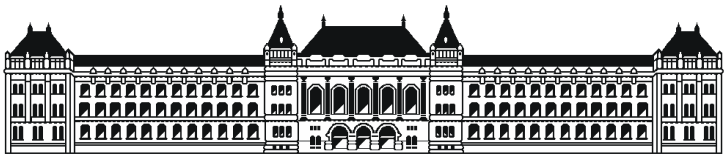


BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM
HÍRADÁSTECHNIKAI TANSZÉK

Privátszférát erősítő technológiák mobiltelefonokra

PRIVÁTSZFÉRÁT ERŐSÍTŐ TECHNOLÓGIÁK (BMEVIHIAV00)

Készítette
Rádi Attila (VTV2CQ)

Konzulens
Földes Ádám Máté

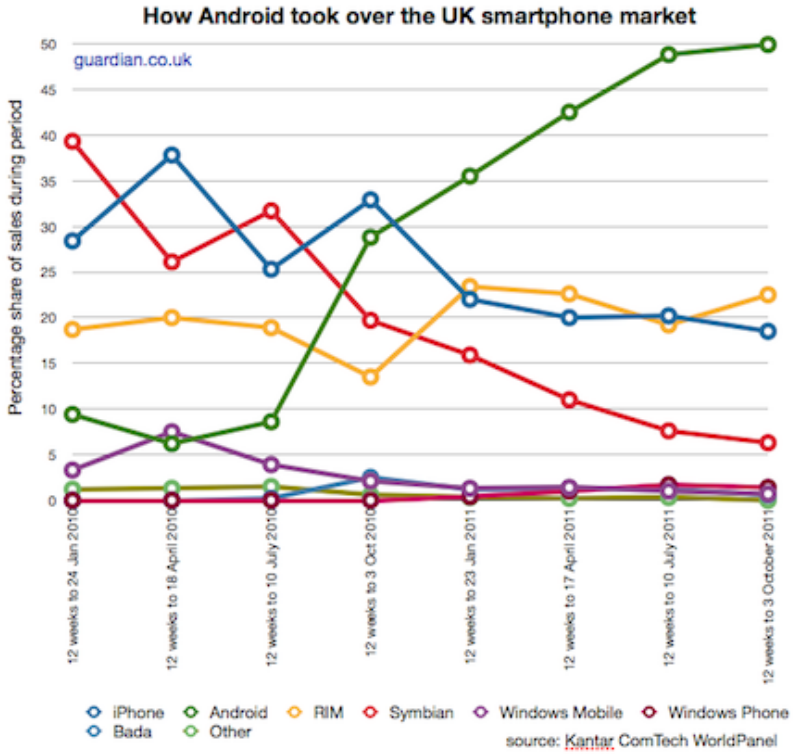
2012. június 25.

Tartalomjegyzék

1. fejezet

Bevezetés

A mobiltelefonok az utóbbi időben teljesen más arculatot alakítottak ki a mobil világban, mint az ezelőtt 10 évvel megjelent elődeik. Most már nem csupán kommunikációra alkalmas eszközökként vannak jelen a piacon, hanem multifunkcionális feladatok ellátására is alkalmasak. Teljesítményük megfelel egy régebbi számítógépnek a teljesítményével, már saját operációs rendszerrel rendelkeznek, és számos alkalmazási területük létezik. De miért is nevezik okostelefonoknak a mai mobiltelefonokat? Több definíció is létezik a fogalom meghatározására, de talán a legközelebb az a meghatározás áll, hogy az okostelefonokra már harmadik fél is könnyen fejleszthet szoftvert.



1.1. ábra. *A mobilplatformok piaci részesedése*

Számos mobilplatform létezik, és az egyes platformokon belül napról napra jönnek ki az újabb frissítések. Általában az jellemző egy platformra, hogy lehetőséget biztosít alkalma-

zásoknak letöltésére egy - a platform üzemeltetője által biztosított - saját alkalmazásbolton keresztül. Ezekbe az alkalmazásboltokba a fejlesztők regisztrálhatnak, és bizonyos feltételek mellett feltölthetik alkalmazásaikat. Az alkalmazások ezután ellenőrzésen mennek át, és ha nem találtak benne kivetnivalót, akkor a felhasználók megvásárolhatják.

A mobilplatformok bőséges választékában két platformot érdemes kiemelni az Android és iPhone platformot. Ez a két platform napjainkban a legnépszerűbbek a felhasználók körében. Ezt a 1.1 ábra is alátámasztja, mely a fentebb említett két platform piaci részesedését igen előkelő helyen ábrázolja. A kérdés az, hogy mégis mennyire megbízhatóak ezek a mobilplatformok? Mennyire szigorú az alkalmazások ellenőrzése? Mi az ára egy ingyenes alkalmazásnak a letöltésének?

2. fejezet

Mobilplatformok biztonsága

Tipikusan egy mobilfelhasználó a készülékén személyes adatokat is szokott tárolni. Az okostelefonokon már direkt erre kifejlesztett, úgynevezett PIM (Personal Information Manager) funkciók vannak beépítve. Ilyen PIM funkció például a *Contact List* (Névjegyzék), a *Calendar* (Naptár) és a *TODO* lista (Teendők lista) [14]. A PIM funkciók segítségével kényelmesen tudjuk egyként kezelni ezeket az információkat. További személyes adatnak számítanak az SMS-ek, a telefonon tárolt képek és a különböző adatok, melyek nem tartoznak harmadik fél számára. De milyen védelmet biztosítanak a mobilplatformok, hogy még véletlenül se juthassanak hozzá ezekhez az adatokhoz? Sajnos azt kell mondani a személyes adatok védelme mobilplatformokon nagyon gyenge, majdnem azt lehet mondani, hogy nulla. Naponta röppennek fel hírek újabb biztonsági rések felfedezéséről és káros alkalmazásokról, melyek az alkalmazás boltban találhatóak. Azt gondolná a felhasználó, hogy egyes fizetős alkalmazások nem kártékonyak a készülékre nézve, hiszen pénzt adnak ki érte, és ezért cserébe megbízható működést várnak. Azt kell mondani, hogy akár fizetős, akár ingyenes egy alkalmazás ugyanúgy megvan annak az esélye, hogy kártékony kódot tartalmaz.

2.1. Android biztonság

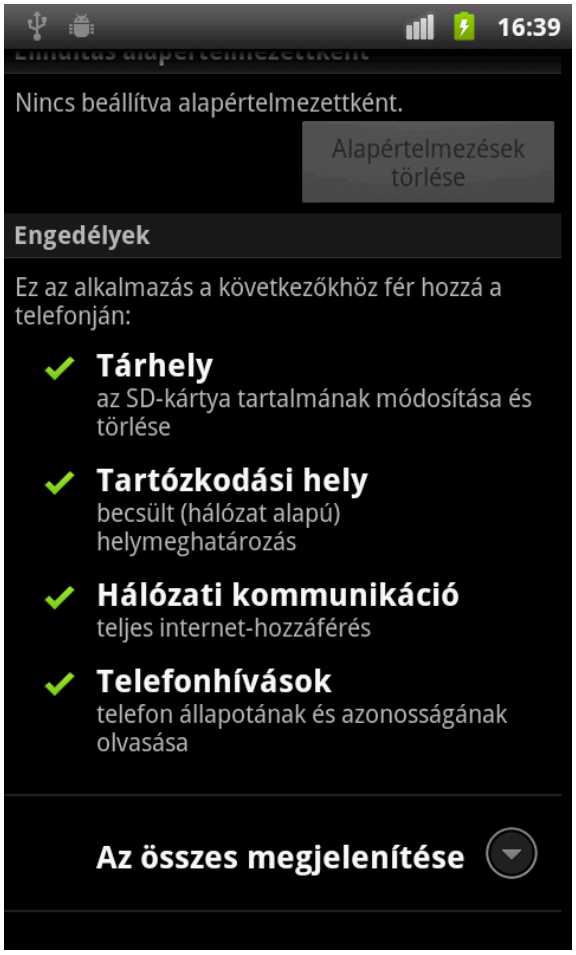
2.1.1. Platformszintű biztonság

Az Android platform vizsgálatával kiemelten foglalkozok az esszében, adatbiztonság és adatvédelem területén. Ez a legdinamikusabban fejlődő platform, és egyre több felhasználó ad bizalmat az ilyen platformmal rendelkező készülékeknek. Többek között azért is ilyen népszerű a felhasználók körében, mert rengeteg ingyenes szolgáltatás érhető el a Google jóvoltából és a Google Play-en (régen Android Market) számos alkalmazást lehet letölteni akár teljesen ingyen.

Azonban érdemes egy kicsit jobban utánanézni, hogy mi az ára az ingyenességnek. [12] Azok az alkalmazások, melyek ingyen letölthetőek a Google Play-ről valójában annyit jelent, hogy bár az alkalmazásért nem kell fizetni, harmadik fél reklámokat küldhetnek a készülékre. Ezt legtöbbször igen trükkösen érik el. Az ingyenes alkalmazások (ezek körülbelül 73%-át teszik ki a Google Play-ről letölthető alkalmazásoknak) többsége (80%)

pozíciófüggő hirdetéseket jelenít meg az alkalmazáson belül. Ezt úgy érik el, hogy a telefon aktuális földrajzi koordinátáit kérdezik le, és ez alapján választják ki a megfelelő hirdetést.

További veszélyei az ingyenes alkalmazásoknak, hogy általában több jogosultságot kérnek el, mint a fizetős alkalmazások. Előfordulhat, hogy egy alkalmazás helymeghatározáshoz, *Contact List*-hez, *Calendar*-hoz, e-mail, illetve SMS üzenetekhez kérhet hozzáférést.



2.1. ábra. *Az Angry Birds által igényelt hozzáférések*

A 2.1 ábrán láthatjuk, hogy a legnépszerűbb mobilos játék az Angry Birds milyen jogosultságokhoz kér engedélyt. Jogosan merül fel a kérdés, hogy egy játék alkalmazásnak miért szükséges a tartózkodási helyhez és a telefonhívásokhoz hozzáférni. Igaz egy alkalmazás feltelepítése előtt kiírja a felhasználó számára, hogy milyen jogosultságokat követel, a felhasználók általában nem tudják, hogy milyen veszélyeket rejthet, ha egy alkalmazás mondjuk a *Contact List*-hez fér hozzá. A *DreamDroid* nevezetű trójai alkalmazás ingyenes játék csomagként terjedt, közben meg IMEI, IMSI, UserID, telefon típus és szolgáltató információkat gyűjtött a készülékről [17].

2.1.2. A Google adatkezelése [1]

Megoszlanak a vélemények a Google új adatkezelésével kapcsolatban is. A több, mint 60 jogosultsági dokumentumból összesen 1 darab jogosultsági dokumentumot fognak készíteni. Ezáltal a Google által készített szolgáltatások (Gmail, Calendar, Search) és a Youtube adatkezelése azonos lesz. Nem titkolt célja a felhasználói élménynek a növelése azáltal, hogy a kereső funkciók rugalmasabban, az adott személyre szabottan fognak megjelenni. Arról persze nem nyilatkoznak, hogy ezt hogyan valósítják meg. Az új dokumentum bevezetése hatással lesz az Android-os mobiltelefonokra is.

A dokumentum alapján az alábbi adatokat gyűjtheti a Google Android-os okostelefonokról:

- **Eszköz információk**: Lehetőségük van eszköz specifikus információkat gyűjteni, mint a készülékben található hardver, operációs rendszer verzió, egyedi eszköz azonosítók, hálózati információk és telefonszám.
- **Napló információk**: Telefóniai adatok lekérdezése, mint saját telefonszám, felhívott telefonszámok, továbbított telefonhívások, hívások időpontja és hívás időtartam, SMS kézbesítés.
- **Helymeghatározási információk**: GPS-en, cellainformáción, Wi-fi kapcsolaton, adatkapcsolaton keresztüli pozíciók gyűjtése.
- **Helyi tár**: lekérdezhetik a webböngésző által tárolt, illetve az alkalmazások gyorsítótárában tárolt adatokat.

2.2. Privátszféra védelem

Az előbbi bekezdésben látható, hogy az emberek privát szférája több forráson keresztül sérülhet. Ugyanis nem csak harmadik fél, hanem maga a szolgáltató is hozzáférhet érzékeny adatokhoz. Az elméleti, illetve konkrét megoldások előtt bemutatnám, hogy kik azok, akik megszerezhetik és felhasználhatják az adatainkat saját céljaikra.

- **Tolvajok**: Időrendi szempontból a lopási tevékenységek ellen védekeznek legrégebb óta. Bár általában a lopási tevékenységek célja a telefon megszerzése, majd értékesítése, nem hagyhatjuk figyelmen kívül, hogy a telefonon levő összes adatunkhoz hozzáférhetnek. Ha az áldozat mobil előfizetéssel rendelkezik egy szolgáltatónál, akkor amíg nem tiltja le a telefonszámot a tolvaj az áldozatnak további károkat okozhat.
- **Szolgáltatók**: Nem csak a platform tulajdonosai, hanem a telefonszolgáltatók is képesek bizonyos információk lekérdezésére. Habár, egy szolgáltatóval kötött szerződésben szerepel, hogy monitorozzák a hívásokat, és az adatforgalmat, ezt pedig nem továbbítják harmadik fél számára, a gyakorlatban nem tudni, hogy pontosan mire használják fel ezeket az adatokat.

- **Alkalmazások fejlesztői**: Láthattuk, hogy az alkalmazások fejlesztői milyen egyszerűen kaphatnak jogosultságokat bizonyos funkciók eléréséhez. Külön veszélyt jelentenek az alkalmazásbolton kívüli alkalmazások telepítése a mobiltelefonra, mivel ez kevesebb szűrőn ment keresztül. Akár egy hivatalos, akár egy nem hivatalos alkalmazás tartalmazhat kártékony kódot, ami adatok eltulajdonításán kívül akár a készülékben is kárt tehetnek. Léteznek már olyan trójai programok, melyek a telefonnak a számítási kapacitását használja fel egy későbbi DDOS (Distributed Denial of Service) támadásra.

3. fejezet

PET-ek mobiltelefonokra

Egy mobiltelefonon alkalmazott privátszférát erősítő technológia függ attól is, hogy milyen fajta támadások ellen kívánjuk az adatainkat elrejteni. Először néhány általános megoldást ismertetek, utána rátérek platformspecifikus megoldásokra.

Vannak bizonyos technológiát biztosító szabványok (ISO 15408 'Common Criteria'), de ezeknek a szabványoknak az alkalmazása még nagyon kezdetleges [15]. A szabványokon túl minden egyes állam külön rendelkezik az adatvédelmi szabályozásokkal, amelyek a személyes adatok védelmén kívül az adatok kereskedelemben való használatának jogkörére is kiterjed.

3.1. Illetéktelen hozzáférés elleni védelem

A technológiák ezen fajtái a felhasználókat attól védik meg, hogy eltulajdonított készülékeket csak nehezen tudja a tolvaj működésre bírni. Ilyenfajta védelmeket a mai napig használnak, viszont ez nem védi meg a felhasználókat, ha egy harmadik fél egy alkalmazáson, vagy a kommunikációs csatornákon keresztül adatot próbál tőlük gyűjteni.

3.1.1. Kód alapú védelem [9]

Ameddig nem terjedtek el a mai okostelefonok, gyakorlatilag csak a tolvajoktól kellett tartani, hogy adatainkat megszerezhetik. A telefonok SIM kártyái alapértelmezetten rendelkeznek PIN kóddal, de magát a készüléket is el lehet látni kód alapú védelemmel. A kód alapú védelem tulajdonképpen egy számsor, melynek számjegyei 0-tól 9-ig vehetnek fel értékeket. Meg lehet adni, hogy a kód alapú védelem ne csak belépéskor, hanem minden billentyűzár feloldásakor kérje a jelszót. Ez a megoldás nagyon kényelmetlen a felhasználók számára, mert mindig azzal kell bajlódniuk minden egyes billentyűzár feloldáskor, hogy beírják a saját kódjukat.

3.1.2. Érintőképernyős védelem

Érintőképernyős mobiltelefonokon általában nem kód alapú védelmet szoktak használni. Ez alól kivétel a PIN kódnak a bekérése, ami - az a gombokkal rendelkező telefonok

analógiájára - egy virtuális billentyűzeten kéri be a kódot. A telefon zárolását általában egy a képernyőn végigrajzolt egyedi minta segítségével lehet feloldani. Nyilvánvaló, hogy a mintát csak a telefon tulajdonosa ismeri.

3.1.3. Arcfelismerés [19]

Ezt a funkciót csak akkor lehet használni, ha a telefon rendelkezik kamerával. Alapja, hogy aktivizálódása után egy arcképes fényképet készít a felhasználóról, mely a továbbiakban referenciául fog szolgálni kép összehasonlítás szempontjából. A felhasználó minden egyes belépésekor a telefon ismételten egy arcképes fényképet készít a felhasználóról, majd kép összehasonlítási algoritmusok segítségével vizsgálja az egyezést az imént készített fényképről és a memóriában tárolt referenciáról.

Ez a módszer még nem számít kiforrott technológiának. Hibázási lehetőség lehet az összehasonlító algoritmusok működésében. Lehetséges egy jól fényképezett arcképet hibásnak vesz, mert az algoritmust megzavarja a háttérkörnyezet, vagy nem megfelelő a fényintenzitás, esetleg valami árnyék vetül a felhasználóra. További korlát a kamerák képessége, ugyanis a telefonokba általában olcsóbb, rosszabb minőségű kamerákat szoktak beépíteni, mint fényképezőgépekbe. A harmadik hibázási faktor meg abból adódik, hogy a felhasználó nem megfelelő referencia képet készít a védelem aktiválásakor.

3.2. Helymeghatározás alapú védelem [16]

A helymeghatározás alapú védelem célja, hogy harmadik fél akár egy alkalmazáson keresztül, akár illetéktelenül a szenzor adatokhoz hozzáférve ne legyen képes meghatározni a telefon földrajzilag értelmezett helyzetét. A célra olyan protokollokat dolgoztak ki, melyek bizonyos alkalmazási terület számára nyújtanak védelmet.

3.2.1. Nearby Friend Protocol

A szituáció, hogy létezik egy alkalmazás, mely helymeghatározási adatokat közvetít ismerőseink felé. Az ismerőseink szintén elküldik az adatokat a pozíciójukról és az alkalmazás segítségével kiszámolhatjuk a két készülék közötti távolságot. Van-e valami lehetőség, hogy biztosan megkapjuk ismerőseinktől az információkat?

Első megközelítés: Legyen Alice és Bob a két résztvevő a kommunikációban, és a kommunikáció során használjanak nyilvános kulcsú titkosítást.

1. Alice titkosítja pozícióját saját publikus kulcsával, és elküldi a titkosított adatot Bobnak.

2. Bob titkosítja pozícióját Alice nyilvános kulcsával, és a két értéken modulo összeadás segítségével kiszámolja a távolságot (*homomorphic encription*).

3. Bob visszaküldi a titkosított távolság értéket Alice-nek.

4. Alice a privát kulcsa segítségével dekódolja az adatot. Ezáltal Alice mindig tudja, hogy Bob milyen távol tartózkodik tőle, miközben Bob nem tud Alice-ről semmit! Sőt Alice tudja Bob-nak a pozícióját is.

A szerző ennek a problémának a megoldására a Közeli Barát Protokollt (*Nearby Friend Protocol*) alkalmazza. A protokoll lényege, hogy Bob mikor kiszámolja a titkosított távolságot nem küldi el azonnal az üzenetet Alice-nek, hanem generál egy véletlen értéket, amit szintén Alice publikus kulcsával titkosít, majd az így titkosított véletlen értéket modulo összeadja a titkosított távolság értékkel. Így Alice amikor megkapja a titkosított üzenetet, a dekódoláskor nem tudja meg mi a pontos eredmény. A protokollban részt vesz egy harmadik személy is, őt hívjuk Trent-nek. Alice elküldi a dekódolt üzentet, míg Bob a generált véletlen értéket küldi el Trent-nek. Trent a kapott adatokból ki tudja számolni a távolság értékét, amit elmond Alice-nek.

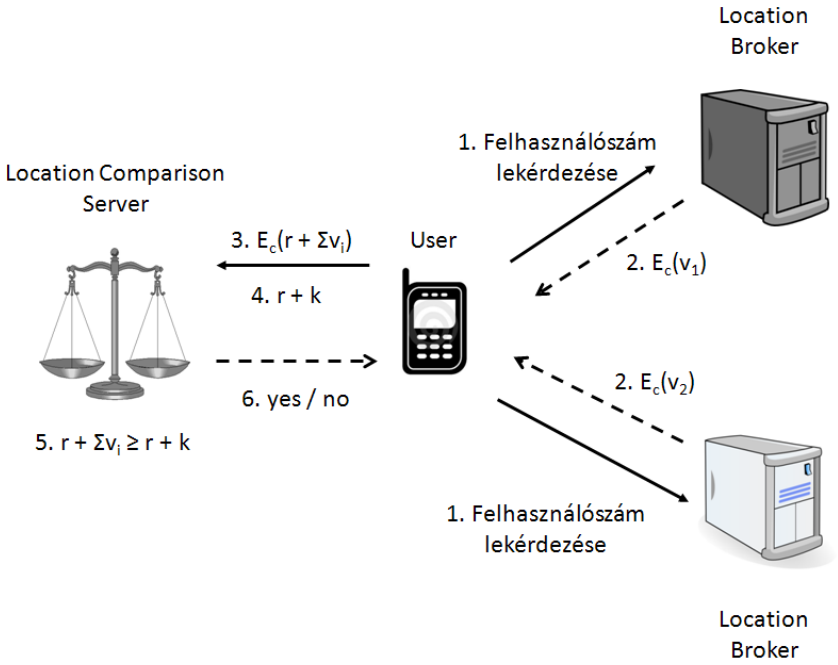
A kérdés az, hogy okvetlenül szükséges-e harmadik személynek a jelenléte? Egy megoldás lehet, ha a területet négyzet alapú mezőkre osszuk, és a mezők távolságát számítjuk ki. Ekkor ha Alice és Bob közötti távolság 0, akkor azt jelenti, hogy azonos mezőn tartózkodnak. Ez esetben Bob ismételten generál egy véletlen értéket, de a véletlen értéket megszorozza a kiszámolt távolság értékkel. Ez azt jelenti, hogy ha ugyanabban a mezőben tartózkodik Alice-szel, akkor a 0-át kell szoroznia valamilyen generált értékkel, ami szintén 0 lesz, más esetben meg Bob egy véletlen értéket fog kódolni, amit Alice nem tud dekódolni. Bob elküldi az így kiszámolt értéket, amit Alice ha dekódol csak annyit tud meg, hogy Bob-bal azonos mezőn tartózkodik-e, vagy sem. A protokoll segítségével Bob akár azt is megadhatja, hogy az n-el távolabb levő mezőn tartózkodik úgy, hogy a kiszámolt titkosított távolságból levon n-et (ez esetben 0 lesz a távolság és n különbsége), amivel a generált értéket fogja megszorozni.

3.2.2. Privacy-Friendly Distributed k-Anonimity Protocol

A következő protokoll, abban nyújt segítséget, hogy elrejtse az eszközt a helymeghatározás alapú szolgáltatásoktól (*Location Cloacking*). Ez a technológia a k-anonimitás képességet használja ki: A rejtést használó terület elrejti a felhasználó pozícióját egészen addig, amíg legalább $k-1$ felhasználó tartózkodik a területen. A protokollt meg lehet valósítani központosítottan és elosztottan is. Központosított esetben a *Location Broker* kezeli az elrejtett területeket, ami azt jelenti, hogy 1. megbízhatónak kell lennie a *Location Broker*-nek, 2. a *Location Broker* egy Single Point of Faiulure, tehát kiesése esetén nem biztosított a pozíció elrejtése. Elosztott esetben a a felhasználók megtanulják egymás pozícióját, és megbíznak egymásban, hogy a rejtést fenntartsák.

A protokoll megadja, hogy a felhasználók együttműködnek a *Location Broker*-ekkel, és félig megbíznak bennük, hogy fenntartsák a rejtőzködést k ember között. A protokoll működését a 3.1 szemlélteti:

1. A felhasználó eszköze a *Location Broker*-ektől lekérdezi, hogy mennyi felhasználó tartózkodik a rejtett területen.



3.1. ábra. *Elosztott k-anonimitás protokoll*

2. A *Location Broker*-ek titkosítják a saját területükön tartózkodó felhasználók számát a *Location Comparison Server* nyilvános kulcsával, és visszatérnek ezzel a titkosított értékkel a kérdezőnek.

3. A kérdező felhasználó generál egy véletlen számot, amit szintén titkosít a *Location Comparison Server* nyilvános kulcsával, majd a véletlen számot, és a *Location Broker*-ektől kapott értékeket modulo összeadja. Ezt az összeget küldi tovább a *Location Comparison Server*-nek.

4. A felhasználó nyíltan elküldi a generált ért és az általa preferált k-anonimitási szint összegét a *Location Comparison Server*-nek.

5. A *Location Comparison Server* dekódolja a kapott adatot a saját privát kulcsával, így megkapja a felhasználó generált értékét és a *Location Broker*-ek által meghatározott felhasználók számát.

6. A *Location Comparison Server* összehasonlítja a dekódolt adatot a felhasználó által küldött k-anonimitási szint igénnyel. Mivel mind a kettő értékhez hozzá van adva a felhasználó által generált érték, így az összehasonlítást nem fogja befolyásolni. Ha a dekódolt adat nagyobb, vagy egyenlő, mint a felhasználó igénye, akkor "igen" válasszal tér vissza, hogy teljesíthető a rejtőzködés. Ellenkező esetben "nem"-el tér vissza a felhasználónak.

Kihívások a protokollal kapcsolatban:

- Problémát vet fel, hogy a *Location Comparison Server* tudomást szerez a felhasználó által igényelt rejtési szintről.

Megoldás: Input elrejtése a GT-SCOT Protocol szerint. A felhasználó titkosítja az általa generált érték és az igényszintje összegét a saját publikus kulcsával, amit elküld a *Location Comparison Server*-nek. A szerver miután dekódolta a titkosított adatot a *Location Broker*-ek által ismert felhasználók számáról, kódolja a felhasználó publikus kulcsával ezt az adatot. Az így kapott két értéken bitenkénti összehasonlítást végez el. Az algoritmus további lépéseit nem ismertetném, mert részletes bemutatásra lenne szükség.

- A felhasználó könnyen megtudhatja, hogy hány másik felhasználó tartózkodik a rejtett területen. Ehhez a k igény paraméterét kell csak módosítania, és folyamatosan a *Location Comparison Server*-nek elküldenie.

  Megoldás: Bináris keresés elkerülése. A *Location Broker*-ek nem a titkosított felhasználószámmal tér vissza a felhasználónak, hanem felhasználószámmal és egy r érték összegének titkosításával, továbbá egy ticket-el. A ticket valójában egy időbélyeg, ami tartalmazza az r értéket, amit a *Secure Comparison Server* fel tud használni, hogy kivonja a titkosított adatból. A ticket-et az érvényességéig eltárolja a szerver, ezáltal ha ugyanaz a ticket érkezik be a felhasználótól megtagadja a válaszadást a felhasználó felé.

- A felhasználó többször is megpróbálhat regisztrálni egy *Location Server*-hez. Ebben az esetben ki kell szűrni az ilyen duplikált kéréseket, ugyanis ezzel a felhasználók meghamisíthatják a rejtésnek a szintjét.

  Megoldás: Duplikált regisztráció elkerülése. Alapja az e-cash problémán alapul.

- A *Location Comparison Server* kiszámolhatja a rejtett területen tartózkodó felhasználók számát.

  Megoldás: Ütközés valószínűségének csökkentése.

3.3. Személyazonosság kezelés [15]

A szolgáltatásoknak általában nincsen szükségük egy személy összes adatának a felhasználására, így ennek a módszernek a célja az, hogy a szolgáltatások számára csak a céljuknak megfelelő, feltétlenül szükséges adatokat kerüljenek továbbításra.

3.3.1. IDEMIX

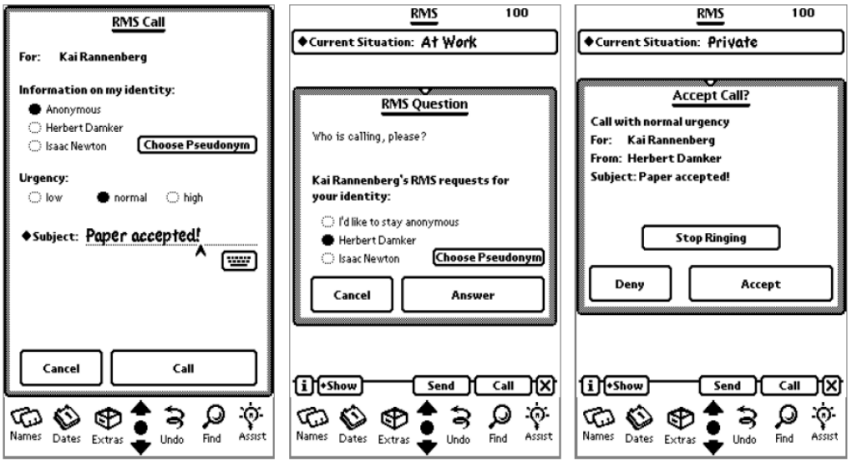
Az IBM által fejlesztett IDEMIX egy olyan rendszer mely erős anonimitási és pszeudonimitási szolgáltatásokat tesz lehetővé kriptográfiai protokollok és adatformátumok segítségével. A személyes adatoknak az elrejtését *zero-knowledge* protokollok segítségével valósítják meg. Ennek alapján lehetőség van, hogy konkrét adatok nem kerülnek továbbításra, hanem például csak annyit mond meg, hogy az adott személynek van jogosítványa, és elmúlt 25 éves.

3.3.2. Liberty Alliance identity federations

A Liberty Alliance 2001-ben alapult, körülbelül 30 szervezet volt ekkor a tagja. Céljuk, hogy a személyazonosság kezeléssel kapcsolatban szabványokat, javaslatokat dolgozzanak ki. Web szolgáltatásokkal kapcsolatban előírják, hogy milyen személyazonosság kezeléssel kapcsolatos intézkedéseket kell tenniük. Nem lehet kimondottan PET-nek nevezni, mivel csak specifikációkat fogalmaznak meg személyazonosságok kezelésére. A specifikációk előnye, hogy nyíltak, így folyamatosan bővülnek a szolgáltatások funkciói, illetve a specifikációt bárki szabadon implementálhatja egy cél rendszerre.

3.3.3. Reachability management

Az elérhetőség kezelés eszközeinek a feladata, hogy kezelje a "privátszféra inváziónak" a fenyegetését. Ezt úgy érik el, hogy mérik a nem kívánatos információ, illetve kommunikáció mértékét. Inváziónak nevezzük a gyakran kézbesített hirdetéseket, SPAM üzeneteket, nem kívánt telefonhívásokat, illetve az egyéb formáját az elektronikus kommunikációnak. Ezt a technológiát legtöbbször a kommunikációs csatornán végzett személyazonosság kezeléssel kombinálva használják.



3.2. ábra. *Elérhetőség kezelés mobiltelefonon*

A 3.2 ábrán látható a protokoll prototípusának megvalósítása egy 1997-ben forgalomban lévő PDA-ra. Az ábrán az látható, hogy a felhasználó a bejövő hívások fogadása előtt szabályokon keresztül megadhatja, hogy milyen információkat adjon ki más személyeknek.

4. fejezet

PET-ek Android platformra

A biztonsági kérdéseknél részletesen tárgyaltam az ingyenességnek a veszélyeit, az Android platformon keresztül. A továbbiakban megvizsgáljuk, hogy milyen lehetősége van egy Android-os felhasználónak megvédeni a saját privát szféráját.

4.1. Platform szintű módosítások

Az illetéktelenektől való védelem egyik módszere, ha az Android operációs rendszeren végeznek olyan módosításokat, melyek megerősítik a személyes adatok védelmét. Ezeket firmware-eknek nevezzük, melyek a gyári operációs rendszert cseréli le. Fontos megjegyezni, hogy ez egy veszélyes művelet, mivel nem tudjuk, hogy a harmadik fél által készített firmware tényleg megbízható-e és nincsen benne kártékony kód, ugyanis ezeket a szoftvereket nem ellenőrzi a Google. Tovább, ha valamilyen ok folytán a telefon meghibásodik, és nem a gyári operációs rendszer fut a készüléken, akkor már nem lehet igénybe venni a garanciális cserét akkor sem, ha elméletileg A garancia még fenn áll.

Először kettő nem kiforrott firmware technológiát mutatnék be (MockDroid, TaintDroid), melyek csak prototípus szintjén kutatási projektekben vannak jelen, majd egy gyakorlatban is használt fimrware-t a CyanogenMod-ot.

4.1.1. MockDroid [13]

Ez a firmware nem a hozzáférések szigorításával akadályozza meg az alkalmazások jogosultságait, hanem úgynevezett mock-ok segítségével hamis adatot szolgáltat az alkalmazások számára, amikor azok megpróbálnak elérni bizonyos funkciókat. Más szóval, ha egy alkalmazás megpróbálja elérni a személyes adatbázist vagy szenzor adatot lekérdezni, akkor a firmware mindig ettől eltérő információt fog szolgáltatni. Például, ha egy alkalmazás GPS koordinátákat akar lekérdezni, akkor a MockDroid segítségével a felhasználó beállíthatja, hogy mindig hamis adatokat szolgáltasson, illetve akár azt is megadhatja, hogy nem áll rendelkezésre helyzetinformáció, holott a GPS tisztán veszi az adatokat. Ennek a funkciónak a segítségével befolyásolni lehet helymeghatározás alapú szolgáltatások helyes működését.

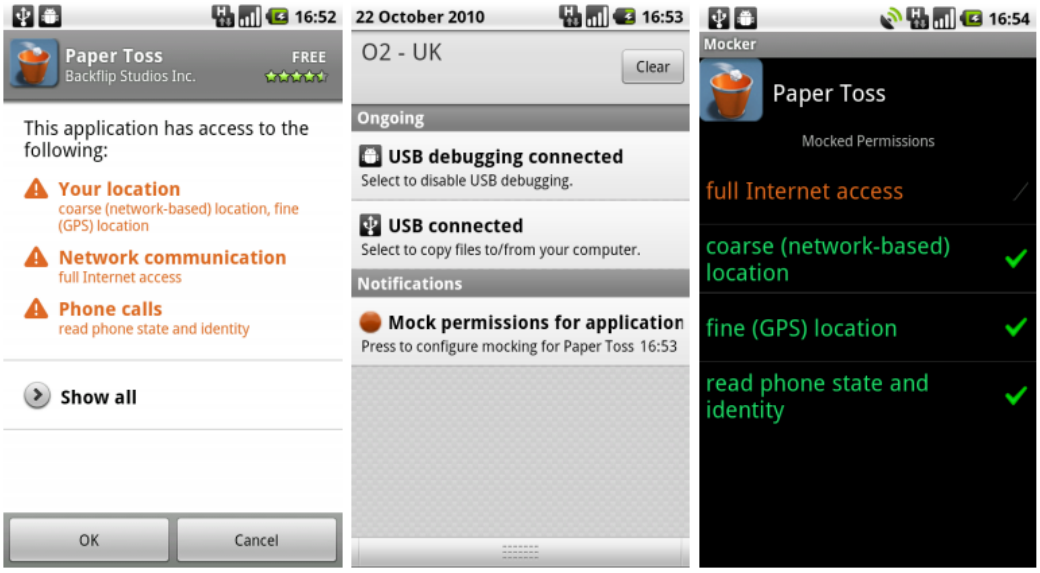
Nézzük meg, hogy általánosságban az operációs rendszer mely funkcióit érdemes módo-

sítani, hogy a privát adatoknak magasabb legyen a védelmi szintje, illetve a MockDroid milyen megoldásokat kínál fel megoldásként:

- **Opcionális funkciók korlátozása**: Gyakori, hogy az alkalmazások - opcionális lehetőségként - hozzáférést kérnek különböző API-khoz, hogy ezzel növeljék a szolgáltatásuk hatékonyságát. Például a *Skype* alkalmazásnál be lehet állítani, hogy a *Contact List*-el összeintegrálja az alkalmazást. A firmware segítségével viszont korlátozhatjuk, hogy milyen adatokat szolgáltatunk az alkalmazások számára.
- **Nincsen nem kívánatos adatmegosztás**: Sok alkalmazás amely képes a személyes adatok lekérdezésére, a normális funkció megvalósítása mellett feltölti egy távoli szerverre az elért adatot. Mock-ok segítségével megakadályozhatjuk az ilyen nem kívánatos adatmegosztásokat.
- **Adatelkülönítés**: Az egyes személyes adatoknál praktikusan meg lehet adni egy részhalmazt, amit az alkalmazások elérnek, míg a többi része rejtve marad tőlük. Például egy *Calendar*-nál megadhatjuk, hogy csak publikus eseményeket szolgáltasson.
- **Drága műveletek szabályozása**: Némely API olyan szolgáltatást vesz igénybe, mely pénz elköltésével jár (például a 3G adatkapcsolati kommunikáció, vagy SMS küldés). A firmware segítségével ezeket a fizetős szolgáltatásokat lehet letiltani úgy, hogy az adott alkalmazás továbbra is tökéletesen működjön.
- **Új funkciók**: Az alkalmazásoknak a felhasználói élmény növelése érdekében vannak olyan funkciói, melyek a készüléknek egy adott állapotától függ. Ilyen például a képernyő orientáció megváltoztatása. A képernyő elforgatása a gyorsulásérzékelő által mért adatoktól függ. Viszont, ha megadjuk, hogy egy mock hamis adatokat szolgáltasson a gyorsulásmérő által mért adatok helyett, akkor a képernyő elfordulása nem fog megtörténni.
- **Tesztelés**: Alkalmazásfejlesztés szempontjából is hasznos, ha a személyes adatok helyett mock-ok által szolgáltatott adatokat használnak bizonyos funkciók letesztelésére.

Érdemes megjegyezni, hogy az Android platform alapesetben úgy lett megtervezve, hogy a funkciók eléréséhez négy jogosultsági fokozatot definiáltak:

- *Normál*: Minden package hozzáfér az adott funkcióhoz.
- *Veszélyes*: A felhasználó számára veszélyes, vagy költséges művelet (például telefonhívások).
- *Aláírt*: A funkciót csak azok a packege-k érik el, melyek ugyanazzal a digitális aláírással vannak ellátva, mint az elérni kívánt funkció.



(a) (b) (c)

4.1. ábra. *A MockDroid működés közben: 4.1(a) A PaperToss alkalmazás telepítésekor olyan jogosultságokat kér, melyek látszólag szükségtelenek; 4.1(b) Ha a felhasználó mock-olt jogosultságot használ, akkor az Android egy notification-ban jelzi számára; 4.1(c) Mock-olt jogosultságok kezelése*

- *Aláírt vagy rendszer*: Tulajdonképpen ugyanaz, mint az *aláírt* jogosultsági szint, csak régebbi platformverziókban maradt benne.

A 4.1 ábrán látható a MockDroid firmware futás közben. A letöltött PaperToss alkalmazás olyan hozzáférési jogosultságokat kér, melyek teljes értékű működéséhez szükségtelenek. A MockDroid a privát adatokhoz való hozzáféréskor, a hozzáférést átirányítja a beállított mock-nak.

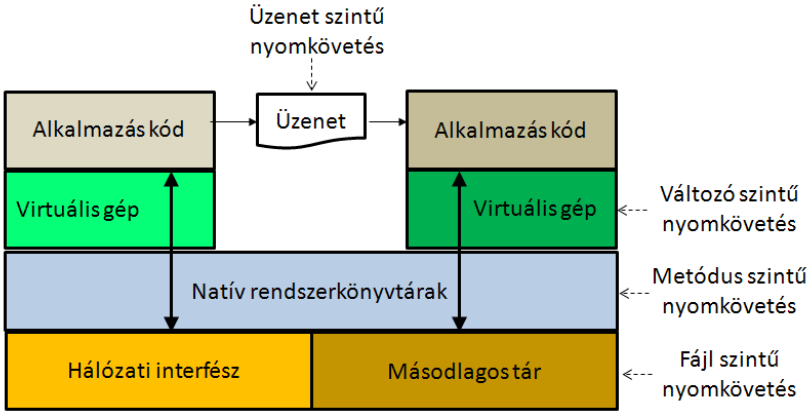
A MockDroid az alábbi funkciókat valósítja meg:

- **Durva- és finom-szemcsézettségű helymeghatározás**: Az alkalmazások sosem kérnek a pozícióról részletesebb adatokat, ha kevés információ áll rendelkezésükre.
- **Internethasználat**: Ha az Internet kapcsolat mock-olva van sosem csatlakozik vezeték nélküli hálózathoz, mindig időtúllépéssel tér vissza.
- **SMS, MMS, Calendar, contact-ok**: A személyes adatokhoz való hozzáférést úgy kezeli, mintha egy új készüléknek üres adatbázisa lenne jelen, illetve minden írási művelet tiltva van.
- **Készülék azonosító**: A készülékazonosító lekérdezésekor mindig egy hamis konstanssal tér vissza.
- **Broadcast Intent-ek**: Ha a valamely alkalmazásnak olyan jogosultságra van szüksége, hogy *Broadcast Intent*-et küldjön, akkor ez az Intent sosem lesz elküldve. Illetve, ha valamely alkalmazásnak olyan jogosultsága van, hogy *Broadcast Intent*-eket fogadhat, akkor sosem kapja meg azokat.

4.1.2. TaintDroid [20]

Az Android jogosultság kezelésének hátránya, hogy csak durva-szemcsézettségű lehetőségek vannak kezelésükre. Nézzünk egy egyszerű példát: Egy alkalmazás hozzáférést kér a készülék pozíciójához, de azt már nem tudjuk ellenőrizni, hogy mikor kérdezte le az aktuális koordinátákat és küldte el azt harmadik fél számára. A TaintDroid segítségével megfigyelhetjük, hogy az alkalmazások mikor veszik igénybe privát adatainkat, és milyen műveletet végeznek velük. Ezáltal tudomásunkra jut, hogy érzékeny adataink mikor kerülnek ki a mobiltelefon környezetéből. A TaintDroid automatikusan címkékkel látja el a következő érzékeny adatokat: változók, fájlok, folyamatok közti üzenetek. Naplózza ezeknek az adatok feldolgozásának tényét, úti célját így azt is, hogy az adat mikor hagyja el a mobiltelefon környezetét. Ezeket a kiegészítéseket úgy oldották meg, hogy a lehető legkevesebb többlet erőforrásra legyen szükség, így a módosítások változó-, metódus-, üzenet- és fájl szinten történtek. Mindezekkel együtt a processzor teljesítménye csak 14%-al romlott.

A keretrendszer alapja a dinamikus jelölés[1] analízis. Az adat kiindulási helye a jelölés forrás (*taint source*), ahol először az adat megjelölése történik. A privát adatot a jelölés kezdeményező (*taint marking*) látja el a megfelelő plusz információval, hogy a továbbiakban követni lehessen. Ezek után a jelölt adat a telefon környezetében nyomon követhető, míg végül egy jelölés elnyelőhöz (*taint sink*) ér, ahol megszűnik a privát adat követése (általában a nyelő egy hálózati interfész szokott lenni). Viszont a jelölés követésnek számos hátránya van. A hatékonyság az adat-instrumentálási[2] szinttől függ, ami szerencsétlen esetben akár 20-szoros működési és feldolgozási többletmunkát okozhat a rendszeren belül, mert a jelölt adat megfigyelése valós időben működik.



4.2. ábra. *A TaintDroid platform architektúrája*

A 4.2 ábrán látható, hogy milyen módon lehet a jelöléskövetést instrumentálni. Először a virtuális gép instrumentálása szükséges, így az egyes változók manipulációjának folyamatát lehet megfigyelni. Ezek a változtatások az adatokon, és nem kódban fognak létrejönni. Másodszor az alkalmazások közötti üzenetváltás instrumentálását kell megoldani. Ennek a

[1] tain jelentése: fertőzés, mivel nincsen magyar megfelelője, így a továbbiakban jelölésnek nevezem

[2] adat-instrumentálás: adat ellátása olyan kiegészítő információkkal, amelyek egy adott rendszeren kívülről elérhetőek

célja, hogy a teljesítménycsökkenést minimalizáljuk, ugyanis a monitorozás nem az üzenet egyes adatain, hanem a teljes üzeneten történik meg. Harmadjára a rendszer szintű natív könyvtárak metódusait kell instrumentálni. Végül pedig a fájl szintű instrumentációval érjük el, hogy a nyomkövetés lekérdezhető, és visszajátszható legyen.

Az egész architektúra valójában a firmware integritásán múlik. Ugyanis a nyomkövető rendszernek meg kell bíznia a nem megbízható alkalmazás által betöltött natív rendszerkönyvtárakban, illetve a virtuális gép által igényelt felhasználói területben. Pont ezért a TaintDroid biztosítja, hogy mindig a natív rendszerkönyvtárak töltődjenek le, és ne az alkalmazás töltse le magának a szükséges osztálykönyvtárakat. Egy korai 2010-es felmérés szerint, az Android Market[3]-ről letölthető, kategóriánként az 50 legnépszerűbb ingyenes alkalmazások (kb. 1100 alkalmazás) 4%-a vesz igénybe külső forrásból osztálykönyvtárat (`.so` kiterjesztésű fájl). Később ezt a felmérést újból elvégezték, és az ilyen alkalmazások száma 5%-ra ugrott.

A TaintDroid architekturális működése:

1. A privát adatot megjelöli egy megbízható alkalmazás.
2. A jelölő interfészen keresztül egy natív metódushívás történik a Dalvik virtuális gép felé, amivel elérheti a jelölőkészletet.
3. Ezt a megjelölt adatot a megbízható alkalmazás felhasználhatja.
4. Folyamatok közti kommunikáció során a megbízható alkalmazás ugyanúgy belerakja az üzenetbe a megjelölt adatot (`parcel`), mint az többi elküldendő részletet.
5. Ezt az üzenetet megkapja a nem megbízható alkalmazás.
6. Mivel a *Binder Library* módosítva van, így a nem megbízható alkalmazás a jelölt adatot is ugyanolyan formában fogja kiolvasni, mint ahogy elküldésre került.
7. A nem megbízható alkalmazásban a különböző adathozzáférések hatására a megjelölt adat szétterjed az alkalmazásban.
8. Amikor a nem megbízható alkalmazás olyan natív metódushívást hajt végre, melyben jelölő elnyelő található, akkor eltávolítja a jelölő információkat a jelölt adatról.
9. Az elnyelő által végzett művelet naplózásra kerül.

Az architektúra implementálásához az alábbi komponenseket kell megtervezni:

- Jelölés tároló
- Interpretált kód jelölés nyomkövetés
- Natív kód jelölés nyomkövetés
- Folyamatok közötti jelölés nyomkövetés
- Másodlagos tár jelölés nyomkövetés

[3] most Google Play

4.2. CyanogenMod [5]

A CyabogenMod egy nyílt forráskódú firmware, melynek célja a készülék teljesítményének fokozása. Számunkra viszont azon funkciói az érdekes, melyek segítségével növelni tudjuk saját adataink védelmét.

Mint már fentebb ismertettem egy Androidos alkalmazás felsorolja, hogy milyen jogosultságokat vesz igénybe működése során. A CyabogenMod fejlesztői alapvetően azt vették figyelembe, hogy ezeket jogosultságokat az alkalmazás telepítése után lehessen módosítani.

A biztonsági beállításoknál aktiválni kell a CyanogenMod által nyújtott lehetőségeket. Ekkor a telefon képernyőjén megjelenik egy figyelmeztetés, hogy az egyes jogosultságoknak a tiltásával lehetséges, hogy nem érjük el az alkalmazás összes funkcióját. Ezek után lehetőségünk van az alkalmazások jogosultságait letiltani, illetve később akár engedélyezni is. A módosítások elvégeztével a telefont újra kell indítani, ugyanis az elvégzett módosítások ekkor lépnek érvénybe.

Ha valaki szeretné a CyanogenMod-ot feltelepíteni első körben a készítő honlapján található eszköztámogatási listában kell megnézni, hogy a a firmware-t fel lehet-e telepíteni a készülékére. Fontos még egyszer megjegyezni, hogy a CyanogenMod, vagy bármely más firmware feltelepítése esetén a készülék garanciája elvész.

4.3. PET alkalmazások

A privát szférának a védelmére nem csak operációs szintű módosítások léteznek, hanem számos alkalmazást kifejlesztettek, melyek az ismert problémákra próbál megoldást nyújtani. Előnye az, hogy nem kell az operációs rendszert részlegesen, vagy teljesen lecserélni, viszont a hatékony alkalmazásokat legtöbbször éves előfizetési díj fejében lehet csak alkalmazni. A továbbiakban néhány Android-os alkalmazást mutatnék be, melyek növelik a felhasználó privát szférájának védelmét.

4.3.1. Droidhunter [6]

Ez az alkalmazás egy teljes védelmi rendszer, mely vírusirtó, behatolás detektáló, online szolgáltatás menedzselő funkcióval rendelkezik. Hasonlít a desktop környezetben megtalálható védelmi rendszerekhez, annyi különbséggel, hogy Android specifikus támadásokra és veszélyekre van kiélezve. Az alkalmazásnak létezik egy ingyenes és egy fizetős változata is. A fizetős változatban lehetőség van elvesztett, illetve ellopott telefon bemérésére.

4.3.2. App Protector [3]

Az alkalmazás segítségével kiválaszthatjuk azokat az alkalmazásainkat, amelyeket csak az általunk megadott jelszóval, vagy képernyőn kirajzolt mintával lehet elérni. Az uninstaller alkalmazásoknak kötelező megadni ezt a védelmet, ugyanis enélkül egy tolvaj törölheti az *App Protector*-t, így az alkalmazásokon lévő védelem megszűnik. Az alkalmazást 7 napig ingyen ki lehet próbálni, a további használata előfizetéshez kötött.

4.3.3. Photo Vault [10]

Ennek az egyszerű alkalmazásnak a segítségével fényképeket, illetve fénykép albumokat láthatunk el általunk megadott jelszavakkal.

4.3.4. Video Vault [11]

Hasonló a *Photo Vault* alkalmazáshoz, csak képek helyett videók jelszavas védelmét teszi lehetővé.

4.3.5. aFirewall [2]

Ezzel a tűzfal alkalmazással tiltani lehet a megadott számokról érkező hívásokat, illetve SMS-eket. Az alkalmazás lehetővé teszi azt is, hogy a bejövő kommunikációt bizonyos időszakokra korlátozzuk, de akár létrehozhatunk egy fekete listát is a nem kívánatos telefonszámokról. A tűzfalbeállításokat elmenthetjük, módosíthatjuk, és bármikor aktivizálhatjuk. Az alkalmazásnak létezik egy ingyenes és egy fizetős változata, melynél csak az alkalmazás árát kell kifizetni és teljes értékű Android tűzfalként működik.

4.3.6. Lookout Mobile Security [7]

Az egyik legnépszerűbb alkalmazás, amit a Google Play-ről letöltöttek (kb. 15 millió letöltés csak az Android platformra). Az alkalmazás ingyenes verziója az alábbi funkciókat tartalmazza:

- **Vírusirtó**: Kártékony-, kémprogramok és trójai vírusok detektálására alkalmas. A víruskeresést el lehet végezni bizonyos időközönként automatikusan, de egyéni keresés kezdeményezésére is van lehetőség.
- **"Találd meg a telefonom"**: Követi a telefon pozícióját. Ha a GPS ki van kapcsolva, akkor aktivizálni lehet távolról, hogy a funkció működjön.
- **Biztonsági mentés**: Elmenti és helyreállítja a *Contact List*-et.

Az alkalmazás fizetős változata havidíj fizetése ellenében vehető igénybe. A fizetős verzió a további szolgáltatásokat tartalmazza:

- **Biztonság**: Online antivírus védelem és személyes adat tanácsadó.
- **"Találd meg a telefonom"**: Megkeresi és zárolja a telefont.
- **Biztonsági mentés**: A *Contact List*-en kívül a hívás naplót és a képeket is el lehet menteni, illetve helyreállítani.

4.3.7. Orweb [8]

Ez egy olyan böngésző alkalmazás, amely webes szűrőket és tűzfalakat alkalmaz, miközben a felhasználó anonim marad az internet kapcsolat idejére. Funkció közé tartozik:

csak a fehér listán szereplő cookie-kat fogadja, nem tárol előzményeket, kikapcsolja a Flash-t, illetve csak azokat a weboldalakat engedélyezi melyekhez összesen az internet hozzáférési jogosultság szükséges. Meggátolja, hogy illetéktelen személy kinyomozza milyen weboldalakat néz felhasználó, illetve weboldalak nem kérdezhetik le a telefon földrajzi pozícióját.

5. fejezet

PET-ek iOS platformra

Valójában az iOS platform tág fogalom, mert az Apple által elkészített összes termék e platform felett fut. Az esszében kifejezetten a iPhone készülékek privátszféra kezelését és támogatását vizsgálom. Alapvetően a probléma ugyanaz, mint az Android-os készülékek esetén: a szolgáltató, a gyártó képes privát adatokhoz hozzáférni a felhasználó tudta nélkül. Persze iPhone esetében a legtöbb alkalmazást meg kell vásárolni a felhasználók mégsem érezhetik magukat biztonságban. Például az utóbbi időkben olyan kártékony alkalmazásokra bukkantak, melyek a felhasználó fotóalbumához férhetnek hozzá [4].

Az iOS platform biztonságának növelése érdekében hasonló lehetőségek vannak, mint Android platform esetén. Ugyanúgy léteznek harmadik fél által készített firmware-k, melyek root jogot adnak a készülék használatához, illetve lehetőség van különböző PET alkalmazások letöltésére is. Az alkalmazásokra nem térnék ki, mert nagyon hasonlítanak az Android platformon elérhetőkhöz, sőt sok iPhone-ra elérhető alkalmazást portoltak át Android-ra, így gyakran elérhető ugyanaz az alkalmazás mind a két platformra.

5.1. iPhone biztonsági beállítások [18]

Ebben a fejezetben néhány iPhone készüléken alkalmazható biztonsági beállítást ismertetek. Igaz a beállítások iPhone specifikusak, de más platformokon is hasonló intézkedéseket lehet végezni.

5.1.1. Biztonsági kód bekapcsolása bekapcsolása

A funkció hatására az inaktív állapotban lévő telefon egy bizonyos idő eltelte után kérni fogja a felhasználó által megadott biztonsági kódot, aminek helyes megadása után a telefon ismét üzemképes lesz.

A beállítást az alábbi módon lehet elvégezni:

1. Settings $\Longrightarrow$ General $\Longrightarrow$ Passcode Lock.
2. Turn Passcode On.
3. Ekkor kétszer egymás után be kell gépelni egy négy számjegyből álló jelszót.

Az automatikus zárolás időtartamát 1 és 5 perc között adhatjuk meg. Továbbá beállíthatjuk, hogy 10 sikertelen kísérlet után a telefon törölje a személyes adatokat. Ebben az esetben előfordulhat, hogy a telefont nem lopták el, hanem például valamelyik ismerős, vagy rokon próbálkozik a telefon aktiválásával. Ekkor az első pár próbálkozás után 1 perces zárolás történik a következő újrapróbálkozásig, majd ez az időtartam minden egyes sikertelen próbálkozás után folyamatosan nő, egészen 30 percig. Ez a módszer csökkenti annak az esélyét, hogy az adatok valamilyen véletlen folytán elvesszenek.

5.1.2. Biztonsági kód nélküli funkciók kikapcsolása

iPhone telefonon alapértelmezett beállításként meg van adva néhány funkció, melyek a biztonsági kód kérése nélkül elérik a telefon bizonyos szolgáltatásokat. Az alábbiakban megnézzük melyek ezek, és mit lehet velük kezdeni.

Hangtárcsázás

Ha a lezárt iPhone-on megnyomjuk a Home gombot, akkor hozzáférünk a hangtárcsázáshoz. A funkció alapértelmezetten nem kéri a biztonsági kódot, aminek hatására bárki a *Contect List*-ből felhívhatja egy ismerősünket, vagy lejátszhatja a telefonon található zenéket.

A kikapcsoláshoz a következőket kell tenni:

1. Settings $\Longrightarrow$ Password Lock $\Longrightarrow$ Voice Control.
2. Turn Voice Dial to OFF.

SMS előnézet

Az SMS előnézet megjelenítése azért lehet veszélyes egy felhasználó számára, mert rengeteg szolgáltatás (például bankok, megrendelési adatok) küld SMS-ben érzékeny információkat (jelszavak, megerősítési kód). Célszerű, hogyha ezek az információk nem jutnak még véletlenül sem illetéktelen személy tudomására.

A kikapcsolást az alábbi módon lehet megtenni:

- Settings $\Longrightarrow$ Messages $\Longrightarrow$ Show Preview $\Longrightarrow$ kikapcsolás.

5.1.3. Személyes adatok kezelése

Az iPhone készülékek gyakran tárolnak el a felhasználó tudta nélkül olyan adatokat és információkat, amelyeket illetéktelen kezek könnyen felhasználhatnak. Ezekre a funkciókra fokozottan ügyelnünk kell, nehogy hanyagság miatt kiszivárogjanak a személyes adataink.

Billentyű gyorsítótár

A készülék minden egyes billentyűlenyomást 12 hónapon keresztül tárol a `/var /mobile /Library /Keyboard /dynamic-text.dat` fájlban. A gyorsítótár tárolja az összes bil-

lentyűzetkódot, amit a oldalak regisztrációkor kellett megadni, viszont a jelszómezők tartalmát sosem menti el.

Gyorsítótár törlése:

1. General $\Longrightarrow$ Reset.
2. Reset Keyboard Dictionary.
3. Megerősítés.

Automatikus képernyőkép készítés

A Home billentyű lenyomásakor a készülék egy képernyőképet készít, hogy a billentyű lenyomása után történő effekteket végre tudja hajtani. A telefon valószínűleg az alkalmazás bezárása után törli az elkészített képet, de a legtöbb esetben ez nem maradandó törlést jelent. Az eltárolt képernyőkép azért lehet veszélyes, mert tartalmazhat érzékeny, illetve személyes adatokat, amiket így illetéktelen személy is megtudhat. Egyik fajta elkerülés az lehet, hogy a Home billentyű megnyomása előtt a felhasználó elnavigál egy másik ablakba, ahol nincsen érzékeny adat, és ezen az oldalon nyomja meg a Home billentyűt.

Tapasztaltabb felhasználók akár ki is kapcsolhatják ezt a funkciót. Ehhez egy feltört iPhone-ra van szükség.

Geotagging

Egy képen belül tárolt szélességi és hosszúsági fokokat nevezik geotagging-nek. Az iPhone készülék az általa készített képeket automatikusan ellátja geotagging paraméterekkel. Ez olyan veszélyekkel járhat, mint például a felhasználó a saját házáról készült képet ellátva paraméterrel jóformán a címet is megtudhatják illetéktelen személyek.

Ezt a funkciót az iPhone készüléken belül a helymeghatározás alapú szolgáltatások menüpont alatt lehet kikapcsolni:

- Settings $\Longrightarrow$ Location Services $\Longrightarrow$ Camera kikapcsolás.

Helykövetés

Az iPhone készülékek korábban titkosítva naplózták a GPS koordinátákat, amit a `consolidated.db` fájlban tároltak, majd a fájl mindig szinkronizálódott az iTunes-on. Most már a fájl titkosítatlan formában elérhető a készüléken.

A GPS-es nyomkövetés kikapcsolására az alábbi lehetőséget lehet tenni:

1. Cydia tört telefonok esetén az Untracked alkalmazás törli az adatokat a `consolidated.db` fájlból.
2. iPhone készülék titkosítása az iTunes-on.
3. Settings $\Longrightarrow$ Location Services $\Longrightarrow$ kikapcsolás.
4. Migráljuk az iOS verziót 4.3.3 verzióra, vagy nagyobbra.

5.1.4. Összes adat törlése garanciális csere-, szervizbe leadás-, eladás előtt

Az iPhone készülékek Restore (gyári beállítások visszaállítása) funkcióját alkalmazva önmagában nem jelent biztonságos törlést, ugyanis ezek után még mindig lehetőség van a perzisztens adatok visszanyerésére (ilyen eszközöket használnak törvényszéki vizsgálatoknál is).

Nézzük meg, hogy milyen műveleteket érdemes elvégezni, mielőtt megválunk a készüléktől:

1. Változtassuk meg a készüléken tárolt e-mail fiókok, közösségi oldalak, banki oldalak jelszavait.
2. Settings $\Longrightarrow$ General $\Longrightarrow$ Reset.
3. Reset All Settings és ennek megerősítése.
4. Settings $\Longrightarrow$ General $\Longrightarrow$ Reset $\Longrightarrow$ Erase All Content and Settings.
5. Hajtsuk végre az iTunes-on a telefon újratelepítését.
6. Az iTunes segítségével vegyük ki az összes szinkronizációt a fotókra, videókra, zenékre és egyéb tartalmakra.
7. Készítsünk három elkülönült lejátszási listát. Mindegyiknek a mérete legyen akkora, mint a telefonnak a háttértár kapacitása.
8. Szinkronizáljuk a telefonra az első lejátszási listát. Utána a második lejátszási listát szinkronizáljuk a telefonra, majd a harmadikat.
9. Ismételjük meg az előző műveletet még kétszer.
10. Ezek után telepítsük újra a telefont az iTunes-on.

Egy másik módszer, hogy ha az iErase alkalmazást használjuk, miután letöröltük a személyes adatokat és elvégeztük az újratelepítést az iTunes-on.

5.1.5. Rendszeresen frissítsük a készülék firmware-ét

Minden egyes firmware frissítéssel a rendszerben felfedezett újabb hibákat, és biztonsági lyukakat javítanak ki, amivel a rendszer stabilabb működését érik el. Az iPhone készülék firmware verziójáról információt az alábbiakban érhetjük el:

- Settings $\Longrightarrow$ General $\Longrightarrow$ About

A legfrissebb verzióról az információt az Apple honlapján, vagy Twitter-én lehet elérni, a frissítést az iTunes-on keresztül lehet elvégezni.

5.1.6. A tört firmware dilemmája

A tört verziójú (jailbreaked) firmware olyan iOS firmware, ami egyedi kernelt használ, hogy javítsa, illetve megváltoztassa az gyári operációs rendszer tulajdonságait. A tört firmware lehetővé teszi, hogy a felhasználó olyan alkalmazásokat is feltelepíthessen és futtathasson, melyeket az Apple nem hitelesített. A tört verziók használatát 2010-ben a DMCA-n alatt legálissá tették.

A tört verzió hátrányai:

- A tört verziójú firmware alkalmazásával nagyobb az esély, hogy féreg, vagy kártékony kód kerül a készülékre, pont azért, mert nem szükséges, hogy az Apple által hitelesítve legyenek az alkalmazások.
- Az alkalmazások root jogosultsággal futnak elhagyva ezzel a gyári verzió által biztosított felügyelt környezetet.
- Mindig újra kell törni a készüléket, ha a gyári firmware-nek újabb verziója jelenik meg. Ilyenkor a tört alkalmazásokat is újra kell telepíteni.

A tört verzió előnyei:

- A tört verziójú firmware-re gyorsabban jönnek a frissítések, mint a gyári verzióra.

A tört firmware használata semmiképpen sem javasolt tapasztalatlan felhasználóknak.

5.1.7. Safari böngésző biztonsági beállításai

A beállításokat a következőképpen érhetjük el: Settings $\Longrightarrow$ Safari.

Felugró ablakok blokkolása

A következőképpen blokkolhatjuk a felugró ablakokat: Settings $\Longrightarrow$ Safari $\Longrightarrow$ Block Pop-ups bekapcsolása.

Automatikus kiegészítés kikapcsolása

Alapértelmezetten a Safari böngésző megjegyzi az űrlapokon kitöltött adatokat. Kikapcsolásával elkerülhetjük, hogy címek, jelszavak, illetve egyéb érzékeny adatok tárolásra kerüljenek. Ilyen automatikus kiegészítő funkciókkal rendelkeznek az asztali környezetben ismeretes böngészők is. A kikapcsolást a következőképpen végezhetjük el: Settings $\Longrightarrow$ Safari $\Longrightarrow$ Auto Fill kikapcsolás.

Figyelmeztetés gyanús weboldalak esetén

Bekapcsolásával a böngésző figyelmeztetheti a felhasználót ismeretlen honlapok esetén, hogy gyanús tevékenységet (adathalászat, social engineering) végeznek. Bekapcsolása a következőképpen történik: Settings $\Longrightarrow$ Safari $\Longrightarrow$ Fraud Warning bekapcsolás.

Sütik, előzmények, gyorsítótár törlése

Ha olyan weboldalt böngésztünk (banki, közösségi oldalak) érdemes az törölni az alkalmazás által elmentett adatokat. A törlő funkciók megtalálható a Settings $\Longrightarrow$ Safari menüpont alatt.

Adatbázisok törlése

A HTML 5 weboldalak esetén további intézkedéseket kell tennünk saját privátszféránk védelme érdekében, ugyanis néhány weboldal adatbázisokat készít a készüléken. A következőképpen kezelhetjük a weboldalak által igényelt adatbázisokat:

1. Settigns $\Longrightarrow$ Safari $\Longrightarrow$ Databases.
2. Töröljük azokat az adatbázisokat, melyek tudtunk nélkül jöttek létre.

5.1.8. Bluetooth, Wi-fi, E-mail biztonságos használata

A használaton kívül levő Bluetooth portot mindig kapcsoljuk ki, ugyanis a támadó felhasználhatja az aktív portot, hogy kártékony kódot küldjön a telefonunkra. Ilyen támadás lehet a BlueJacking meg a BlueSnarfing. Az iPhone készülékek nem teszik lehetővé, hogy kikapcsoljuk a készülék felderíthetőségét, ezért a Bluetooth portot minden használat után kapcsoljuk ki. Az alábbit kell tenni a kikapcsoláshoz: Settings $\Longrightarrow$ General $\Longrightarrow$ Bluetooth kikapcsolás.

Hasonlóan a Wi-fi kikapcsolása is ajánlott. Miután ez megtörtént a készülék átvált cella alapú adatkapcsolatra, ami azon kívül, hogy az akkumulátort jobban kíméli még a telefon nyomon követhetőségét is megnehezíti. Alapértelmezetten az iPhone készülék megpróbál automatikusan csatlakozni egy regisztrált Wi-fi hálózathoz, amint annak hatósugarába ért. Ezt a következőképpen tudjuk kikapcsolni:

1. Settings $\Longrightarrow$ General $\Longrightarrow$ Wi-fi.
2. A listában szereplő Wi-fi kapcsolat SSID-jára kattintás.
3. Forget this Network.
4. Ismételjük meg mindegyik felvett Wi-fi kapcsolatra.

Ha Hotmail-t, vagy Gmail-t használunk, akkor győződjünk meg arról, hogy az SSL (Secure Socket Layer) be van kapcsolva. Enélkül az üzenet titkosítatlanul kerül továbbításra, egy hozzáértő könnyen megszerezheti az elküldött e-mail-eket.

1. Settings $\Longrightarrow$ Mail, Contacts, Calendar.
2. Egy aktív e-mail fiók kiválasztása.
3. Advanced.
4. Use SSL ON.

5.1.9. Find My iPhone szolgáltatás bekapcsolása

A szolgáltatás az Apple által biztosított ingyenes funkció iOS 4.2. verziójú operációs rendszerek esetén. Az alkalmazás GPS-en keresztül segíti az elvesztett, vagy ellopott mobiltelefon nyomon követését. Az alkalmazást először telepíteni kell, majd be kell állapítani az alábbi módon:

1. Az AppStore-ból le kell tölteni a Find My iPhone alkalmazást.
2. Settings $\Longrightarrow$ Mail, Contact, Calendar $\Longrightarrow$ Accounts, majd itt adjuk hozzá a MobileMe fiókot.
3. Lépjünk be a MobileMe fiókba az Apple azonosító és jelszó segítségével.
4. Kapcsoljuk be a Find My Phone szolgáltatást.
5. Menjünk a telepített alkalmazásokhoz majd lépjünk be az Apple azonosító és jelszó segítségével.

Amint elvégeztük a beállításokat a MobileMe weboldalon keresztül nyomon követhetjük saját telefonunkat ha netán elveszítenénk. Fontos, hogy a szolgáltatás használatához használjunk biztonsági kódot, mert különben a tolvaj könnyen eltávolítja a nyomkövető alkalmazást.

6. fejezet

Összefoglalás

Mobiltelefonon a személyes adatok megvédése alapvetően nehéz dolog. Mikor adatainkat védeni szeretnénk nem elég csak arra gondolni, hogy a lopások által okozott károkat minimalizáljuk, hanem a kommunikációs csatornákon érkező támadásokat is hatékonyan ki kell tudni védeni. Márpedig ha egy készülék csatlakozik az internethez - és mobiltelefonoknál ez különösen igaz - akkor veszélynek van kitéve a legkülönb támadásoknak. Védekezni nem elégséges pusztán a rosszindulatú harmadik fél ellen, hanem a szolgáltatók, operátorok és készülék gyártók ellen is el kell rejtenünk személyes adatainkat, ugyanis a tudtunk nélkül monitorozzák a mobiltelefonokat.

Az esszé során végighaladtam egy folyamaton, kezdve attól, hogy miért szükségesek a privátszférát erősítő technológiát használni, konkrét, platformspecifikus megoldásokig. A legelején megvizsgáltam, hogy a különböző szolgáltatók mit garantálnak személyes adataink biztonságáért, és katasztrofális eredményekre jutottam. Tulajdonképpen a szolgáltatók potenciális veszélyt jelentenek a felhasználók számára, így tőlük jobban kell félni, mint a tolvajoktól, hiszen - kicsit sarkítva - legálisan hozzáférhetnek az érzékenyebb adatokhoz. Külön kitértem az ingyenes alkalmazások veszélyeire, azon belül is a Google adatvédelmi szabályait, illetve az Android platformon lévő alapértelmezett biztonsági szintet vizsgáltam.

A továbbiakban néhány általános védekezési módszert ismertettem különböző esetekre. Külön kitértem, hogy a tolvajok ellen milyen védekezési módszereket alkalmaznak, majd néhány protokollt mutattam be, amik a helymeghatározás alapú szolgáltatások, illetve a személyazonosság kezelés szempontjából elrejti a felhasználónak az adatait harmadik fél számára.

A következőkben azt vizsgáltam, hogy privát adatainkat hogyan védhetjük meg Android és iPhone platformon. A két platform a népszerűsége, a rengeteg ingyenes alkalmazás, és a nyílt forráskódú operációs rendszer (iPhone esetén tört verzió) miatt érdemel külön hangsúlyt. Ha az operációs rendszer szintjén létesítenek változtatásokat, azok valamennyi alkalmazásra hatással lesznek, így egységesen kezelhetjük mindegyiket, még azokat is, amelyek csak később jelennek meg. Ezzel szemben az alkalmazások magasabb szintről próbálnak védelmet nyújtani, sokszor nem is látnak az operációs rendszer szintjéig. Ilyen, vagy ehhez hasonló alkalmazások más mobilplatformokon is fellelhetőek sőt, a népszerűbb

alkalmazásokat - és a biztonsági alkalmazásokkal is így van - Android platformra is megírják. iPhone esetén bemutattam néhány bárki által elvégezhető biztonsági intézkedést. Ezek között találhatóak olyan megoldások is, melyek más platformon is könnyűszerrel alkalmazhatóak.

Összességében a privátszférát erősítő technológiák mobiltelefonokon még nagyon gyerekcipőben járnak. Jogi szabályozásokra lenne szükség, hogy a szolgáltatók ne tudják a felhasználók személyes adatait megszerezni. Ennek hiányában sajnos a személyes adatoknak a védelmét technológiai eszközökkel kell megoldani. Viszont a legtöbb esetben a felhasználókat hiába védik szigorú adatvédelmi szabályok, illetve hiába rendelkeznek a legerősebb adatbiztonsági protokollokat megvalósító alkalmazásokkal, a felhasználók saját hiszékenységük és óvatlanságuk következtében szivárogtatják ki az adatokat egy harmadik fél számára. Az ilyen fajta támadásokat social engineering[1]-nek nevezzük.

[1] http://en.wikipedia.org/wiki/Social_engineering_(security)

Ábrák jegyzéke

Irodalomjegyzék

[1] Adatvédelmi irányelvek. http://www.google.com/policies/privacy/.

[2] afirewall. http://androidfirewall.appspot.com/.

[3] App protector. http://www.carrotapp.com/portfolio_1/app-protector/.

[4] Congress has more questions over apple iphone privacy. http://www.forbes.com/sites/connieguglielmo/2012/03/14/congress-has-more-questions-over-apple-iphone-privacy/.

[5] Cyanogenmod. http://www.cyanogenmod.com/.

[6] Droidhunter. http://droidhunter.wordpress.com/.

[7] Lookout mobile security. https://www.mylookout.com/.

[8] Orweb: Proxy+privacy browser. http://guardianproject.info/apps/orweb/.

[9] Personal identification number. http://en.wikipedia.org/wiki/Personal_identification_number.

[10] Photo vault. http://www.pacificsoftwaresolutions.net/photovault.html.

[11] Video vault. http://www.pacificsoftwaresolutions.net/videovault.html.

[12] What is the price of free? http://www.cam.ac.uk/research/news/what-is-the-price-of-free/.

[13] Ripduman Sohan Nicholas Skehin Alastair R. Beresford, Andrew Rice. Mockdroid: trading privacy for application functionality on smartphones. Technical report, Computer Laboratory, University of Cambridge, 2011.

[14] Dr. Forstner Bertalan. *Bevezetés a mobil szoftverfejlesztésbe - PyS60 PIM funkciók elérése előadásfóliák*. BME VIK AUT, 2008. In Hungarian.

[15] Lothar Fritsch. State of the art of privacy-enhancing technology (pet). Technical report, Norsk Regnesentral, Norwegian Computer Center, 2007.

[16] Urs Hengarten. *Privacy-Enhanced Technologies for Mobile Applications előadásfóliák*. David R. Cheriton School of Computer Science, University of Waterloo. Előadás: http://www.youtube.com/watch?v=7d6-vyOTJsg.

[17] Dr. Ekler Péter. *Az Android platform bemutatása*. BME VIK AUT, 2011. In Hungarian.

[18] Kunjan Shah. Top 10 iphone security tips. Technical report, 2010. www.mcaffe.com.

[19] Kowashik Prakasam Shashank Hegde. Golden eye: A face recognition based authentication system for smartphone applications. Technical report, University of California Santa Barbara, Norwegian Computer Center, 2007.

[20] Byung-Gon Chun Landon P. Cox Jaeyeon Jung Patrick McDaniel Anmol N. Sheth William Enck, Peter Gilbert. Taintdroid: An information-flow tracking system for realtyme privacy monitoring on smartphones. Technical report, 2010. http://appanalysis.org/.

HALLGATÓI NYILATKOZAT

Alulírott Rádi Attila (VTV2CQ), jelen esszé dolgozat házi feladatként történő benyújtásával kijelentem, hogy a művet meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

☐ Hozzájárulok,

☐ Nem járulok hozzá,

hogy a dolgozatom a beadott formájában, vagy átdolgozás után megjelenhessen a PET Portálon.

Kelt: Budapest, 2012. 03. 28.

.....
Rádi Attila (VTV2CQ)