Mobil ad hoc hálózatok biztonsága

GÉMESI ROLAND[1]–IVÁDY BALÁZS[1]–ZÖMBIK LÁSZLÓ[2]

gemesiro@sch.bme.hu–ivadyb@sch.bme.hu–laszlo.zombik@eth.ericsson.se

BME, Távközlési és Telematikai Tanszék[1], Ericsson Magyarország Kft.[2]

Az ad hoc hálózatok nem igényelnek kiépített és fix infrastruktúrát, minden résztvevő egyenrangú. A hálózat működésének biztosítása együttes erővel történik, ám teljesen elosztott és önszervező hálózatról lévén szó, a kommunikáció számos veszélyforrásnak van kitéve. Cikkünk célja, hogy bemutassa az ad hoc hálózatok biztonsági kérdéseit, rávilágítva néhány felmerült megoldási lehetőségre.

1. Bevezetés

A hagyományos hálózatokkal szemben az ad hoc hálózatok nem igényelnek kiépített infrastruktúrát, a feladatokat az egyenrangú résztvevők elosztva, önszervező módon végzik. Egy ilyen rendszerben nincsenek kitüntetett szerepű eszközök, melyek központilag irányíthatnák vagy ellenőrizhetnék a folyamatokat.

A mobil ad hoc hálózatok különféle speciális tulajdonságokkal rendelkeznek. Dinamikus topológiájuk miatt a felépülő kapcsolatok, így az utak is csak korlátozott ideig érvényesek. A résztvevők mozgása vagy eltűnése nem befolyásolhatja a hálózatműködést. Az egységek rendszerint kisméretű, kézi, hordozható készülékek, melyek korlátozott CPU-, memória- és telepkapacitással rendelkeznek. A vezeték nélküli kapcsolatok gyakran kisebb sávszélességűek, és e csatornák gyakran osztottak és limitáltak. Mind a kapcsolatok, mind a készülékek sokkal sebezhetőbbek, mint vezetékes hálózatok esetében.

Egy ad hoc hálózat elosztottságánál fogva számos olyan veszélyforrásnak van kitéve, melyek a korábbiakban még csak fel sem merültek. A továbbiakban biztonságtechnikai oldalról vesszük szemügyre e hálózatokat, biztonságossá tételükhöz eddig felmerült gondolatok ismertetésére törekszünk.

A 2. fejezetben bemutatjuk az ad hoc útválasztó mechanizmusok típusait, alapelveit, majd két, biztonságtechnikailag különböző konkrét protokollt is ismertetünk. A 3. fejezet a biztonság kérdéskörét részletezi, rámutat az elérendő célokra, valamint bemutat jó néhány fenyegetést jelentő tényezőt. Majd a problémák megoldásához rendelkezésünkre álló biztonsági mechanizmusokat ismertetjük. Ezután az 5. fejezetben rámutatunk arra, miként vezet a biztonságosság kérdése az útválasztás témakörébe, és lehetőségeket mutatunk a biztonságos megoldás eléréséhez.

2. Ad hoc hálózatok útvonal-választási mechanizmusai

A kommunikációhoz megfelelő útvonalon továbbított csomagok szükségesek. Hagyományos hálózatokban léteznek olyan kitüntetett pontok (gateway, szerver...), amelyek a hálózat felépítéséről információt hordoznak, így tudják a helyes továbbítási irányt. Ezzel szemben az ad hoc hálózatokban nincsen fix infrastruktúra, nincsenek központosított szerepek, minden egységnek részt kell vállalnia az útvonalválasztásban és csomagtovábbításban.

Útvonalválasztás protokolljai a mobil kommunikáció területén két nagy csoportba sorolhatók: *proaktív és reaktív* protokollok.

Egy *proaktív* protokoll állandóan figyelemmel kíséri a hálózat változásait és a lehetséges útvonalakat. Hátránya, hogy a résztvevőknek sok adatot kell tárolniuk, valamint a hálózat rendszeres felderítése nagy többletterhelést eredményez. Ezzel ellentétben állnak a *reaktív* vagy más néven igény szerinti protokollok, melyek csak akkor keresnek útvonalat, amikor arra szükség van. Ez jobb megoldás, de késedelmet jelenthet a kommunikáció kezdetekor, hiszen az útvonalat ekkor kell kialakítani. Léteznek ezen túl még e két típust ötvöző *hibrid* útvonalválasztó protokollok is. A következőkben két tisztán reaktív protokollt tekintünk át.

A ***Dynamic Source Routing (DSR)*** protokoll forrás útvonal-választási (source routing) algoritmust használ, vagyis mindig a küldő határozza meg a csomag teljes útját, melyet a csomag fejrészébe épít. A közbülső csomópontok számára ebből egyértelműen kiderül a továbbítás iránya.

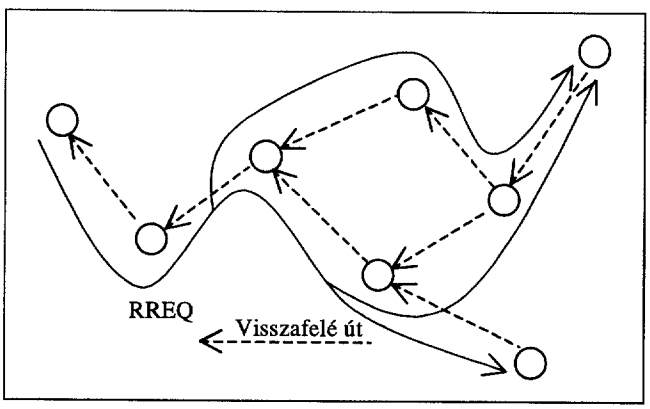
A résztvevők útvonal-gyorsítótárat (route cache) tartanak fenn, melyben bizonyos ideig tárolódnak a használt útvonalak. Megfelelő bejegyzés hiányában egy út-

vonalkérés csomag (route request) indul. Ez az üzenet szétterjed a hálózatban (broadcast message), minden továbbító résztvevő beleírja saját címét. Idővel a kérés eljuthat a célállomáshoz is, aki a fordított úton nyugtázhatja az esélyes útvonalat. Ha nem található útvonal a célállomásig, akkor hibaüzenet keletkezik, később a küldő ismét indíthat útvonalkeresést.

Kis mobilitású hálózatban ez a protokoll meglehetősen jól működik, mivel az útvonal-gyorsítótárban lévő bejegyzések hosszabb ideig is használhatók maradnak. Hátrányuk, hogy a csomagok a közbülső résztvevők címeit is tartalmazzák, amely jelentős többletterhelést (overhead) eredményezhet.

Az ***Ad hoc On-Demand Distance Protokoll (AODV)*** szintén reaktív protokoll, mely a DSR-hez hasonlóan üzenetszórásos útvonalkereső algoritmust használ, de a küldő nem határozza meg a csomag teljes útját. A közbülső résztvevők irányítják a csomagot, minden csomópont csak a következő ugrás irányát dönti el. A hurokmentesség problémáit az üzenetek sorszámozásával küszöbölik ki.

Útvonal-felderítés indítja a folyamatot útvonalkérő (route request – RREQ) üzenettel, ami csak a célállomás és a küldő címét tartalmazza. Minden résztvevő, aki megkapja a RREQ üzenetet, egy visszafelé mutató utat (reverse route) állít be oda, ahonnan az üzenetet kapta, majd szétszórja az üzenetet (1. ábra). Amikor a célhoz megérkezik egy ilyen RREQ üzenet, arra útvonalválasz (route reply – RREP) üzenettel felel. A RREP üzenet a visszafelé vezető útvonalon fut végig, miközben beállítja az előre vezető utakat. Minden útvonal-bejegyzésnek meghatározott élettartama van, adatforgalom hiányában az útvonalak helyi „Hello" üzenetek segítségével tarthatók érvényben.



1. ábra AODV útvonalválasztás

Ha valahol hiba lép fel a kapcsolatban (link error), akkor az elérhetetlenné váló célpontokról útvonalhiba (route error – RERR) üzenet keletkezik. Az AODV képes helyileg korrigálni a kieső kapcsolatokat, amit helyi javításnak (local repair) neveznek. Ha a helyi javítás sikeres, a végpontok nem is észlelik a változást.

Az AODV kis számítás- és memóriaigényű, jól skálázható protokoll, ami nagyobb mobilitású hálózatok esetén is megfelelően működik.

3. Ad hoc hálózatok biztonsági kérdései

Biztonsági célok

A különböző típusú hálózatoknak különféle biztonsági szempontokat kell figyelembe venniük.

Az *elérhetőség vagy használhatóság* (availability) azt jelenti, hogy a hálózat és annak szolgáltatásai mindenféle támadás és hibás működés ellenére mindig hozzáférhetőek és használhatók legyenek.

A *hitelesítés* (authentication) garantálja a résztvevőknek egy adat vagy egy másik résztvevő hitelességét.

A *bizalmasság* (confidentiality) az adat titkosságát biztosítja, ami azt jelenti, hogy az információ nem kerülhet illetéktelen egységek birtokába.

Az *integritás* (integrity) biztosítja az üzenetben történt módosítás felismerését. A módosításokat természetes környezeti hatások vagy szándékos támadások is okozhatják.

A *letagadhatatlanság* (non-repudiation) a részvételt rögzíti, ami egy későbbi bizonyítási folyamathoz lehet szükséges.

Számos kriptográfiai eljárás igényel *kulcs menedzsmentszolgáltatást* a kulcsok adminisztrálásához.

Ezeken kívül más szolgáltatásokra is szükség lehet, mint például a *hozzáférés-védelem* (authorization), ami a rendszer erőforrásokhoz való hozzáférését korlátozza.

Mobil ad hoc hálózatok veszélyforrásai a vezetékes hálózatokhoz képest újabb kérdéseket vetnek fel. A fő gond, hogy egy elosztott hálózatban a kommunikáció résztvevőinek meg kell bízniuk egymásban, ezért a megfelelő biztonság csak nehezen érhető el.

Szolgáltatás elleni támadások (Denial of Service Attacks – DoS) a hálózat működésének meggátlását célozzák. Ez a fajta támadás csökkenti a szolgáltatás elérhetőségét, a hálózat hozzáférhetőségét, nagy késedelmet generálhat. Romboló magatartásról lévén szó, együttműködést igénylő környezetben kiemelt veszélyt hordoznak és csak nehezen küszöbölhetők ki.

A közös kommunikációs csatornát megcélozva, az ellenség zajjal terhelheti a fizikai közeget, mellyel interferenciát vagy nagy információvesztést okozhat. Nehéz feladat a támadó zavarását a csatorna természetes zajától megkülönböztetni. A közeghozzáférés szabályainak be nem tartása szintén gátolhatja a normális kommunikációt.

A mobil eszközök kis CPU-, memória- és telepkapacitása miatt a nagy CPU teljesítményt igénylő folyamatok leterhelhetik a processzort, így gátolva más folyamatok haladását. A telepterhelést is optimalizálni kell, a használaton kívüli időben a készülék egyes részei alvó üzemmódba válthatnak át. Viszont a bejövő kérésekre válaszolni kell, így az energiakimerítési támadás egy másik veszélyforrás lehet.

Az útvonal-választási folyamat is megzavarható- ha a támadó egységek rossz útvonalakat terjeszthetnek szét. Rossz vagy régi információk bekerülve az útkere-

sési folyamatba hosszabb utakat, hurkokat eredményezhetnek, ami teljesítménycsökkenéshez vezethet. Végül az önzőség is gátolhatja a működést, amikor a node kihasználja az együttműködés lehetőségeit, viszont ő maga nem segíti mások kommunikációját, tehát nem továbbít csomagokat, kíméli saját erőforrásait.

Megszemélyesítés (impersonation) az, ha egy támadó másnak adja ki magát, mint aki valójában. Hitelesítés nélküli rendszerekben ez egy triviális támadás lehet, ám tanúsítványok használatával kiküszöbölhető. E tanúsítványok összerendelik a résztvevőket azonosítóikkal. A központi egység hiánya miatt ez egy nehéz feladat, mivel nincs egyetlen elem, mely tárolhatná ezen összerendeléseket. A megszemélyesítés másik fajtája a „man-in-the-middle" támadás, amikor egy ellenséges node beépül az útvonalba, majd szimulálja a másik oldalt.

Bizalmasság megsértésének (Confidentiality Violation) első esete, ha egy passzív lehallgató csak figyeli a kommunikációs csatornát, de egy aktív lehallgató módosíthatja is az átvitt adatokat. Ha a kommunikáció titkosított, kódtöréses támadás jöhet szóba, ezért elkerülendő a gyenge algoritmusok használata.

Üzenet megváltoztatása is lehet a támadó célja. Próbálkozhat új üzenetek beszúrásával, az információk megváltoztatásával vagy fontos adatok eltávolításával is.

Üzenetletagadás veszélye, hogy a kommunikációban való részvétel utólagos letagadhatósága teljesen lehetetlenné teszi egy későbbi bizonyítás és szankcionálás folyamatát.

Anonimitás megsértése is lehetséges, mert a hálózat résztvevői információkat gyűjthetnek szomszédaikról, a médium ugyanis osztott, és az identitás egy részének (cím) mindenképpen elérhetőnek kell lennie. Ezáltal egy célpont azonosíthatóvá vagy lokalizálhatóvá válhat. Ha az egységek azonosító tanúsítványokkal rendelkeznek, képesek egymásról információkat gyűjteni. Az azonosítókat lehallgatók is megszerezhetik.

Fizikai támadás esetén a támadó megszerezhet (pl. lopás) egy készüléket, melyben akár hardverben, akár szoftverben módosításokat végezhet. Gyenge védelem esetén így megtudhat titkos információkat (pl. kulcsok), de akár vírus vagy trójai faló is telepíthető a készülékbe.

Központi egység hiánya miatt a rendszer nem sebezhető a központ ellen induló támadásokkal. Így viszont számos probléma válik nehezebben megoldhatóvá.

Útvonalválasztás manipulálásával, vagyis rossz információk terjesztésével a támadó jelentős zavarokat okozhat. Megnövekedhet a csomagküldési idő, fontos adatok veszhetnek el, de akár a támadó is fontos információkat szerezhet meg.

A kulcsok kompromittálódása is komoly veszélyforrás, mert a titkosítási eljárások gyakran titkos információkat igényelnek.

4. Biztonsági mechanizmusok

A hagyományos biztonsági megoldások (pl. autentikáció, digitális aláírás, kódolás) megfelelő biztonságot nyújthatnak, ám legtöbbjük valamiféle menedzsmentszolgáltatást igényel. Ezen szolgáltatások ellátásához nagyrészt központi egység szükséges, ám ad hoc környezetben ilyenekre nem támaszkodhatunk. A feladatokat is elosztva kell végezni, ráadásul a rendszernek redundanciát is hordoznia kell, mivel résztvevők eltűnhetnek, szakadozhat a kapcsolat és a kompromittálódás veszélye is fennáll.

Nyilvános kulcsú infrastruktúra (Public Key Infrastructure – PKI) esetén minden résztvevő rendelkezik egy nyilvános és egy titkos kulccsal. A nyilvános kulccsal kódolt üzenet csak a neki megfelelő titkos kulccsal dekódolható, és a két kulcs egymásból nem származtatható.

Létezhet ezen felül egy központi Certificate Authority (CA), amely tanúsítványokat (certificate) állít ki nyilvános kulcsok és a megfelelő csomópontok összetartozásáról. A CA-nak on-line elérhetőnek kell maradnia, hogy ezen összerendeléseket biztosítani tudja, követnie kell a kulcsváltozásokat, valamint képesnek kell lennie tanúsítványok visszavonására.

A *küszöb kriptográfia (Thresold cryptography)* a bizalom szétosztásának egy lehetséges módja. Egy (n, t+1) séma lehetővé teszi egy kriptográfiai feladat n résztvevő közötti szétosztásának lehetőségét úgy, hogy azt bármely t+1 tagja sikeresen el tudja végezni, de ennél kevesebb tag már nem birkózhat meg a feladattal.

Ez esetben a kulcs menedzsmentszolgáltatás n szervere osztja szét egy bizonyítvány aláírásának jogát, vagyis a szolgáltatás titkos kulcsát. Ezt a szétosztást a k kulcs (n, t+1) szétosztásának nevezzük. Így minden egyes szerver az aláírásnak csak egy részét képes előállítani, melyek még akkor is sikeresen összekombinálhatók, ha t szerver kompromittálódott.

A küszöb kriptográfia egy nagyon hasznos tulajdonsága, hogy képes a résztvevők információdarabjainak frissítésére. Ez a mozgó támadók kivédését teszi lehetővé, melyek egymás után törik fel a szervereket, melyek mennyisége egy bizonyos idő elteltével meghaladhatná a kritikus t mennyiséget. Szétosztásfrissítéskor (share refresh) egy új küszöb kriptográfiai séma jön létre, mely kiválóan alkalmas a hálózat változásaihoz való alkalmazkodásra. Mivel e frissítés nem túl bonyolult művelet, egy kialakított konfiguráció hosszabb ideig is képes a hálózat változásait követni. Viszont ha túl sok résztvevő kompromittálódott, új sémát kell kialakítani.

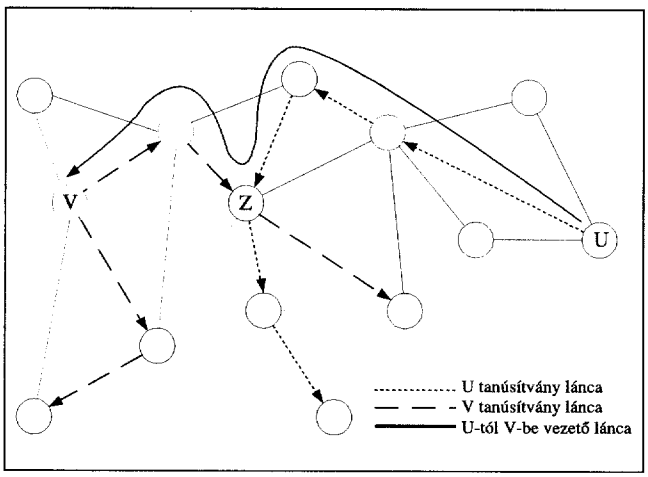
A megoldás súlyos problémája, hogy feltételezi a résztvevők szinkronitását, amely csak a legritkább esetben áll fenn. Egy csomópont megszüntetheti a kapcsolatot vagy lelassulhat egy DoS támadástól, és eközben előfordulhat, hogy a többiek véghezvisznek egy szétosztásfrissítést. Ezek után már nem lesz képes visszakapcsolódni a folyamatba, mert azóta egy újabb konfiguráció alakult ki.

A rendszer kialakítása is probléma, amikor még csak néhány résztvevővel rendelkezik a hálózat. Ki kezdemé-

nyezheti ekkor egy szétosztott CA létrehozását? Ráadásul később, ha egy kialakult rendszer diszjunkt részekre szakad, ezek külön folytathatják működésüket.

Önszervező PKI (Self Organizing Public Key Infrastructure) nem igényel megbízott harmadik felet, mely a tanúsítványalapú hitelesítés hátránya ad hoc környezetben. Léteznek bizonyítványalapú autentikációs megoldások (pl. PGP – Pretty Good Privacy), melyekben a bizonyítványokat a felhasználók saját maguk készítik el és terjesztésüket is megoldják.

Az önszervező PKI-ben minden felhasználó rendelkezik egy kisméretű helyi tanúsítványtárolóval, melyben korlátozott számú tanúsítványt képes tárolni. Amikor az *u* és *v* felhasználók kommunikálni szeretnének, egyesítik e tárolójukat, melyben *u* próbál keresni egy megfelelő tanúsítványláncot *v*-hez (2. ábra). Ilyen jó eséllyel biztosítható még akkor is, ha a tároló mérete kicsi a hálózat résztvevőinek számához képest.



2. ábra Önszervező PKI tanúsítványlánca

Ezen megközelítés jó megoldás lehet a teljes önszerveződés eléréséhez. Problémát a skálázhatóság jelent, mivel a hálózat növekedésével egyre növekszik a valószínűsége annak, hogy nem létezik megfelelő tanúsítványlánc.

Az *identitásalapú (ID-Based)* titkosítás alapötlete, hogy a nyilvános kulcs szerepét az egységek azonosítására szolgáló szintén nyilvános azonosítók töltik be, és ezzel a CA által biztosított összerendelő feladatkör szükségtelenné válik. Azonban a nyilvános kulcsú titkosítás biztonságossága éppen abból adódik, hogy a nyilvánosból a titkos kulcs (illetve fordítva) nem származtatható. Ez esetben egyetlen központi entitás képes arra, hogy egy publikus kulcsból (identitás) elkészítse annak titkos párját. Ezt a műveletet csak egyszer, a felhasználók regisztrációjakor kell elvégezni. A rendszerben részt vevő, egymást nem ismerő elemek ezután autentikusan képesek egymással kommunikálni, mégpedig harmadik féllel való interakció nélkül.

A valóságban viszont gondolni kell a kompromittálódott és később újra regisztrálni akaró résztvevőkre is, tehát a regisztráló központnak mégis elérhetőnek kell maradnia. További veszélyforrás, hogy e központ ismerni fogja a résztvevők titkos kulcsait is. A titkos kulcs elkészítése erőforrás-igényes feladat, hosszú időt vehet igénybe.

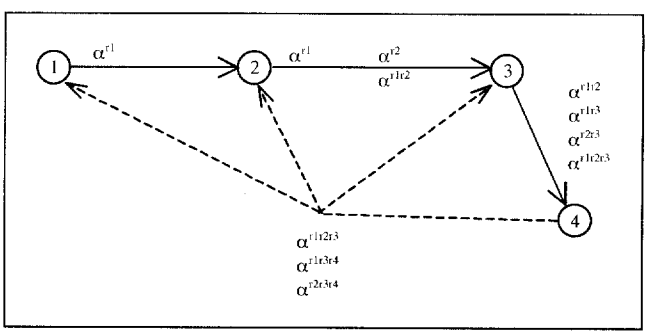
A *titkos (közös, szimmetrikus) kulcsú architektúra (Common key architecture)* az előzetesen megtörtént azonosítás után a titkosság megvalósításához egy kódolt kapcsolat felépítésére van szükség. Ilyen célra alkalmazzák a közös kulcsú titkosítási megoldásokat. Ehhez szükséges a résztvevőknek egy közös (osztott, szimmetrikus) titok birtoklása, melyet egyaránt használnak a kódolás és a dekódolás folyamatában. Titkos kulcsú architektúra alkalmazásával lehetővé válik a titkos multicast is. Korábban kialakultak már kulcs menedzsmentprotokollok, de az ad hoc hálózatok számos más igényeket is támasztanak.

Diffie-Hellman (DH) kulcs cserealgoritmus segítségével nyilvános csatornán lehetőség nyílik két résztvevő közötti közös titok kialakítására. Egy harmadik résztvevő, aki teljes egészében hallja a kommunikációt, nem lesz képes ugyanazon közös kulcs generálására.

A két résztvevő (*A* és *B*) megállapodnak egy q rendű G ciklikus véges csoporton és a csoport generátorelemén. Ezután mindkét résztvevő véletlenszerűen választ egy titkos kitevőt (A és B). Ezután A kiszámítja α^A-t, *B* α^B-t és átküldik egymásnak az így megkapott értékeket. Mivel a kitevők titokban maradtak, az átküldött értéket erre a hatványra emelve kialakul a α^{AB} közös titok, de egy lehallgató nem tudja meg magukat a kitevőket, így képtelen a kulcs létrehozására.

A kettőnél több résztvevős esetben is lehetséges a kulcs kialakítása. Ezek közül tekintünk most át néhányat.

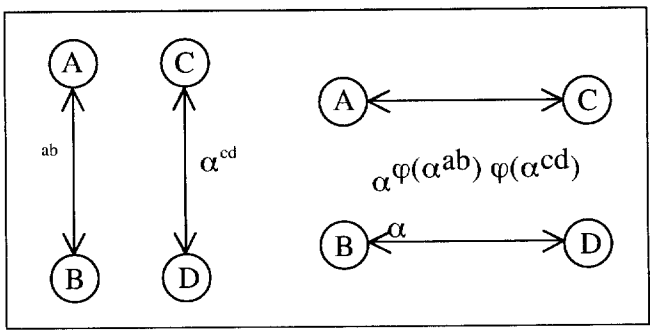
A *GDH.2 (Generalized Diffie Hellman)* a Diffie-Hellman kulcs csere algoritmus általánosítása több résztvevő esetére. A résztvevők egy láncot alkotnak, majd az első tag elküld egy Diffie-Hellman-szerű üzenetet a következő tagnak, majd az kiegészítés után továbbküldi. Ez a folyamat az utolsó elemig folytatódik, ami már képes lesz a közös titok létrehozására, és minden egyes résztvevőnek eljuttatja a számára hiányzó információtöredéket (3. ábra). A titok csak a megfelelő végpontok számára lesz ismert.



3. ábra A GDH.2 üzenetváltásai

Ezen eljárás fő problémája, hogy az utolsó entitás kitüntetett szereppel rendelkezik. Ezen túl viszonylag nagy mennyiségű adatot kell átvinni a csatornán.

A *Hypercube* algoritmus alapötlete párok kialakítása, melyek létrehozzák a közös titkaikat. Ezután a párokat is párokba rendezzük, melyek között ismét elvégezhető a kulcscsere kettesével, és így tovább (4. ábra). A probléma az, hogy a résztvevők száma csak 2^n lehet.



4. ábra A Hypercube üzenetváltásai

Az Octopus protokoll segít a fenti problémán, mely létrehoz egy Hypercube magot, amit karokkal egészít ki. Először e karok végeznek DH kulcscserét a nekik megfelelő központi elemekkel, majd e központ alakít ki Hypercube struktúrát, végül pedig közlik a karokkal a kialakult új kulcsot. A gond ezzel az, hogy a Hypercube mag központi szerepet játszik, valamint új résztvevők bevonása bonyolult.

Csoport kulcs menedzsmentprotokoll (Group Key Management Protocol-GKMP) a résztvevők egy csoportjának szimmetrikus kulcs menedzsmentfunkcióit látja el. A GKMP kulcs kialakítási mechanizmusa bármely két résztvevő közötti kooperatív eljárás, mint például a Diffie-Hellman kulcscsere. A kulcs létrehozása után a GKMP elterjeszti e csoportkulcsot az arra jogosult résztvevőknek. Lehetővé teszi ezen túl új résztvevő beléptetését, tag törlését és a csoport teljes újrakulcsolását. A GKMP teljes jogosultság-ellenőrző rendszert is tartalmaz, mivel a kulcsolás alkalmával jogosultsági tanúsítványok (Permission Certificate – PC) is létrejönnek. Bármely csomópont bárki jogosultságát ellenőrizheti, de módosítani nem képes azt. GKMP-vel lehetséges a kompromittálódott résztvevők kizárása, mivel a feltört résztvevők listáját (CRL – Compromise Recovery List) szétterjeszti a hálózatban.

Ez a protokoll megpróbálja a lehető legtöbb feladatot kiosztani a csoportnak, tehát igyekszik elkerülni bármiféle központi beavatkozást. Ennek ellenére néhány funkció, mint például a jogosultságok osztása, továbbra is központosított feladatkör maradt.

Az előzőkben láttuk, hogy a kommunikáció titkosságának biztosítása nem jelent komolyabb problémát, mivel közös titok, vagyis titkos csatorna kialakítására és kulcsainak létrehozására léteznek megoldások (DH, GDH.2, Hypercube, Octopus). A közös kulcs a végpontok között kialakítható, így a köztes csomópontok és más lehallgatók számára a kommunikáció rejtve marad.

Nagyobb kihívás az egyes résztvevők azonosítása, vagyis az autentikáció megvalósítása. Ad hoc hálózatokban nem támaszkodhatunk egy központi tanúsítványokat biztosító entitásra (Certificate Authority), bár láttuk, hogy megoldási lehetőségek elméletben már léteznek (Küszöb kriptográfia, Önszervező PKI, ID-Based PKI).

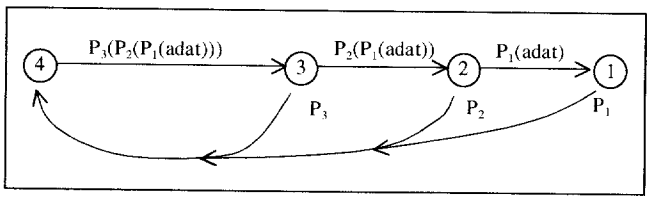
5. Biztonságos útvonalválasztás

Az ad hoc hálózatok biztonságosságának vizsgálata elvezet az útvonalválasztás kérdéskörébe, bár a jelenlegi útvonal-választási megoldások biztonsági szempontokat még nem vesznek figyelembe. A támadó megpróbálhat beépülni egy útvonalba, és őt ebben a jelenlegi megoldások egyáltalán nem korlátozzák. A támadónak elegendő lehet akár megközelíteni az útvonalat és egy onnan indított DoS támadással szabotálhatja a kommunikációt. Egy DoS támadás sokszor teljesen megkülönböztethetetlen a csatorna természetes minőségcsökkenésétől. Úgy is tekinthetjük a kérdést, hogy egy megfelelő minőségű csatornát szeretnénk felépíteni, ahol az útvonal megbízhatósága a rendszernek egy QoS paramétere.

A lényeg tehát az, hogy olyan útvonal-választási folyamatra van szükség, amelynél a kialakuló úton átvitt adatok megadott biztonsági szintje biztosítható. Emellett fontos még, hogy az útvonal igény szerint meghatározható legyen. Ezt elegendő információ birtokában előírhatja egy kijelölt node (pl. a küldő), de elosztott algoritmus is elképzelhető.

Az *Onion routing* egy olyan biztonságos útvonal-választási megoldás, mellyel egy üzenet csakis a megadott útvonalon utazhat. Ezt nyilvános kulcsú titkosítással oldják meg, melynek során a küldő összegyűjti a közbülső csomópontok nyilvános kulcsait és mindegyikkel egymás után kódolja az üzenetet. Az egyes publikus kulcsok hatását csakis a neki megfelelő titkos kulcs, vagyis a megfelelő közbülső résztvevő képes eltávolítani (5. ábra). A címzetthez eljutó adat csak abban az esetben feldolgozható, amennyiben végigutazott a meghatározott útvonalon. Ez akkor alkalmazható, ha már a küldő pontosan tudja, hogy csomagja milyen útvonalon fog végighaladni, amely a forrás útvonalválasztások sajátossága (pl. DSR).

A *Security-aware routing (SAR)* nemcsak az optimális útvonal megtalálását tűzi ki célul (legrövidebb vagy leggyorsabb), hanem meghatározott biztonsági igények teljesülését is szem előtt tartja. Egy útvonal biztonságosságának leírására többféle attribútum alkalmazható, mint például a megbízhatóság szintje (trust level) vagy biztonságosság szintje (security level). Az útvonalválasztás ezeket figyelembe veszi, és csak a megfelelő biztonsági jellemzőjű résztvevők szerepel-



5. ábra Onion routing

hetnek az útban. Természetesen e szinteknek manipulálhatatlannak kell lenniük ahhoz, hogy sem a csomópontok szintjei, sem az igényelt szint ne kompromittálódhasson.

SAR alkalmazásakor a küldő, aki útvonal-felderítést kér, az igényelt biztonsági szintet hozzáépíti a kéréshez. A közbülső résztvevők csak akkor továbbítják a kérést, amennyiben megfelelnek a biztonsági előírásoknak. Ha eljut egy kérés a címzetthez, akkor kialakul egy megfelelő biztonságú útvonal. A SAR kiegészítés szinte bármely igény szerinti (on-demand) útvonalválasztó protokollhoz kapcsolható.

A SAR megoldások nehézsége, hogy az egyes szinteket autentikálni kell, semmiképpen sem lehet a résztvevők felelőssége saját szintjükről nyilatkozni. Láttuk korábban, hogy ad hoc hálózatokban az autentikáció kérdésének megoldása nem triviális feladat, úgymint az sem, hogy megakadályozzuk a nem megfelelő biztonságú (és esetleg támadó) csomópontok beépülését az útvonalba. Mindezek tetejére már az is veszélyt jelenthet, ha egy node biztonsági paraméterei kiolvashatók, mivel ezek általában szoros összefüggésben vannak annak fontosságával.

A *Watchdog* a résztvevők jóságának folyamatos nyomon követését teszi lehetővé. A rádiós csatorna osztottsága révén minden résztvevő figyelemmel kísérheti közvetlen szomszédainak viselkedését, így könnyen észrevehet egy megbízhatatlan elemet. Ez a passzív figyelés a hálózatra nézve nem okoz további terhelést, de sajnos tévedhet (pl. aszimmetrikus link, ütközés), illetve megtéveszthető (pl. irányított antenna).

A *Pathrater* eljárás úgy valósítja meg a veszélyes elemek kirekesztését a felépülő kommunikációból, hogy az egyes utakhoz működésük során felállított statisztikák alapján jósági értékeket rendel. Az egyes utak jósági mutatóját megfelelő működés esetén folyamatosan növeli, míg hiba esetén csökkenti. Ilyen adatok ismeretében az útvonalválasztás során lehetőség van egy megfelelő út kiválasztásra.

Az eddig áttekintett biztonságos útvonal-választási mechanizmusok mind arra törekedtek, hogy a kialakuló utakban ne szerepelhessenek megbízhatatlan résztvevők. Egy támadó fejével gondolkozva rájöhetünk, hogy semmiféle hátrányba nem kerülünk, legfeljebb nem kell majd a későbbiekben nagy fontosságú csomagokat továbbítani.

Az ilyenfajta önző viselkedésminta csábító lehet például a telep kímélése céljából, de tömeges méretekben a hálózat működésének megszűnését is eredményezheti. Kiküszöbölése egyfajta virtuális fizetőeszköz, a *nuglet* alkalmazásával történhet. Ezzel a fizetőeszközzel vásárolják meg a résztvevők egymás szolgáltatását a hálózatban, így az önző egyedek nuglet-jei pedig idővel elfogynak.

A kommunikáció során tehát a nem megbízható résztvevők olyan szankcionálására van szükség, mellyel csakis a jó viselkedés kifizetődő, vagyis amely alkalmazásával a támadók idővel ellehetetlenülnek. Ez azt jelenti, hogy a nem megfelelően viselkedő résztvevők kommunikációs igényeinek segítését a rendszernek meg kell tagadnia.

6. Összegzés

Az ad hoc hálózatok számos veszélynek vannak kitéve, de vannak lehetőségek arra, miként érhető el mégis biztonságos kommunikáció. Sok kérdés még mindig nyitott, de e területen napjainkban is intenzív kutatás folyik.

Irodalom

Földesi András, Homolya György, Horváth Cz. János, dr. Imre Sándor: Bevezetés a mobil ad hoc útvonalválasztó protokollok világába (Híradástechnika, 2001. május)

Kata Molnár, László Zömbik: Security Issues in Mobile Ad hoc Networks. (Evolution in the military communications systems – trends and challenges in the XXI. century, 2001.)

Lidong Zhou, Zygmunt J. Haas: Securing Ad Hoc Networks (IEEE Network Magazine, vol. 13, no.6, November/December 1999)

Jean-Pierre Hubaux, Levente Buttyán, Srdan Capkun. The Quest for Security in Mobile Ad Hoc Networks (ACM Symposium on Mobile Ad Hoc Networking and Computing, MobiHOC 2001)

Ueli M. Mauer, Yacov Yacobi: A Non-interactive Public-Key Distribution System (Designs, Codes and Cryptography, 1996)

Maarit Hietalahti: Key Establishment in Ad hoc Networks (Proceedings of the Helsinki University of Technology, Seminar on Network Security fall 2000)

Seung Yi, Prasad Naldurg, Robin Kravets: Security-Aware Ad hoc Routing for Wireless Networks (Technical Report UIUCDCS-R-2001-2241(ps/pdf), August 2001)

Sergio Marti, T.J. Giuli, Kevin Lai, Mary Baker: Mitigating Routing Misbehavior in Mobile Ad Hoc Networks (In Proceedings of MOBICOM, 2000)

Sonja Buchegger, Jean-Yves Le Boudec: Nodes Bearing Grudges: Towards Routing Security, Fairness, and Robustness in Mobile Ad Hoc Networks (In Proceedings of the Tenth Euromicro Workshop on Parallel, Distributed and Network-based Processing, January 2002)