-20000228/signature-
example.xml">
 <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
 <DigestValue>UrXLDLBIta6skoV5/A8Q38GEw44=</DigestValue>
</Reference>
```

<DigestMethod> element identifikuje algoritmus použitý na spočítání digestu. Tedy v tuto chvíli pouze počítám jednotlivé digesty a vždy k odkazu na podpisovaný zdroj připojím algoritmus (DigestMethod) a příslušný digest. (DigestValue).

3. Spojení, spočítání Reference elementů
Spojí se <Reference> elementy (spolu s jejich přičleněnými digesty) do <SignedInfo> elementu jako např.

```
<SignedInfo Id="foobar">
 <CanonicalizationMethod
   Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
 <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1" />
 <Reference URI="http://www.abccompany.com/news/2000/03_27_00.htm">
  <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
  <DigestValue>j6lwx3rvEPO0vKtMup4NbeVu8nk=</DigestValue>
 </Reference>
 <Reference
   URI="http://www.w3.org/TR/2000/WD-xmldsig-core-20000228/signature-
example.xml">
  <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
  <DigestValue>UrXLDLBIta6skoV5/A8Q38GEw44=</DigestValue>
 </Reference>
</SignedInfo>
```

Dalším přidaným prvkem je tzv. kanonizační metoda.

<CanonicalizationMethod> element udává, že byl použit **kanonizační algoritmus** na <SignedInfo> element. Různé datové proudy se stejnou XML informací mohou totiž dát různé textové reprezentace, např. se mohou lišit např. ve white spaces, v prázdných tagách, vložených řádcích, komentářích, oddělovačích atd. Proto, abychom zabránili rozdílným výsledkům při ověřování podpisu, je nutno, aby XML informace byly nejdříve před podepisovacím procesem kanonizovány. Aby byl podpis

jednoznačný, **musí se podepisovat a ověřovat zcela identický proud bitů (bytů).** Sebemenší změna ve vzkazu (message), ze kterého se počítá digest, má za následek změnu tohoto digestu. To je jedním ze základních kritérií pro kvalitní hash funkce. Ovšem pro použití v souvislosti s XML to přináší zásadní problémy. **Ačkoliv jsou 2 XML dokumenty logicky stejné, mohou se lišit textově**. To vše nemá vliv na logickou strukturu, ovšem hash je pak jiný. Kanonizace přesně popisuje a normuje jak vyrobit tzv. kanonizační formu (přesně definovaná fyzická struktura dokumentu).

<SignatureMethod> element indikuje, který z **podepisovacích algoritmů byl** použit pro podpis. Jde tedy o to, který algoritmus byl použit pro převedení kanonizovaného elementu <SignedInfo> na hodnotu <SignatureValue.>

4. Podepsání

Spočítá se digest <SignedInfo> elementu, podepíše se digest a hodnota podpisu se dá do <SignatureValue> elementu. Tedy se vlastně **může podepisovat několik digestů najednou**, které byly předtím nakumulovány do jednoho <SignedInfo> elementu.

```
<SignatureValue>MC0E LE=</SignatureValue>
```

5. Přidání informací o klíči

Pokud se přidávají **informace o klíči**, tak se umístějí do <KeyInfo> elementu. Zde uvedený příklad např. obsahuje **X.509 certifikát** odesílatele a tento certifikát obsahuje veřejný klíč nutný pro ověření podpisu. Tento element je volitelný, neboť podepisující nemusí mít vždy zájem prozrazovat svůj veřejný klíč všem stranám.

```
<KeyInfo>
 <X509Data>
  <X509SubjectName>CN=Ed
Simon,O=XMLSecinc.,ST=OTTAWA,C=CA</X509SubjectName>
  <X509Certificate>MIID5jCCA0+gA...lVN</X509Certificate>
 </X509Data>
</KeyInfo>
```

6. Složení do Signature elementu

A teď to celé poskládáme dohromady.
Umístí se <SignedInfo>, <SignatureValue>, a <KeyInfo> elementy do <Signature> elementu . <Signature> element zahrnuje XML signature.

```
<?xml version="1.0" encoding="UTF-8"?>
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
<SignedInfo Id="foobar">
<CanonicalizationMethod
  Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
<SignatureMethod
  Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1" />
<Reference URI="http://www.abccompany.com/news/2000/03_27_00.htm">
<DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
<DigestValue>j6lwx3rvEPO0vKtMup4NbeVu8nk=</DigestValue>
```

```
</Reference>
<Reference
  URI="http://www.w3.org/TR/2000/WD-xmldsig-core-20000228/signature-
example.xml">
<DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<DigestValue>UrXLDLBIta6skoV5/A8Q38GEw44=</DigestValue>
</Reference>
</SignedInfo>
<SignatureValue>MC0E~LE=</SignatureValue>
<KeyInfo>
<X509Data>
<X509SubjectName>CN=Ed Simon,O=XMLSec
Inc.,ST=OTTAWA,C=CA</X509SubjectName>
<X509Certificate>
MIID5jCCA0+gA...lVN
</X509Certificate>
</X509Data>
</KeyInfo>
</Signature>
```

7. Ověření XML podpisu

Jak se XML podpis ověří?
a. **Ověří se podpis <SignedInfo> elementu.** Abychom to provedli, tak se znovu spočte digest <SignedInfo> elementu (použije se digest algoritmus specifikovaný v <SignatureMethod> elementu) a použije se veřejný ověřovací klíč, abychom ověřili, že hodnota <SignatureValue> elementu je korektní digest <SignedInfo> elementu.

b. Jestliže tento krok proběhne, **spočte se digest z referencí** obsažených uvnitř <SignedInfo> elementu a porovná se to s digest hodnotami vyjádřenými v každém <Reference> elementu odpovídající <DigestValue> elementu.

XS je tedy **schéma XML pro uplatnění digitálního podpisu v XML**, kde digitální podpis zaručuje integritu (resp. zjištění porušení integrity), nepopíratelnost (nemožnost popřít odeslání) a autenticitu. Je možno podepsat jak celý XML dokument, části dokumentu nebo externí datové objekty, na které XML odkazuje.

Na XML Signature spolupracovalo **W3C konsorcium a IETF**. Cílem bylo vyvinout syntaxi takové části jazyka XML, která umožňuje podpisy webových zdrojů, dále jsou definovány procedury pro počítání a ověřování podpisů a je definována tzv. kanonizace. Hlavní výhodou je **možnost podepisovat pouze určité části dokumentu**, nemožnost této akce byla dříve hlavním problémem použití digitálního podpisu v reálné praxi. Za nevýhodu je považováno to, že jsou pro podpis nutné zdroje ze sítě. Poskytování autentizačních služeb by časem mohlo vést až k monopolu určitého výrobce SW.

XML Encryption - XML šifrování

Předpokládejme, že chceme poslat XML soubor společnosti, která publikuje knihy. Tento soubor obsahuje detaily o knize, kterou chceme koupit. Navíc obsahuje informace o kreditní kartě zákazníka. Pro komunikaci o takto soukromých údajích

chceme samozřejmě použít bezpečnou komunikaci. XML dokument, stejně tak jako jiné dokumenty, může být zašifrován vcelku např. SSL a poslán jednomu nebo více příjemcům.
Mnohem zajímavější ale je, jak řešit situaci, kdy různé části stejného dokumentu potřebují různé zacházení. Možností je XML šifrování. XML šifrování není alternativou SSL/TLS. Pokud aplikace vyžaduje zabezpečit celou komunikaci, je lepší SSL. Na druhé straně XML šifrování je nejlepší možností, pokud aplikace vyžaduje kombinaci bezpečné a nebezpečné komunikace. Tj. část bude vyměňována zabezpečeně a část nezabezpečeně. Navíc ovšem SSL nezajišťuje trvalou ochranu tj. při uložení na disku. Cílem vyvíjeného standardu je také schopnost rozlišit, zda je podpis aplikován na zašifrované části nebo naopak.

Příklad – XML, které budeme zašifrovávat

```
<purchaseOrder>
        <Order>
                <Item>book</Item>
                <Id>123-958-74598</Id>
                <Quantity>12</Quantity>
        </Order>
        <Payment>
                <CardId>123654-8988889-9996874</CardId>
                <CardName>visa</CardName>
                <ValidDate>12-10-2004</ValidDate>
        </Payment>
</purchaseOrder>
```

Tento XML soubor je velmi jednoduchý, tak aby na něm mohly být ukázány rysy vztahující se k šifrování. Skutečný XML soubor ve spolupráci s webovými službami bude mít podobnou strukturu, ale bude mnohem komplikovanější. Např. WSDL (Web Services Definition Language) a SOAP (Simple Object Access Protocol) jsou jazyky založené na XML, které jsou často užívané v B2B integraci. Jak WSDL tak i SOAP také mohou použít XML šifrování.

XML Encryption poskytuje široké možnosti. Jde zašifrovat celý soubor, jednotlivý element, obsah dokumentu. Dále ukázka z**ašifrování jednotlivého elementu pomocí XML šifrování**

Z nějakého důvodu můžeme chtít zašifrovat pouze 1 element z příkladu, např. element Payment . V tomto případě je výsledek ilustrován následujícím příkladem

```
<?xml version='1.0' ?>
<PurchaseOrder>
        <Order>
                <Item>book</Item>
                <Id>123-958-74598</Id>
                <Quantity>12</Quantity>
        </Order>
        <EncryptedData Type='http://www.w3.org/2001/04/xmlenc#Element'
        xmlns='http://www.w3.org/2001/04/xmlenc#'>
                <CipherData>
                        <CipherValue>A23B45C564587</CipherValue>
```

```
                    </CipherData>
          </EncryptedData>
</PurchaseOrder>
```

Zašifrování dat, která nejsou XML

Na následujícím příkladu je ukázáno, jak se zašifruje JPEG soubor pomocí XML šifrování. Celý zašifrovaný JPEG soubor je frekvence bytů, která je zde obsahem elementu CipherValue .

Příklad - Šifrování libovolných ne XML dat

```
<?xml version='1.0' ?>
<EncryptedData xmlns='http://www.w3.org/2001/04/xmlenc#'
                                Type='http://www.isi.edu/in-notes/iana/assignments/media-
types/jpeg' >
          <CipherData>
                    <CipherValue>A23B45C56</CipherValue>
          </CipherData>
</EncryptedData>
```

XE je schéma pro šifrování dat. Je možno šifrovat celý XML dokument, část nebo externí datové objekty, na které XML odkazuje. XE tedy umožňuje selektivní šifrování. Dále vyvíjený standard [5] by měl umožnit rozlišit části, které byly podepsány před zašifrováním a po zašifrování, a tedy se s nimi musí zacházet rozdílnými způsoby. XE je koordinováno konsorciem W3C. Za hlavní nevýhody nutno považovat to, že zašifrováním včetně tagů se ztrácí určité informační vlastnosti XML.

Kombinace XML podpisu a XML šifrování - některé obecné problémy pro použití bezpečnostních metod v XML

XML dokument, stejně tak jako jiné dokumenty, může být zašifrován vcelku a poslán jednomu nebo více příjemcům. To je situace např. s SSL a TLS. Ale mnohem zajímavější je, jak řešit situaci, kdy různé části stejného dokumentu potřebují různé zacházení. Výhodou XML je to, že XML dokument může být poslán jako celek, ale pak se s ním dá zacházet po částech, čímž se omezuje síťový provoz. Ale to pak stejně navozuje otázku, jak řídit autorizovaný náhled na rozdílné skupiny elementů. Obchodník potřebuje znát jméno a adresu zákazníka, ale nikoliv nějaké detaily o jeho kreditní kartě (ani to není žádoucí), naopak banka nepotřebuje vědět informace o zboží, které zákazník kupoval. Vědec by neměl vědět osobní informace ze zdravotní karty (měly by mu stačit anonymizované údaje), naopak administrátor nemocnice by měl vědět osobní informace a ne medicínskou historii. Lékař nebo sestra naopak potřebují medicínské detaily, ale ne úplné personální detaily. Např. v oblasti universitního informačního systému, pokud bude XML dokument použit jako záznam o studentovi, nepotřebuje pedagog znát informace např. o zdravotním stavu studenta, naopak je nežádoucí aby měl k takovým informacím přístup. Naopak měl by vidět některé jiné informace. Pedagog by měl samozřejmě podepisovat pouze části dokumentu, které se ho týkají. Stejně tak lékař nebo sestra.

Kryptografie nyní slouží nejen k ukrývání informací. Digest potvrdí integritu textu, digitální podpis podporuje autentizaci odesílatele a s tím spojené mechanismy jsou užívány k tomu, aby zajistily, že provedená transakce nebyla později odmítnuta jinou stranou.

Z obecného hlediska není problém s podepsáním XML dokumentu jako celku. Problém nastává, pokud části dokumentu je nutno podepsat, navíc různými lidmi a tato potřeba je společně dohromady se selektivním šifrováním. Může být totiž těžké nebo i nemožné určit zašifrování částí určenými lidmi v určitém pořadí. Další problém nastává tehdy, pokud je nutno nějakou část podepsat, ale ta je zašifrována někým jiným. Není totiž samozřejmě rozumné podepisovat něco zašifrovaného. Navíc data, která jsou již zašifrována, mohou být dále zašifrována jako součást větší množiny (části). Čím komplikovanější formulář bude a čím bude procházet více aplikacemi (např. workflow) a přes více uživatelů, tím to bude celé složitější.

Další problém je zakotven v samotném XML, pokud budeme šifrovat vše včetně tagů, ztrácí se schopnost vyhledávat v XML podle tagů, tedy vlastně vypovídací schopnost XML. Navíc pokud zašifrujeme celé tagy, je to vhodný materiál pro prolomení (známe strukturu, co jsme zašifrovali)**.**

WS Security

WS security je obecný mechanismus, který je stavebním kamenem pro spojení a využití jiných bezpečnostních řešení jako např. XML podpisu a XML šifrování. Definuje, jak vložit zašifrovaná data atd. do zprávy protokolu SOAP. WS Security byl vyvinut v roce 2002 společně Microsoftem, IBM a firmou Very Sign. Umožňuje zlepšení protokolu SOAP tak, aby poskytoval integritu, utajení a autentizaci vzkazu. Kombinuje SOAP s XML Encryption a XML Signature a je připraven i pro jiné bezpečnostní modely a jiné technologie.

XKMS

XML Key Management Specification (XKMS) je Zpráva W3C, byla vyvinuta společně W3C a IETF, je zde specifikován protokol pro klíčové hospodářství tj. registraci a distribuci veřejných klíčů. Tedy XKMS je určeno pro spojení s XML podpisem a XML šifrováním (XML Signature and XML Encryption). Definuje důvěryhodné služby pro management kryptografických klíčů.

XKMS má 3 části :
XML Key Information Service Specification (X- KISS) a **XML Key Registration Service Specification** (X – KRSS).
X KISS podporuje služby pro používání kryptografických klíčů.
X KRSS podporuje služby užívané držitelem kryptografických klíčů (registrace, obnova klíče atd.)
Bulk Key Regisration (X- Bulk) je rozšíření X-KRSS pro hromadnou registraci
Tyto protokoly mohou být použity dohromady se SOAP pro bezpečnou distribuci klíčů a nalezení informací o klíčích.

XACML

XACML (Extensible Access Control Markup language) je iniciativa vedená skupinou OASIS [3], je určená na vyjádření **bezpečnostní politiky pro přístup** (autentizaci a

autorizaci) k XML dokumentům a datovým zdrojům. Souvisí se SAML (viz dále) a to tak, že SAML poskytuje mechanismus pro šíření autentizačních a autorizačních informací mezi servery a službami, zatímco **XACML je autentizační a autorizační informací.** Idea XACML je ta, že XML dokument nebo samotný SOAP vzkaz může popisovat politiku přístupu, kdo má mít přístup k čemu atd.
Cílem je **standardizovat jazyk pro popsání autentizace a přístupových politik** v XML syntaxi. Standardní jazyk pro kontrolu přístupu vede k nízkým nákladům, protože není potřeba vyvíjet jazyk pro určitou aplikaci nebo psát politiky kontroly přístupu ve více jazycích. Pomocí XACML je možné vytvářet politiky kontroly přístupu z těch, které byly vytvořeny jinými stranami. XACML definuje slovník pro specifikaci předmětu, práv subjektu a podmínek. Jeden standardní jazyk pro řízení kontroly přístupu může nahradit několik jiných jazyků jednotlivých aplikací. XACML je jazyk pro řízení přístupu, poskytuje syntaxi (definovanou v XML) pro řízení přístupu ke zdrojům.
XACML je standard OASIS, který popisuje jednak **jazyk pro politiku a jednak jazyk pro řízení přístupu založený na dotaz / odpověď** (obojí je napsané v XML). XACML (extensible access control markup language) je tedy jazyk založený na XML. Jazyk politiky je použit na vyjádření politiky řízení přístupu – popisuje obecné požadavky řízení kontroly přístupu. A má standardní rozšiřitelné body pro definování nových funkcí, datových typů, kombinační logiky atd. Je zde tedy možno definovat, kdo může kdy dělat co. Jazyk dotaz / odpověď umožňuje vyjádřit dotaz, zda speciální akce může být dovolena (vyžádána) a interpretovat výsledek. Odezva (reakce) vždy zahrnuje odpověď, zda požadavek může být povolen za použití 4 hodnot:

Permit - povolit
Deny- zakázat
Inderterminate- neurčené, vyskytuje se chyba nebo chybí nějaké hodnoty a nemůže být určena odpověď
Not Applicable – žádost nemůže být odpověděna touto službou

V typickém XACML scénáři **subjekt** (tj. uživatel nebo pracovní stanice) chce udělat nějakou akci na speciálním zdroji. Subjekt podá žádost entitě, která chrání zdroj (např. web server, souborový systém atd.). Tato entita se nazývá **PEP** (Policy Enforcement Point). PEP zformuluje dotaz (za použití XACML jazyka) založený na atributech subjektu, akci, požadovaném zdroji a dalších informacích náležejících dotazu. PEP potom pošle tento dotaz **PDP** (Policy Decision Point), který se podívá na dotaz a nějakou politiku, kterou aplikuje na dotaz a přijde s odpovědí, zda může být přístup povolen. Odpověď vyjádřená v XACML jazyku je vrácena PEP, který pak povolí nebo zakáže přístup. PEP i PDP mohou být buď v jednotlivé aplikaci nebo mohou být distribuovány na několika serverech. Navíc k poskytnutí dotaz / odpověď a jazyku politik, XACML také poskytuje další části tohoto vztahu, speciálně nalezení politiky, která se aplikuje pro daný dotaz a ověření, zda porovnáním žádosti a dané politiky je přístup povolen nebo ne.

XACML má mnoho výhod nad jinými jazyky (existuje mnoho proprietárních a pro nějakou aplikaci specifických jiných jazyků) pro politiku řízení přístupu

1. Jeden standardní jazyk pro řízení kontroly přístupu může nahradit několik jiných jazyků jednotlivých aplikací. Použitím standardního jazyka se bude užívat něco, co užívá široká komunita expertů a uživatelů, nebude se muset

stále znovu zavádět nové jazyky, bude snazší spolupracovat s jinými aplikacemi používajícími stejný standardní jazyk.

2. Je generická. To znamená, že místo pokoušet se o accesss control pro speciální prostředí nebo speciální zdroj, může to být užito v jakémkoliv prostředí. Může být napsána jedna politika, která může být užita mnoha různými druhy aplikací a pokud je užit jeden společný jazyk, management politik je snazší.
3. Je to distrubuované tj. může se napsat politika, která se po řadě odvolává na jiné politiky, které se vyskytují v libovolném okolí. Výsledkem je to, že místo „managování" monolitické politiky různí lidé nebo různé skupiny mohou mít svoje politiky a XACML ví, jak zkombinovat. Výsledky z jednotlivých politik do jednoho řešení.
4. Je to mocný nástroj- je zde mnoho cest, jak jazyk rozšířit, mnoho prostředí. Mnoho datových typů, funkcí, pravidel, atd. Jsou zde různá pravidla, jak kombinovat výsledky různých politik. Dále pracovní skupiny pracují na rozšíření a profilaci XACML do dalších standardů jako SAML a LDAP, které zvýší množství cest pro použití XACML:

Jedna XACML politika může pokrýt mnoho zdrojů, to zabrání nekonsistentním politikám. XACML dovoluje jedné politice odkazovat na jiné, to je důležité pro velké organizace.

Příklad žádosti o přístup

Předpokládejme, že universita jménem University AB (UnivAB.cz) má politiku kontroly přístupu, která kterémukoli uživateli, jehož email patří do UnivAB.cz umožní provést jakoukoli akci nad jakýmkoli zdrojem.
Takto podobně by vypadala žádost o rozhodnutí (anglicky decision request), v níž si uživatel Dagmar Brechlerova , s adresou DB@UnivAB.cz chce přečíst záznam o studentovi XY UnivAB.:

```
<Request xmlns=... >
<Subject>
<Attribute AttributeId="urn:oasis: ... :1.0:subject:subjected"
DataType="urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name">
<AttributeValue>DB@UnivAB.cz </AttributeValue>
</Attribute>
</Subject>
<Resource>
<Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:ufspath"
DataType="http://www.w3.org/2001/XMLSchema#anyURI">
<AttributeValue>/UnivAB /record/student/XY</AttributeValue>
</Attribute>
</Resource>
<Action>
<Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
<AttributeValue>read</AttributeValue>
</Attribute>
</Action>
</Request>
```

SAML

SAML umožňuje přechod autentizačních a autorizačních informací mezi zúčastněnými stranami. SAML poskytuje tzv. "prosazení " důvěry. Takže tato aplikace může prosadit, že jde o určitého uživatele a ten má navíc určitá privilegia. SAML dokument může být digitálně podepsán pomocí XML signature. SAML poskytuje distribuci informace mezi určitou platformou a organizací a je proto jedno, kolik bodů prochází.. Např. nějaký portál autentizuje Alici a ví, že Alice má určitou roli. Portálová aplikace toto připojí do tvrzení v SOAP zprávy s dotazem na další webovou službu. Další webová služba se podívá na portálovou identitu, ověří digitální podpis portálu a povolí nebo zakáže přístup uživatele vzhledem k jeho roli. SAML je Security Assertion Markup Language of Structured Information: tento jazyk je vyvíjen OASIS. Cílem příslušné skupiny OASIS je vyvinout standard pro **výměnu autentizačních a autorizačních informací**. Jedním slovem SAML je konstrukce, systém založený na XML pro výměnu autentizačních a autorizačních informací.

Má 3 části:

1. definuje syntaxi a sémantiku XML zpráv obsahujících tvrzení (assertion) ve formě XML.
2. definuje protokoly žádostí a odpovědí mezi žádající a vydávající stranou pro výměnu bezpečnostních informací
3. definuje pravidla pro užití tvrzení se standardy pro transport, např. definuje jak SAML tvrzení můžeme transportovat ve zprávě SOAP přes http.

Tvrzení SAMLu neprovádějí autentizaci, ale slouží k obalení, zapouzdření tohoto procesu a jeho přenosu.

Použití SAMLu – existují celkem 3 různé scénáře a to

Jediné přihlášení SSO – uživatel se přihlásí na tom.com a je autentizován. Později se chce přihlásit na joe.com. Bez užití SSO by musel své údaje zadávat znovu. Pokud je užit SAML, pak joe.com pošle požadavek na tom.com s dotazem, zda se již uživatel na tom.com autentizoval. tom.com odpoví prohlášením, že ano, uživatel je autentizován. Poté joe.com zpřístupňuje své zdroje, aniž vyžaduje znovu přihlašovací informace.

Distribuovaná transakce

Uživatel použije přihlášení k nějaké službě na www.a1.cz a poté chce jinou službu od www.b1.cz. Uživatel poté může předat informace o svém profilu na www.a1.cz serveru www.b1.cz. Ten pošle tzv. SAML tvrzení serveru www.a1.cz, ve kterém bude chtít veškeré informace, které o uživatelovi má. www.a1.cz tyto informace pošle ve formě tzv. tvrzení.

Autorizační služba

Pomocí této služby je možno zasílat tvrzení, zda je někdo k něčemu autorizován, např. k platbě, objednání atd.

Liberty Alliance Project

Byl definován skupinou korporací s cílem ochránit soukromí zákazníků a definovat standard pro federální síťovou identitu (USA) pro SSO přes sítě, domény, organizace atd. Jde tedy o to, že se dá tímto způsobem dát záruka o identitě .

Je možno se jednou přihlásit a dalším členům ve skupině se pak už uživatelé dál hlásit nemusejí. Zahrnuje SAML, XML Encryption, XML Signature.[1]

Spolupráce těchto iniciativ

Jak již bylo dříve uvedeno, SAML vyměňuje autentizační a autorizační informace, tj. se jedná o velmi důležitý druh informací, které je nutno zajistit proti cizímu zásahu případně i proti vyzrazení. Proto tvrzení SAML mohou být digitálně podepsána pomocí XML Signature a pro zajištění utajení mohou být stejná tvrzení zašifrována pomocí XML Encryption. Veřejný klíč použitý k digitálnímu podpisu a šifrování může být ověřen a registrován pomocí XKMS, neboť právě XKMS definuje důvěryhodné služby pro management kryptografických klíčů.
Strana vydávající tvrzení SAML může využít XACML k definování politiky kontroly přístupu jako základ pro manipulaci s požadavky na tvrzení SAML.

Alice použije XML digitální podpis a XML šifrování k podepsání a zašifrování objednávky, která má formu XML dokumentu. Tento dokument pak odešle dodavateli Bobovi, nejspíše protokolem SOAP, jehož struktura hlavičky je definována bud ve WS-Security nebo ve standardu ebXML Message Service. Příjemce dokumentu Bob může využít XKMS k vyhledání a kontrole veřejného klíče, který patří Alici. Také ale může použít jiných mechanismů. Po ověření, že klíč je důvěryhodný, Bob ověří a dešifruje objednávku.

Závěr

V oblasti XML dochází v posledních letech k významným činům na poli bezpečnosti. Některé tyto iniciativy jsou teprve na úrovni vývoje, jiné jako např. XML podpis a XML šifrování jsou již i implementovány v produktech. Dá se očekávat, že tyto nové možnosti zvýší bezpečnost webových služeb a umožní některé doposud těžko řešitelné akce jako např. selektivní podepisování záznamů o pacientech, studentech apod.
Některé z výše uvedených bezpečnostních technologií jsou teprve ve stádiu normalizace, jiné ve stadiu návrhu. Jiné jsou již i implementovány. Nejdále je zatím XML podpis, rychlý rozvoj je dále u šifrování, v poslední době jsou také nové pokroky u XACML. Vzhledem k zájmu o tuto oblast je možno očekávat velmi rychlý pokrok a nasazení v nejbližší době.

Literatura

[1] http://xml.coverpages.org/ni2002-07-16-a.html
[2] http://www.w3.org/Signature/
[3] http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml
[4] http://www-106.ibm.com/developerworks/xml/library/x-xmlsecuritysuite/index.html
[5] http://www.w3.org/TR/xmlenc-core/
[6] http://www-106.ibm.com/developerworks/xml/library/x-seclay1/