Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar

Biztonsági célú átfedő hálózat

Doktori értekezés tézisfüzete

Szerző: Czirkos Zoltán
okleveles villamosmérnök

Témavezető: Dr. Hosszú Gábor
egyetemi docens
műszaki tudomány kandidátusa

Elektronikus Eszközök Tanszéke
Budapest, 2011.

1. Bevezetés

Az Internetre kötött számítógépeket képzett támadók, rosszakaratú programok, vírusok és férgek támadásai érik. A támadások számának növekedése, és a védendő rendszerek komplexitása miatt megjelent az igény az automatikus betörésérzékelő és -védelmi rendszerek létrehozására.

Ezek megvalósítása azonban rengeteg elméleti és gyakorlati problémát vet fel. A támadások egy része nem jár együtt önműködően észlelhető jelekkel. Amelyek igen, azok érzékeléséhez gyakran hatalmas mennyiségű adatot kell feldolgozni. A támadások jelentős része hálózati szintű, vagyis azok nem egy, hanem több gazdagépet vagy esetleg azok egy alhálózatát érinthetik egyszerre [1]. Ebben az esetben előfordulhat, hogy egy adott támadáshoz köthető különféle események a hálózat különböző pontjain keletkeznek. Ilyenkor az érzékelő rendszerben az érzékelőknél rögzített események összegyűjtését, és az azok közötti kapcsolat felismerését is meg kell oldani.

A betörésérzékelés már egy önálló érzékelő esetén is komoly számítási kapacitást igényelhet. A feldolgozandó adatok egy részének figyelmen kívül hagyásával csökkenthető ugyan a feldolgozáshoz szükséges idő, azonban ezzel az érzékelés hatékonysága romlik. Egyes esetekben a rendszer a hiányos adatok miatt figyelmen kívül hagyhat támadásokat, máskor hamis pozitív riasztásokat indíthat azok miatt. A több érzékelőtől származó adatok feldolgozása esetén az igényelt számítási, hálózati és tárolási kapacitás is meredeken növekszik az érzékelők számával.

A kutatások célkitűzései

A jelenleg elterjedt betörésérzékelő rendszerek nagy részének egyik gyengesége, hogy az egyes érzékelt események maguknál az érzékelőknél maradnak [2]. A kutatás célja egy olyan új elvű rendszer kidolgozása, amely ezek megosztását hatékonyan képes elvégezni. Így lehetővé

válik az újrahasznosításuk – egy felismert támadó ellen nem csak az érzékelővel összekötött gazdagép vagy alhálózat, hanem más védendő egyedek is védekezni tudnak. Az önmagukban támadásra még nem utaló, de összetett támadások részét képező események megosztása pedig lehetővé teszi a hálózati szintű támadások érzékelését.

A kifejlesztett, érzékelésre alkalmazott elosztott rendszernek megbízhatóan és stabilan kell működnie hálózati hibák esetén. Különösen fontos ez amiatt, mert egy támadás alatt álló hálózat megbízhatósága alacsonyabb lehet. Fontos célkitűzése ezért a rendszer tervezésének az is, hogy képes legyen az érzékelésekkel kapcsolatos feldolgozást, és az azáltal okozott terhelést minél jobban elosztani az egyedei között.

2. Felhasznált eszközök és vizsgálati módszerek

A P2P átfedők kísérleti vizsgálata a nagy egyedszám miatt általában nehezen kivitelezhető. A nagytérségi hálózatokon történő tesztelés komoly erőforrásokat és együttműködést igényel a kutatóhelyektől [3]. A tervezett eljárásokat ezért gyakran kísérletek helyett szimulációval vagy matematikai úton igazolják.

Az értekezésem témáját képező Kademlia átfedő hálózat [4] vizsgálatát is ezen a két úton végeztem. Az elérhető szimulátorok hiányosságai miatt saját szimulátor alkalmazást fejlesztettem, amely megvalósítja a Kademlia szimulációjához szükséges eljárásait. A szimulátor a fizikai hálózat hibáit a szakirodalomban bemutatott mérések, hibaeloszlások alapján modellezte. A valós, fizikai hálózat hibáinak eloszlását közelítésekkel figyelembe véve a bemutatott modell ellenőrizhető analitikusan is.

A Kademlia átfedőre kidolgozott üzenetszórás algoritmus vizsgálata is bizonyos közelítések mellett lehetséges volt analitikusan. Ezen közelítések miatt azonban megbízhatóság növeléséhez alkalmazott replikáció esetében már pontatlanná váltak az eredmények. Emiatt szükségessé vált az algoritmus szimulátorban történő vizsgálata is. Ezt a szimulációt is saját alkalmazásomban implementáltam,

a szakirodalomban közölt, valós hálózatokon mért késleltetési idő és hibaeloszlásokon.

A bemutatott betörésvédelmi eljárás működését kísérleti úton vizsgáltam. A létrehozott teszt átfedő három éven keresztül végzett betörésérzékelést és adatgyűjtést. Az így összegyűlt adatokat elemezve ellenőrizhetővé vált az eljárás működése, és igazolhatóvá annak hatékonysága.

3. Új tudományos eredmények

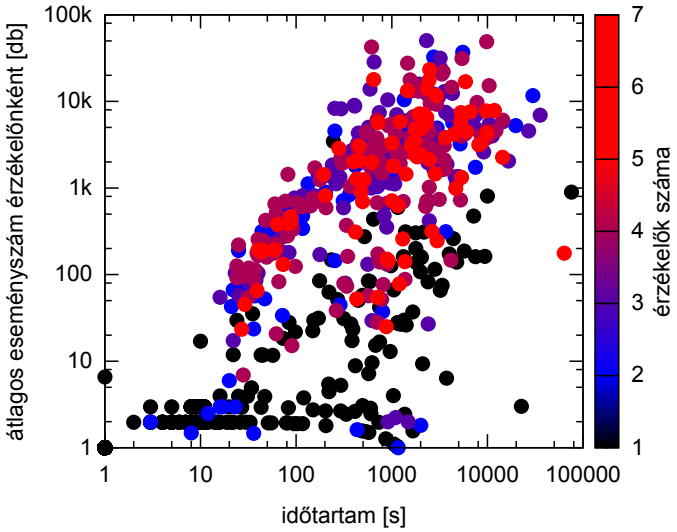
3.1. Első tézis

1. tézis. *Kidolgoztam egy hálózati biztonsági eljárást (Komondor), amellyel a hálózat eltérő pontjain történő betörésérzékelések összesített tapasztalatait felhasználva javítható a gazdagépek védelme. [J1, J4, J7, J6, J8, C4, C5, B1]*

A bemutatott *Komondor* eljárás a veszélyt jelentő hálózatot használja arra, hogy az onnan érkező támadások ellen a védekezést hatékonyabbá tegye.

Lényege, hogy a védendő gazdagépek önműködően egy *alkalmazási szintű hálózatot* hoznak létre, amelyen keresztül kapcsolatot tartanak egymással. Az érzékelt betörésekre vagy azok kísérletére utaló jeleket egy jelentésként elküldik a hálózatba. Ha az összegyűlt tapasztalatok támadásra utalnak, akkor az összes résztvevő értesítést kap erről. Ilyenkor azok a szükséges védekező lépéseket megtehetik.

A Komondor hatékonysága abban rejlik, hogy a gyanús, betörési kísérletekre utaló események összegyűjtésekor keletkező tudásbázis nem marad az egyes egyedeknél kihasználatlanul, hanem az összes résztvevő számára hasznossá válik. A több gazdagépet érő támadások esetén (1. ábra) így egymás védelmét erősíthetik. A módszerrel lehetővé válik támadások megelőzése is, hiszen egy érzékelés után a többi résztvevő a védelmet azelőtt kiépítheti, hogy a támadás azokat is érintené.

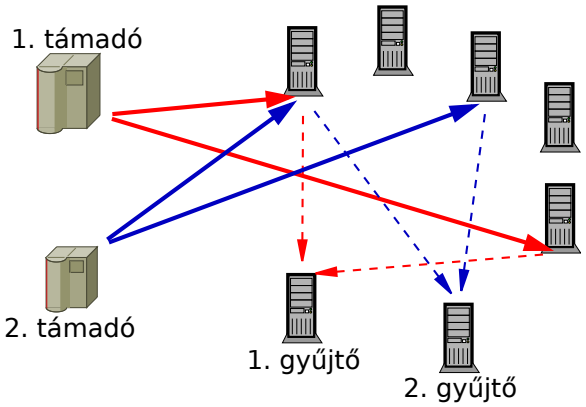


1. ábra. Támadó féregprogram bejelentkezési kísérletei [5], nem létező felhasználói nevekkel és jelszavakkal

1.1. altézis. *Kifejlesztettem egy módszert, amellyel a különböző helyeken érzékelt betörési kísérletekből nyert tapasztalatok összegezhetők, és azokból egy értékelési eljárással elosztott adatbázis építhető. Bebizonyítottam, hogy a kidolgozott eljárás hatékony működésének feltétele, hogy az egyes csomópontokban működő példányok közötti kommunikáció egy DHT típusú átfedő hálózatra épüljön. [J1, J2, J4, C1, C2, B4, B5, B6, B7, B9]*

A *támadásokra* általában több *gyanús esemény* érzékelésével és a közöttük lévő *kapcsolat felismerésével* (attack correlation) lehet következtetni. A kapcsolat felismerésére a szakirodalomban több módszert ismertetnek [6, 7, 8], amelyek eltérő jellegű támadások esetén használhatóak. A Komondor eljárás lehetővé teszi, hogy ezeket a módszereket elosztottan lehessen használni.

A Komondor egyedei minden olyan gyanús eseményt rögzítenek,



2. ábra. Betörésérzékelés a Komondor eljárással

amelyik támadás része lehet. Az érzékelt eseményekhez az egyedek egy *pontszámot* és egy vagy több *kulcsot* és rendelnek. A kulcsok az eseményekhez egyes tulajdonságaik alapján rendelhetők, és azok adják a kapcsolat felismerésének alapját. Eltérő helyeken keletkező, de egymással esetleg kapcsolatban lévő eseményekhez ugyanazt a kulcsot kell rendelni. Ez lehet például az érzékelt esemény forráscíme (IP címe) – így ha a konkrét támadóhoz köthető események külön érzékelőknél keletkeznek, a rendszer akkor is érzékelni fogja a támadást.

A pontszám az esemény fontosságát jelzi, amely annak jellege alapján határozható meg. Az érzékelés szempontjából fontosabb események magasabb pontszámot kapnak. Ha a pontszámok összege egy adott határértéket meghalad, az eseményeket támadásként kezeli a rendszer.

A Komondor egyedek által létrehozott alkalmazási szintű hálózat egy *strukturált peer-to-peer (P2P) átfedő*, amely egy *elosztott hasító táblát* (distributed hash table, DHT) valósít meg. A DHT átfedők kulcs-érték párokat tárolnak. Minden adat (érték) tárolására az átfedőben egy egyedet jelölnek ki a hozzárendelt azonosító (kulcs) alapján. A Ko-

mondor eljárásban ez a támadáshoz rendelt kulcsok alapján történik. Egy adott támadással vagy támadóval kapcsolatos események, mivel ugyanazt a kulcsot rendelik hozzájuk az érzékelők, ugyanahhoz az egyedhez kerülnek. Ez a *gyűjtő egyed* (2. ábra). Ez végzi az események feldolgozását, és indítja el a riasztást az átfedőn, ha az összegyűlt tapasztalatok támadásra utalnak.

Ennek a felépítésnek számos előnye van. Egyrészt a P2P modell egy stabil, önszerveződő és megbízható átfedő hálózatot biztosít a tapasztalatok megosztásához. Hálózati hiba vagy egyed kiesése esetén az átfedő automatikusan újrarendeződik, és a kiesett egyedet pótolja. A gyűjtő egyed meghibásodása esetén annak szomszédja veszi át az adott kulcshoz tartozó események feldolgozását. Másrészt különböző kulcsok tárolására más-más egyedet jelöl ki az átfedő. Ezzel biztosítja a terheléselosztást ebben a betörésérzékelő alkalmazásban is, mert eltérő támadók adataiért eltérő gyűjtő egyedek felelnek; ugyanakkor egy adott támadás adatai, származzanak bármely érzékelőtől, a hozzárendelt kulcs miatt mindig ugyanahhoz a gyűjtő egyedhez kerülnek. Az adatok feldolgozása így hatékony, a P2P átfedők stabilitását ötvözi a központosított rendszerek előnyeivel.

1.2. altézis. *Igazoltam, hogy a DHT átfedőre épülő betörésérzékelés során létrejövő hálózati forgalmat a Kademlia hálózat hatékonyabban kezeli, mint az egyéb ismert DHT átfedők. [J8, C1, B1, B7, B8]*

A szakirodalomban ismertetett P2P átfedők topológiája eltérő, és így azok eltérő jellegű terhelésnél más hatékonysággal működnek. A *Kademlia* az egyedeit egy bináris fába rendezi, amelyben iteratív útválasztást használ [4]. Eszerint az egyedek (a többi DHT átfedőtől eltérően) az eltárolt adatokat nem egymás közt továbbítva juttatják el a kulccsal kijelölt egyedhez, hanem a tárolást kezdeményező egyed a többi segítségével megkeresi a tárolásra kijelölt egyedet, és utána közvetlenül kommunikál azzal.

A betörések érzékelésénél gyakori, hogy sok, ugyanazzal a támadóval kapcsolatos esemény keletkezik az érzékelő egyedeknél (1. ábra).

Átfedő	Chord	Kademlia
Egyed megkeresése	0	$O(\log_2 N)$
Első tapasztalat küldése	$O(\log_2 N)$	$O(1+\log_2 N)$
n **db tapasztalat ua. kulccsal**	$O(n \cdot \log_2 N)$	$O(n+\log_2 N)$
Üzenetszám határértéke tapasztalatonként	$O(\log_2 N)$	$O(1)$

1. táblázat. A protokollüzenetek száma a betörésérzékelésben, n egyedszámú átfedőben

Az azonos hozzárendelt kulcs miatt ilyenkor a feldolgozásért felelős gyűjtő egyed ugyanaz. A Kademlia használatával az első esemény adatainak küldése után a gyűjtő egyed hálózati címe bekerül az érzékelő gyorsítótárába. A továbbiakban az üzenetküldés már csak az érzékelő és a gyűjtő egyed között folyik, vagyis nem terheli tovább az átfedőt. A hálózati terhelés ezáltal mindössze egy üzenet tapasztalatonként (1. táblázat). Ez független az átfedő méretétől is, így nagytérségi alkalmazás, azaz nagy egyedszámú átfedő esetén a Kademlia használata különösen indokolt.

3.2. Második tézis

2. tézis. *Új eljárást dolgoztam ki a Kademlia DHT átfedő rendszerszintű működési jellemzőinek beállítására. [J1, J2, C1, B2]*

A Kademlia átfedőben az iteratív útválasztási eljárás miatt egy egyed akár minden egyes lekérdezésnél más egyedekkel léphet kapcsolatba. Két egyed között azonban nem mindig lehet összeköttetést létrehozni. Ennek oka az, hogy a tűzfalak és hálózati címfordítás mögött lévő egyedek nem tudnak bejövő kapcsolatokat fogadni, csak

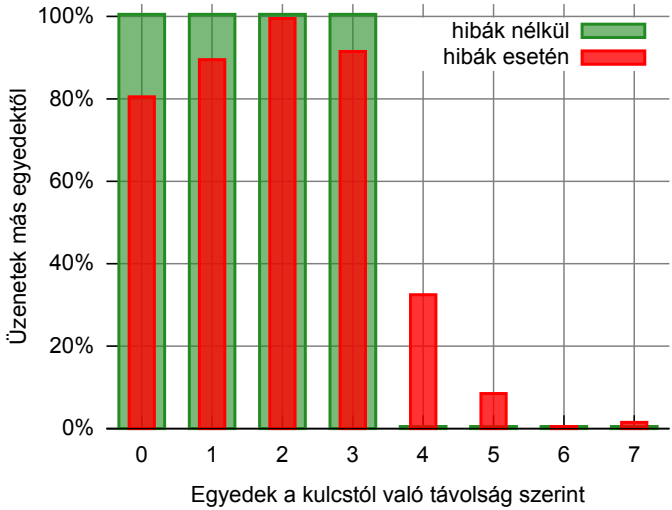
kimenőeket létesíteni [9]. Emiatt a lekérdezések meghiúsulhatnak. Előfordulhat, hogy egy kulcs-érték párt egy egyed tárol, de mások nem tudják lekérdezni azt. Lehetséges az is, hogy eltárolásnál nem a megfelelő egyedhez kerülnek az adatok, mivel az azt kérvényező egyed nem tudja elérni a kijelölt egyedet.

Replikációval (replication), vagyis több helyen történő eltárolással növelhető annak valószínűsége, hogy lesz olyan tároló egyed, amelyet el tudnak érni a lekérdezők. Ez viszont a hálózati forgalmat és az egyedektől elvárt tárolókapacitást is növeli. A kidolgozott eljárás célja ezért egy olyan *minimális replikációs szint* meghatározása, amellyel egy elvárt rendelkezésre állási szint mellett biztosítható az adatok elérhetősége.

2.1. altézis. *Kidolgoztam egy modellt, amely alapján a Kademlia átfedőben analitikusan és numerikus módszerrel is meghatározható a replikáció szintje úgy, hogy az adott hálózati viszonyok, mint peremfeltételek mellett az átfedő biztosítsa az egyedeinél tárolt adatok elvárt rendelkezésre állási szintjét. [J1, C1, B2, B3]*

Az átfedőben a kulcs-érték párok egyedekhez történő hozzárendelése a kulcs hasított (hasító függvény szerinti, hashed) értéke és az egyed azonosítója közötti távolság alapján történik. Minden pár ahhoz az egyedhez kerül, amelynek azonosítója az átfedő címtérében legközelebb van a kulcshoz. Ezt a *kulcshoz legközelebbi egyednek* nevezzük. Replikáció esetén nem csak a pontosan legközelebbi, hanem a legközelebbiek közül többen is tárolják az adatokat. Ezek számát k_r-rel jelölöm.

A lekérdezést végző egyed először a többi egyed segítségével megkeresi a tárolást végző egyed hálózati címét. Az ezt elvégző iteratív útválasztási eljárás helyessége bizonyított [4]. Az adat lekérdezésének sikeressége ezután azon múlik, hogy tud-e kommunikálni az azt tároló egyeddel. Ha nem, a lekérdezés meghiúsul. A lekérdezés sikerességéhez viszont elegendő az is, ha a kulcs közelében egy másik egyed tárolja az adatot, nem szükséges annak pont a legközelebbinek lennie.



3. ábra. A kulcs közelében lévő egyedeknek küldött üzenetek a Kademlia átfedőben, $k_r = 4$-szeres replikáció esetén

A kidolgozott modell alapja az egyedek közötti kapcsolódási lehetőségek vizsgálata. Ha a kulcs közelében lévő egyedek közül bármelyik olyan hálózati kapcsolattal rendelkezik, amely biztosítja, hogy elegendő számú másik egyedtől lehet kapcsolódni és üzenetet küldeni hozzá, akkor az adat lekérdezhető. Ha vannak ilyen egyedek az átfedőben, akkor a kulcs körüli tartományban is találunk ilyeneket, mivel az átfedőbe belépéskor minden egyed véletlenszerűen választ helyet magának a címtérben.

A kapcsolódás sikertelensége esetén, mivel az egyedek nem tudják ellenőrizni, hogy ennek oka a keresett egyed távozása vagy a hálózati hibája, a kulcstól távolabbiakhoz is küldenek üzeneteket (3. ábra). Így k_r-nél szélesebb tartományban szóródnak szét az adatok vagy a lekérdezési kísérletek. A modellel ez a jelenség vizsgálható kvantitatív módon. A szükséges replikáció meghatározása a tartomány k_r széles-

ségének kiszámítása úgy, hogy legyen benne legalább egy megfelelő elérhetőségű egyed.

2.2. altézis. *Megmutattam, hogy a Kademlia átfedőben az azonosítók gyakorlatilag véletlenszerűen történő megválasztása miatt a tárolt adatok elérhetősége a hibák globális eloszlásától függ.*
Bebizonyítottam, hogy egy adott rendelkezésre állás eléréséhez szükséges replikációs szint változatlan hálózati peremfeltételek mellett független az átfedő egyedszámától. [J1, C1]

Az átfedőben tárolt adatok kulcsai és az egyedek között nincsen eredendő összefüggés, hanem csak az átfedők hozzák létre azt. Egy egyedhez akkor lesz egy adott kulcs hozzárendelve, ha az átfedőbe beépüléskor a kulcshoz az összes többinél közelebbi azonosítót választ magának. Mivel az egyed hálózati kapcsolata, és a véletlenszerűen megválasztott átfedőbeli azonosítója között nincsen összefüggés, a hálózati hibák és a tárolt kulcsok között sincsen. Egy adott kulcshoz közeli egyedek a fizikai hálózatnak egymástól távoli pontjain lehetnek. Az átfedő modellezésekor emiatt a becsült vagy mért hálózati hibákat nem kell az egyedekhez hozzárendelni, hanem csak azok globálisan vett eloszlása lényeges.

Az átfedő egyedszámának növelésével növekszik a címtér telítettsége. Mivel az egyedek véletlenszerűen kerülnek a címtérbe, egy adott kulcshoz legközelebbi k_r darab egyed annak telítettségétől függetlenül lesz elszórva a fizikai hálózaton. Ha feltételezzük, hogy az átfedő új egyedei ugyanolyan hálózati kapcsolatokkal rendelkeznek, mint az addigiak (pl. továbbra is 15%-uk rendelkezik címfordítást használó kapcsolattal), akkor a kulcshoz közeli egyedek más egyedek által látható elérhetősége sem változik meg. Emiatt megnövekedett egyedszám mellett ugyanaz a k_r replikációs szint használható, mint előtte.

2.3. altézis. *Megmutattam, hogy a replikáció növelésével a Kademlia átfedő megbízhatatlan egyedek jelenléte esetén is megbízhatóvá tehető. [J1, C1, B2, B3]*

A kidolgozott modell segítségével a szükséges replikáció kiszámítására egy képlet adható meg.

Egy egyed azonosítóját m-mel jelöljük, az egyedhez tartozó hibaarányt (vagyis az azt el nem érő egyedek arányát) pedig a $h(m)$ függvénnyel adjuk meg. Mivel a kulcsok és a hálózati hibák között nincsen összefüggés, az egyedenkénti hibák növekvő sorba rendezésével a $h(m)$ függvény monotonná tehető. Ha a megengedett hálózati hibák aránya β, az m egyed $h(m) \leq \beta$ esetén képes megbízhatóan tárolni az adatot. Ez alapján a $h^{-1}(\beta)$ kifejezés a tárolást elvégezni képes egyedek arányát adja meg, ha az egyedek címeire a $[0,1)$ tartományú folytonos közelítést alkalmaztuk [10]. A hasító függvények a címteret egyenletesen lefedő kimenete miatt ez egyben annak valószínűsége is, hogy egy kulcs megfelelő helyre kerül.

A lekérdezések helyességére a fentiekből lehet következtetni, és az a k_r replikáció növelésével javítható. Ennek szükséges értéke a modell alapján a következő egyenlettel határozható meg:

$$k_r = \left\lceil \frac{\ln(1-P)}{\ln(1-h^{-1}(\beta))} \right\rceil, \tag{1}$$

ahol P az előírt megbízhatóság. Mivel a replikáció csak egész szám lehet, a kapott értéket felfelé kell kerekíteni.

Ha a hibák eloszlása, vagyis a $h(m)$ függvény ismeretlen, $h^{-1}(\beta)$ értéke az elérhető egyedek arányával helyettesíthető. Vagyis ha egy egyed azt érzékeli, hogy az általa indított kapcsolatok 15%-a sikertelen, akkor a $h^{-1}(\beta) = 0{,}85$ közelítést alkalmazhatja a szükséges replikáció becsléséhez. Ez a fenti összefüggést egyszerűen alkalmazhatóvá teszi, hiszen ez az adat működés közben könnyen mérhető.

3.3. Harmadik tézis

3. tézis. *Kidolgoztam egy eljárást a Kademlia átfedőn belül küldött üzenetszórás hatékony megvalósítására. [J1, J2, J3, J5, J9, C3, B1]*

A Komondorban támadás érzékelésekor a gyűjtő egyed a hálózaton minden egyedet értesít, hogy azok a felismert támadó ellen védekezhessenek. Erre az átfedőben üzenetszórás (broadcast) használható. A P2P átfedőkben ritkán valósítanak meg ilyet, azok betörésérzékelési alkalmazásában viszont éppen erre van szükség. A kidolgozott eljárás az átfedő beépített topológiáját használja arra, hogy az üzenetszórást a lehető leggyorsabban elvégezze.

3.1. altézis. *Felismertem, hogy bármely Kademlia átfedő átrendezhető úgy, hogy egy adott egyed egy, a címtérből tetszőlegesen kiválasztott azonosítójú átfedő hálózati pontba kerüljön, miközben az egyedek közötti, az átfedőn ugrásszámban mért távolságviszonyok nem változnak.*
Kidolgoztam egy átrendezési eljárást, amellyel egyszerűsíthető a Kademlia átfedő hálózatokra vonatkozó kommunikációs algoritmusok tervezése.

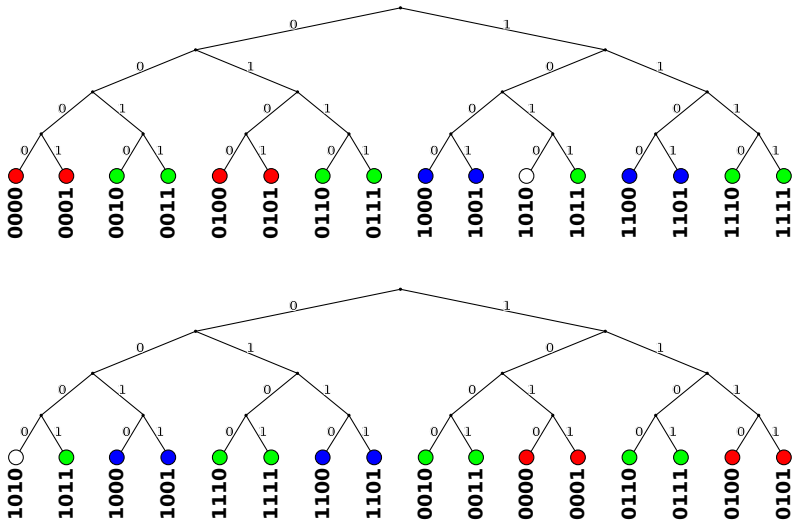
A Kademlia átfedőben az egyedek távolságát a kizáró vagy (XOR) függvénnyel számítják ki. Ezt reprezántálja a bináris fa, amellyel ábrázolják. Minél magasabb helyiértéken van az első 1-es bit két egyed közötti távolságban, annál magasabb az a részfa, amely a címtér legalacsonyabb, mindkettőt tartalmazó részfája.

Ha egy egyed N azonosítóját az átfedő összes egyedének azonosítóival kizáró vagy kapcsolatba hozzuk, azzal létrehozunk egy transzformált átfedőt (4. ábra). Ebben minden egyed azonosítója megváltozik. Mivel $N \otimes N \equiv 0$, a kiválasztott egyedé éppen úgy, hogy az pontosan a 0 azonosítójú helyre kerül. Eközben a transzformált átfedőben az egyedek közötti távolság nem változik meg, amit a következő összefüggés biztosít:

$$D = A_{új} \otimes B_{új} = (A \otimes X) \otimes (B \otimes X) = A \otimes B. \tag{2}$$

Ezért az átalakított átfedő az eredetivel azonos útválasztási táblázatokkal rendelkezik, hiszen azok a távolságoktól függenek.

A transzformáció előnye, hogy az átfedő algoritmusainak tervezését egyszerűsíti. Az algoritmusok (pl. az itt bemutatott üzenetszórás)

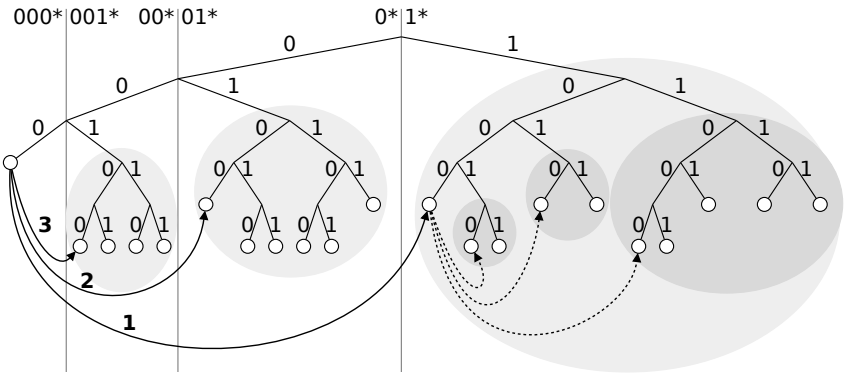


4. ábra. A Kademlia átfedő átrendezése. Az átrendezéssel a fehérrel jelölt, 1010 azonosítójú egyed a 0 pontba került, miközben az egyedek távolsági viszonyai nem változnak

úgy tervezhetők meg, mintha az azokat végrehajtó egyed a 0 azonosítójú pontban helyezkedne el. Így általános esetben is használható algoritmusokat kapunk, mivel egy ilyen átfedő visszaalakítható az eredeti azonosítókat használó alakjára.

3.2. altézis. *Elméleti úton igazoltam, hogy a kidolgozott üzenetszórási eljárással az üzenet küldése az egyedszámhoz viszonyítva lépésen belül megtörténik, és az a hálózat fenntartásának hálózati költségét nem terheli. A kidolgozott algoritmus elméletileg meghatározott eredményeit szimulációval is igazoltam. Az vizsgálatokat elvégeztem különböző, gyakorlati eseteknek megfelelő peremfeltételek mellett. [J1, J2, J3, J5, J9, C3, B1]*

A kidolgozott algoritmus működésének lényege, hogy az üzenetszórást indító egyed az átfedő egyedeit tartalmazó bináris fát egyre



5. ábra. Az üzenetszórási algoritmus működése a Kademlia átfedőben

kisebb, egyforma méretű részekre osztja. A minden lépésben két részre osztott fa egyik felében benne van önmaga is, a másikban pedig nem. Az előbbiekben önmaga végzi el az üzenetszórást, az utóbbiakból pedig kijelöl egy egyedet, amely a címtér azon részében felel annak elvégzéséért.

A részfákra bontás révén az algoritmus igazodik a Kademlia átfedő útválasztási eljárásához, így használhatja az azáltal karbantartott táblázatokat. Ez látható az 5. ábrán egy transzformált átfedőben, amelyben az üzenetszórást indító egyed a 0 azonosítóval rendelkezik. Első lépésben a küldő kétfelé osztja a címteret. A tőle távoli fél fában a kijelölt felelős egyed végzi el az üzenetszórást ugyanezzel az algoritmussal, a sajátjában pedig önmaga. A saját felét újra két részre osztva két negyed fát kap; a távoli negyedből felelőst jelöl ki, a közelebbiért pedig önmaga felel.

Az egyes felelősök egymástól függetlenül hajthatják végre a nekik kijelölt részfában az algoritmust, nem kell egymásra várniuk. Mivel az átfedő átmérője, vagyis két egyed között a legrövidebb lépésszám az átfedő méretével csak logaritmikusan arányos, ezért az üzenetszórás

is elvégezhető ennyi lépésben.

Az algoritmust saját Kademlia szimulátor programban valósítottam meg. A helyességre és végrehajtási időre adott becsléseket ezzel ellenőrizni lehetett. A szimulációban a fizikai hálózat modellezése a szakirodalomból vett adatok alapján történt [9, 11].

3.3. altézis. *Bebizonyítottam, hogy replikáció alkalmazásával az üzenetszórás hibákkal terhelt fizikai hálózaton, valamint rosszindulatú egyedek jelenléte esetén is megbízhatóvá tehető.*
Meghatároztam az egy adott megbízhatóság eléréséhez szükséges replikációs szintet a hálózati hibaarány és az egyedszám függvényében. [J1, J3, J5, J9]

Az üzenetszórásban az egyes egyedek eltérő lépésszám után kapják meg az üzenetet. Némelyik egyed közvetlenül az indító egyedtől, mások egy kijelölt felelőstől, vagy több ilyenen keresztül. Ha hálózati hibák miatt ezek az üzenetek elvesznek, vagy esetleg rosszindulatú egyedek jelenléte miatt nem kerülnek továbbításra, akkor a megbízhatóság csökken. Az elvesző üzenetek eltérő nagyságú hibát okozhatnak: egy olyan üzenet, amelyik fél részfáért felelős egyedet jelölt ki, akár 50%-osat.

Egy adott egyed felől nézve a megbízhatóság attól függ, hogy mennyire van távol az üzenet eredeti kiindulási pontjától. Minél több közvetítőn keresztül jut hozzá el az, annál valószínűbb, hogy útközben elveszik. Mivel az üzenettovábbítások során az üzenet minden lépésben távoli fába kerül, az azonosító egy bitben változik meg lépésenként. Emiatt a közvetítők száma a két azonosító Hamming-távolsága, amelyből az algoritmus helyessége, b-vel figyelembe véve a címbitek számát, és P-vel a helyes továbbítások valószínűségét, így becsülhető:

$$m = \frac{\sum_{i=0}^{b} \binom{b}{i} P^i}{2^b}. \tag{3}$$

A megbízhatóság replikációval növelhető. Ez azt jelenti, hogy minden részfából több felelős egyedet jelölünk ki. Azok egymástól füg-

getlenül lehetnek hibásak, ezért csökken annak a valószínűsége, hogy egy felelős sem fogja elvégezni a műveletet az adott részfán belül. A replikáció P_h valószínűséggel elvesző üzenet esetén egy részfán belül a felelős meglétét $P = 1 - P_h^{k_b}$-re javítja. Ebből meghatározható az üzenetszórásban alkalmazandó replikáció szintje:

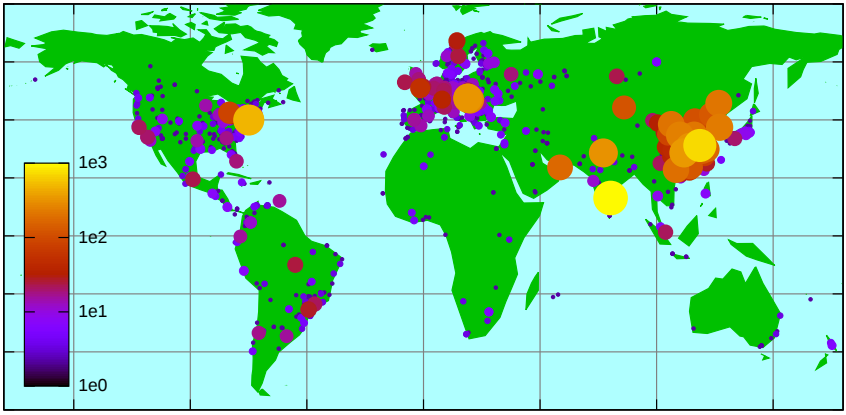
$$k_b = \left\lceil \frac{\ln\left(2\left(1 - \sqrt[b]{n}\right)\right)}{\ln P_h} \right\rceil, \tag{4}$$

amelyben b az átfedő N egyedszámát is tartalmazza ($b = \log_2 N$), P_h a csomagvesztési arány, n pedig az elérendő helyességi arány. Az átfedő egyedszáma meghatározható az útválasztási táblázatokból, a címtér telítettségét vizsgálva, így az az adat minden egyed számára rendelkezésre áll.

Az algoritmus által generált hálózati forgalom, vagyis az átfedő egyedszáma és az üzenetszáma közötti összefüggés szimuláció eredménye alapján határozható meg. A replikáció nélküli esetben az algoritmus $N - 1$ üzenetet igényel. Ez értelemszerűen az elérhető minimum, hiszen ez egyedenként egy üzenetet jelent, kivéve az indító egyedet. Replikáció használata esetén az üzenetszám megnövekszik, és függ a fizikai hálózat egyedek közötti késleltetési idejeitől is. Ennek oka az, hogy egy egyed felé több úton haladhatnak üzenetek, amelyek akár eltérő méretű részfákért teszik azt felelőssé. A duplikátumok száma így eltérő lehet, attól függően, hogy melyiket kapja meg előbb.

4. Az eredmények gyakorlati alkalmazásai

A Komondor eljárás általánosan alkalmazható a gazdagépeket érintő támadások érzékelésében (6. ábra). Segítségével a különböző helyeken érzékelt, támadásokra utaló események adatai gyűjthetőek össze hatékonyan. Az eljárásban létrehozott DHT alapú átfedő hálózat segítségével az érzékelés által okozott hálózati terhelés elosztása javítható.



6. ábra. A Komondor rendszer által érzékelt támadások forrásai és azok száma a világ térképén

A Kademlia átfedőre kidolgozott eljárások bármely arra épülő alkalmazás esetén használhatóak. A második tézis modelljével a Kademlia átfedő egy rendszerszintű konfigurációs paramétere, a k_r replikációs szintje határozható meg úgy, hogy az átfedő hálózati forgalma alacsony maradhasson, mégis az eltárolt adatok visszakereshetőségét biztosítani tudja. Az ehhez használt 1. egyenlet $h^{-1}(\beta)$ paramétere, amely az egyedek megbízhatóságát adja meg, az átfedő működése közben mérhető adattal becsülhető.

A bemutatott üzenetszórási eljárás is bármely Kademlia topológiájú átfedőben megvalósítható. Segítségével kivitelezhető olyan információk közlése, amelyek az összes egyed számára fontosak. Ilyen a Komondor eljárásban egy felismert támadó IP címe. Ezen kívül az eljárás kiegészíti a Kademliában használt kikeresések lehetőségeit, mert a globális üzenetküldésre alapozva tetszőleges keresési feltételek szerinti lekérdezések valósíthatóak meg. Az algoritmus megbízhatósága a kívánt mértékben javítható. Az ehhez szükséges replikáció mérté-

ke a 4. egyenlet alapján meghatározható. Az egyenlet paraméterei, az egyedszám és a fizikai hálózat hibái az egyedek által becsülhető és mérhető információk, amelyek megszerzéséhez az átfedő globális ismerete nem szükséges.

Megjelent folyóiratcikkek

[J1] Zoltán Czirkos and Gábor Hosszú. Peer-to-peer alapú betörésérzékelés. *Híradástechnika*, 63:29–36, 2008. Pollák–Virág díjas cikk.

[J2] Zoltán Czirkos and Gábor Hosszú. Distributed Detection of Intrusions. *Informatika – a Gábor Dénes Főiskola Közleményei*, XII(2):37–40, 2010.

[J3] Zoltán Czirkos and Gábor Hosszú. Peer-to-peer Based Intrusion Detection. *Infocommunications Journal*, LXIV(I):3–10, 2009.

[J4] Zoltán Czirkos, Loránd Lehel Tóth, Gábor Hosszú, and Ferenc Kovács. Novel Applications of the Peer-to-Peer Communication Methodology. *Journal on Information Technologies and Communications – Research, Development and Application on Electronics Telecommunications and Information Technology*, E-1(1(5)):59–70, 2009.

[J5] Zoltán Czirkos and Gábor Hosszú. Enhancing the Kademlia P2P Network. *Periodica Polytechnica*, kézirat elfogadva, megjelenés alatt.

[J6] Zoltán Czirkos and Gábor Hosszú. Műveleti rendszerek egyenrangú közlésen alapuló védelme. *Informatika – a Gábor Dénes Főiskola Közleményei*, 8(4):9–21, 2005.

[J7] Zoltán Czirkos. Operációs rendszerek egyenrangú közlésen alapuló védelme. *Linuxvilág*, VII(5):65–69, 2006.

[J8] Zoltán Czirkos and Hosszú Gábor. A Distributed Intrusion Network Based on Kademlia. *Computers & Security*. Kézirat elküldve.

[J9] Zoltán Czirkos and Hosszú Gábor. Pseudo Reliable Broadcast in the Kademlia P2P system. *Computer Networks*. Kézirat elküldve.

Konferenciakiadványban megjelent előadás

[C1] Zoltán Czirkos, Lóránd Lehel Tóth, and Gábor Hosszú. Komondor – P2P Intrusion Prevention, poster. In Róbert Szabó and Attila Vidács, editors, *HSN Workshop 2009*, Balatonkenese, May 2009.

[C2] Zoltán Czirkos and Gábor Hosszú. Az elosztott betörésérzékelés hatékonysága. In *Informatika Korszerű Technikái*, Dunaújváros, 2010. Dunaújvárosi Főiskola.

[C3] Zoltán Czirkos and Gábor Hosszú. Üzenetszórás modern P2P hálózatokban. In *Informatika Korszerű Technikái*, pages 13–23, Dunaújváros, 2008. Dunaújvárosi Főiskola.

[C4] Zoltán Czirkos and Gábor Hosszú. P2P alapú betörésvédelem. In *Informatika Korszerű Technikái*, pages 45–52, Dunaújváros, 2007. Dunaújvárosi Főiskola.

[C5] Zoltán Czirkos. P2P alapú biztonsági szoftver fejlesztése. In *Információvédelem menedzselése XXII. Szakmai fórum*, pages 43–47, Budapest, 2006. Hétpecsét Információbiztonsági Egyesület.

Könyvfejezet

[B1] Zoltán Czirkos and Gábor Hosszú. Usage of Broadcast Messaging in a Distributed Hash Table for Intrusion Detection. In Peyman

Kabiri, editor, *Privacy, Intrusion Detection and Response: Technologies for Protecting Networks*. IGI Global, Hershey, 2011.

[B2] Zoltán Czirkos and Gábor Hosszú. Reliability Issues of the Multicast-Based Mediacommunication. In Margherita Pagani, editor, *Encyclopedia of Multimedia Technology and Networking*, pages 1215–1223. Information Science Reference, Hershey, 2009.

[B3] Zoltán Czirkos and Gábor Hosszú. On the Stability of Peer-to-Peer Networks in Real-World Environments. In Antonio Cartelli and Marco Palma, editors, *Encyclopedia of Information Communication Technology*, pages 622–630. Information Science Reference, Hershey, 2008.

[B4] Zoltán Czirkos and Gábor Hosszú. Application of the P2P Model for Adaptive Host Protection. In Margherita Pagani, editor, *Encyclopedia of Multimedia Technology and Networking*, pages 54–60. Information Science Reference, Hershey, 2009.

[B5] Zoltán Czirkos and Gábor Hosszú. Peer-to-Peer Methods for Operating System Security. In Goran D. Putnik and Maria Manuela Cunha, editors, *Encyclopedia of Networked and Virtual Organizations*, pages 1185–1191. Idea Group Inc., Hershey, 2008.

[B6] Zoltán Czirkos, Gábor Hosszú, and Kovács Ferenc. E-Collaboration Enhanced Host Security. In Ned Kock, editor, *Encyclopedia of E-Collaboration*, pages 172–177. Information Science Reference, Hershey, 2008.

[B7] Zoltán Czirkos and Gábor Hosszú. Intrusion Detection Based on P2P Software. In Mehdi Khosrow-Pour, editor, *Encyclopedia of Information Science and Technology*, pages 2232–2238. Information Science Reference, Hershey, 2008.

[B8] Zoltán Czirkos and Gábor Hosszú. A Novel Application of the P2P Technology for Intrusion Detection. In Antonio Cartelli and

Marco Palma, editors, *Encyclopedia of Information Communication Technology*, pages 616–621. Information Science Reference, Hershey, 2008.

[B9] Zoltán Czirkos and Gábor Hosszú. Network-based intrusion detection. In Mário Freire and Manuela Pereira, editors, *Encyclopedia of Internet Technologies and Applications*, pages 353–359. Idea Group Inc., Hershey, 2007.

Hivatkozások

[1] Hervé Debar and Andreas Wespi. Aggregation and Correlation of Intrusion-Detection Alerts. In Wenke Lee, Ludovic Mé, and Andreas Wespi, editors, *Recent Advances in Intrusion Detection*, volume 2212 of *Lecture Notes in Computer Science*, pages 85–103. Springer Berlin / Heidelberg, 2001.

[2] C.V. Zhou, C. Leckie, and S. Karunasekera. A survey of coordinated attacks and collaborative intrusion detection. *Computers & Security*, 29(1):124–140, 2010.

[3] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman. Planetlab: an overlay testbed for broad-coverage services. *ACM SIGCOMM Computer Communication Review*, 33(3):3–12, 2003.

[4] Petar Maymounkov and David Mazières. Kademlia: A Peer-to-peer Information System Based on the XOR Metric. 2002.

[5] Christian Seifert. Analyzing Malicious SSH Login Attempts. `http://www.symantec.com/connect/articles/analyzing-malicious-ssh-login-attempts`, November 2010.

[6] Chenfeng Vincent Zhou, Shanika Karunasekera, and Christopher Leckie. A Peer-to-Peer Collaborative Intrusion Detection System.

In *Networks, 2005. Jointly held with the 2005 IEEE 7th Malaysia International Conference on Communication., 2005 13th IEEE International Conference on*, volume 1, page 6. IEEE, 2006.

[7] Alfonso Valdes and Keith Skinner. Probabilistic Alert Correlation. *Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection*, pages 54–68, October 2001.

[8] S.J. Templeton and K. Levitt. A requires/provides model for computer attacks. In *Proceedings of the 2000 workshop on New security paradigms*, pages 31–38. ACM, 2001.

[9] S.A. Crosby and D.S. Wallach. An Analysis of BitTorrent's Two Kademlia-based DHTs. Technical Report TR-07-04, Department of Computer Science, Rice University, 2007.

[10] M. Naor and U. Wieder. Novel Architectures for P2P Applications: the Continuous-Discrete Approach. *ACM Transactions on Algorithms (TALG)*, 3(3):34–es, 2007.

[11] The „king" data set. `http://pdos.csail.mit.edu/p2psim/kingdata/`.