HADMÉRNÖK

XI. Évfolyam 1. szám - 2016. március

NAGY Dániel
nagy.daniel@operculum.hu

KRIPTOGRÁFIAI KIHÍVÁSOK A VEZETÉK NÉLKÜLI SZENZORHÁLÓZATOKBAN

Absztrakt

A felügyelet nélküli szenzorhálózatok egyértelműen a jelen és a jövő hadviselésének kellékei. Használatukkal kevesebb katonával lehet nagyobb területet ellenőrzés alatt tartani. Mint minden informatikai rendszer ezek is támadhatók, és e támadások nagy részének kivédésében kriptográfia használatos. Írásomban áttekintés adok a támadási lehetőségekről, majd három olyan kriptográfiai módszert mutatok be, amely a szenzorhálózatok adta korlátozott teljesítményű környezet jelentette kihívások közt is alkalmazható.

Unattended ground sensors are inevitable instruments of present's and future's warfare. By using them less manpower can supervise a larger territory. Just like any communication network, these sensor networks use cryptography as a main measure against attacks. In this writing a primer is given on the attack possibilities, and then I present three cryptographic methods, that are used in the constrained field of sensor networks.

Kulcsszavak: *titkosítás, kriptográfia, WSN, UGS ~ encription, cryptography, WSN, UGS*

BEVEZTÉS

A félvezetők, később az integrált áramkörök feltalálása és kifejlesztése indította el az emberiséget azon az úton, amely néhány évtized leforgása alatt jelentősen átformálta életünket. Mára korábban elképzelhetetlen számítási kapacitást vagyunk képesek rendkívül kisméretű és kis fogyasztású eszközökbe sűríteni. Az adatátvitel ezzel összhangban fejlődött, sebessége, annak megbízhatósága, lokális és globális rendszere a közelmúlt tudomány és technológia adta lehetőségeivel megvalósíthatatlan lett volna. Az új készülékek új hordozható energiaforrásokat igényeltek, olyan akkumulátorok jöttek létre, amelyek a korábbiakhoz képest lényegesen nagyobb energiasűrűségre és teljesítménysűrűségre lettek képesek. Mindezen műszaki vívmányok megnyitották az utat a nagy számítási kapacitású, kommunikálni képes, kisméretű, mobil eszközök előtt.

A WSN[1] vagy vezeték nélküli szenzorhálózat fogalma a ma értelmezésében nem más, mint viszonylag sok, kisméretű, önállóan üzemelő, valamilyen fizikai mennyiséget érzékelni képes számítógépek összessége, amelyek vezeték nélküli kommunikációs hálózatba kapcsolódnak és kommunikálnak. Ezen rendszereket földön, vízen, levegőben, világűrben telepíthetőek, amelyek aztán jellemzően egy átjárón[2] keresztül kommunikálni képesek a hálózatot üzemeltető emberrel is.

UGS[3]-EK A HADSEREGBEN

Jelen írás elsősorban a szenzorhálózatok egy speciális és igen elterjedt válfajával a felügyelet nélküli, földi telepítésű szenzorhálózatokkal foglalkozik. Ezeket a rendszereket tipikusan kézzel telepítik, a felszín közelébe vagy betemetve. Jellemzően területvédelemre, behatolás észlelésre, megfigyelésre használják őket. A hadsereg igen nagy érdeklődéssel tekint e rendszerekre, az ok pedig triviális: ezeket a szenzorokat megfelelő módon kitelepítve, személyes jelenlét nélkül távolról és azonnal nyerhetünk információt az adott területen történő eseményekről, például arról, ha a területre valaki gyalogosan vagy gépjárművel behatol. A személyes jelenlét hiánya egyfelől nem teszi ki a megfigyelést végző katonát veszélynek, másfelől nem vonja el őt az egységétől, illetve számos egyéb előnyt is nyújt.[1]

Korábban a hadsereg technikai fejlesztései a civil felhasználás előtt jártak, és sok esetben a katonai fejlesztés megszelídített változata került civil felhasználásra. A jelenben ez a tendencia elsősorban a telekommunikációs, elektronikai fejlesztések tekintetében megváltozott. A hadsereg fejlesztési, tesztelési kapacitása messze elmarad a telekommunikációs világcégek képességei mögött. Nincs ez máshogyan az UGS-ekkel kapcsolatban sem. Nem meglepő tehát, hogy a hadsereg már meglévő megoldásokhoz nyúl. Konkrétan két területvédő UGS-ről tudható, hogy az Egyesült Államok hadserege érdeklődésére tart számot.[2] Ezek az *1. ábrán* látható MicroObserver[3] és a *2. ábrán* látható Scorpion II[4].

Mindkét rendszerről általánosságban elmondható, hogy tartalmaznak szeizmikus szenzorokat, látható fényű, illetve infravörös kamerákat. Ezeket könnyedén és gyorsan telepítve, a rendszer önmagától hálózattá alakul, amely hálózatot az átjárón, a Scorpion II esetében a műholdas Iridium rendszeren keresztül is el lehet érni. A kihelyezett eszközök akkumulátor kapacitása tipikusan jó pár hónapot, maximálisan két év üzemidőt tesz lehetővé. Működésmódjukra jellemző, hogy a szeizmikus szenzorok a közeledő gyalogost vagy járművet néhányszor tíz, maximum pár száz méteren belül érzékelik, ennek folyományaként aktiválják

1 WSN: Wireless Sensor Network - Vezeték nélküli szenzorhálózat
2 átjáró: gateway, különböző hálózatok között biztosít átjárást
3 UGS: Unattended Ground Sensor - Felügyelet-nélküli földi szenzor

az esetlegesen kihelyezett kamerákat, és értesítik a rendszer kezelőjét, aki innen „élőben” követheti a védendő területen történteket.

1. kép. A MicroObserver rendszer eszközei

2. kép. A Scorpion II rendszer összetevői

Ezek a rendszerek tehát speciális információkommunikációs rendszerek, amelyek csakúgy, mint bármely ilyen, számos módon támadhatók. Ezen támadások célja a működés megzavarásától, annak bénításán át egészen a rendszer feletti kontroll teljes átvételéig terjedhet.

Mielőtt megvizsgálnánk a védekezési lehetőségeket, azon belül is a kriptográfiai kihívásokat, tegyük fel a kérdést, hogy miért kell a szenzorhálózatokat egyáltalán védeni? Mondhatjuk azt, hogy a szenzorhálózatok nem tesznek mást, mint a körülöttük létező adatokat gyűjtik, feldolgozzák és továbbítják. Ezen adatok szabadon rendelkezésre állnak, a szenzorhálózatban ilyenformán semmi „titkos” nincs. Miért kellene szeizmikus, képi vagy bármi egyéb információhoz az ellenség szenzorhálózatán keresztül, azt feltörve hozzájutni, amikor egyéb módon valószínűleg sokkal egyszerűbb lenne? A válasz, hogy a szenzorhálózat a modern értelemben vett C4I rendszerek egyik *érzékszerve*. A szenzorhálózat egyike azon rendszereknek, amelyek segítségével a vezetés információt szerez egy bizonyos területen zajló tevékenységekről. Ezen rendszerbe bejutva kideríthető, hogy a szenzorhálózat üzemeltetője mit „lát”, továbbá, a szenzorhálózat működését megzavarva, a szenzorhálózat üzemeltetőjét tudom megzavarni, megtéveszteni. A szenzorhálózat lehallgatásával kideríthető, hogy milyen ingerre hogyan reagál a rendszer. Példaként, ha betörtem a rendszerbe, egyszerűen ki tudom deríteni, hogy a közeledésemet, a belépésemet a területre, amelyet a szenzorhálózat figyel, detektálja rendszer vagy sem. Manipulálva azt, el tudom titkolni a közeledésemet olyan módon, hogy azt a szenzorhálózat üzemeltetője nem veszi észre, vagy ami még veszélyesebb lehet, téves riasztást indíthatok, amellyel elterelem a szenzorhálózat tulajdonosának erőit egy támadás előkészítéseként.

A feltört, a szembenálló fél által kontrollált szenzorhálózat tehát nem csak haszontalanná válik az üzemeltetője számára, hanem egyenesen ellene is fordítható. Nem kérdés tehát, hogy ezeket a rendszereket informatikai és egyéb módszerekkel védeni kell, ezen rendszereknek információbiztonsági szempontból minél ellenállóbbnak kell lenniük.

MŰKÖDÉS KORLÁTOZÁSI LEHETŐSÉGEK

Az informatikai rendszerek biztonságát az alábbi öt szempont szerint vizsgálhatjuk: bizalmasság, sértetlenség, rendelkezésre állás, hitelesség és számon kérhetőség.[5]

Bizalmasságnak[4] nevezzük azt a tulajdonságot, amely jogosultsági rendszer segítségével csak az arra felhatalmazott felhasználót engedi az információhoz hozzáférni. Bizalmasság elvesztése azt jelenti, hogy az információ olyan kezekbe juthat, ahova nem szeretnénk.

Sértetlenségnek[5] nevezzük azt a tulajdonságot, amely biztosítja, hogy az információ a tárolás során ne változzon, sérüljön, illetve ne semmisüljön meg. A sértetlenség elvesztése a szándékolatlan adatmódosítást, adatvesztést jelenti.

Rendelkezésre állásnak[6] nevezzük azt az állapotot, amikor az információt a megfelelő időben és sebességgel el tudjuk érni. A rendelkezésre állás elvesztése azt jelenti, hogy az információ elérésében hosszabb-rövidebb nehézségek történnek. Nevezzük még időbeliségnek is.

Hitelességnek[7] nevezzük, ha az információ eredeti, forrása megbízható, ellenőrizhető. A hitelesség elvesztése azt jelenti, hogy az információ forrása nem megbízható illetve ellenőrizhető.

Számon kérhetőségnek[8] nevezzük, hogy a rendszerben a hozzáférések, változtatások visszakövethetőek, az események forrása és időpontja beazonosítható. Ezen képesség elvesztése azt jelenti, hogy nem tudjuk megmondani, hogy rendszerünk korábban milyen állapotban volt.

A működés korlátozása, és így a harcászati előny kivívása, a fenti tulajdonságok megsértésén keresztül valósulhat meg. Az alábbiakban egy rövid, de jó áttekintést adó leírást olvashatunk az informatikai rendszerek támadási lehetőségeiről. Ezekből kiemelem azokat, amelyek az UGS-ek szempontjából különösen relevánsak. Nem célom a támadásokat, azok mechanizmusát és egyéb tulajdonságait részletezni, hanem egy áttekintés szeretnék adni, amely érezteti, hogy kriptográfia mennyire fontos területe egy UGS-nek, de szinte bármilyen informatikai rendszernek.

Fizikai támadás

Mivel a node-ok[9] a legtöbb esetben a szabadban vannak telepítve és az esetleges álcán kívül semmi nem védi őket, esélye van annak, hogy a szemben álló fél a node-ot megtalálja és birtokába veszi. A fizikai támadás jelentheti a node egyszerű elpusztítását, de működésének tanulmányozása előkészíthet ennél nagyobb hatású beavatkozásokat is. A fizikai támadás egyik legnagyobb veszélye nem is a node elpusztítása, még csak nem is annak működési analízise, hanem a benne lévő kriptográfiai kulcs kinyerése, amire jó esély van, megfelelő műszerezettség esetén. A fizikai támadás a kulcs kinyerésével kombinálva igen nehezen védhető támadást okoz, hiszen a szemben álló fél képes lehet egy saját node-ot telepíteni a mi hálózatunkba.

Útvonalválasztás támadások

A szenzorhálózatok igen speciálisak abban a tekintetben, hogy egységei nem csak végpontok, de az *általuk használt hálózat útvonalválasztói* is egyben. Az útvonalválasztást célzó támadások, a hálózat által továbbított csomagokat térítik el, késleltetik, vagy semmisítik meg oly módon, hogy ezt a hálózat nem veszi észre. Az útvonalválasztás támadásával összességében

---

4 bizalmasság: angol terminológiában confidentiality
5 sértetlenség: angol terminológiában integrity
6 rendelkezésre állás: angol terminológiában availability
7 hitelesség: angol terminológiában authenticity
8 számon kérhetőség: angol terminológiában accountability
9 node-nak vagy mote-nak szokás nevezni a szenzorhálózat egyes elemeit

az adatcsomagok olyan hálózati pontokra kerülhetnek, ahova normális esetben nem, és ez valami miatt kedvez a támadónak. Az útvonalválasztásban természetesen nagy szerepe van az adatcsomagok headerjének[10]. Alapvető kriptográfiai kérdés, hogy csak az adattartalmat, vagy a headert is titkosítjuk. Erről részletesebben írok a Védekezési lehetőségek c. fejezetben.

Forgalom analízis

Az útvonalválasztás támadásával rokon fogalom, amely a szenzorhálózatok esetében kiemelt jelentőségű. A szenzorhálózatokkal történő kommunikáció, ahogyan azt a bevezetőben említettem, átjárón keresztül történik. Így legtöbb esetben igaz az, hogy minél közelebb esik egy node az átjáróhoz, annál több és valószínűleg értékesebb adat továbbítása is ráhárul. Az efféle támadással tehát felderíthetjük az átjáróhoz legközelebb eső node-okat, amelynek manipulálása aztán végzetes következményekkel jár a szenzorhálózat tulajdonosának a szemszögéből. Mit ér a szenzorhálózatunk, ha nem tudunk vele kommunikálni?

Szinkronitás támadás

Bár valójában „egyszerű” hálózat támadás, a szenzorhálózatok esetében kiemelt fontosságú a node-ok belső órájának és üzenetszámlálójának szinkronja. Ezen támadásokkal az időbeniség tulajdonságra lehet csapást mérni, ami mélyebb támadást tud előkészíteni. A legtöbb óraszinkron protokoll gyengének nevezhető a biztonság tekintetében, így könnyen támadható. Az óraszinkron szétcsúszása pedig a szenzorhálózat funkcionális széteséséhez vezet. Ugyanez elmondható az üzenetszámlálóra is. A témáról bővebben írok a Védekezési lehetőségek c. fejezetben.

Sybil

Ez az elnevezést használja az informatika azokra az esetekre, amelyben egy identitás (esetünkben node) nem a rendszer legitim része, hanem a támadó hozta létre művileg. Ez lehet egyszerű hamis node, de lehet hamis node-ok összessége illetve azokat szimuláló egység. Jelentősége abban rejlik, hogy ha a valójában vagy virtuálisan a sybil node-ok többségben lesznek, a hálózatot normál működésben is befolyásolni tudjuk. Példa lehet hamis routing táblák létrehozása és azok propagálása. A Sybil és az útvonalválasztás támadás egyik keveréke, az ún. HELLO támadás. Ilyen esetben egy, a szemben álló fél által telepített nagyobb teljesítményű rádióval felszerelt node képes „meggyőzni” távoli node-okat arról, hogy ő a legközelebbi node. Az ilyen node sok node forgalmát tudja magához vonzani, amellyel aztán a célnak megfelelő módon kezel.[6] A Sybil támadások ellen óriási jelentősége van a kriptográfiának azon részének, amely a hitelesség megvédésére irányul.

Mellékcsatorna támadás

Ide soroljuk azokat a támadási kísérleteket, amelyek nem közvetlenül az információs csatornát célozzák, hanem valamely azt kiszolgáló erőforrást. Ennek azért van értelme, mert ezen kiszolgáló erőforrások tulajdonságainak vizsgálatával következtethetünk arra az információra, amelyet el szeretnénk érni. Ilyen például, ha egy rendszer tápfeszültségének, tápáramának ingadozását monitorozzuk, és ebből következtethetünk arra, hogy a rendszer mikor milyen üzemmódban van. Hasonlóan a rendszer valamilyen lekérdezései és a válaszok között eltelt idő is adhat információt, amelyet felhasználhatunk. Szenzorhálózat esetén például abból a tényből, hogy a node-ok intenzívebben kezdenek kommunikálni egymással, következtethetünk, hogy behatolást észleltek, a kommunikáció részleteinek megértése nélkül. Nem kell tehát feltörni a kommunikációt, mégis meghatározható, hogy a hálózat milyen hatósugarú. Érezhető, hogy az ilyenfajta támadás ellen a kriptográfia sok esetben mit sem ér.

[10] header: egy adatcsomag metaadatai. Forrás, cél, stb. tartoznak ide.

Túlterheléses támadás

Jól ismert támadási forma, amely azon alapszik, hogy a rendszert akár legális, akár nem legális kérések sokaságával bombázzuk, amely ennek következtében nagyon lelassul, vagy akár le is áll. Ennek a támadásnak az a célja, hogy a rendszer a szolgáltatást ne tudja a megfelelő színvonalon nyújtani. Sok esetben a túlterheléses támadás, lehet egy más típusú támadás előkészítése, amely a túlterhelés miatt speciális üzemállapotba került rendszer gyengeségét használja ki. Nem tipikusan kriptográfiai vonatkozású támadás.

Programhiba támadás

Ez a támadás feltételezi, hogy valamilyen módon képesek vagyunk kommunikálni az általunk támadott rendszerrel. Legjellemzőbb formája a buffer túlcsordulás támadás, melynek során a rendszernek legális csatornán olyan nem legális bemenetet adunk, amely (vélt vagy valós) programhiba következtében memória korrupciót okoz. Hibás működést, fagyást, szélsőséges esetben olyan állapotot eredményezhet, amelyben a támadó saját kódot futtathat a megtámadott rendszeren. Nem tipikusan kriptográfiai vonatkozású támadás.

A WSN-ek biztonsági szempontból különösen érzékenyek a következő három sajátosság miatt:

1. A kommunikációra rádióhullámokat használnak, amely rádióhullámok nagy területen felfoghatók.

2. A fenti miatt is, a node-ok viszonylag könnyen felderíthetők, elmozdíthatók, szétszerelhetők, vizsgálhatók, visszafejthetők. A fizikai védelmi lehetőség közel nullára redukálódik, így szinte kizárólagos hangsúly fektetődik az információbiztonsági megoldásokra.

3. Az információbiztonsági megoldásokra kiváló, jól bejáratott módszerek léteznek, ám ezek a szenzorhálózatok jelentette speciális környezetben problémássá válnak. Elsősorban számítási kapacitás, és működési időtartam adta korlátokra kell gondolni.

VÉDEKEZÉSI LEHETŐSÉGEK

Miért speciális a szenzorhálózat kriptográfiai védelme?

Ahogyan azt a támadási lehetőségek bemutatásánál láthattuk, sokuk kizárható, vagy hatásuk minimalizálható, amennyiben a hálózat kommunikációja valamilyen mértékben és módon titkosított. Fogalmazhatunk úgy is, hogy a kriptográfia egy katonai célú szenzorhálózat megbízható működésének egyik alapeleme. A szenzorhálózatok sajátosságai azonban kérdőjelekhez vezetnek a kriptográfia megszokott alkalmazását illetően. Ezek a sajátosságok nagyrészt a node korlátozott számítási és működési kapacitásából következnek. A támadás nem csak a konkrét adatkommunikációt érhet, hanem a hálózat alapvető működését is. Ilyen a szinkronitás vagy az útvonalválasztás támadása. Mindezen veszély fennáll, ha a hálózati kommunikáció nem teljes egészében titkosított. Nincsen ez másképpen a megszokott asztali számítógépes környezetben sem. Ha egy banki weboldallal kommunikálok titkosított HTTPS[11] protokoll segítségével, attól még a hálózat alapvető működését nyújtó alsóbb rétegek protokolljai, mint IMCP[12], ARP[13] stb. titkosítatlan marad. Ezt nevezzük end-to-end titkosításnak.

Dönthetünk úgy is, hogy az egész kommunikációt titkosítjuk, a titkosítást az OSI rétegben minél lejjebb helyezve. Ennek megvalósulása a link-layer titkosítás, amelyre példa lehet a

11 HTTPS: HyperText Transfer Protocol Secure
12 IMCP: Internet Message Control Protocol
13 ARP: Address Resolution Protocol

WiFi-t kiszolgáló titkosítások. Ebben az esetben a hálózati kommunikáció teljes egészében titkosított lesz, azt lehallgatva még routing információkhoz sem juthatunk. Ennek nagy hátránya azonban a WSN-ek esetében, hogy mivel minden node egyben router is, minden node-nak vissza kell fejtenie a csomagokat, kinyerni a routing- és egyéb hálózatadminisztrálási információt, majd azt újra titkosítani és továbbküldeni.

A fent leírtak szorosan összefüggnek a kriptográfiai kulcskezelés problémájával is. Előbbi esetben kulcs csak a küldő és a végpont között szükséges, amely végpont lehet a szenzorhálózat információit kezelő/megjelenítő számítógép is. Utóbbi esetben azonban a szenzorhálózat minden node-jának ismernie kell a kulcsot, illetve nem utolsó sorban minden csomag továbbításánál kriptográfiai műveletet kell, végrehajtson. Azon túlmenően, hogy a kulcs egy befogott node-ból megfelelő eszközökkel kinyerhető, a nagyobb probléma, hogy a szenzorhálózatban a számítások processzor- és energia kapacitásának végessége miatt az ilyen számításigényes működésmód erősen korlátozza a hálózat teljesítményét és a node-ok működési idejét.

Összefoglalva tehát a WSN-ek esetében a kriptográfia elengedhetetlen fontosságú, használata azonban a kis processzorteljesítmény, korlátozott memória, korlátozott energia miatt új megoldásokat igényel, kimondottan szenzorhálózatok igényeit szem előtt tartó kriptográfiai módszereket. Az alábbiakban ezekből mutatok be néhányat, kitérve arra, hogy miért speciális.

Három, speciálisan WSN-hez kifejlesztett kriptográfiai módszer bemutatása

SPINS

A fent nevezett protokollt a Berkely egyetemen fejlesztették ki [7], kimondottan olyan node-okhoz, amelyek erőforrásai igen korlátosak. A használt node-ok fontosabb paraméterei az *1. táblázatban* láthatók. Bár a kutatás több mint tíz éves, így ma ennél nagyobb teljesítményű node-okkal számolhatunk, mégis jól példázza a szenzorhálózatok korlátos erőforrásait, és az egész problémakör sajátosságait. A SPINS valójában két protokollt foglal magába: a SNEP-et, amely a bizalmasságot, a hitelességet illetve a rendelkezésre állást biztosítja, valamint a µTESLA-t, ami pedig a kis teljesítmény igényű üzenetszórásos[14] adásmódot tesz lehetővé.

1. táblázat. A SPINS fejlesztéséhez használt WSN egyik nodejának specifikációja

CPU	8bit, 4MHz
Tár	8 kbájt utasítástár 512 bájt RAM 512 bájt EEPROM
Kommunikáció	916 MHz
Sávszélesség	10 Kbps
Oprendszer	TinyOS
OS méret	3500 bájt
Elérhető tár	4500 bájt

A SNEP sajátosságai az alábbiak:

1. Mindössze 8 bittel növeli az adatcsomag méretét.
2. Nem továbbítja az üzenetszámlálót az adatcsomagban.
3. Szemantikai titkosítást használ. (Lentebb kifejtve.)
4. Bizalmasságot, hitelességet és rendelkezésre állást biztosít.

[14] Üzenetszórásos adásmód: angol terminológia szerint broadcasting, olyan adásmód, amikor egy adó és sok vevő van.

A kriptográfiai algoritmust akkor nevezzük szemantikusnak, ha egy támadó, akkor sem tud a plaintext-re[15] következtetni a cyphertext-ből[16], ha lehetősége van bármilyen rövid plaintext-et adni, illetve azon bármilyen kis változtatást követően a cyphertext-et újra tanulmányozni. Az algoritmus nem adhat ugyanarra a plaintext-re ugyanolyan cyphertext-et, illetve a plaintext kis változásai is nagy változást kell, hogy okozzanak a cyphertext-ben. E tulajdonság elérésének egy alapvető eleme a nonce-ok[17] generálása és felhasználása. A hagyományos módszer szerint egy számláló érték kerül az üzenettel egyidejű elküldésre, amely megvalósítástól függően vagy valamilyen módon kódolva van a véletlen számmal vagy nem. A lényeg, hogy a vevő egy sorszámot is kap az üzenethez, és így tud az üzenetek sorrendiségéről meggyőződni. Ha kimaradnak értékek a sorban, tudja, hogy üzenetek vesztek el, ha többször jön ugyanaz, hálózati hibára vagy újrajátszás támadásra lehet gyanakodni. Az ilyen módon létrehozott üzenet azonban nagyobb, több energiát igényel előállítása és elküldése, mint ami a szenzorhálózatok esetében rendelkezésre áll. Ezért a SNEP máshogyan éri el a szemantikai titkosítást. A sorszámok elküldése helyett küldő és vételi oldalon is egy önálló számlálót valósít meg. A sorszám továbbra is része a titkosításnak, így amikor a visszafejtéshez szükség van a számláló értékre, ez esetben nem az üzenettel jön, hanem a vevő saját értékét használja fel. Ha nem sikerül az üzenet dekódolása, a vevő gyanakodhat arra, hogy elcsúszott a szinkron, és egy külön számláló szinkronizálásra kifejlesztett protokollal kérheti az üzenetszámlálója szinkronizálását. Tehát amíg nincs túl sok csomagkiesés, addig e protokollnak nyilván való az előnye: egyszerűen kispórolja az üzenetből a számlálót. Ha sok csomagkiesés van, a gyakori számláló kérelmek miatt hatása ellentétes lehet.

A protokollról elmondhatók az alábbiak:

Szemantikus titkosítás: A számláló vételi és adó oldalon egyaránt minden adatcsomag után nő, amely nonce-ként használatos, tehát minden adatcsomag máshogyan titkosítódik. A számláló maximális értéke elegendően nagy ahhoz, hogy a gyakorlati felhasználásban ne ismétlődhessen.

Hitelesség: Amennyiben a SNEP üzenet-hitelesítő protokollja az üzenetet megerősíti, a vevő bizonyos lehet abban, hogy az üzenet a feladótól származik.

Újrajátszás állóság: A számláló biztosítja, hogy ugyanazt az üzenetet ne lehessen tárolni, majd támadási szándékkal újra kiküldeni.

Időbeliség: A szenzorhálózatokban, akárcsak bármely csomagkapcsolt hálózatban, nem lehet garanciát nyújtani arra, hogy az üzenetek a küldés sorrendjében érkeznek a vevőhöz. Amennyiben az üzenet ellenőrzés sikeres, akkor a vevő a beérkező üzeneteket sorrendbe tudja állítani.

Alacsony teljesítményigény: A számláló nem kerül elküldésre üzenetenként.

A protokoll-csomag hitelesített üzenetszórásos adásmódjáért felel a µTESLA protokoll. A µTESLA a TESLA protokollt fejlesztette kimondottan a szenzorhálózatok jelentette csökkentett erőforrású környezethez. A TESLA protokoll az adást digitálisan aláírja, amelyhez aszimmetrikus kulcsrendszerre van szükség. Ezek az algoritmusok nagy kulcsot követelnek, amely már önmagában jelentős területet vesz el a rendelkezésre álló kis memóriából. A TESLA 24 bájttal növeli egy csomag méretét, amely PC-s környezetben nem probléma. Azonban a szenzorhálózatok tipikusan nagyon kisméretű csomagjaihoz viszonyítva ez az érték akár azt is

[15] A kriptográfiában plaintext-nek hívják a titkosítás előtti adatot. (Nem kell feltétlen szöveges állományra gondolni.)

[16] A kriptográfiában cyphertext-nek nevezik a kriptográfiai eljárással a plaintext-ből létrehozott, titkosított adatot.

[17] nonce: egyszer használatos véletlen szám.

jelentheti, hogy a hálózatot, így az adót és a vevőt 50%-ban csak a kriptográfia jelentette overhead-del[18] terheljük, ami nyilván valóan célszerűtlen.

Ennek megfelelően a µTESLA az alábbi fejlesztéseket tartalmazza:

1. A TESLA aszimmetrikus kulcsú rendszerével szemben szimmetrikus kulcsokat használ.

2. A TESLA minden üzenetbe kulcsot illeszt, a µTESLA csak egyszer küldi el azt.

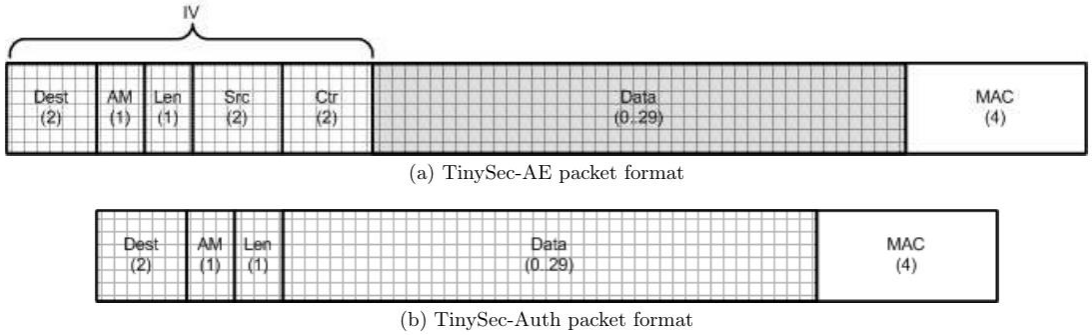
3. A küldők kulcsait tárolni költséges egy node képességeihez viszonyítva, így a µTESLA esetében korlátozva van a küldők száma, amelyet maximálisan hitelesíteni képes.

TinySEC

A TinySEC[8] sok tekintetben hasonlít a fentebb említett SNEP-re. Hitelességet, bizalmasságot, adat sértetlenséget és időbeliséget nyújt. Ám a TinySEC fejlesztői szerint a SNEP-et nem fejezték be teljes egészében ahhoz, hogy éles körülmények között is lehessen használni. Azon túlmenően, hogy a TinySEC bevethetőségét is bizonyította, jelentős különbsége, hogy nem használ számlálókat a kriptográfiai folyamatban. A titkosításhoz CBC[19]-t használ, az hitelesítéshez pedig CBC-MAC[20]-et.

A CBC olyan kriptográfiai eljárás, amely egy következő blokk plaintext-et kizáró-vagy kapcsolatba hozza a jelenlegi cyphertext-tel. Ilyenformán minden cyphertext blokk függ a korábbi plaintext blokkoktól. Amennyiben az első kizáró-vagy művelethez inicializáló vektort használunk a teljes titkosítási művelet egyedi lesz, kielégíti a korábban említett szemantikus titkosítás feltételét is. A CBC egyúttal hitelességet is nyújt (CBC-MAC), amennyiben zérus inicializáló vektort használunk.

A TinySEC két csomagformátumot használ. A TinySEC-auth az üzenet autentikációhoz, valamint az TinySEC-AE, az hitelesített és titkosított üzenetekhez. Ez utóbbi esetben a csomag adattartalma 29 bájt lehet, amelyhez 8 bájtos header tartozik. Csak az adattartalmat titkosítják, de a MAC számításba a headert is beleveszik. A TinySEC-auth csomag ettől annyiban különbözik, hogy 4 bájttal rövidebb a header, valamint az adattartalom nincs titkosítva. (1. ábra)



1. ábra. A TinySEC-AE és TinySECAuth adatcsomagok felépítése

A 4 bájtos MAC a CBC-MAC esetében rövidnek mondható, hiszen 8-16 bájtot használnak más protokollok. A MAC mérete erősen összefügg a protokoll biztonságával. Négy bájt 2^{32} lehetőséget, jelent, azaz ennyi próbálkozásból egy támadónak biztosan sikerül egy érvényes MAC-et létrehoznia, így az üzenet a vételi oldalon hitelesnek tekintett. Ez elsőre igen gyenge védelemnek tűnik, de fejlesztői érvelése szerint, a protokoll által megcélzott kis teljesítményű szenzorhálózatok tipikusan 19200 bps sebességű kommunikációt használnak, így egy ilyen elárasztásos támadás megközelítőleg 20 hónap alatt vinne bizonyos sikerre. A

[18] Overhead: az értékes adattartalmon felüli, de a kommunikációhoz szükséges része az adatcsomagnak.

[19] CBC: Cipher Block Chaining - Kódblokk láncolás

[20] MAC: Message Authentication Code - üzenethitelesítő kód

protokollt bemutató publikáció szerint ez az idő sokkal több, mint egy átlagos node akkumulátor adta működési időkerete, így elegendőnek mondható.

Mivel az UGS-ek folyamatos kommunikáció mellett biztosan lemerülnek, így ez a támadási lehetőség nem jelent veszélyt. A MINISEC fejlesztői azonban nem látták ennyire optimistán a kérdést, így ezen a hiányosságon javítottak.

MINISEC

A MINISEC fejlesztését[9] két tényező motiválta. A fentebb említett TinySEC sebezhetőségét komolyabbnak értékelték, mint az azt fejlesztők, illetve a biztonság tekintetében elégségesnek gondolt ZigBee protokollt túlságosan erőforrás igényesnek ítélték. Olyan protokollt akartak tehát alkotni, amely magában foglalja a TinySEC csekély erőforrásigényét, és a ZigBee komolyabb biztonságát is.

A célt alapvetően három eszköz szolgálja. Az egyik OCB[21] módusú titkosítás. Ahogyan azt a TinySEC-nél láthattuk két külön adatcsomagra van szükség a bizalmasság és a hitelesség eléréséhez. Ez szenzorhálózatos környezetben nagy áldozatot jelenthet, a nagyobb adásigény miatt. Az OCB mód egyetlen CBC-ban egyesíti e kettőt, a következő módon. Jelöljük M-mel a tetszőleges hosszúságú plaintext üzenetet, H-val a headert, K-val a kulcsot, és N-nel a nonce-ot (esetünkben ez egyet jelent a számlálóval). Első lépésben M, K és N felhasználásával generálódik a cyphertext magja, jelöljük C-vel. Mindeközben az algoritmus elkészít egy előre meghatározott hosszúságú toldalékot is. Végső soron az OCB_K (N, M, H) függvény eredménye a (C,toldalék) páros lesz. A vevő a kulcs ismeretében visszafejti C-t M-é, és megvizsgálja a toldalékot is. Amennyiben a vevő más toldalékot számol ki, mint amely az üzenetben jött, akkor az érvénytelennek tekintendő.

A második jelentős fejlesztés az inicializáló vektort érinti, helyesebben szólva az üzenetszámlálót. A TinySEC megvalósításában a számláló maga az inicializáló vektort, így az minden üzenettel elküldésre kerül. A SNEP, ahogyan fentebb írtam nem küld számlálót, hogy az erőforrásokon spóroljon. A MINISEC megítélése szerint ez két szükségtelen végletnek tűnik. A kompromisszum egy SNEP-hez hasonló számlálókezelés, azonban annak csak az utolsó n bitjét küldik el az üzenettel. Így a vevő az utolsó 2^n üzenet időbeliségéről tud meggyőződni. Addig, amíg az egymás után elveszett üzenetek száma kevesebb, mint 2^n, ezt a vevő könnyen észreveszi, és szinkronizálja saját üzenetszámlálóját. Amennyiben ezen időkeretet meghaladja a kieső csomagok száma, működésbe léphet a SNEP-ban megismert szinkronizáló protokoll. Ez a megoldás tehát kisebb overhead-et jelent, mint minden üzenetben számlálót küldeni, de sokkal nagyobb biztonságot a SNEP túlságosan nagyvonalúnak nevezhető számláló nélküli megoldásához. A szenzorhálózatok adatcsomagjai tipikusan néhányszor tíz bájtosak, így jelentős változás, ha egy számláló 32 bit helyett 16 vagy akár csak 4 bit méretű.

A harmadik vívmány annak felismerésén alapszik, hogy célszerű más titkosítást használni unicast (MINISEC-U) illetve broadcast (MINISEC-B) üzemmódokban. Unicast esetén célravezetőbb a fentebb írt csökkentett bitszámú számlálós megvalósítást használni, míg broadcast esetén Bloom szűrőt[10] használnak. A Bloom szűrő nagy energia-hatékonysággal meg tudja állapítani, hogy egy keresett elem bizonyosan nincs-e egy halmazban. (Az ellenkezőjét nem.) Ezt a tulajdonságát felhasználva az ún. újrajátszási támadásokat lehet jól kiszűrni, amelyek során a támadó rögzít egy korábbi legitim üzenetet és újraküldi azt. Végső soron biztonsági szempontból ugyanez a szerepe a számlálóknak is az üzenetekben.

A MINISEC protokoll a fentebb leírt továbbfejlesztésekkel impozáns eredményeket hozott, amely a 2. táblázatban látható összefoglalva. Az összehasonlításhoz választott másik protokolloknál kisebb energiaigényű és biztonságosabb is. Egyetlen tekintetben marad el, ez pedig a szükséges memória mérete. A technikai fejlődést tekintve megfigyelhető, hogy a

[21] OCB: Offset Codebook mode

processzorok, memóriák kapacitása lényegesen gyorsabban fejlődik, mint az akkumulátoroké. Ilyen perspektívába helyezve jó kompromisszumnak látszik valamivel több memóriáért cserébe erősebb titkosítást, és hosszabb üzemidőt nyerni.

2. táblázat. A MiniSec sarokparaméterei, más protokollokkal összehasonlítva.

	Payload (B)	Packet Overhead (B)	Security Overhead (B)	Total Size (B)	Energy (mAs)	Increase over TinyOS
TinyOS	24	12	–	36	0.034	–
TinySec	24	17	5	41	0.0387	13.9%
SNEP	24	20	8	44	0.0415	22.2%
MiniSec	24	15	3	39	0.0368	8.3%

ÖSSZEGZÉS

A szenzorhálózatok, szűkebb értelemben az UGS-ek a hadsereg aktív eszköztárába tartoznak. Nagy területek megfigyelését lehet rájuk bízni, ilyen módon helyőrségeket, őrjáratokat lehet „megspórolni”. Mivel a C4I rendszerek érzékszervének tekinthetők, különös jelentősége van annak, hogy a szemben álló fél elektronikai ellentevékenysége ellen védjük ezt a rendszert. A szenzorhálózat megfigyelésével, de különösen annak informatikai rendszerébe történő behatolással, saját fegyverünket fordítják ellenünk, hiszen ami eddig érzékszerv volt, onnantól a szemben álló fél használhatja a mi megtévesztésünkre. Ezen rendszerek védelmében kiemelt jelentőséggel bírnak a kriptográfiai megoldások.

Az UGS-ek olyan információtechnológiai alapmegoldásokra épülnek, amelyek régóta körülvesznek bennünket, a számítástechnikával és a modern matematikával együtt fejlődtek és mára olyan stabillá váltak információbiztonsági érelemben, hogy az emberiség gyakorlatilag bármit rájuk mer bízni. (Gondoljunk a bankrendszerre vagy akár katonai vezetési rendszerekre.) Az katonai UGS-ek esetében az igény az erős biztonságra tehát magas, ám a szenzor node-ok adta szűk erőforrásokkal a bejáratott megoldások nem valósíthatók meg. Olyan megoldások szükségesek, amelyeket ehhez a korlátozott és speciális környezethez alakítottak.

Három fejlesztést mutattam be, és igyekeztem a terjedelem adta korlátokon belül azokra a pontokra fókuszálni és azokat röviden megmagyarázni, ahol tetten érhető, hogy pontosan milyen módon különbözik a szenzorhálózatok adta környezet, milyen irányú fejlesztésekre van szükség ahhoz, hogy a kriptográfia bonyolult rendszere működhessen bennük.

A szenzorhálózatok rendkívül kis adatcsomagokat küldenek, az adás pedig rádióhasználatot jelent, amely az egyik legkritikusabb erőforrást, az akkumulátort használja jelentős mértékben. Nem túlzás azt állítani, hogy a kiküldött adatcsomagokban „minden bit számít”, ha a kiküldött csomag akár néhány bittel rövidebb, az már jelentős eredmény a node kommunikációs telephasználatát illetően. A bemutatott eljárások olyan megoldásokat példáznak, amelyek létező kriptográfiai módszereket alakítanak át oly módon, hogy azok rövidebb üzenetet eredményezzenek.

Felhasznált irodalom

[1] Palm, B.C. & Richter, R.P.: Mobile situational awareness tool: unattended ground sensor-based remote surveillance system (Msc Thesis). (2014).

[2] ERIK SCHECHTER: Army explores new networked ground sensors. C4ISR & Networks [Online] http://archive.c4isrnet.com/article/20140829/C4ISRNET08/308290003/Army-explores-new-networked-ground-sensors (2015.04.05.)

[3] Textron Systems: MicroObserver®. [Online] http://www.textronsystems.com/products/weapon-sensor/microobserver (2013.12.26.)

[4] NORTHROP GRUMMAN CORPORATION: SCORPION II Unattended Target Recognition Systems. [Online] http://www.northropgrumman.com/Capabilities/SCORPIONII/Pages/default.aspx (2014.01.03.)

[5] William Stallings: Cryptography and Netrwork Security. V.. . 1. Lake Street, Upper Saddle River, NY 07458, Pearson Education Inc. (2011)

[6] Chris Karlof, D.W.: Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures. AdHoc Networks Page: 299-302., (2003).

[7] ADRIAN PERRIG, ROBERT SZEWCZYK, J.D.TYGAR, VICTORWEN, DAVIDE.CULLER: SPINS: Security Protocols for Sensor Networks. ACM Journal of Wireless Networks, (2002).

[8] C. Karlof, N. Sastry, and D. Wagner: TinySec: a link layer security architecture for wireless sensor networks. 2nd international conference on Embedded networked sensor systems , (2004).

[9] M. Luk, G. Mezzour, A. Perrig, and V.Gligor: MiniSec: A Secure Sensor Network Communication Architecture. IEEE International Conference on Information Processing in Sensor Networks (IPSN'07), (2007).

[10] Wikipedia: Bloom filter. [Online] http://en.wikipedia.org/wiki/Bloom_filter (2015.04.16.)