Nemzeti Információs Infrastruktúra Fejlesztési Iroda

Grid biztonsági megoldások

Készítette:
Szalai Ferenc Attila
Sándor István
Szigeti Szabolcs

Készült az Magyar Informatikai Erőforráshalózat (grid közmű) alapjai (MEGA) NKFP OM-00263/2004 projekt keretén belül

NIIF Iroda, Budapest
2006. augusztus 30.

Tartalomjegyzék

1. fejezet

Bevezetés

Sajnálatos módon a grid rendszerek tervezésénél, fejlesztésénél a 90-es évek elején elkövették azt a hibát, hogy nem nem fogalmazták meg kellően alaposan és körültekintően a rendszer biztonsági alapjait, valamint nem alkalmaztak kellően formalizált biztonsági fejlesztési keretrendszert (pl.: CCIT). Mára nyilvánvalóvá vált, hogy a grid rendszerek számos olyan követelményt támasztanak az elosztott azonosítási, jogosultság kezelési és elszámolási rendszerekre (AAA) melyeket azok csak nagyon nehezen képesek kielégíteni. Ezeket a problémát felismerve csak az utóbbi pár évben terelődött - az általános igényekkel párhuzamosan - a figyelem a grid rendszerek biztonsági követelményeinek meghatározására és olyan rendszerek tervezésére amelyek ki is elégítik ezeket a követelményeket. Ebben a dokumentumban bemutatjuk a jelenlegi helyeztet és fejlesztési lehetőségeket valamint meghatározzuk a saját grid közművünkkel szemben támasztott biztonsági követelményeket is.

2. fejezet

Grid biztonsági követelmények

Annak érdekében, hogy meghatározhassuk a grid közművel szemben támasztott biztonsági követelményeket fel kell mérnünk milyen komponenseket szeretnék védeni és ezeket a komponenseket konkrétan milyen fenyegetettségnek vannak kitéve. Számos módszer és metodológia létezik a fenyegetések és azok mértékének meghatározására mi most elsősorban a fenyegetések jellegét szeretnénk bemutatni

2.1. Biztonsági veszélyek grid rendszerekben

Az alábbiakban kísérletet teszünk arra, hogy összegyűjtsük milyen veszélyeknek lehet kitéve grid közmű rendszerünk. A grid működésből fakadó veszélyekre koncentrálunk, mivel a grid rendszer informatikai eszközöket kapcsol össze így értelem szerűen az általános informatikai erőforráskora (szerverekre, kliensekre stb.) vonatkozó fenyegetéseket nem kívánjuk részletezni, hiszen az nem speciális grid működésből fakad, valamint számos általános összefoglaló elérhető ebben a kérdésben.

A biztonsági veszélyeket azok forrása szerint az alábbi csoportokba sorolhatjuk:

- *Fizikai*: mivel a grid rendszer fizikai eszközöket kapcsol össze így azok ugyanazoknak a veszélyeknek vannak kitéve mint minden fizikai eszköz a grid viselkedés nem ró e tekintetben rájuk speciális követelményt illetve a grid viselkedésből nem származik speciális fizikai veszély.

- *Informatikai*: konkrét informatikai rendszerek ellen irányuló szándékos vagy véletlen események következtében kialakuló veszélyhelyzetek. Ezeket általában a rendszer felépítésekor használt technológiák határozzák meg.

- *Emberi*: számos esetben elhanyagolt veszélyforrás, pedig különösen a nagy földrajzilag elosztott rendszerek, mint például a grid rendszerek esetén különösen nagy figyelmet kell fordítani erre a veszélyforrásra. A grid rendszer többnyire emberek használják és üzemeltetik akik nem tévedhetetlenek, számolunk kell a grid felhasználók és üzemeltetők szándékos vagy véletlen tevékenysége folytán felmerülő veszélyekkel.

- *Szervezeti*: mivel a grid rendszer nem működik magától (különösen a nagy földrajzi kiterjedésűek) fontos teher és veszélyforrás az a szervezeti rendszer és bizalmi, kapcsolati viszony ami a grid rendszerben lévő erőforrás szolgálatokat, grid rendszerüzemeltetőket és felhasználókat összeköti. Ezt a szervezetet is érhetik, azt kompromittáló fenyegetések.

2.1.1. Informatikai veszélyek

Az informatikai veszélyek két szinten jelentkezhetnek. Egyfelől a gridbe kapcsolt erőforrások oldalán, másfelől a grid szolgáltatásokat nyújtó köztes-réteg szintjén.

Az erőforrás általában valamilyen többfelhasználós, többfeladatos operációs rendszerrel van ellátva, így erre az operáció rendszerre a szokásos veszélyek leselkednek (illetéktelen hozzáférés, vírusok, dos stb.) a grid felhasználástól függetlenül is. A grid felhasználásból adódik néhány speciális veszély is:

V1 *kártékony kód*: a grid feladatokkal szándékosan vagy véletlenül kártékony viselkedésű kód kerülhet a rendszerünkbe (pl.: helyi root exploit, vírus stb.). Ez nem csak a grid, de a mögöttes infrastruktúra működését is veszélyezteti.

V2 *hibás kód*: a grid feladatokkal olyan kód is érkezhet ami szándékosan vagy valamilyen programozási hiba folytán indokolatlan erőforrás felhasználásba kezd (megeszi a memóriát, végtelen a futási ideje stb.)

V3 *jogosulatlan adathozzáférés*: megeshet, hogy grid feladatok jogosulatlanul férhetnek hozzá más grid feladatok illetve az erőforráson található helyi adatokhoz.

V4 *helyi grid azonosító kompromittálóik*: általánosan elmondható, hogy az erőforrásokon nem minden grid felhasználó rendelkezik egyedi azonosítóval. Viszont a feladatokat valamilyen helyi felhasználó jogosultsági szintjén lehet csak futtatni. Így számos módszer alakult ki a különféle grid rendszerekben hogy hogyan képeznek állandó vagy ideiglenes lokális felhasználókat a grid feladatok futtatására. Veszélyt jelenthet magára a grid feladatra és annak eredményére ha illetéktelen hozzáférhet ezekhez a grid feladatok futtatására fenntartott helyi azonosítókhoz.

V5 *erőforrás felhasználók hozzáférnek grid feladatok adataihoz*: grid feladatok eredményeihez a grid erőforrás tulajdonosa, rendszergazdája a legtöbb esetben hozzáférhet ezt megakadályozni meglehetősen nehéz. Alacsonyabb jogosultsági szinttel rendelkező helyi felhasználók grid feladatok adataihoz való hozzáférése azonban nemkívánatos eredményekre vezethet. Itt megjegyezzük, hogy a grid felhasználók és grid erőforrás üzemeltetők közötti bizalmi viszonyra és annak sérülésére különös figyelmet kell fordítani a grid rendszerek üzemeltetésekor, különösen ha a grid feladatok kényes adatokon végeznek feldolgozást (pl.: orvosi adatbázison, adatbányászati alkalmazás).

V6 *fekete lyuk erőforrások*: könnyen elképzelhető olyan erőforrás mely szándékosan vagy hiba miatt a valódi helyzettől eltérő információkat hirdet magáról (végtelen processzor szám, soha el nem fogyó diszk, azonnal lefutó alkalmazások stb.). Mindez azt okozhatja, hogy az ütemező ezen hamis információk alapján ütemezhet és a beütemezett feladat nem fog soha elvégződni.

A grid rendszereket általában az azokon futó köztes-réteg szoftverek teszik speciálissá. Ezek a rendszerek jelenleg X509 tanúsítványok segítségével azonosítják a grid rendszer szereplőit (erőforrás, felhasználók, feladatok stb.). Ezek a szereplők csoportokba virtuális szervezetekbe rendeződnek.

V7 *jogosulatlan szolgáltatás hozzáférés*: az egyes szereplők, szándékosan, vagy véletlenül olyan szolgáltatásokhoz is hozzáférnek, amikre nem lenne joguk. A szolgáltatások, vagy a jogosultság kezelési rendszer hibás működésé vagy hibás beállítása okozza ezt a veszélyt.

V8 *jogosulatlan tanúsítvány használat*: elképzelhető, hogy a felhasználók gondatlanságból vagy szándékosan odaadják tanúsítványaikat arra jogosulatlanoknak.

V9 *szolgáltatás, vagy feladat olyan jogosultságot szerez mely nem lett ráruházva*

V10 *hozzáférés szabályozási szolgáltatás kompromittálóik*, ez az egész hozzáférési rendszer összeomlását okozhatja

V11 *jogosulatlan VO tagság*: egy résztvevő nem kívánatos VO tagságot szerez például a VO szolgáltatás hibás működése, vagy annak üzemeltetőjének gondatlansága miatt.

V12 *megszemélyesítés*: számos esetben szükség lehet proxy jellegű szolgáltatásokra (pl.: grid szintű ütemező). Ilyen szolgáltatások esetén általában a felhasználónak bizonyos jogokat át kell ruházni, a szolgáltatásra, hogy annak nevében járjon el az. Ilyen esetekben előfordulhat, hogy a nem valódi vagy nem jogosult felhasználótól ékező kéréseket, műveleteket jogosult felhasználó nevében hajt végre a proxy jellegű szolgáltatás.

2.2. Emberi veszélyek

V13 *tanúsítvány kompromittálódás*: a felhasználó vagy grid szolgáltató tanúsítványát illetéktelen megszerzi, vagy azt szándékosan vagy véletlenül átruházza illetéktelen más felhasználóra.

V14 *tanúsítvány hatóság kompromittálódása*: szükségtelen hangsúlyozni, hogy mekkora veszéllyel járhat a tanúsítvány hatóság kompromittálódása. Itt jegyezzük meg, hogy számos esetben maga a felhasználó is lehet ideiglenesen tanúsító hatóság (pl.: a saját feladatai számára kibocsátott tanúsítványok számára).

2.2.1. Szervezeti veszélyek

V15 *grid szolgáltatások üzemeltetői kompromittálódnak*

V16 *kommunikációs csatorna sérül*: grid szolgáltatások üzemeltetői közötti kommunikációs csatorna kompromittálóik vagy nem működik.

2.3. Kockázatelemzés

Ebben a fejezetben sorra vesszük az előzőekben bemutatott fenyegetések, veszélyek kockázatait és esetleges bekövetkezésük esetén az okozott károkat. A bekövetkezés valószínűségének megítélésere az alábbi ötfokozatú skálát használjuk:

- *kivételes*: A bekövetkezés csak kivételes esetkben fordul elő. Ilyen lehet például természeti katasztrófa, vagy igegen állam által támogatott hacker támadás.
- *ritka*: A bekövetkezés csak ritkán fordul elő. Ilyen például egy célzott támadás felkészült hackerek által, tűz, jelentős áramszünet.
- *átlagos*: A veszély megvalósulása nem rendkívüli, átlagosan előfordulhat. Ilyen például hardver hiba vagy célzott hacker támadás.
- *gyakori*: A bekövetkezés gyakori, ilyen lehet például a vírusfertőzés vagy a nem célzott hacker támadás.
- *folyamatosan fennálló*: A veszély folyamatosan létezik, bármikor előfordulhat. Ide tartoznak a felhasználói hibák, az automatizált kártékony kódok (férgek, botnetek).

Az okozott kár mértékét (a megvalósuló veszély hatását a rendszerre és aműködtető valamint felhasználó szervezetekre nézve) pedig az alábbi kategóriákba soroljuk:

- *elhanyagolható*: a veszély bekövetkezte nem okoz pénzügyi veszteséget, a szolgáltatási képességek olyan mértékben és időtartamban csökkennek, hogy ez a normál működés számára nem észlelhető illetve további következmények nélkül elviselhető. A rendszer alkotóelemeinek károsodása elhanyagolható. Személyes adatok, üzleti és egyéb bizalmas információk sérülése nem történik.
- *kicsi*: a veszély bekövetkezte minimális pénzügyi veszteséget okoz, a szolgáltatási képességek olyan mértékben és időtartamban csökkennek, hogy ez a normál működés számára észlelhető, de jelentős következmények nélkül elviselhető. A rendszer alkotóelemeinek károsodása gyorsan javítható. Személyes adatok, üzleti és egyéb bizalmas információk sérülése megfelelő ellenintézkedések hiányában valószínűsíthető.
- *közepes*: a veszély bekövetkezése komoly hatásokkal jár. Jelentős pénzügyi veszteséget okoz. A szolgáltatási képesség olyan mértékben és időtartamra csökken, hogy az elsődleges funkciók elvégzésének hatásossága jelentősen csökken. Az eredeti helyzet helyreállítása a határidők betartását veszélyezteti. Személyes adatok, üzleti és más bizalmas információk védelme azonnali ellenintézkedése hiányában sérülhet.
- *jelentős*: a veszély bekövetkezése jelentős hatásokkal jár. Hosszabb távon is jelentős pénzügyi veszteséget okoz. A szolgáltatási képesség olyan mértékben és időtartamra csökken, hogy az elsődleges funkciók elvégzése időlegesen lehetetlenné válik. Az eredeti helyzet helyreállítása miatt a határidők betartása lehetelen. Személyes adatok, üzleti és más bizalmas információk védelme súlyosan sérül.

- *katapsztrofális*: a veszély bekövetkezése végzetes hatással jár. Olyan pénzügyi vagy más veszteséget okoz, amely a működtető szervezetre nézve végzetes vagy hosszútávú hatással bír, az elsődleges funkcióját nem képes betölteni, az eredeti helyzet helyreállítása nem lehetséges. A rendszer lényegileg károsodik. Személyes adatok, üzleti és egyéb információk védelme súlyosan és jóvátehetetlenül sérül.

A kockázat a *bekövetkezés valószínűsége* és az *okozott kár* alapján becsülhető és a szükséges ellenintézkedések rangsorolhatóak:

valószínűség - kár	katasztrofális	jelentős	közepes	kicsi	elhanyagolható
ritka	jelentős	magas	alacsony	minimális	minimális
átlagos	kritikus	jelentős	alacsony	minimális	minimális
gyakori	kritikus	jelentős	magas	alacsony	minimális
folyamatos	kritikus	kritikus	jelentős	magas	minimális

V1 Mivel a grid rendszerbe kerülő kódok jelenleg nem esnek át alapos minőségellenőrzésen (ez általában jellemző az informatikai termékekre), így ennek a fenyegetésnek a kockázatát *átlagosnak* minősítjük. Amennyiben a futtató erőforrások dedikált, felügyelt rendszerek, úgy a védelem jól automatizálható és gyorsan elhárítható ezért ebben az esetben az okozható kár *kicsi*. Ha az asztali gépek, szórt, desktop szerű erőforrások vannak többségben a grid rendszerben, akkor a bekövetkezés valószínűsége megnő, ismert okok miatt, *folyamatosan fennállóra*, valamint az okozott kár mértéke is *jelentős*re fokozódik. Ilyen esetben tehát különös figyelmet kell fordítani az adat és futatási környezet izolációjára.

V2 Mivel a grid felhasználók, akik jelenleg rendszerint fejlesztők is általában nem professzionális program fejlesztők, különösen nem rendelkeznek több platformon egyen szilárdságú ismeretekkel, tapasztalatokkal, a hibás kód *gyakori* jelenség. Ezekből a hibás kódokból adódó károk mértéket a programfuttató környezet biztonsági szintje határozza me, de a közvetlen kár az erőforrások pazarlásában nyilvánul meg. A károkozás mértékét ennek megfelelően *közepesnek* értékeljük.

V3 Mivel nem csak a grid rendszerek, hanem az azokat futtató operációs rendszerek is rendelkeznek az adatvédelemre, szeparációra vonatkozó eszközökkel, így ennek a fenyegetésnek a mértékét *kicsire* minősítjük. Rá kell mutatnunk azonban, hogy a jelenlegi grid rendszerek nem rendelkeznek megfelelő integrációval az operációs rendszerek alacsony szintű jogosultság kezelési mechanizmusaival, így a konfigurációs hibákból adatódó jogosulatlan hozzáférés lehetősége nem elhanyagolható kockázat. A károkozás mértékét illetően egyértelmű, hogy különösen ipari környezetben az anyagi és erkölcsi kár is *jelentős* lehet.

V4 Felügyelt rendszerek esetén a bekövetkezés valószínűsége megegyezik a normál felhasználói azonosítók kompromittálódásának esélyével, ezért ezt *átlagos* szintűnek minősíthetjük. Azonban rá kell mutatnunk arra, hogy a helyi privilégizált felhasználói azonosítók sérülése a helyi rendszeren túlmutató *jelentős* veszélyt is jelent, különösen desktop rendszerek esetén, ahol az egyszerű felhasználók is gyakran privilégizált felhasználók az operációs rendszer szempontjából. Az okozott kárt *jelentősnek* értékeljük, mivel az ideiglenes futtatási eredmények legtöbbször a helyi rendszereken keletkeznek.

V5 lsd. V4

V6 Mivel a felhasználók gyors visszajelzéseket adnak az üzemeltetők felé a hiányos adatokról ill. a lassú alkalmazásokról, valamint megfelelő teszt alkalmazásokkal ezeknek az hibás erőforrásoknak a felfedezése automatizálható is (ilyen jellegű felfedezési mechanizmussal a jelenlegi rendszerek nem rendelkeznek), így *ritkának* tekintjük az előfordulásukat. Nyilvánvalóan, desktop alapú rendszerek esetén a felügyelet hiánya miatt ezeknek a hibás erőforrásoknak a száma jelentősen megnőhet. A kiesett adatfeldolgozási idő *közepes* kárt okoz.

V7 A grid köztes-réteg rendszerek készítésekor sajnos nem a biztonsági szempontok az elsődlegesek számos esetben a finom hozzáférés szabályozási mechanizmusok is teljesen hiányoznak. Ezért ennek a fenyegetésnek a mértéket *átlagosnak* minősítjük. A hibás működésből fakadó károk mértéke függ az adott szolgáltatás jellegétől:

- biztonsági szolgáltatás: jelentős.
- számítási, adattárolási erőforrásokat reprezentáló szolgáltatások esetén közepes ill. jelentős az erőforrás értékétől függően.
- információs rendszer: kicsi, mivel ebben általában nincsenek csak publikus adatok

V8 : a jelenlegi tipikus felhasználási mintázat, hogy a felhasználók a tanúsítványaikat és privát kulcsaikat állományokban tárolják valamilyen rendszeren. Ennél a felhasználási módnál *átlagos* szintű kockázatot jelent, hogy a felhasználói tanúsítványokhoz illetéktelenek hozzáfernek. Ez a kockázat jelentősen csökkenthető hardver támogatott, speciális tároló eszközökkel (pl.: smart card, usb token stb.). A keletkezett kár akkor haladja meg a *kicsi*t, amennyiben fenáll még V5 és/vagy V6.

V9 : jelenlegi grid rendszerekben *gyakran* előfordulhat ez a szituáció a jogosultsági rendszerek hiányosságai miatt. Ugyanakkor *kicsi* kárnál csak akkor keletkezhet több, ha fennáll V1 és/vagy V2.

V10 : mivel a hozzáférés szabályozási szolgáltatásokat felügyelt rendszereken üzemeltetik, ráadásul ezek a rendszerek általában nem számítási vagy adattárolási erőforrások is egyben, ezért *kicsi* az esélye, hogy azok kompromittálódjanak. Ha azonban a hozzáférés szabályozási rendszer sérül akkor *jelentős* kár keletkezhet. A kár jelentősen csökkenthető, ha a hozzáférés szabályozási rendszer adatbázisa elosztott autonóm egységekre bomlik ill. ha a el van választva a CA szolgáltatástól.

V11 : amennyiben a VO túl általános funkciókkal rendelkezik ill., a VO tagság megszűnése nincs szabályozva és/vagy automatizálva a kockázata ennek e sérülékenységnek *átlagos*. Amennyiben fennáll még V1 és/vagy V2 is, akkor a kár *kicsi*nél több is lehet.

V12 : a jelenlegi grid rendszerek hozzáférés szabályozási eljárásainak hiányosságai miatt *gyakori* a valószínűsége ilyen hiba bekövetkezésének. Ugyanakkor az okozott kár csak akkor nagyobb *kicsi* szintnél, ha fennáll V1 és/vagy V2 is.

V13 : mivel a jogosult körben a tanúsítványok könnyen beszerezhetőek, így *kicsi* az veszélye annak, hogy valaki átruházza azt. Ez a veszély tovább csökkenthető, ha a tanúsítvány közvetlen értéket képvisel a felhasználó szemében (pl.: fizet érte). Az okozott kár, akkor nagyobb *kicsi*nék, amennyiben fennáll még V1 és/vagy V2.

V14 : az erős fizikai és logikai végelem miatt, amit a különféle szabályok és előírások kényszerítenek a CA üzemeltetőkre, továbbá a CA-kat folyamatos ellenőrzési mechanizmus is terheli, ezért *alacsony* a kompromittálódásuk esélye. A CA sérülése *katasztrófális* következménnyel járhat ráadásul nem csak a grid rendszert éri ilyenkor kár.

V15 : *átlagos* kockázat, különösen, ha a szolgáltatások üzemeltetőinek nincs közvetlen anyagi felelőssége. Az okozott kár *jelentős*, mivel tömegesen fennáll V7-V12 minden.

V16 : mivel nincs indíték ezért ennek veszélye *kicsi*, különösen, ha az üzemeltetők erősen titkosított kommunikációs csatornát használnak. A kár *közepes*, mivel eloszotott és redundáns kommunikációs csatorna van.

3. fejezet

Feladat anonimitás és titkosítás alkalmazása grid rendszerekben

A grid felhasználók részéről, különösen ipari környezetben természetes igényként jelentkezik, hogy grid rendszer egyszerű és biztonságos lehetőséget biztosítson a felhasználó feladatainak elválasztására más felhasználók feladatairól. Ugyanez vonatkozik a feladatok által felhasznált és generált adatokra is. Habár ez egy természetes igény a jelenlegi grid rendszerek kevéssé foglalkoznak a problémával. Az alábbiakban röviden áttekintjük a problémát és konkrét megoldási javaslatokat próbálunk megfogalmazni.

3.1. Feladat anonimitás és szeparáció

A jelenlegi grid rendszerekben általánosan elfogadott gyakorlat, hogy a X509 tanúsítványokkal azonosított felhasználókat az abban szerepló DN alapján egy statikus ún. gridmap állományt felhasználva, helyi felhasználókra képeznek le.

Ez a módszer több sebből is vérzik:

- a gridmap állomány egy központosított és meglehetősen sérülékeny elem a rendszerben.
- a gridmap állományban szereplő lokális felhasználókat létre kell hozni, meg kell szüntetni, általában kezelni kell őket és a grid felhasználók száma lehet igen magas is, így egy átlagos, esetleg kis felhasználói bázissal rendelkező, rendszerre is nagy adminisztratív terhelést ró.
- a felhasználói szinten kell a hozzáférés szabályozást megvalósítani, nem feladat szinten. A felhasználónak adott legtágabb jogosultsági körrel rendelkezik, a felhasználó minden feladata.
- nem skálázható

Amennyiben ezekre a felmerült problémákra választ akarunk adni, világosan látunk kell milyen szinten és milyen módszerekkel valósítható meg a feladatok dinamikus elválasztása:

- *operációs rendszer szint*: ezen a szinten kétféle megoldás létezik: Mandatory Access Control (MAC) és a különféle virtualizációs megoldások.

- *felhasználó szinten*: létrehozhatunk az alkalmazások futási idejéig létező dinamikus felhasználói azonosítókat, amik a feladat jogosultsági körének megfelelő hozzáférésekkel rendelkeznek.
- *kommunikációs csatorna szinten*: a kommunikációs csatorna megfelelő titkosítása (lsd. alább)
- *grid rendszer szinten*: maga a ClusterGrid hálózati architektúra jó példa arra, hogyan lehet elvásztani a felhasználókat az erőforrásoktól és jól ellenőrzött, de elosztott belépési pontokon kontrollálni a tevékenységüket.

A fenti módszerek együttes felhasználásával erős szeparáció biztosítható a felhasználók és azok feladatai között is.

3.2. Feladat titkosítás

A grid feladatok általános esetben bemeneti és kimeneti adatokkal dolgoznak ill. hálózaton kommunikálnak a grid rendszer más elemeivel. Ennek megfelelően a titkosítással több szinten élhetünk:

- *kommunikációs csatorna* : itt kétféle megoldás képzelhető el:
  - rendszertől, rendszerig: ekkor a host gépek közötti csatorna titkosított. Jellemzően valamilyen IPSec VPN alapú megoldásokat szoktak ilyenkor alkalmazni.
  - szolgáltatástól, alkalmazásig: a host gépek közötti titkosítatlan kapcsolaton, titkosított csatornát TLS segítségével lehet kialakítani. Ez ipari környezetben is széles körben használ, jó teljesítmény mutatókkal rendelkező technológia. Fel kell hívni a figyelmet, hogy a hosszú életű TLS session-ök sebezhetőek lehetnek.
  - üzenet alapú titkosítás (message level encryption): többen felvetettek, hogy jobb biztonságot nyújt a csatorna titkosítás helyett az üzenet titkosítás. Figyelembe véve, hogy a grid rendszerek természetese működés az üzenet alapú kommunikáció és annak is aszinkron jellege a domináns, a felvetés jogos. Ugyanakkor a üzenet alapú titkosítás erőforrás igényes, kevés alkalmazói környezet támogatja és számos esetben alkalmazás specifikus megoldások vannak csak rá.
- *adat*: erős titkosítási algoritmusokkal lehet élni, ám mindegyiknek a feltétele, hogy a feladat rendelkezzen a kikódoláshoz szükséges kulccsal, amit ily módon el lehet tőle lopni.
- *algoritmus*: lehetőség van általános esetben, hogy az elvégzendő feladat teljes algoritmusát valamilyen titkosító transzformációnak vettjük alá, ekkor a transzformált algoritmus a transzformált adatokon működik és transzformált eredményt állít elő, ezt az eredményt csak a transzformáció birtokában lehet értelmes eredménnyé visszaalakítani.

3.3. Bizalmi viszony elemzése

A feladat-anonimitással és titkosítással kapcsolatos kérdések felvetnek egy általánosabb, a bizalmi viszonyokra vonatkozó problémát is. Ennek a problémának a gyökere az, hogy a gridben található bármely erőforrás gazdája teljes jogosultsággal rendelkezik a saját rendszerén, így az oda érkező mindenféle adaton is. A kérdés, hogy tudunk-e olyan általános rendszert készíteni, aminél nem kell megbíznunk az erőforrások üzemeltetőiben? A válasz, az, hogy nehezen. Jelenleg az egytelen lehetséges mód erre az előző részben említett algoritmus szintű titkosítás. Ez azonban nem általános, drága és nehezen automatizálható feladat.

Valamilyen szintű bizalomra az erőforrás üzemeltetőkkel szemben tehát szükség van. Ha figyelembe vesszük az általános grid felhasználási mintázatokat, láthatjuk, hogy számos esetben a megfelelő bizalmi szint megválasztásával az algoritmus szintű titkosítás használatának kényszere elkerülhető:

- *közösségi*: nagy közösségi kezdeményezések hatására létrejövő, többnyire önkéntes felajánláson alapuló, sok desktop erőforrást tartalmazó rendszerekben (Seti@Home, LHC@Home, Boinc projektek stb.), a feldolgozandó adat általában közkincs vagy ahhoz közel álló értéket képvisel. Itt nem aggódunk az eredmények ellopása , megsemmisítés miatt. Ezekben a rendszerekben az eredmények integritásának ellenőrzésére általában ugyanazt a műveletet többszöri végrehajtásával szűrik ki a szándékos vagy véletlen hibákat.

- *ipari*: olyan erőforrásokat integrálnak, amik többnyire az adott ipari cég vagy tárulás saját tulajdona. Az ipari konzorciumok között a hagyományos együttműködési, jogi, technikai szabályok, garanciák érvényesek.

- *szolgáltatás*: ha valaki számítási szolgáltatást akar nyújtani akkor a szokásos bizalmi viszony alakul ki szolgáltatást igénybe vevő és szolgáltató között. Bármilyen visszaélés nyilvánosságra kerülése azonnali és maradandó bizalomvesztéssel jár, melynél a várható veszteség nagyobb, mint a nyereség.

Ha feladatoknak mindenképpen kényes adatokat kell feldolgozniuk, akkor jobb modell , ha a feladat nem viszi magával az adatot, hanem közvetlenül írja, olvassa egy felügyelt adatbázisból. Ugyanis ebben az esetben, ha az erőforrás kompromittálódása nem közvetlenül a grid feladat kényes adatainak megszerzésére irányul, akkor gyorsabban és hatékonyabban lehet védekezni, mivel az adatforrás a felügyeletünk alatt áll.

4. fejezet

Grid rendszerekben alkalmazott delegációs technikák

Napjaink grid rendszerei újszerű követelményeket támasztanak a jogosultságkezelő rendszerekkel szemben. A grid környezetben több olyan probléma is felmerül, amelyek a hagyományos kliens-szerver modell alapján működő rendszerekben általában nem. A problémák forrása leginkább az hogy ilyen méretű elosztott rendszerekben sem a jogosultságok nyilvántartása, sem a felhasználók, szolgáltatások és feladatok azonosítása nem központosítható.

Az egyik ilyen probléma a jogosultságok átruházása, más néven delegálás. A felhasználók által indított programoknak (joboknak) ugyanis maguknak is szükségük van jogosultságokra, hogy feladataikat elvégezhessék. A felhasználók által indított programok esetében leginkább fájl olvasási és írási jogosultságokról van szó. Ha ezek a programok egy olyan számítógépen futnak, amelyen a felhasználó rendelkezik saját felhasználói fiókkal (account), akkor a szokásos megoldás szerint a felhasználó által indított folyamat „örökli" a felhasználó jogosultságait, tehát minden olyan fájlhoz hozzáférhet, amihez a felhasználó maga is. A felhasználó folyamatai tehát az ő „nevében" futnak, a felhasználó *minden jogosultságát átruházza* az általa futtatott folyamatra. Ez a mechanizmus (kiegészítve néhány „kiskapuval": setuid, setgid) kielégítő megoldást biztosít egy számítógép esetére, illetve arra az esetre, ha a felhasználók csak olyan gépeken kívánnak programokat futtatni, amelyeken rendelkeznek fiókkal.

Grid rendszerekben azonban nem ez a helyzet. A gridben bármely feladat (job) elméletileg a rendszer bármely gépén futhat, és ilyen gépekből sok ($10^2 - 10^7$ db) lehet. Lehetetlenség megkövetelni, hogy a grid minden felhasználójának minden gépen legyen fiókja. Szintén nem jó az a széles körben alkalmazott megoldás, hogy minden globális felhasználói névhez rendelünk egy lokális felhasználói nevet (grid-mapfile), mivel a gépek és felhasználók számának növekedésével a rendszert egyre nehezebb karbantartani.

A gridben is célszerű lenne tehát azt megvalósítani, hogy a felhasználó által indított feladatok az ő „nevében" fussanak, tehát hogy a felhasználó átruházhasson bizonyos jogosultságokat a sajátjai közül ezekre a feladatokra, vagy akár más felhasználókra is. Ezt valamilyen globális, az egyes gépek helyi jogosultságkezelésétől független mechanizmussal kell biztosítani.

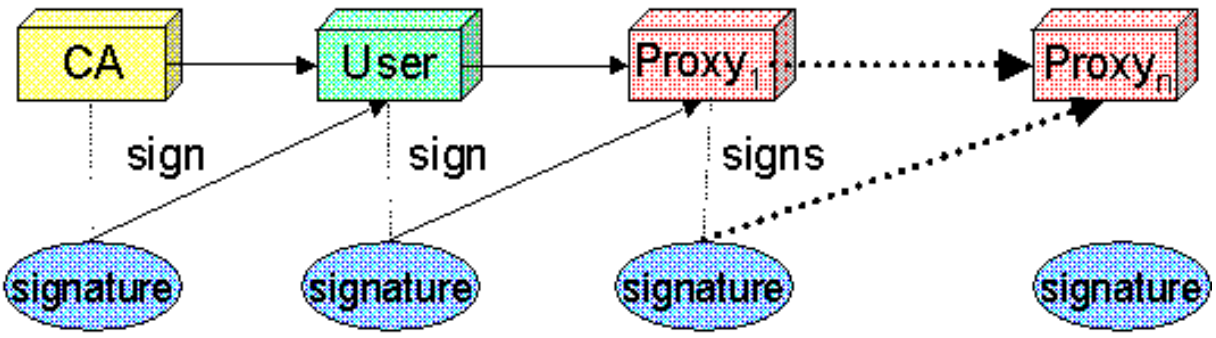
4.1. Proxy tanúsítványokon alapuló jogosultság átruházás

A felhasználók azonosításának egyik módja az X.509 tanúsítványok (certificate) használata. Az X.509 tanúsítványok a hagyományos felhasználónév/jelszó alapú azonosítással szemben több előnnyel is rendelkeznek, ezért alkalmazásuk széles körben elterjedt. Az eredeti elgondolás szerint a tanúsítványok egy RSA publikus kulcs és egy név között létesítenek hiteles (megbízható entitások, úgynevezett CA-k (Certificate Authority) által tanúsított) kapcsolatot. A tanúsítványok használatára több népszerű, biztonságos kommunikációt lehetővé tevő protokoll épül, például az SSL (v2/v3) és TLS [12].

A tanúsítványok felépítéséről és használatáról részletesen olvashatunk az Interneten [11], a további tárgyalás miatt itt csak annyit jegyeznék meg, hogy minden tanúsítvány tartalmaz *Subject* mezőt, ami a tanúsítvány tárgyát (a tanúsítvány tulajdonosának nevét, ez lehet URL vagy bármilyen más azonosító is) tartalmazza. Ezen kívül a tanúsítvány tartalmazza a *Subject* nyilvános kulcsát és az érvényesség idejét is, illetve egyéb mezőket. Így tehát a tanúsítvány a benne foglalt publikus kulcsot köti a *Subject* mezőben található névhez, a megadott időtartam erejéig. A tanúsítvány tulajdonosa a nyilvános kulcshoz tartozó titkos kulcs birtokában hitelesen igazolni tudja magát.

4.1.1. GSI proxy tanúsítványok

A Globus GSI-ben (Globus Grid Security Infrastructure) a delegálást proxy tanúsítványokkal oldják meg. A proxy tanúsítványok abban különböznek a hagyományos tanúsítványoktól, hogy aláírójuk nem a CA hatóság, hanem bármely felhasználó lehet. A proxy tanúsítvánnyal aztán további tanúsítványokat lehet létrehozni hasonló módon és így tovább. A proxy tanúsítvány hitelességét a másik fél úgy ellenőrzi, hogy először megvizsgálja a felhasználó aláírásának hitelességét, majd ellenőrzi, hogy a felhasználó nyilvános kulcsát tényleg a CA állította-e ki. A proxy tanúsítványok tehát csak akkor érvényesek, ha a hitelességi lánc végig követhető egészen a CA-ig. A proxy tanúsítványok működését szemlélteti a 4.1.1. ábra. A proxy tanúsítványok használata nem csak a grides alkalmazásokkal kapcsolatban merült fel, és mára a szabvány részévé váltak [7].



4.1. ábra. A proxy tanúsítványok működése

A GSI-ben, ha a felhasználó jobot (feladatot) indít a griden, kiállít a job számára egy proxy

tanúsítványt, amelynek segítségével a job az ő nevében járhat el. A proxy tanúsítvány *Subject* mezője ugyanazt a (DN - Distinguished Name) nevet tartalmazza, mint a felhasználó saját tanúsítványa, amelyet a grid CA-ja állított ki. Így a felhasználó feladatai ugyanazokkal a jogosultságokkal rendelkeznek, mint a felhasználó maga. A különbség a felhasználó saját tanúsítványa és a proxy tanúsítvány között az, hogy mivel az előbbit a felhasználó maga használja, a hozzá tartozó titkos kulcs jelszóval védett, míg a job által használt proxy tanúsítványhoz tartozó titkos kulcs titkosítatlan. A proxy tanúsítvány titkos kulcsát nem volna értelme jelszóval védeni, mivel akkor a jelszót kellene titkosítatlanul tárolni és így tovább. A proxy tanúsítvány tehát alkalmas a felhasználó megszemélyesítésére (impersonation), a felhasználó nevében történő eljárásra, hiszen ez a célja. A proxy tanúsítvány nyilvános, a hozzá tartozó titkos kulcsot pedig legjobb esetben is csak fájlrendszer szintű hozzáférési jogosultságok védik. A rosszindulatú támadó tehát viszonylag könnyen átveheti a felhasználó *összes* létező jogosultságát.

Ezt a jelentős biztonsági problémát a GSI-ben úgy próbálják orvosolni, hogy korlátozzák a proxy tanúsítványok érvényességi idejét, általában a kiállítástól számított 12 órára. Azonban sajnos semmilyen garancia nincs arra, hogy az adott feladat 12 órán belül befejeződik, sőt gyakran kívánatos lenne ennél hosszabb futási idejű feladatokat is indítani. Amint a proxy tanúsítvány érvényét veszti, a feladat működésképtelenné válik, mivel nem fér többé hozzá a grid erőforrásaihoz. Ha meg akarjuk hosszabbítani a működési időt, újabb proxy tanúsítványt kell kiállítani, ehhez pedig a felhasználó aktív közreműködése szükséges (be kell ütnie a jelszavát). Ha automatikusan hosszabbítjuk meg a proxy tanúsítványt (pl. egy proxy tanúsítvány kiállító szerverrel), akkor az időkorlátozás értelmét veszti, és ismét az előbbi helyzettel állunk szemben, amikor is a rosszindulatú támadó idővel a grid bármely felhasználóját képes lesz megszemélyesíteni.

Fejlesztések

A koncepció továbbfejlesztése jelenleg a részleges jogosultság átruházás irányába folyik. Az előbbi megoldásnál a proxy tanúsítvány a kiállító teljes nevét tartalmazta, így jogot adott az ő megszemélyesítésére. Vannak olyan megoldások, melyeknél a proxy tanúsítvány nem nevet, hanem tetszőleges információkat, ún. attribútumokat tartalmaz. Attribútum lehet például: munkahely neve (pl. BME IIT tanszék), beosztás, becenév, stb. tulajdonképpen tetszőleges olyan információ, ami alapján jogosultsági döntést lehet hozni. Az ilyen tanúsítványokat attribútum tanúsítványoknak neveik (Attribute Certificates). Így egyetlen tanúsítvány csak a felhasználó egy „részének" megszemélyesítésére elegendő.

Ez a megoldás annyiban jobb az előzőnél, hogy finomabb jogosultságkezelést tesz lehetővé, azonban a proxy tanúsítványok alapvető problémáját, az érvényesség problémáját nem oldja meg. Elegendően hosszan futó feladatok esetében idővel a támadó az előbb említett módon összegyűjtheti az attribútumok tetszőleges halmazát, és így bármely erőforráshoz hozzáférhet. Egy proxy tanúsítvány vagy korlátozottan érvényes, és akkor a felhasználónak folyamatosan hosszabbítgatnia kell azt, vagy korlátlanul érvényes és akkor jelentős biztonsági kockázatot jelent.

4.1.2. Megjegyzések

Single Sign On

A proxy tanúsítványok jól használhatók viszont Single Sign On (egyszeri bejelentkezés) megvalósítására. Itt arról van szó, hogy a felhasználónak ne kelljen minden egyes olyan művelet végrehajtásakor begépelnie a jelszavát, amikor a titkos kulcsára szükség van. A megoldás egy proxy tanúsítvány létrehozása korlátozott (pl. 1/2 óra) időtartammal. Mivel ilyenkor a proxy tanúsítványra csak addig van szükség, amíg a felhasználó műveleteket hajt végre, a korlátozott időtartam nem jelent problémát.

Egyéb problémák

A proxy tanúsítványok és attribútum tanúsítványok széles körű használatát tulajdonképpen a visszavonás nehézségei korlátozzák. Emellett az attribútumok használatával is problémák vannak. Ahhoz, hogy attribútumok alapján hozhassunk jogosultsági döntéseket, a szolgáltatónak és a tanúsítvány kibocsátónak egységes attribútumokat kell használniuk. Ez utóbbi problémára például a Shibboleth-ben alkalmazott Federated Trust modell jelent megoldást, ahol a szolgáltatók és tanúsítvány kibocsátók szövetségeket (federation) alkotva megállapodnak a használandó attribútumokról. Attribútumok használatával azonban ésszerű korlátokon belül nem lehet fájl szintű hozzáférés szabályozást megvalósítani.

5. fejezet

Grid rendszerekben alkalmazott autorizációs módszerek

5.1. Autentikáció és autorizáció

A hagyományos egygépes rendszerekben az autentikáció (azonosítás) és autorizáció (jogosultság meglétének bizonyítása) nem különül el élesen egymástól. Miután a felhasználó azonosította magát - megadta a jelszavát vagy bemutatta a megfelelő tanúsítványt - a rendszerben nyilvántartott jogosultságainak megfelelően hozzáférhet (vagy nem férhet hozzá) a kívánt erőforrásokhoz. A jogosultságok a felhasználó nevéhez vannak kötve, és a rendszer minden hozzáférés engedélyezése előtt megvizsgálja, hogy az adott névhez tartozik-e megfelelő jogosultság.

Az autorizáció és autentikáció elkülönülését jól szemlélteti egy a valós életből vett példa. Amikor például szeretnénk megnézni egy mozielőadást, nem az történik, hogy a pénztárnál befizetjük a jegy árát és ott felírják a nevünket egy listára (jogosultság felvétele az adatbázisba), majd a terem bejáratánál a személyazonosságunkat igazolva (autentikáció) ellenőrzik, hogy rajta vagyunk-e a jogosultak listáján, hanem a pénzünkért cserébe kapunk egy jegyet, ami feljogosít minket az adott film megtekintésére (autorizáció). A terem bejáratánál csak a jegyet kell felmutatnunk, nem kell azonosítanunk magunkat. Az első változat a fentebb vázolt hagyományos működést mintázza, az utóbbi változat a tisztán autorizáción alapuló hozzáférés engedélyezés.

Megjegyzendő, hogy külön autorizációs lépésről akkor érdemes beszélni, ha a jogosultság megléte valamilyen tárgyi vagy szoftver eszközzel (jegy, bankkártya, X.509 tanúsítvány, SmartCard stb.) igazolható anélkül, hogy ez a felhasználót azonosítaná. Bizonyos esetekben tehát célszerű elkülöníteni egymástól az autentikációt és az autorizációt, más esetekben - pl. egyetlen gép esetében - nincs. A tisztán autorizáción alapuló hozzáférés kezelésnek nyilvánvaló előnye, hogy a jogosultság igazolásához a felhasználónak nem kell felfednie a személyazonosságát, ami elméletileg nagyobb biztonságot nyújt számára.

5.2. SPKI

A fentihez hasonló kérdések a grides alkalmazásoktól függetlenül is felmerültek, és születtek javaslatok ezek megoldására. Az egyik ilyen volt az SPKI - Simple Public Key Infrastructure [1], ami a PKI továbbfejlesztéseként annak számos problémájára próbált megoldást keresni. A legfontosabb fejlesztések:

- a globálisan egyedi DN (Distinguished Name) nevek koncepciójának elvetése
- autentikáció helyett autorizáció, attribútum tanúsítványok
- jogosultságok átruházása (delegálás)
- kísérlet a tanúsítványok visszavonásával kapcsolatos problémák megoldására

Nevek

Az PKI tervezői abból a feltételezésből indultak ki, hogy mindenkinek lesz egy globálisan egyedi DN (Distinguished Name) neve és egy nyilvános kulcsa, ami felhasználható az illető azonosítására. A PKI tanúsítványok a DN nevek és nyilvános kulcsok között teremtettek hitelesített kapcsolatot. Az elképzelés az volt, hogy hasonlóan a telefonkönyvhöz, ahol minden név és cím mellé meg van adva a telefonszám, elő lehet majd állítani a globálisan egyedi DN nevek jegyzékét is, ahol minden név mellett szerepel majd egy nyilvános kulcs is. Ez az elképzelés több szempontból sem reális, ezért az SPKI lokális neveket használ, amelyek egy adott nyilvános kulcsra vonatkoztatva tehetők globálissá. A lokális nevek nem kapcsolódnak szorosan az autorizáció kérdéséhez, bővebben az RFC2693-ban [1] olvashatunk róluk.

Autentikáció helyett autorizáció

A másik hallgatólagos feltételezés az volt, hogy a szolgáltatók majd képesek lesznek a tanúsítványokban szereplő DN nevek alapján hozzáférési jogosultságokat meghatározni („ha tudjuk egy illető nevét, akkor tudjuk azt is, hogy kicsoda"). Ez azonban csak akkor igaz, ha az adott szolgáltató a hozzáférést kérő felhasználóról rendelkezik nyilvántartással. Minél szélesebb körben akar tehát a szolgáltató a szolgáltatásához PKI alapú hozzáférést biztosítani, annál nagyobb lesz a rá nehezedő adminisztrációs teher. Az SPKI háromféle tanúsítványt definiál:

1. (hagyományos PKI) név tanúsítvány: `<név, kulcs>`
2. autorizációs tanúsítvány: `<autorizáció, kulcs>`
3. attribútum tanúsítvány: `<név, autorizáció>`

„autorizáció" itt bármilyen, a tanúsítványban feltüntethető, jogosultsági döntés meghozására alkalmas információ lehet. Megjegyzendő, hogy a legtöbb alkalmazásban *attribútum tanúsítvány* alatt a 2. pont szerinti, itt *autorizációs tanúsítványnak* nevezett tanúsítványt értik, a továbbiakban attribútum tanúsítvány kifejezést én is a 2. pont szerinti tanúsítványra használom. Ennek oka, hogy a tanúsítványban található autorizációs információ általában valamilyen attribútum.

Az *attribútum tanúsítvány* tehát egy attribútum és egy nyilvános kulcs között teremt kapcsolatot. Ezek a tanúsítványok alkalmasak a nyilvános kulcsú titkosításon alapuló kommunikációs protokollokban történő használatra. Továbbá lehetővé teszik, hogy a szolgáltatók anélkül hozzanak jogosultsági döntéseket, hogy a felhasználókról nyilvántartást kelljen vezetniük. Ez a felhasználók adatainak védelme szempontjából is előnyös.

Jogosultságok átruházása

Az attribútum (autorizációs) tanúsítványok használatának előnye, hogy könnyen megoldható a jogosultságok átruházása. A tanúsítvány tulajdonosa ugyanis egyszerűen készíthet proxy tanúsítványt [7] az eredetiről anélkül, hogy az eredeti tanúsítvány kibocsátójával kapcsolatba kellene lépnie.

Tanúsítványok visszavonása

A PKI tanúsítványok legnagyobb problémája a visszavonhatóság kérdése. Előfordulhat ugyanis, hogy egy tanúsítványt még az érvényesség (a tanúsítványban szereplő időtartam) lejárta előtt visszavon a kibocsátó hatóság. A hatóságnak semmilyen eszköze nincs a kint lévő tanúsítvány tényleges bevonására, ezért ún. CRL-eket (Certificate Revocation List) bocsát ki, amelyek felsorolják az időközben érvénytelenített tanúsítványokat. Ahhoz, hogy egy szolgáltató hitelesnek fogadhasson el egy tanúsítványt, szüksége van az összes addig visszavont tanúsítvány listájára (az összes CRL-re). A szolgáltató biztonsági igényeitől függően rendszeres időközönként (naponta, óránként, percenként stb.) frissítenie kell a CRL-eket. A CRL-ek elég hosszú időre visszamenőlegesen is tárolni kell, ez jelentős kapacitást vesz igénybe. Ezen a problémán az SPKI sem tudott javítani.

SPKI implementációk

Az SPKI nagyrészt megmaradt elmélet szintjén, bár készültek implementációk [2], ezek használata nem terjedt el. Az SPKI ötletei azonban hatással voltak más rendszerekre, különösen az „autentikáció helyett autorizáció", továbbá a jogosultságok átruházásának elvét alkalmazzák széles körben.

5.3. Shibboleth

A Shibboleth [3] elosztott jogosultságkezelő rendszer az Internet2 kezdeményezés része. A rendszer egyrészt a felhasználók, másrészt az erőforrás szolgáltatók (tartalomszolgáltatók, email, tárhely szolgáltatás stb.) dolgát hivatott megkönnyíteni. Hagyományos esetben a felhasználónak minden olyan helyen regisztrálnia kell, ahol valamilyen védett szolgáltatást kíván igénybe venni. Ez azt jelenti, hogy minden ilyen helyen meg kell adnia néhány személyes adatát és egy jelszót. A személyes adatok kiadása biztonsági kockázatot jelenthet a felhasználó számára, a sok jelszó megjegyzése pedig problémát okoz, és végeredményben csökkenti a jelszavas védelem hatékonyságát (mindenki egyszerűen megjegyezhető jelszavakat választ). Másrészt a szolgáltatóknak

is problémát jelent a felhasználók nyilvántartása. Ezeket a problémákat kívánja megoldani a Shibboleth rendszer.

5.3.1. A Shibboleth működése

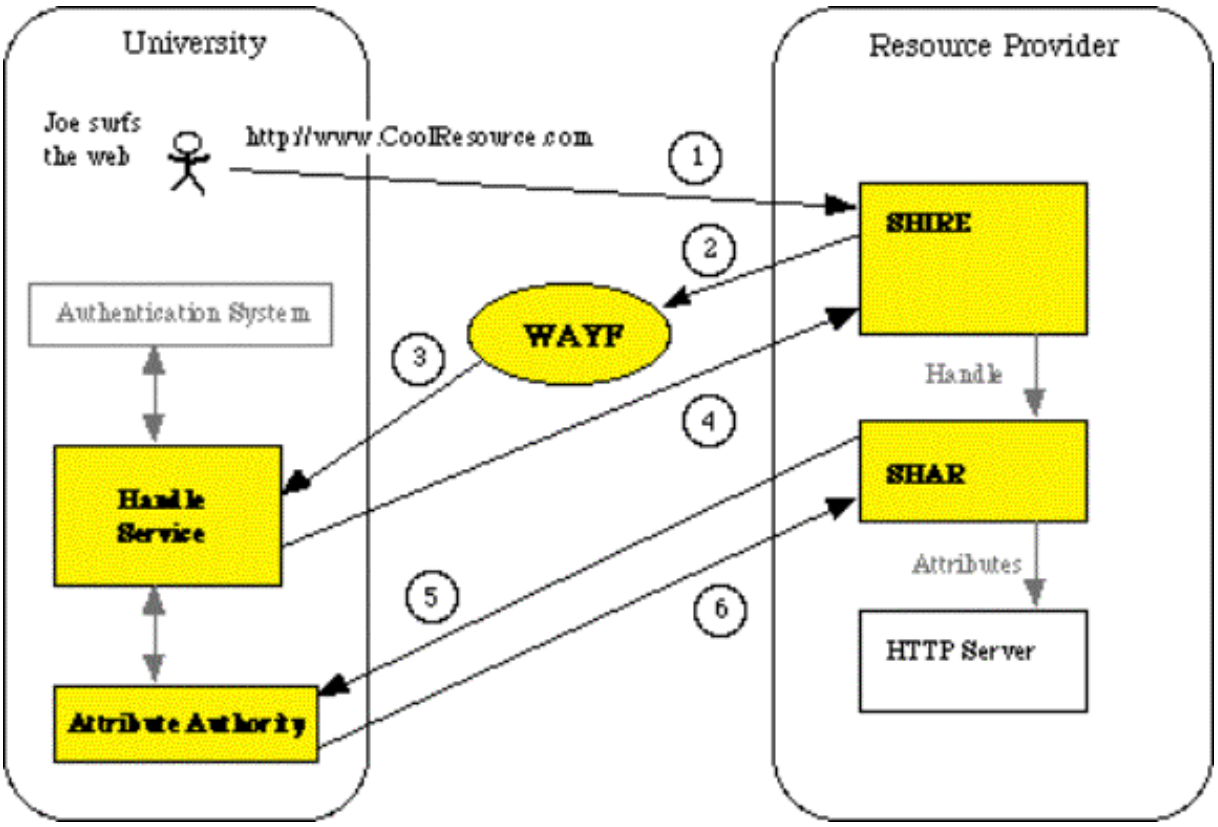
A Shibboleth rendszer feltételezi, hogy a felhasználó a védett tartalmakhoz webböngésző használatával fér hozzá. A Shibbolethben minden felhasználónak van ún. *származási helye*, ahol az adatai (attribútumai) nyilván vannak tartva. Vannak továbbá *erőforrás szolgáltatók*, akik szeretnének különböző korlátozások mellett hozzáférést biztosítani bizonyos tartalmakhoz. Az erőforrás szolgáltatók nem vezetnek nyilvántartást a felhasználóktól, hanem megbízva a származási helyek csoportjában, a tőlük kapott attribútumok alapján hoznak jogosultsági döntéseket. A Shibboleth feladata ebben a környezetben az, hogy biztonságosan eljuttassa a megfelelő attribútumokat az erőforrás szolgáltatókhoz, amikor azokra szükség van.

A rendszer felépítése [4]

Minden származási helyen van legalább egy attribútum hatóság (továbbiakban AA - Attribute Authority), amely felelős a felhasználók attribútumainak kiadásáért. A felhasználók biztonságát tovább növeli, hogy az AA-nál - előnyösen webes interfészen keresztül - beállíthatják, mely szolgáltatók milyen adatokhoz férhetnek hozzá. A szolgáltatóknál található a SHAR (Shibboleth Attribute Requester) komponens, amely a megfelelő AA-tól beszerzi a szükséges attribútumokat. Az adott felhasználóhoz megfelelő AA megállapítására a Shibboleth egy összetett névfeloldási mechanizmust használ. A SHAR a SHIRE (Shibboleth Indexical Reference Establisher) komponenshez fordul. A SHIRE megkéri a WAYF (Where Are You From) szolgáltatást, hogy derítse ki számára a felhasználó származási helyét. A WAYF először megkérdezi a felhasználót, hogy honnan jött (webböngésző átirányítása a WAYF-hez, felhasználó begépeli a származási helyet pl. BME IIT), majd kapcsolatba lép az adott származási hely HS (Handle Service) komponensével. A HS egy ideiglenes azonosítót generál a felhasználó számára, és megadja ezt az ideiglenes azonosítót, valamint a megfelelő AA címét a SHIRE-nak, az pedig továbbadja a SHAR-nak. Innen kezdve az AA és SHAR kicserélik egymás között a kívánt attribútumokat (amennyiben a felhasználó ezt az AA-nál megengedte). Az attribútum csere SOAP alapú SAML (Security Assertions Markup Language) üzenetekkel történik. A folyamat során az adott felhasználóra végig az ideiglenes azonosítóval hivatkoznak, így a szolgáltató előtt titok marad a felhasználó valódi személyazonossága. A névfeloldási mechanizmus azért ilyen összetett, hogy a lehető legkevesebb információt kelljen kiadni a felhasználóról a szolgáltatónak. A rendszer felépítése látható az 5.3.1. ábrán.

Attribútumok

A kapott információk alapján a szolgáltatónak el kell tudnia dönteni a hozzáférési kérelem jogosságát. Ez azt jelenti, hogy a szolgáltatónak (ill. a szolgáltatók egy csoportjának) és a származási helynek (ill. a származási helyek egy csoportjának) előzetesen meg kell állapodniuk az attribútumok halmazáról annak érdekében, hogy a szolgáltatók az AA-tól kapott információkat egyáltalán értelmezni tudják. A Shibboleth használatához a szolgáltatóknak és származási helyeknek



5.1. ábra. A Shibboleth működése

(egyetemek, hivatalok stb.) ún. szövetségekbe (federation) kell tömörülniük. A szövetség tagjai egymásban megbíznak, és megállapodnak egy közös attribútum-készletben. Ez az ún. Federated Trust modell. Az attribútumokra jó példa a Shibboleth-tel ismerkedők számára létrehozott InQueue szövetség [5], az itt definiált attribútumok többek között:

```
* urn:mace:dir:attribute-def:cn
* urn:mace:dir:attribute-def:sn
* urn:mace:dir:attribute-def:telephoneNumber
* urn:mace:dir:attribute-def:title
* urn:mace:dir:attribute-def:initials
* urn:mace:dir:attribute-def:description
* urn:mace:dir:attribute-def:carLicense
* urn:mace:dir:attribute-def:departmentNumber
* urn:mace:dir:attribute-def:displayName
* urn:mace:dir:attribute-def:employeeNumber
* urn:mace:dir:attribute-def:employeeType
* urn:mace:dir:attribute-def:preferredLanguage
* urn:mace:dir:attribute-def:manager
```

```
* urn:mace:dir:attribute-def:roomNumber
* urn:mace:dir:attribute-def:seeAlso
* urn:mace:dir:attribute-def:facsimileTelephoneNumber
* urn:mace:dir:attribute-def:street
* urn:mace:dir:attribute-def:postOfficeBox
* urn:mace:dir:attribute-def:postalCode
* urn:mace:dir:attribute-def:st
* urn:mace:dir:attribute-def:givenName
* urn:mace:dir:attribute-def:l
* urn:mace:dir:attribute-def:businessCategory
* urn:mace:dir:attribute-def:ou
* urn:mace:dir:attribute-def:physicalDeliveryOfficeName
```

Shibboleth és GSI

Történtek próbálkozások a Shibboleth rendszer gridben történő alkalmazására, főként a Globus projekt keretében [6]. Mivel a Globus saját biztonsági rendszere, a GSI amúgy is az attribútum tanúsítványok és az attribútum alapú hozzáférés-szabályozás irányába mozdul el, kézenfekvőnek tűnik a Shibboleth-tel történő összekapcsolása. A jelenlegi elképzelés azonban jelentős megkötéseket tartalmaz:

- a Shibboleth egyik előnyének számító névfeloldási mechanizmust elhagyták, ami kézenfekvő döntés, mivel a folyamat a felhasználó közreműködését igényli
- ennek eredményeképpen a grid szolgáltatásokba „be van drótozva" az AA fellelhetősége, tehát ez a szolgáltatás „kvázi" központosított, mivel az AA-k hozzá vannak rendelve a grid szolgáltatásaihoz
- az AA a proxy tanúsítványokban található globálisan egyedi DN neveket a saját nyilvántartásában szereplő nevekre képezi le, ami szintén arra mutat, hogy az AA központosított

Jelenleg úgy tűnik a Shibboleth-ből csak a SAML alapú attribútum üzenetek maradtak meg, a fenti problémák kiküszöbölésére jelenleg is folyik a fejlesztés.

5.3.2. A Shibboleth alkalmazhatósága grid rendszerekben

A Shibboleth gridben történő alkalmazhatóságának legnagyobb korlátját nyilvánvalóan az jelenti, hogy a rendszert webes alkalmazásokhoz tervezték. A Shibboleth legnagyobb előnye, az elosztott nyilvántartási rendszer és a kifinomult névfeloldási mechanizmus a felhasználó közreműködését igényli, ami a web böngészése közben nem jelent problémát, a gridben viszont pont azt szeretnénk elérni, hogy a jogosultsági kérdések eldöntéséhez ne legyen szükség a felhasználó beavatkozására. Konkrétan a WAYF (Where Are You From?) kérdésnél szükséges a felhasználó beléptetése. Ezt a problémát még át lehetne hidalni például úgy, hogy a gridben minden job, szolgáltatás magával hordozná a származási helyének a címét.

Attribútum alapú vagy erőforrás szintű hozzáférés szabályozás

Még nagyobb probléma azonban az attribútum alapú hozzáférés szabályozás. Az attribútumok kiválóan alkalmasak webes tartalmak, e-mail fiókok hozzáférési jogosultságainak meghatározására. Egy adott oldalt például csak egy adott szervezet munkatársai tekinthetnek meg. Egy e-mail fiókhoz csak az adott e-mail címmel rendelkező tulajdonos férhet hozzá. A munkahely, e-mail cím stb. mind a felhasználó adatai, az ő attribútumai. A gridben viszont *erőforrás szintű* hozzáférés szabályozásra van szükség, konkrétan például fájl szintű hozzáférés szabályozásra. Azt kell eldönteni adott esetben, hogy egy folyamat írhat/olvashat-e egy adott fájlt, nem pedig azt, hogy egy felhasználó hozzáférhet-e adott információkhoz. Adott esetben elképzelhető lenne, hogy minden egyes fájl hozzáférési jogosultság a felhasználó egy-egy attribútuma legyen. Ez azonban egyrészt nagyon megnövelné az attribútumok számát, másrészt a fájlrendszer változásainak követését is szinte lehetetlenné tenné. Az egész attribútum-rendszer kezelhetetlenné válna.

A Shibboleth hasznos tulajdonságai

A Shibboleth leghasznosabb tulajdonsága, hogy lehetővé teszi a jogosultságok elosztott adminisztrálását. Egy hasonló megoldás alkalmas lehet fájl hozzáférési jogosultságok tárolására is.

5.4. Globus GSI

Grid rendszerekben széles körben használatos a nyílt forráskódú Globus Toolkit. A Globus Toolkit biztonsági szolgáltatásait a GSI (Grid Security Infrastructure) biztosítja. A Globus Toolkit legújabb 4.0-s verziója webszolgáltatás alapú, a nem webszolgáltatás alapú régebbi verziókból is tartalmaz komponenseket. Bár a GSI jelenleg nem autorizáció, hanem csak autentikáció alapú hozzáférés szabályozást valósít meg, továbbfejlesztése egyértelműen az autorizáció alapú hozzáférés szabályozás irányába folyik.

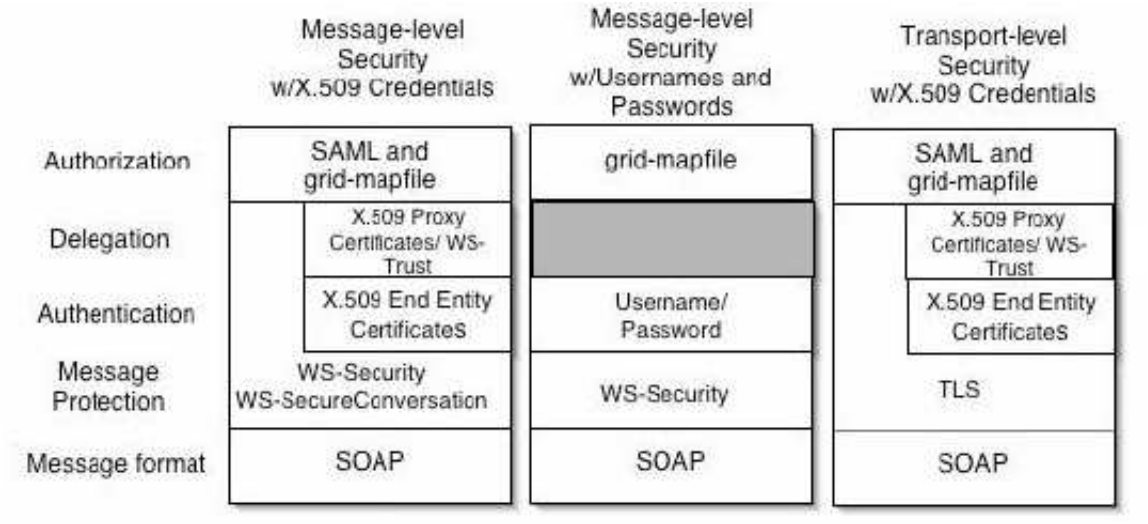
Alapértelmezésben a GSI legújabb verziója is a TLS (Transport Layer Security) protokollal, vagyis a szállítási rétegben biztosítja az üzenetváltások titkosságát. A GSI képes üzenetszintű titkosítás alkalmazására is, ez azonban a GSI dokumentáció [8] szerint még kiforratlan, és alkalmazása visszafogja a rendszer teljesítményét.

Alapértelmezésben tehát az 5.4. ábrán látható harmadik változat szerint működik a rendszer. A jelen tárgyalás szempontjából lényeges részek a TLS protokoll, ezzel kapcsolatban az X.509 végfelhasználói tanúsítványok (EEC - End Entity Certificate), proxy tanúsítványok, és a gridmapfile.

5.4.1. A TLS protokoll és a tanúsítványok

Nyilvános és titkos kulcsok

A TLS protokoll [9] széles körben használatos biztonságos kommunikációs csatornák létrehozására többek között például biztonságos webes hozzáféréshez (https). A TLS protokoll különféle magasabb szintű protokollok - a GSI esetében például a SOAP (Simple Object Access Protocol) - „becsomagolására” alkalmas. A GSI-vel védett grid rendszerben részt vevő összes felhasználó,



5.2. ábra. A GSI rétegei

feladat (job) és szolgáltatás egymás között TLS protokoll segítségével kommunikál. A TLS használatához szükséges, hogy a kommunikációs kapcsolat felépítésekor mindkét fél rendelkezzen egy-egy érvényes, a másik fél számára elfogadható tanúsítvánnyal. A tanúsítvány egyrészt egy hitelesített (általában RSA) nyilvános kulcsból, és a hozzá tartozó titkos kulcsból áll. Tulajdonképpen a nyilvános kulcsot magában foglaló és annak hitelességét igazoló szöveges fájlt szokás tanúsítványnak nevezni, de ennek használatához a titkos kulcsra is mindenképpen szükség van.

Azért fontos hangsúlyozni a titkos kulcs szerepét, mert valójában ennek a kulcsnak a tikossága biztosítja a rendszer biztonságosságát. Minél „titkosabb" a titkos kulcs, vagyis minél nehezebben juthat illetéktelenek kezébe, annál biztosabbak lehetünk benne, hogy a kommunikáció valóban titkos marad. A titkos kulcsot ezért általában kódolva, jelszóval védve tárolják. Ennek a kulcsnak, és ezzel együtt a hozzá tartozó tanúsítványnak a használatához tehát a felhasználó közreműködésére (a jelszó begépelésére) van szükség. Különböző megoldásokat (proxy tanúsítványok, agent programok) dolgoztak ki arra, hogy a felhasználónak minél kevesebbszer kelljen ténylegesen begépelnie a jelszavát, ez a probléma Single Sign On (SSO) témakörébe tartozik.

Tanúsítványok a GSI-ben

A GSI-ben tehát az összes felhasználó, feladat (job) és szolgáltatás a TLS protokoll segítségével kommunikál egymás között. Így minden felhasználó, feladat (job) és szolgáltatás rendelkezik a saját személyazonosságát igazoló tanúsítvánnyal és titkos kulccsal. A titkos kulcsot azonban csak a felhasználók esetében van értelme jelszavas védelemmel ellátni. A szolgáltatások és jobok titkos kulcsai így jobb híján titkosítatlanul tárolódnak, és csak a fájlrendszer hozzáférési jogosultságok védik őket. A szolgáltatások és jobok tanúsítványai így valamivel kevésbé biztonságosak, mint a felhasználókéi.

Minden egyes GSI tanúsítvány (a proxy tanúsítványok is) pontosan egy darab grid szintű globális azonosítót tartalmaz. A tanúsítvány tehát a használóját globálisan azonosítja, a megfelelő

jogosultságok érvényesítéséről egy másik mechanizmus (grid-mapfile) gondoskodik.

A tanúsítványok kiosztása

Mindezidáig csak egyfajta „tanúsítványról" volt szó, a GSI azonban kétfajta tanúsítványt használ: a végfelhasználói tanúsítványt (továbbiakban egyszerűen tanúsítvány), és a proxy tanúsítványt. Az azonosítás szempontjából a kétfajta tanúsítvány nem különbözik egymástól, mindkettő egy-egy globális azonosítót tartalmaz. A különbség a kettő között a kibocsátó személyében van. A végfelhasználói tanúsítványt a CA (Certificate Authority) bocsátja ki, proxy tanúsítványt azonban bármely (végfelhasználói) tanúsítvány tulajdonos készíthet. A végfelhasználói tanúsítványokat a felhasználók és szolgáltatások használják (mivel ezek viszonylag „állandó" entitások a gridben), az ő személyazonosságukat tehát a CA igazolja. A jobok az őket indító felhasználótól kapnak proxy tanúsítványt, és ezzel személyazonosságot.

A proxy tanúsítványok hitelességének ellenőrzése a „normál" tanúsítványokétól némileg eltérő mechanizmust igényel [7], ez azonban jelen tárgyalás szempontjából nem lényeges. Lényeges viszont, hogy a GSI-ben jelenleg a proxy tanúsítvány kiállításakor a kiállító személy (felhasználó) globális azonosítója kerül át a proxy tanúsítványba, kiegészítve némi utalással arra, hogy ez egy proxy tanúsítvány. A felhasználó tehát átruházza a személyazonosságát és ezzel minden jogosultságát a proxy tanúsítvány tulajdonosára. Mivel a proxy tanúsítványt jobok használják, a hozzá tartozó titkos kulcs a jobbal együtt védtelenül „utazik" a hálózaton. A felhasználó minden job indításakor új proxy tanúsítványt készít, és ezzel folyamatosan növeli a veszélyét annak, hogy jogosultságai illetéktelen kezekbe kerülnek.

A proxy tanúsítványok tehát önmagukban jelentős biztonsági kockázatot jelentenek. Ezt a GSI tervezői úgy próbálták meg mérsékelni, hogy korlátozták a proxy tanúsítványok érvényességének időtartamát, alapértelmezésben a kibocsátástól számított 12 órára. Ennek a megoldásnak a hátránya, hogy nem teszi lehetővé 12 óránál hosszabb futási idejű feladatok indítását. Ha ennél hosszabb futási időt szeretnénk engedélyezni, a felhasználónak 12 óránként új proxy tanúsítványt kellene kiállítania, ami nem csak a felhasználók kényelme szempontjából problémás. A rendszer lehetővé teszi, hogy 12 órásnál hosszabb élettartamú proxy tanúsítványokat is kiállítsunk, az időtartam növelésével azonban a biztonsági kockázat nő. Továbbá, mivel semmiféle garancia nincs arra, hogy egy adott feladat adott időn belül befejeződik, bármilyen időkorlátot választunk is, bizonyos feladatokhoz az sem lesz elegendő. Látnunk kell, hogy a probléma nem a megvalósításban van, hanem a proxy tanúsítványok lényegét érinti, az ilyen tanúsítványokkal megvalósított jogosultság átruházás - legalábbis a grid környezetben - jelentős biztonsági hiányosságokkal jár.

Grid-mapfile

Ahhoz, hogy a grid egy adott gépén egy adott feladat futhasson, a feladatnak (jobnak) szüksége van egy helyi felhasználói fiókra. A GSI rendszerben a helyi felhasználói fiókok és a globális azonosítók között a grid-mapfile teremt kapcsolatot. A grid-mapfile szerepe a tanúsítványokban található globális azonosítók leképezése lokális azonosítókra, azaz helyi login-nevekre. A grid-mapfile rendszeres frissítésétől eltekintve ez a leképezés viszonylag statikus.

Bár a grid-mapfile (a Globus dokumentáció szerint) nem tartozik szorosan a GSI-hez, tulajdonképpen ez a mechanizmus biztosítja a jogosultságok globális azonosítókhoz történő hozzá-

rendelését a lokális azonosítókon keresztül. Tehát a GSI-ben használt különböző tanúsítványok az azonosítást (autentikáció) szolgálják, míg a jogosultság hozzárendelés (autorizáció) a grid-mapfile-on keresztül történik.

A grid-mapfile mechanizmus legnagyobb hátránya, hogy nehezen skálázható. A grid-mapfile-t rendszeres időközönként frissíteni kell annak érdekében, hogy biztosítsuk, hogy a grid bármely felhasználója a grid bármely gépén futtathasson jobokat. Emiatt a gridben részt vevő gépek számának növekedésével egyre nehezebb lesz a grid-mapfile-ok szinkronizálása. A felhasználók számának növekedésével a grid-mapfile mérete is nő, és egyre több felhasználói fiókra van szükség az egyes gépeken. A rendszer méretének növekedésével a felhasználók, szolgáltatások és jobok halmaza egyre dinamikusabban változik, aminek következtében a grid-mapfile-t is egyre gyakrabban kell frissíteni. Továbbá, mivel a jogosultságok kezelése az azonosítástól elkülönülő, helyi mechanizmusokra van bízva, nincs pontos hozzárendelés a grides felhasználók és jogosultságaik között, így a pontos jogosultság-kezelés is nehézkes.

Ezekkel a problémákkal természetesen a Globus Toolkit fejlesztői is tisztában vannak, a grid-mapfile helyettesítésére irányuló fejlesztések jelenleg az attribútum alapú jogosultságkezelés irányába folynak.

Attribútumok és SAML

A SAML (Security Assertion Markup Language) [10] szabványosított, XML alapú nyelv úgynevezett biztonsági kijelentések (Security Assertion) megfogalmazására. A biztonsági kijelentések egy adott tárgyról (Subject - lehet személy, szolgáltatás, job, bármi amihez azonosító van rendelve) tartalmaznak állításokat. Az állítások azonosítási információkból, többek között attribútumokból állnak. Az attribútumok név-érték párok, amelyek a tárgy (Subject) egy-egy tulajdonságát határozzák meg (pl. név, cím, munkahely, tudományos fokozat stb.).

A SAML segítségével szabványos úton kommunikálhat egymással az erőforrás szolgáltató, akinek az adott entitás jogosultságaira szükség van a hozzáférés engedélyezése érdekében, és a jogosultság kezelő, aki a jogosultságokat meghatározza és tárolja. A hozzáférés engedélyezése alapesetben a következő séma szerint történik: Az erőforrás szolgáltató a saját tulajdonában lévő erőforrásokat kínálja használatra, de saját maga nem ismeri a hozzá forduló felhasználókat, és azok jogosultságait sem. A felhasználók és jogosultságaik a jogosultság kezelőnél vannak nyilvántartva. Az erőforrás szolgáltató úgy határozza meg a hozzá bejövő kérések jogosságát, hogy elkéri az ehhez szükséges attribútumokat a jogosultság kezelőtől, a jogosultság kezelő pedig egy szabványos attribútum kijelentés formájában átadja azokat. Ez a séma alkalmas nagyméretű elosztott jogosultság kezelő rendszerek létrehozására (lásd pl. Shibboleth).

A fenti sémát a GSI-re a következőképpen kívánják alkalmazni: Hagyományos tanúsítványok helyett attribútum tanúsítványokat alkalmaznának. Ezek nem a szokásos globális azonosítót tartalmazzák, hanem különböző attribútumokat. Az attribútum tanúsítványok elvben ugyanúgy alkalmazhatók a TLS protokollban (némi átalakítás után), azonban nem a felhasználó személyazonosságát tanúsítják, hanem azt, hogy a felhasználó (vagy az adott attribútum tanúsítványt használó entitás) rendelkezik a tanúsítványban megadott attribútumokkal. A másik fél, például a grid adott gépén futó helyi ütemező ilyenkor az attribútumok alapján hozzáférési döntéseket hozhat. Ezzel a grid-mapfile kiküszöbölhető, és a jogosultságok nyilvántartása is átkerül az attribútum kibocsátó hatósághoz.

Ezzel a megoldással azonban több probléma is van, és valószínűleg ezért maradt meg még a Globus Toolkit 4.0-s verziójában is a hagyományos grid-mapfile. Ahogy a Shibboleth-el kapcsolatban említettem, az attribútumok kiválóan alkalmasak webes tartalmak hozzáférés szabályozására. Itt azonban fájl szintű (de legalábbis könyvtár szintű) hozzáférés szabályozásra lenne szükség. Mik legyenek tehát az attribútumok? Túl sok attribútum megnehezíti az adminisztrációt, túl kevés attribútum nem nyújt megfelelően finom hozzáférés szabályozási lehetőséget. Emellett a gridben részt vevő összes szervezetnek azonos formátumú és azonos jelentésű attribútumokat kellene használnia, ezeket tehát célszerű lenne szabványban rögzíteni. Még ha sikerülne is ezeket az akadályokat leküzdeni, a legfontosabb probléma megoldatlan marad: a tanúsítványok visszavonása és a proxy tanúsítványok élettartam korlátozása. A jogosultság átruházás ugyanis továbbra is proxy tanúsítványok segítségével folyna, bár ezek már proxy attribútum tanúsítványok lennének. A hosszabb élettartamú proxy tanúsítványokkal azonban folyamatosan „szivárognának" az attribútumok a rendszerből, amelyekből a rosszindulatú támadó idővel összeállíthatná a számára megfelelő attribútum profilt.

Irodalomjegyzék

[1] SPKI tanúsítványok leírása
http://www.ietf.org/rfc/rfc2693.txt

[2] SPKI implementációk
http://theworld.com/~cme/html/spki.html#Code

[3] Shibboleth Project
http://shibboleth.internet2.edu

[4] A Shibboleth felépítése
http://shibboleth.internet2.edu/draft-internet2-shibboleth-arch-v05.html#_Toc23129736

[5] InQueue attribútumok
http://inqueue.internet2.edu/policy.html#attributes

[6] Globus Gridshib porjekt
http://gridshib.globus.org/
http://grid.ncsa.uiuc.edu/papers/gridshib-pki06-draft.pdf

[7] Proxy tanúsítványok
http://www.ietf.org/rfc/rfc3820.txt

[8] Globus Toolkit biztonsági architektúra
http://www.globus.org/security
http://www.globus.org/toolkit/docs/4.0/security/GT4-GSI-Overview.pdf

[9] TLS protokoll 1.0
http://www.ietf.org/rfc/rfc2246.txt

[10] SAML v2.0 specifikáció
http://www.oasis-open.org/specs/index.php#samlv2.0

[11] PKI (Public Key Infrastructure) dokumentáció linkek
http://www.ietf.org/html.charters/pkix-charter.html
http://www.pki-page.org/#DOC

[12] SSL és TLS protokollok
http://www.openssl.org