Privát adatok publikálása a weben

Paulik Tamás — Földes Ádám Máté — Gulyás Gábor György*

Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Híradástechnikai Tanszék
1117 Budapest, Magyar tudósok körútja 2., I ép. IB 113.
Telefon: +36 1 463–3227, Fax: +36 1 463–3263
e-mail: paulik.tamas.email@gmail.com

Absztrakt

A Web 2.0-s szolgáltatások egyik fő jellemzője, hogy a felhasználók önkéntes adatszolgáltatására építenek, ami manapság igen elterjednek tekinthető, hiszen a közösségi szolgáltatások térnyerésével ez a jelenség jócskán felerősödött. Azonban az internetre felkerülő adatok eltávolítására a publikálás után jellemzően nincs már mód, így azok végérvényesen elérhetővé válnak mindenki számára. Ennek kapcsán felmerül a kérdés, hogy a közösségi, vagy az egyéb információ megosztásra használt oldalakon, mint például a mikroblogok, hogyan szabályozzuk az információkhoz való hozzáférést. Erre a kérdésre válaszként több megoldás is született már. Tanulmányunkban elemezzük a felhasználókra leselkedő veszélyeket, majd áttekintjük azokat a technikákat, amelyek segítségével az önkéntes adatszolgáltatás során közzétett információkhoz való hozzáférés valamilyen módon szabályozható, és ezekre a technológiákra egy osztályozási szempontrendszert és egy taxonómiát is javasolunk.

Kulcsszavak: web, profilírozás, megfigyelés, adatgyűjtés, titkosítás

1. Bevezető

Manapság egyre gyakrabban olvasni olyan esetekről, amikor a privátszféra sérülésére visszavezethető okokból ért kár embereket. Elég csak a Facebook[1]-bejegyzések miatt kirúgott alkalmazottakra [12][13] gondolnunk, hogy belássuk: a világhálóra felkerülő, személyes jellegű információk komoly veszélyeket hordozhatnak magukban, hiszen ezek

* Paulik Tamás, a Budapesti Műszaki és Gazdaságtudományi Egyetem elsőéves MSc-s mérnök informatikus hallgatója; Földes Ádám Máté, a BME Villamosmérnöki és Informatikai Kar Híradástechnikai Tanszékének doktorandusz hallgatója; Gulyás Gábor György, a BME Villamosmérnöki és Informatikai Kar Híradástechnikai Tanszékének doktorandusz hallgatója.

[1] A Facebook hivatalos honlapja: http://www.facebook.com

eltávolítására jellemzően nincs utólag mód, így azok bárki számára kereshetővé, gyűjthetővé válnak.

Az egyetlen személyhez tartozó, az internet különböző forrásaiból fellelt információk egyesítését nevezzük profilírozásnak. Az ilyen tevékenységet folytató szereplőket alapvetően három kategóriába sorolhatjuk.

Az elsőbe az *információs szuperhatalmak* tartoznak – példaképp a Google-t hozhatjuk fel, amely cég a széles, ráadásul ingyenesen kínált termékpalettája (keresőmotor, GMail, Buzz (Zümm), Calendar (Naptár), Groups (Csoportok) stb.) segítségével ösztönzi felhasználóit szolgáltatásainak használatára, mely előbb-utóbb természetszerűleg személyes információk teljesen önkéntes átadását fogja eredményezni. A begyűjtött adatok sokfélesége okán a szolgáltató részletes profilt képes alkotni a felhasználó érdeklődési köréről, szokásairól, körülményeiről, melyet többek közt célzott reklámok küldésére vagy dinamikus árazásra használhat [19], arról nem is beszélve, hogy – az adatkezelési nyilatkozattól függően – akár harmadik fél részére is kiszolgáltathat. Megjegyzendő, hogy az információs szuperhatalomnak nem feltétlenül szükséges kifinomult technikákat alkalmaznia a profil felépítéséhez, hiszen a felhasználó voltaképpen önként profilírozza magát.

A második kategória a *publikus profilírozók* csoportja. Ezek a szereplők szintén az önkéntes adatszolgáltatásra építenek: azok az információk érdeklik őket, melyeket a különböző Web 2.0 szolgáltatások segítségével nyilvánosságra hoztunk. Motivációként elképzelhető például egy hiteligénylő hátterének részletesebb megismerése, mielőtt a bank döntene a kölcsönnel kapcsolatban. A publikus profilírozók többféle szolgáltatáson is kutathatnak egy adott személy után, és könnyen előfordulhat, hogy különböző pszeudonimek használata esetén is képesek az azonos felhasználóhoz tartozó publikus profilok összekötésére [21]. A profilírozók célja lehet továbbá egy személy kapcsolati hálójának feltérképezése, melyhez aktív és passzív támadásokat indíthatnak a célba vett szolgáltatások ellen. A támadáshoz felhasználhatóak például a közösségi hálózatok anonimizált exportjai, és megfelelő technikákkal lehetséges deanonimizálni azokat. Megvalósítható továbbá az azonos felhasználókhoz tartozó csomópontok összekötése is két különböző exportban.

A harmadik kategóriát a *nyomkövetéses profilírozók* alkotják. Ezen szereplők célja, hogy a látogatót azonosítsák a weboldalon, és megpróbálják az így nyert azonosítót előzményekhez csatolni, és ezt a tevékenységi-, ízlésprofilt tovább bővíteni. A „klasszikus” profilírozás harmadik fél által a látogató gépén elhelyezett http sütiken alapszik. Ezek olyan kisméretű fájlok, melyekben a profilírozó fél valamilyen weblapba ágyazott tartalom – például egy hirdetés – segítségével elhelyez egy egyedi azonosítót. Ha egy másik weblap szintén beágyazza ugyanezt a tartalmat, a profilírozó a sütit lekérdezve értesül arról, hogy a két lapletöltés ugyanazon böngészőprogramtól származik. Némiképp kifinomultabb

technikának tekinthető a Flash sütik[2] használata. Ezek is hasonlóan kis fájlok, mint a http sütik, és a céljuk is hasonlatos, azonban van néhány fontos különbség. Egyrészt nem olyan triviális törölni őket, lévén a Flash lejátszó tárolja őket, nem pedig a böngészőprogram (ezáltal pedig a mai modern böngészők „privát böngészés" opciója sincs rájuk hatással). Másfelől viszont az is feltételezhető, hogy a felhasználók egy jelentős csoportja nem is hallott még a Flash sütikről. Végül pedig lehetséges a kétféle sütit egyszerre „üzemeltetni": a Flash sütiből egy speciális Flash animációval újrateremthető egy normál http süti, ha utóbbit a felhasználó esetleg kitörölné [6].

Elmondhatjuk tehát, hogy a profilírozók erős ellenfelei a privátszférának, és jogosan érezhetjük úgy, hogy bizonyos szituációkban védekezni kell ellenük. Jelen munkánkban azon profilírozási technikák elleni védekezési lehetőségeket vizsgálunk, melyek valamilyen publikusan elérhető információhoz kapcsolódnak.

2. Támadás a privátszféra ellen publikus információforrások alapján

Az információk gyűjtésében, felhasználásában és megosztásában számos szereplő vesz részt – ezt elemezzük a következő fejezetben.

2.1. Szereplők bemutatása, támadó modell

A támadó modell felépítéséhez kiindulási alapként az anonim böngészőknél kialakított szerepeket tekintettük [2][14], és ebből kiindulva határoztuk meg a publikus források alapján történő profilírozáshoz tartozó szerepköröket. Ezek sajátossága, hogy a kapcsolódó adatgyűjtés további bennfoglalt szerepkörök kialakulását hozza magával. A szerepkörök kialakításánál az elsődleges szempontot az információkhoz való hozzáférés (és a hozzáférés kezelése) jelentette.

A szereplőket az alábbiak kategóriákba sorolhatjuk:

- Adatalanyok: azok a felhasználók, akik adatokat tesznek magukról közzé.
- Információszolgáltatók: olyan szolgáltatások, amelyek az adatalanyokról információkat publikálnak, és ezekhez mások is hozzáférhetnek – akár regisztráció nélkül is.
- Publikus források felhasználói: olyan szereplők, akik felhasználják az információkat szolgáltató szereplők által készített anyagokat, vagy a tevékenységük részben erre is épülhet.
- Korlátozási tevékenységet végzők: azok a szervek, szolgáltatók, amelyek a publikus információkat azok, vagy más tartalmak, médiumok blokkolására használnák fel.

[2] A szaknyelvben Local Shared Object, LSO.

- Döntési, választási lehetőséget befolyásolók: bizonyos szereplők a tartalomhoz való hozzáférést ugyan nem korlátoznák, de a profilt a döntési, választási lehetőségek befolyásolására használhatják fel.

A következő fejezetekben a kategóriákba sorolt szerepköröket kifejtve tárgyaljuk.

2.1.1. Felhasználó

A felhasználó első számú célja, hogy a saját adatai felett teljes hatáskörrel rendelkezzen, azaz, hogy képes legyen – legalább korlátozott mértékben – a már publikált adatok visszavonására, illetve a publikálás előtt magas granularitással, áttekinthető és könnyen kezelhető módon meg tudja határozni, hogy kik férhessenek hozzá a publikálandó információhoz. Hasonlóan elvárás a részéről, hogy az egyes adatok hozzáférési körét a későbbiekben is módosíthassa, azaz ha például egy csoport kapott jogosultságot egy blogbejegyzés elolvasására, akkor akár egy-egy személy hozzáférési jogát is vissza lehessen vonni a csoporton belülről.

A megfelelő szabályozás kialakításához meg kell állapítani, hogy ki az adat tulajdonosa, és milyen jogkörrel bír az adat felett. Ez nem minden esetben egyértelmű, és főleg a metaadatok területén vannak még tisztázatlan területek, valamint hiányzik a kialakult gyakorlat is. Például, ha egy közösségi oldalra feltöltött képen megjelölnek valakit, aki még nincs regisztrálva az oldalra, akkor hogyan tudja igazolni, hogy valóban őt jelölték meg a képen, hogy el tudja távolítani a bejelölést? Ez az azonosítás általában nem csak technikai szinten, de más módon sem kidolgozott, így nem kivitelezhető.

2.1.2. Keresőszolgáltatók

A keresőszolgáltatók központi szerepet töltenek be a web szerepében: ezek az információkhoz való hozzáférés „ugródeszkái” és elsődleges hírforrásként is szolgálhatnak. A keresőszolgáltatók robotjai folyamatosan gyűjtik az egyes weboldalakon megjelenő információkat, és ezeket a lehető leghamarabb elérhetővé teszik a keresőmotorjukban. A legújabb trendek szerint az egyik következő cél, hogy az információk elérhetősége valós idejű legyen [20] – azaz, amikor valami felkerül a webre, az azonnal meg is jelenjen a keresési találatok között. Ez mindenképp hatással van a privátszférára, hiszen kizárja a visszavonás lehetőségét.

2.1.3. Hirdetők

A hirdetők szerepe, célkitűzései hagyományosnak mondhatóak. Céljuk a magasabb bevétel elérése, amit célzott, személyre szabott hirdetésekkel érhetnek el a leghatékonyabban. Ez utóbbihoz pedig profilokra, információs oldalakról összegyűjtött adatokra is szükség van,

nem csupán a hagyományos nyomkövetés által készített profilokra. Profilokat több forrásból is gyűjthetnek, például keresők alapján, profiladatbázis-kereskedelemmel, vagy akár a history stealing módszer segítségével is beazonosíthatják a felhasználó ízlésprofilját (a history stealing technikát részletesen a 2.2. fejezetben tárgyaljuk).

2.1.4. Webes boltok

A webes boltokat a vizsgálat szempontjából két fő csoportra bonthatjuk. Vannak a centralizált web boltok, amelyekben egy eladótól vásárol több felhasználó, illetve a felhasználók közötti kereskedelmet engedélyező weboldalak (aukciós oldalak, apróhirdetést támogató rendszerek, virtuális piacok és kiskereskedelmi oldalak). A kettő közötti lényeges különbség, hogy ez utóbbinál a jó hírnév ápolása érdekében minél több mindent érdemes megosztani, így a felhasználói profilok, tevékenységek és valamennyi előzmény általában publikus – így azonban több információt lehet gyűjteni. A centralizált webes boltok esetén pedig – a hirdetőkhöz hasonlóan – hasznos lehet a profilok építése és alkalmazása a célzott hirdetések, dinamikus árlisták előállításához.

2.1.5. Profilírozók

Míg a nyomkövetéses támadások esetén a profilírozók alapvető technikája a web poloska [10] használatán alapuló vagy hasonló jellegű megfigyelés (pl. audit szolgáltatók, hirdetésközvetítő harmadik felek), a publikus források felhasználásával új lehetőségek nyílnak meg. A history stealing technika lehetőséget ad egyrészt a felhasználó közösségi profil alapú azonosítására, másrészt – az előzményekhez való hozzáférés által – a részletesebb ízlés- és érdeklődésprofil felállítására. Az előzmények alapján lehetséges a felhasználót egyedi módon azonosítani (a látogatott weboldalak egyedi „mintázata” szerint), így ha a hagyományos követésre használt azonosítók elvesznének, vissza lehet őket állítani. Ehhez csak elég az előzmények egyedi mintázatát leíró információkat és az aktuális azonosító párt eltárolni, és a visszaállítás később automatikusan megvalósítható.

2.1.6. Információ megosztó oldalak, információforrások

A csoportba tartozó szolgáltatók közös jellemzője, hogy a felhasználói közreműködésen, önkéntes adatszolgáltatáson alapulnak. Mindegyik ilyen weboldal központi funkciója, hogy a felhasználók töltik fel a tartalmat, információkat osztanak meg, amelyek ugyan nem szükségszerűen csak hozzájuk kapcsolódnak, de az ezek által generált metainformációk végül szintén megosztásra kerülnek. Ide sorolhatjuk tehát az alábbi szolgáltatásokat:

- Közösségi oldalak
- Profilépítésen alapuló szolgáltatások (linkmegosztás, ajánló szolgáltatások, tartalomszortírozó alkalmazások)

- Publikálási, tartalommegosztó felületek (kép-, zene-, videó megosztás, blogok, mikroblogok, dokumentum- és fájlmegosztás, fórumok)
- Felhasználói közreműködésen alapuló szolgáltatások (ismerettárak, mint pl. Wikipedia, Google keresési találatok befolyásolása, online térképek)
- Keresőből elérhető egyéb források (pl. intézményi, vagy egyéni honlapok)

Ezek közül a későbbiekben első sorban a szöveges, nem metainformáció jellegű tartalmak megosztására alkalmas oldalakkal foglalkozunk.

2.1.7. Cenzúrázó szervek

A cenzúrázó szervek célja az adott elvek mentén kiválasztott személyekről, csoportokról, szervezetekről, témákról való adatgyűjtés, hogy ezeket az információkat bizonyos tevékenységek, információk vagy médiumok blokkolására használják fel. Ide sorolhatjuk az erre szakosodott állami szerveket is.

2.1.8. Kis Testvérek

Bizonyos esetekben a lokális szolgáltatók (pl. internetkávézó, könyvtár, munkahely) is hasonló cenzúrázó funkciót tölthetnek be, mint az állami szervek, vagy más céllal figyelik meg bizonyos csoportok, felhasználók tevékenységét. Például egy munkáltató utánanézhet a közösségi oldalakon egy állásinterjúra jelentkező személy előéletének, vagy pedig az állítólagos betegszabadságon lévő munkavállaló után is önálló „nyomozásba” kezdhet.

2.2. Közösségi profilok támadása history stealing módszerrel

A history stealing (magyarul előzménylopásként lehetne fordítani) módszer már évek óta ismert támadási forma. A támadás lényege, hogy bár a böngésző által tárolt böngészési előzménylistát a meglátogatott weboldal nem képes lekérdezni, megpróbál kitalálni bizonyos elemeket belőle, tulajdonképpen azáltal, hogy igen-nem jellegű válaszadásra készteti a böngészőt az egyes elemeket illetően. Erre egy lehetőség, hogy a weboldal rejtve új linkeket szúr be a weboldalba, és megnézi, hogy az adott link milyen színnel jelenik meg – így kiderülhet, hogy a felhasználó már járt-e az adott webcímen vagy sem[3].

Ez a technika több célból is használható: a látogató hosszabb távú profilírozására, vagy akár közvetlenül dinamikus árazásra, célzott hirdetések megjelenítésére. Vegyünk egy egyszerű példát: egy kertészeti termékekkel foglalkozó cég webes áruháza ezzel a módszerrel detektálja, hogy a látogató már több konkurens cég honlapján is járt, sőt, azt is

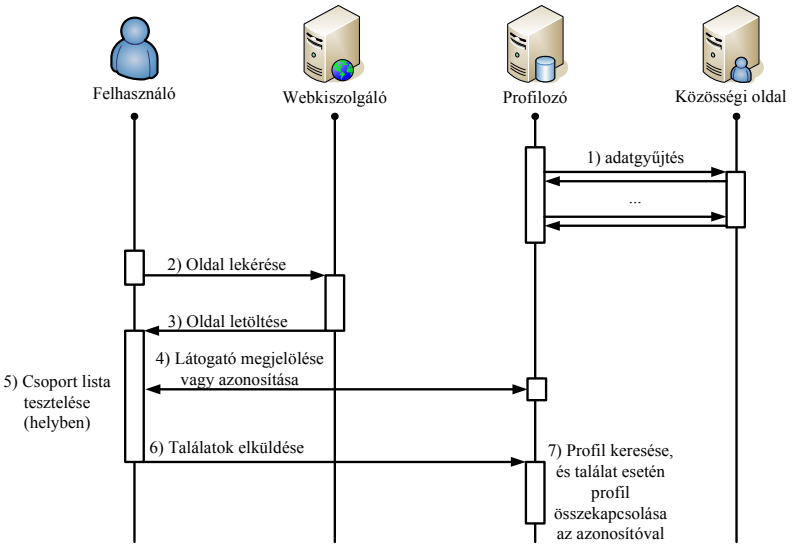
[3] Ez a technikát használja a http://whattheinternetknowsaboutyou.com címen elérhető próba program is.

meghatározza, hogy mely termékeket nézte ott meg (feltéve, hogy minden termék és termékcsoport URL címe egyedi). Ez alapján pedig a látogatót fokozottan érdeklő termékeket olcsóbban kínálhatja, vagy a hirdetések között felhívhatja ezekre a látogató figyelmét.

Természetesen ez a módszer alkalmas arra is, hogy a felhasználót egyedi azonosítóval lássák el, például az alapján, hogy milyen ismertebb Wikipedia cikkeket nézett meg, és milyen nyelven (hiszen minden cikknek egyedi URL címe van). Azonban a legújabb kutatási eredmények szerint ennél is több rejlik a módszerben [9], és felhasználók közösségi oldalon elhelyezett profiljának beazonosítására is alkalmas. A kutatók szerint ez azért kivitelezhető, mert a felhasználók csoportfeliratkozásai nagy valószínűséggel egyedi mintát adnak ki. Az általuk vizsgált Xing közösségi oldalon[4] 1,8 millió felhasználóból kb. 750 000 felhasználónak volt egyedi csoportfeliratkozása, azaz ebben a halmazban nem volt két olyan felhasználó, akik pontosan ugyanazokba a csoportokba lettek volna feliratkozva[5]. Mivel a csoportok adatlapja egyedi URL-en helyezkedik el, a history stealing módszerrel lehetséges tesztelni, hogy a látogató esetleg mely csoportokat látogatta meg, ez alapján pedig meg lehet becsülni a profilját a közösségi hálózaton. A támadás működési elvét az 1. ábra szemlélteti.

[4] A közösségi oldal címe: http://www.xing.com

[5] Vélhetően az sem okoz problémát, ha vannak átfedések, hiszen a további finomítás egyéb megfigyelések alapján lehetséges lehet.



1. ábra: History stealing módszer alkalmazása közösségi profil azonosítására.

A módszer lépései a következők:

1. A profilírozó automatizált módon begyűjti egy közösségi oldal publikus profiljait és a kapcsolódó csoporttagságokat. Ismerve az utóbbi információk eloszlását, eldönti, hogy mely csoporttagságokat érdemes tesztelni, hogy relatíve kevés lekérdezéssel, nagy valószínűséggel egyedileg azonosítható legyen a látogató (ehhez a csoportok lapjainak URL-jeit is ismerni kell).
2. A látogató ellátogat a profilírozóval együttműködő weboldalra.
3. A weboldal betöltődik, amelybe egy (pl. JavaScript nyelvű) tesztelő program is be van építve.
4. A profilírozó egy hagyományos módszerrel megjelöli a látogatót (például egy web poloskán keresztül elhelyez egy sütit a számítógépén, egy egyedi azonosító számmal). A később beazonosított profilt ehhez fogja kapcsolni.
5. A támadó programja helyben teszteli a kiválasztott csoportok URL-jeit.
6. A tesztelt URL listát visszaküldi a program a profilírozónak, aki a háttérben ellenőrzi a mintát az adatbázisában.
7. Ha az ellenőrzés sikeres, akkor a korábban megadott azonosítót össze lehet kapcsolni a közösségi profillal.

Ez a gyakorlatban azt jelenti, hogy amikor a látogató meglátogat egy másik weboldalt, amely szintén együttműködik a profilírozóval, akkor az képes lesz a látogatót akár név szerint azonosítani szinte valós időben, hiszen ez után további számításokra nincs szükség.

Ezek alapján kijelenthetjük, hogy ez a módszer egyesíti a hagyományos nyomkövetéses támadásokat és a publikus források alapján történő profilírozást, és így egy új, összevont kategóriát hoz létre. (Meg kell jegyeznünk, hogy a támadás kivitelezése ezen túlmenően magában hordoz nem triviális kihívásokat is, mint például a hibatűrés megvalósítása, vagy a megfelelő csoportlista kiválasztása.) Arvind Narayanan továbbfejlesztette elméletben ezt a támadási módszert [11], és az alábbi három közösségi platformra általánosította.

Közösségi oldalak feliratkozási lehetőséggel. Bármely olyan közösségi oldalra általánosítható a támadás, amelyen vannak feliratkozást kínáló helyek, mint például a Twitteren a követés funkció, YouTube oldalaknál a feliratkozás, Facebooknál pedig a csoportok. Ha léteznek nagy létszámú ún. hubok (gyűjtőhelyek), akkor ezekkel a fentiekhez hasonlóan kivitelezhető a támadás a közösségi oldalon lévő regisztráció azonosítása céljából.

Megosztásra szakosodott közösségi oldalak. Ez esetben az az alapfeltételezés, hogy a megosztott tartalom URL-je megtalálható a böngésző előzményében, illetve jellemző funkció, hogy az egyes felhasználók megosztásai visszamenőleg is lekérdezhetőek (pl. RSS csatorna segítségével). Ezek begyűjtésével tesztelhető az egyes profilok közötti egyezés, bár kissé nehezebb a támadó dolga, mint az előző esetben, mivel nehezebb jó URL listát készíteni a teszteléshez. Itt a cél a megosztó oldalon lévő profil azonosítása.

Központi azonosítást alkalmazó szolgáltatások. Egyes szolgáltatások központi azonosítást alkalmaznak, és ennek köszönhetően az azonosítók több helyen is megjelenhetnek, például amikor a felhasználó hozzászól egy blogbejegyzéshez. Így egy blog bejegyzéseit végigjárva egy felhasználólistához lehet jutni, és mivel általában a hozzászólás beküldéséhez egyedi URL-eket kell meglátogtni, a history stealing módszerrel kideríthető, hogy mely bejegyzéseknél szólt hozzá a látogató – amiből pedig jó eséllyel leszűkíthető a potenciális azonosítók köre.

Korábban utaltunk arra a lehetőségre, hogy két támadási mód összevonható a history stealing módszerrel. Azonban, ha ezt egy információs szuperhatalom végzi, akkor mindhárom támadási módszer egyesített változatával kell szembenéznünk, és a profilírozási lehetőségek újabb lehetőségei nyílnak meg.

Nem mindig lesz lehetőség a history stealing használatára, ugyanis a Mozilla 2010. tavaszán bejelentette – feltehetőleg a history stealing hirtelen megnőtt népszerűsége miatt –, hogy a Firefox böngészőből el fogja távolítani a támadásra lehetőséget nyújtó funkciókat[6].

[6] A Mozilla hivatalos bejelentése: http://blog.mozilla.com/security/2010/03/31/plugging-the-css-history-leak/

Ez nem fogja a history stealing végét jelenteni közvetlenül, csupán akkor, ha a további főbb böngészők is hasonlóan járnak el; addig a támadás továbbra is kivitelezhető marad.

3. Védekezési lehetőségek

Ahogy megmutattuk, a tartalomelemzésen, adatbányászaton alapuló profilépítési technikák új, az eddigiektől eltérő, ugyanakkor hatásosan alkalmazható támadási formák. Ráadásul a követésen alapuló technikákkal szemben itt nem elegendő az anonimitás pillanatnyi biztosítása, hiszen az elhelyezett tartalmak a későbbiekben is felhasználhatóak. Nem szükséges tehát a felhasználó folyamatos megfigyelése, a készletező adatgyűjtés.

Mivel a jelenlegi megközelítésben a szolgáltatók is megbízhatatlan támadó félként kezelendők, ezért az adatai feletti kontrollt megtartani kívánó felhasználónak egy olyan megoldást kell keresnie, mely még kliens oldalon biztosítja adatai védelmét, így garantálva, hogy a felhasználó adatait már a szolgáltató sem tudja értelmezni. A továbbiakban az ezekkel szemben támasztható követelményeket vizsgáljuk.

3.1. Követelmények

Ebben a fejezetben ismertetjük az igényeket egy olyan szoftverrel szemben, melyre a felhasználók rá szeretnék bízni személyes adataikat. Ezen követelmények alapvetően két nagy kategóriába sorolhatóak. Elsőként és legfőbb szempontként a biztonsággal kapcsolatos követelményeket vizsgáljuk meg, majd a felhasználói élménnyel kapcsolatosan szóba kerülő használhatósági, kényelmi problémákra koncentrálunk, különös tekintettel arra, hogy milyen kompromisszumokra kényszerítjük a felhasználót akkor, amikor az alkalmazás használata mellett dönt.

3.1.1. Biztonságosság

Kiemelkedő fontosságú, hogy a felhasználó adatait megfelelő erősségű algoritmusok védjék, így garantálva azt, hogy az adatokhoz csak az arra jogosultak, a kulcs birtokosai férhessenek hozzá. Általánosságban elmondható, hogy azon algoritmusokat szeretnénk a szoftverekben látni, melyek biztonságossága nemzetközileg is elismert és elfogadott.

3.1.2. Felfedezhetőség

Sok online szolgáltatás esetén a regisztrációkor elfogadott szerződési feltételek között szerepel, hogy a felhasználónak valós adatait kell közölnie, valamint sok egyéb megkötést is tesznek a feltöltött adatokra nézve. Egy olyan szoftver alkalmazása, mely az adatokat pusztán a titkosított formájában tölti fel az oldalra, nyilvánvalóan egy olyan helyzetet

eredményez, melyben a felhasználó megsérti ezeket a feltételeket, ami könnyen a felhasználó profiljának törléséhez, ott tárolt szolgáltatási hozzáféréseinek, adatainak elvesztéséhez vezethet, így fontos szempont lehet, hogy a program olyan kimenetet generáljon, melyet a szolgáltató valós adatnak vélhet, így nem fenyegeti semmilyen veszély a felhasználó regisztrációját.

3.1.3. Fokozatos alkalmazhatóság

A szolgáltatásban, melyben az adatainkat védeni akarjuk, egészen biztosan lesznek olyan felhasználók, akik nem szeretnének semmilyen megoldást igénybe venni, lesznek ugyanakkor olyanok is, akik különböző megoldásokat akarnak majd használni. Fontos, hogy a szolgáltatás akkor is használható maradjon, ha felhasználói nem ugyanazon védelmi alkalmazás mellett döntöttek. Természetesen az egyes felhasználói körök közötti kompatibilitás nem garantálható, de nagyon fontos, hogy a szoftver működését ne befolyásolja az, ha más felhasználók semmilyen vagy eltérő védelemmel rendelkeznek.

3.1.4. Önállóság

Nyilvánvaló, hogy amennyiben az alkalmazás olyan erőforrásokat is igénybe vesz, melyek felett nem rendelkezik kontrollal, az komoly problémákhoz vezethet. A harmadik fél által üzemeltetett szerverek megbízhatósága nem garantálható, az ellenőrizetlen külső szoftverek felhasználása pedig kompatibilitási problémákhoz vezethet, hiszen azok frissülhetnek, és így inkompatibilissé válhatnak.

Alapvető követelmény tehát, hogy az adataink védelmét ellátó program önálló legyen, ne támaszkodjon semmilyen, általa nem felügyelt erőforrásra, mert azok az egész folyamat kritikus pontjai lehetnek, hiszen megbízhatóságuk, hatékonyságuk, frissességük nem kézben tartható.

3.1.5. Univerzalitás

Ahogy erre a későbbiekben is látunk majd példát, rengeteg olyan szoftverrel lehet találkozni, melyek csak egy-egy meghatározott szolgáltatással hajlandóak együttműködni. Egy alkalmazás univerzalitása alatt tehát azt értjük, hogy hány szolgáltatással működik együtt és milyen mértékben. A cél egyértelműen az, hogy olyan programok készüljenek el, melyek az internet egészén biztosítják a felhasználó adatainak védettségét.

3.1.6. Kompromisszumok

A felhasználók számára az elsődleges fontosságú szempontok között található a program kompromisszummentessége. Természetesen egyes funkciók kiesése nem elkerülhető, ha a cél az, hogy a szolgáltató ne tudja elérni az adatokat. Egy alkalmazástól sem várhatjuk el, hogy úgy titkosítsa a nevünket, hogy a szolgáltató ne férjen hozzá, ugyanakkor a szerveroldalon futó keresésekkor a nevünkre keresve találatként megjelenjünk, hiszen ez a kettő nyilvánvalóan nem kivitelezhető egyszerre. Fontos tehát, hogy minden program esetében tisztában legyünk avval, hogy mely szolgáltatások mely funkcióiról kell lemondanunk akkor, ha adataink védelmére használjuk azt.

3.1.7. Kényelem

Nem szabad a programok tervezőinek megfeledkezniük a felhasználó kényelméről sem. Az a program, melynek használata a felhasználó számára szükségtelen kényelmetlenségekkel jár, alkalmatlan arra, hogy elássa feladatát, hiszen az általános felhasználói hozzáállás a mai világban a kényelmet könnyedén a biztonság elé helyezi – tehát, ha a program nehezen használható, akkor inkább nem fogják használni. Következésképpen a felhasználó számára egy egyszerűen és gyorsan használható felületet kell biztosítani, egyszerű installálhatóság mellett, különben a program használatát a felhasználó mellőzni fogja.

3.2. Az ideális megoldás

Mielőtt a létező megoldásokat vizsgáljuk, vizsgáljuk meg, hogy milyen lehet az ideális kliensoldali adatvédelmi program. Nyilvánvaló, hogy ideálisnak azt az alkalmazást tekinthetjük, amely a fenti követelményeket a lehető legnagyobb mértékben kielégíti.

Egy ilyen program üzembe helyezése nem igényel konfigurációt, használata pedig kényelmes és gyors, működése közben a felhasználótól csak a minimálisan szükséges mennyiségű interakciót követeli meg, azaz a titkosítandó tartalom és a jogosultak körének, így implicit módon a titkosításhoz használt kulcsnak, vagy kulcsoknak, kijelölésén kívül semmi egyebet. A weboldalakban található titkosított blokkok feloldása automatikusan történik, amennyiben rendelkezésre áll a szükséges kulcs.

A program biztonságosságának ellenőrizhetőnek kell lennie a felhasználó által. Ehhez szükséges, hogy a program teljes egészében nyílt forráskódú legyen, valamint tilos olyan erőforrásokat használnia, melyek megbízhatósága és változásai a felhasználó által nem ellenőrizhetőek, mint például harmadik felek által üzemeltetett kiszolgálót.

A programnak az adatok védelmére bizonyítottan erős, megbízható algoritmusokat kell használnia. Lehetőség szerint biztosítania kell a titkosítás tényének elfedését a nem megbízható felek elől, tehát célszerűen szteganográfiát is alkalmaznia kell. Biztosítania kell a lehetőséget arra. hogy egy adott tartalomhoz a hozzáférés jogát egyes személyektől meg lehessen vonni a publikálás után is, valamint engedélyezni lehessen azt újabb jogosultak számára.

A programnak meg kell felelnie a „Fokozatos alkalmazhatóság" feltételének, valamint maximálisan univerzálisnak, szolgáltatásfüggetlennek és kompromisszumoktól mentesnek kell lennie.

3.3. Létező megoldások

A következőkben néhány, a felhasználók számára is elérhető szoftver ismertetésére kerítünk sort, megvizsgáljuk őket, hogy milyen mértékben felelnek meg a felállított követelményeknek, milyen előnyökkel rendelkeznek, és milyen problémákra kell felkészülnünk, ha az adott szoftvert választjuk. Ezeken felül létezik két olyan megoldás is, melyekből nem találtunk kipróbálható verziót, ezek a Lockr [4] és a NOYB-Social Networking [1].

3.3.1. FaceCloak [5]

Az alkalmazás elsődleges célja a felhasználó Facebook-profiljának védelme, kizárólag ezen szolgáltatással kompatibilis, és az itt található szöveges mezők titkosítását végzi el helyettünk. A felhasználó dolga mindössze annyi, hogy a titkosítandó tartalom elé a '@@' előtagot odaillessze. Ekkor az oldal mentésekor a program automatikusan titkosítja azt, majd a titkosított adatot egy külső szerveren eltárolja, az oldalon pedig egy valódinak látszó, de igazából hamis adattal helyettesíti azt.

A későbbiekben azok, akik az illető profilját meglátogatják, és rendelkeznek a megfelelő kulccsal, már automatikusan a valós adatokat fogják látni, mivel a program elvégzi az adatok behelyettesítését. Az alkalmazás nagy előnye a nagyfokú automatizáltsága, valamint az, hogy valósnak tűnő, ugyanakkor hamis adatokkal elfedi a titkosítás tényét. Azonban mindenképpen hátrányként említhető meg, hogy kizárólag a Facebookkal képes együttműködni, valamint, hogy adatainkat egy harmadik fél által üzemeltetett szerverre kell bíznunk, ha el akarjuk rejteni azokat. Sajnálatos módon a jelenleg (2010. március) a szoftver honlapján [18] található verzió nem futott a Firefox legfrissebb stabil verziójával (3.6.2), így nem volt alkalmunk élesben kipróbálni az alkalmazást.

3.3.2. NOYB – Secret Messaging [16]

A NOYB – Secret Messaging egy olyan alkalmazás, melynek elsődleges célja a nagyfokú univerzalitás megvalósítása volt. A program egy Firefox kiegészítésként van megvalósítva, a honlapjáról történő letöltés után pár kattintással telepíthető, majd a beállítások között létrehozhatunk hozzáférési csoportokat, csoportonként egy-egy jelszóval. A jelszavak beállítása után, ha egy tartalmat titkosítani akarunk, akkor elég kijelölnünk azt, majd a környezeti menüből a „NOYB Encrypt" opcióval és a csoport megjelölésével titkosíthatjuk azt. A program ekkor automatikusan kicseréli az eredeti szöveget a titkosítottal.

Ha egy oldalon olyan blokkot találunk, amely NOYB-bal van titkosítva, akkor, miután kijelöltünk azt, a környezeti menüből a „NOYB Decrypt" opciót választva a program visszafejti azt, amennyiben rendelkezik a megfelelő kulccsal. Hatalmas előnye a programnak, hogy nem függ semmilyen szolgáltatástól, és a szöveges bemenetek titkosításával olyan eszközt ad a felhasználó kezébe, mellyel az bármilyen környezetben megvédheti tartalmait. Ennek a rugalmasságnak ugyanakkor az automatizáltság látta kárát, hiszen az egyes blokkok visszafejtését kézzel kell indítani.

3.3.3. FlyByNight [3]

Ez egy olyan Facebook alkalmazás, mellyel a FaceCloakhoz hasonlóan facebookos adatainkat titkosíthatjuk. Ha bejelentkezünk Facebook-fiókunkba, és ellátogatunk a program oldalára [17], akkor onnan telepíthetjük az alkalmazást. A megoldás nagy előnye, hogy böngészőfüggetlen, hiszen egy, az oldalhoz tartozó szolgáltatásként lehet használni.

A program szimmetrikus és privát kulcsos rejtjelezőket is használ a különböző funkciók megvalósítására. A titkosított üzenetek személyre szólóan tárolódnak az alkalmazás saját szerverén, és a program használója onnan kérheti le a neki szóló üzeneteket.

Jelenleg (2010. március) a szoftver nem volt elindítható a Facebook alkalmazások között. (Telepítéskor egy hibaüzenetet helyezett el a képernyőn, és nem sikerült működésre bírni.) Ezen felül azonban meg kell említeni egy sokkal súlyosabb problémát, amely a Facebook alkalmazások működéséből fakad. Amikor a felhasználó betölt egy alkalmazást, akkor az a Facebook szerverein keresztül érkezik el hozzá, így a szolgáltató képes lehet módosítani azt. Így amellett, hogy adatainkat egy harmadik fél szerverén kell tárolnunk, még a kód megbízhatósága sem garantálható! Ezen tények ismeretében a program használatát nem ajánljuk.

3.3.4. FireGPG [7]

A FireGPG egy olyan, igen nagy tudású Firefox kiterjesztés, mely széleskörű PGP funkcionalitást ad a felhasználó kezébe. A felhasználónak lehetősége van privát kulcsú algoritmusok használatával üzeneteit aláírni, másoknak írt üzeneteket titkosítani, de van lehetőség szimmetrikus kulcsú rejtjelezésre is. A program bármilyen szöveges felülettel képes együttműködni, sőt beépítetten támogatja GMail webes felületét is, könnyebb, kényelmesebb használatot biztosítva.

A program telepítése és üzembe helyezése sok odafigyelést és lényegesen több szakértelmet igényel, mint az előzőeké. Elsőként a GNU Privacy Guard [15] letöltését és telepítését kell elvégeznie a felhasználónak, majd letöltheti és telepítheti a FireGPG-t. A FireGPG működése a továbbiakban is függ a GNUPG jelenlététől.

Bár alapvetően nem az általunk megkövetelt funkcionalitásra, az online tárolt adatok védelmére tervezték, hanem az adatok átvitel során történő hitelesítésére és védelmére, gazdag funkcionalitása alkalmassá teheti az alapvető feladatok ellátására ebben a problémakörben is. Komoly hátrányaként a GNUPG-től való függése és a komplikált konfigurációja róható fel, ezek azonban szükségesek ahhoz, hogy a program sokrétűen használható és testre szabható legyen.

3.3.5. Blogcrypt [8]

Ez az általunk fejlesztett alkalmazás szöveges adatok titkosítására képes, akárcsak az előzőek. A titkosítás menete igen hasonló a NOYB-nál látottakhoz, a titkosítandó szöveg kijelölése után, a program menüjének „Encrypt” funkciójával indítható el a titkosítás. Ezt követően ki kell választanunk az alkalmazandó kulcsot, valamint a program lehetőséget ad a beépített AES (Advanced Encryption Standard) titkosító üzemmódjának megválasztására is.

Az eddigi megoldásoktól eltérően a Blogcrypt nagy hangsúlyt fektet a kulcsok egyértelmű használhatóságára. A NOYB-bal ellentétben itt minden kulcshoz megadhatunk egy domaint, ekkor a kulcs kizárólag az adott tartományban lesz érvényes, így kerülve el a különböző oldalakon található, de azonos azonosítóval rendelkező kulcsok ütközését. Természetesen lehetőség van globálisan érvényes kulcsok használatára is, amennyiben a domain mezőbe a „global” értéket írjuk, akkor az a kulcs domainfüggetlen lesz, és a NOYB-bal azonos működést kapunk.

Az oldalak betöltődésekor a program automatikusan megkeresi az oldalban található titkosított blokkokat és a megfelelő jelszó birtokában fel is oldja azokat, így ha a

felhasználó rendelkezik a megfelelő kulcsokkal, akkor böngészés közben észre sem veszi, hogy olyan tartalmakat néz, melyek a szerveren titkosítva voltak.

3.3.6. A megoldások értékelése

A fentiekben ismertetett programok összehasonlítására egy táblázatot készítettünk (1. táblázat), melyben feltüntetjük, hogy az egyes megoldások milyen mértékben felelnek meg az egyes követelményeknek.

1. táblázat

Értékelési szempont	Kategóriák	Lockr [11]	FaceCloak [12]	NOYB [1]	FlyByNight [10]	FireGPG [15]	Blogcrypt	Ideális
Fokozatos telepíthetőség		X	X	X	X	X	X	X
Autonómia	Teljesen autonóm			X			X	X
	Harmadik fél által üzemeltetett kiszolgálót igényel	X	X		X			
	Harmadik fél által fejlesztett szoftvert igényel	X				X		
	Szolgáltatói együttműködést igényel	X			X			
Univerzalitás	Szolgáltatásspecifikus	X	X		X			
	Általános interfészt használó			X		X	X	
	Univerzális interfészt használó							X
Kompro-misszumok	A szolgáltatás funkciói 100%-ban használhatóak maradnak	X						X
	A szolgáltatás funkcióinak többsége használható marad		X	X	X	X	X	
	A szolgáltatás lényegében használhatatlanná válik							
Komfort	A program használata és telepítése kényelmes, egyszerű		X	X	X		X	X
	Használata, telepítése hozzáértést és szakértelmet igényel	X				X		
Kriptográfia	Szimmetrikus rejtjelező	X	X		X	X	X	X
	Aszimmetrikus rejtjelező				X	X		X
	Saját algoritmus			X				
Felfedez-hetőség	Felfedezhető	X		X	X	X	X	
	Rejtést alkalmaz		X					X

3.4. A védekezési lehetőségek értékelése

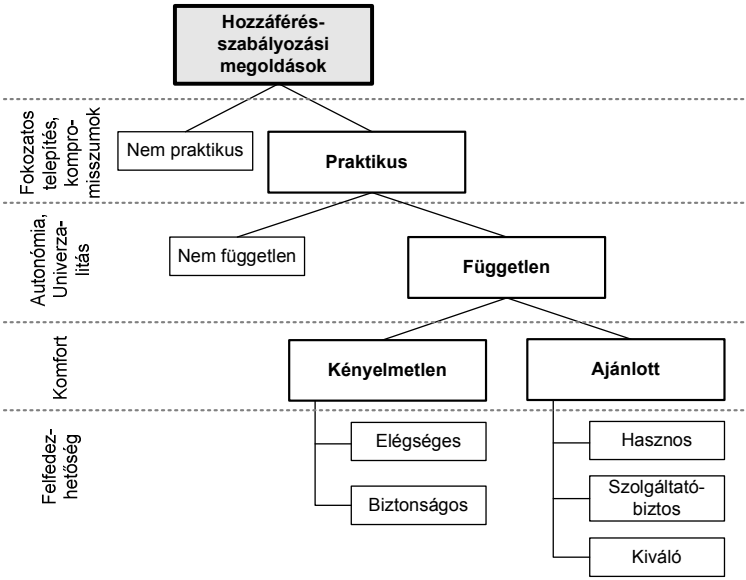
Láthatjuk, hogy sajnálatos módon a felhasználók lehetőségei igen korlátozottak, ha olyan szoftvert keres, mely még a kliensben képes megvalósítani adatai védelmét. Univerzális szoftverek terén lényegileg a NOYB, a FireGPG és a Blogcrypt közül lehetséges választani, ezek közül elsősorban céljainknak megfelelően kell meghoznunk a döntést. Amennyiben célzottan szeretnék egy-egy személynek üzeneteket küldeni, vagy aláírni, akkor a FireGPG bizonyulhat jó választásnak, hiszen a PGP-ben szereplő privát kulcsos technológiák pontosan ilyen jellegű alkalmazásra születtek.

Az egy-a-több jellegű üzenetküldés (például blogolás vagy közösségi hálózatok) esetén a NOYB vagy a Blogcrypt lehet a jó választás, hiszen ezek szimmetrikus kulcsú titkosításukkal, egyszerűségükkel pontosan erre a célra készültek. Utóbbi nagyobb automatizáltságával ráadásul igen kényelmes használatot tehet lehetővé.
Szolgáltatásspecifikus megoldások terén azonban a jelenlegi paletta igen szűkös. Egyedül Facebook alá érhető el olyan alkalmazás, amely a szolgáltatással nagyfokú integráltságot mutat, de ezekből nem sikerült üzemképes példányt találnunk. Végül mindenképpen szükséges megemlíteni azt a tényt, hogy az alkalmazások majdnem mindegyike Firefox kiegészítésként került megvalósításra, hiszen ezeket igen egyszerű fejleszteni, így a más böngészőket használók számára ezek a megoldások nem elérhetőek.

4. Osztályozási szempontrendszer, taxonómia

Ebben a fejezetben ismertetjük az értékelési szempontokra alapozott taxonómiánkat (2. ábra). A választott szempontok a fokozatos telepíthetőség, a kompromisszumok, az autonómia, az univerzalitás, a komfortosság, végül a felfedezhetőség voltak. Célunk az volt, hogy tipizáljuk az elérhető hozzáférés-szabályozási megoldásokat, tulajdonságaik alapján hierarchiába szervezve őket. Ennek megfelelően tehát taxonómiánk az összes publikus információkhoz történő hozzáférést szabályozó szoftverből, mint legbővebb kategóriából indul ki, majd a tulajdonságok fenti felsorolásának sorrendjében bontja szűkebb csoportokra, mígnem végül a dekompozíció elérkezik taxonómiánk elemi típusaihoz.

A taxonómia elkészítésekor a vezérgondolat a felhasználó és a szoftver viszonya volt. Először is, a felhasználó valószínűsíthetően nem fogja használni a hozzáférés-szabályozási szoftvert, ha sérül a fokozatos telepíthetőség követelménye, vagy ha túl sok kompromisszummal jár a program használata. Azon szoftvereket, melyekre a két tulajdonság legalább egyike igaz, a *nem praktikus* típusba soroltuk be, míg a többi a *praktikus* csoportba került.



2. ábra. A hozzáférés-szabályozási szoftverek taxonómiája

Másodsorban, az univerzalitás és az autonómia is a szolgáltatófüggetlenség záloga, mely egyértelműen pozitív tulajdonság a felhasználó szemében, hiszen a program „hordozható". Az autonómia foka ezen kívül még azt is megmutatja, hogy a szoftver működését mennyire tarthatja a felhasználó kontroll alatt. Ezen okfejtés alapján döntöttünk úgy, hogy a szolgáltatásspecifikus vagy távoli harmadik fél közreműködését igénylő megoldásokat a *nem független* típusba soroljuk, míg a többi a *független* csoportba kerül.

Dekompozíciónk tercier szempontjának a komfortot választottuk, mivel ez a szempont határozza meg talán legerősebben, hogy egy felhasználó milyen sokáig fog használni egy szoftvert. A komplikált telepítést vagy túl sok felhasználói beavatkozást igénylő szoftvereket ezért a *kényelmetlen* csoportba soroltuk, míg a többi az *ajánlott* csoport tagja lett.

Felbontásunk végső szempontja a felfedezhetőség volt. A dekompozíció során különféleképpen kezeltük a *kényelmetlen* és az *ajánlott* szoftvereket, mert úgy gondoltuk, hogy a felhasználó számára kedvezőbb tulajdonságokkal bíró programok értékelésekor van szükség igazán nagy granularitásra. A *kényelmetlen* csoport alá tartozó két típus a csak kriptográfiát és a valamilyen kifinomultabb megoldást alkalmazó megoldásoké. A típusok

rendre az *elégséges* és a *biztonságos* neveket kapták. Az *ajánlott* megoldásoknál a csak rejtjelezést használó szoftverek a *hasznos*, a hamis adatokat is behelyettesítők a *szolgáltatóbiztos*, míg a szteganográfiát alkalmazók a *kiváló* típusba soroltattak be. A dekompozíció finomodásának oka az, hogy álláspontunk szerint a szteganográfia használata egyfajta ideális biztonságot jelent. A szteganográfiát „drága" (vagyis természetéből eredően kis kapacitású) volta miatt nehéz beépíteni a gyakorlatban használható, több felhasználóra is jól skálázható megoldásokba – mindazonáltal ezeket a tulajdonságokat már a dekompozíció felsőbb szintjei már lefedik, ezzel az aspektussal a legalsó szinten tehát nem kell foglalkoznunk.

A vizsgált szoftverek típusbesorolásait a 2. táblázatban összegeztük.

2. táblázat. A vizsgált szoftverek besorolása

Szoftver	Elemi típus
Ideális	*kiváló*
Lockr	*nem független*
FaceCloak	*nem független*
NOYB – Secret messaging	*hasznos*
FlyByNight	*nem független*
FireGPG	*elégséges*
BlogCrypt	*hasznos*

Látható, hogy egyik vizsgált szoftvert sem tudtunk taxonómiánk *kiváló* típusába besorolni. Viszonylag sok lett továbbá a *nem független* eredmény – a szerzők álláspontja szerint ugyanis a praktikum és a függetlenség olyan fontos tulajdonságok, melyek sérülése esetén a további attribútumok vizsgálata szükségtelen. Azt is le kell azonban szögeznünk, hogy a vizsgált programok felét a független csoport alá soroltuk be. Ezen belül két szoftver kapott *hasznos* minősítést, amely arra utal, hogy mindössze a felfedezhetőség tekintetében lehet szükség a fejlesztésükre.

5. Konklúzió

Jelen munkánkban bemutattuk a személyes információk nyilvánosságra kerülésének veszélyeit, és rámutattunk, hogy szükség van az olyan megoldásokra, melyek a megosztandó adatokhoz való hozzáférést szabályozzák. A jelenleg elérhető eszközök értékelése után egy taxonómiát építettünk fel, melybe be is soroltuk az eddig kidolgozott szoftvereket és koncepciókat.

Az értékelő táblázatot és a taxonómia besorolásait látva levonhatjuk azt a következtetést, hogy léteznek jól használható, sokoldalú eszközök a publikus helyre feltöltött tartalmak

hozzáférés-szabályozásához. Ráadásul némelyikük, mint például a FaceCloak, nem pusztán rejtjelezést használ, hanem hamis adattal is helyettesíti a rejtjelezett információt. Ez a megközelítés, bár pusztán a biztonság szempontjából nézve rendkívül kedvező, óhatatlanul is szolgáltatásspecifikussá teszi az alkalmazást.

A szteganográfia alkalmazása ezt a korlátot természetszerűleg leküzdi. Bármely szolgáltatás, melyre például képeket tölthetünk fel, alkalmas lehet szteganográfiával rejtett információ közzétételére. A Blogcrypt például kiegészíthető lenne egy szteganografikus információrejtő modullal, melyet a felhasználó opcionálisan felhasználhatna kismennyiségű adat elrejtésére.

A szteganográfia biztonság szempontjából azért előnyös, mert a hordozó közegben foglalt információ létezésének tényéről sem értesül a közzétételt lehetővé tevő szolgáltató. Mindazonáltal ez a technológia egy sor más kihívás elé állítja a kutatót. Fontos például eldönteni, hogy milyen közegbe szeretnénk rejteni. A szövegek alkalmazása azért előnyös, mert így megmaradhatunk a szövegdoboz általános, bevetten használt interfészénél, ellenben az is elmondható, hogy egy rövid szöveg aligha hordoz elég szteganografikus kapacitást számottevő mennyiségű tartalom (pl. egy bekezdésnyi szöveg) elrejtésére. Képek használata esetén viszonylag nagy kapacitáshoz jutunk, viszont nem tudunk többé „helyben” rejteni; képek bevitelére nincsen ugyanis olyan általános beviteli felület, mint amilyen szövegek esetén egy szövegmező.

Úgy gondoljuk tehát, hogy kutatásunk a jövőben a szteganográfiára fog fókuszálni. A taxonómia *kiváló* kategóriájának megvalósítása ugyanis egy sok szolgáltatással együtt használható, megbízható harmadik felet nem igénylő, és nem utolsó sorban igencsak biztonságos eszközhöz juttatná a privátszférára érzékeny felhasználókat. Ha ezek mellett a program még egyszerűen használható is marad, akkor igen komoly felhasználói bázisra is számot tarthat.

Irodalomjegyzék

[1] S. Guha, K. Tang, P. Francis: NOYB: privacy in online social networks, Proc. of the first workshop on Online social networks, 2008. augusztus, 49–54. o.

[2] G. Gulyás, R. Schulcz, S. Imre: Comprehensive analysis of web privacy and anonymous web browsers: are next generation services based on collaborative filtering?, Joint SPACE and TIME International Workshops 2008, Trondheim, Norvégia, 2008. június.

[3] M. M. Lucas, N. Borisov: FlyByNight: mitigating the privacy risks of social networking, Proc. of the 7th ACM workshop on Privacy in the electronic society, 2008. október, 1–8. o.

[4] A. Tootoonchian, K. K. Gollu, S. Saroiu, Y. Ganjali, A. Ganjali, A. Wolman: Lockr: social access control for Web 2.0., Proc. of the first workshop on Online social networks, 2008. augusztus, 43–48. o.

[5] W. Luo, Q. Xie, U. Hengartner: FaceCloak: an architecture for user privacy on social networking sites, Proc. 2009 International Conference on Computational Science and Engineering, 2009. augusztus, 26–33. o.

[6] A. Soltani, S. Canty, Q. Mayo, L. Thomas, C. J. Hoofnagle: Flash cookies and privacy, 2009. augusztus.

[7] M. Cuony: FireGPG – Welcome to the official website of FireGPG. 2009. november 8. http://getfiregpg.org/s/home

[8] Paulik T., Gulyás G., Dörflinger Cs.: Blogcrypt. 2010. január 23. http://pet-portal.eu/blogcrypt/download.html

[9] G. Wondracek, T. Holz, E. Kirda, S. Antipolis, C. Kruegel: A Practical Attack to De-Anonymize Social Network Users, IEEE Security and Privacy, 2010. május.

[10] Hullám G.: A web bug technológia – barát vagy ellenség?, Székely Iván – Szabó Máté Dániel (szerk.): Szabad adatok, védett adatok, Alma Mater sorozat, BME GTK ITM, Budapest, 2005 március.

[11] A. Narayanan: Ubercookies Part 2: History Stealing meets the Social Web. 2010. február 19. http://33bits.org/2010/02/19/ubercookies-history-stealing-social-web/

[12] C. Matyszczyk: Facebook entry gets office worker fired | Technically Incorrect. 2009. február 26. http://news.cnet.com/8301-17852_3-10172931-71.html

[13] World News Australia – Woman fired via Facebook after rant. 2009. augusztus 10. http://www.sbs.com.au/news/article/1070187/Woman-fired-via-Facebook-after-rant

[14] Gulyás G.: Anonim-e az anonim böngésző? Technológiák és szolgáltatások elemzése. In: Tanulmányok az információ- és tudásfolyamatokról 10. Alma Mater sorozat, BME GTK ITM, Budapest, 2006. március.

[15] The GNU Privacy Guard – GnuPG.org. 2010. március 9. http://www.gnupg.org/

[16] NOYB: Posting Secret Messages on the Web. 2009. http://adresearch.mpi-sws.org/noyb.html

[17] flyByNight | Facebook. 2009. http://apps.facebook.com/flybynight/

[18] U. Hengartner: FaceCloak. 2010. február 26. http://crysp.uwaterloo.ca/software/facecloak/

[19] R. Pitofsky, S. F. Anthony, M. W. Thompson, O. Swindle, T. B. Leary: Online profiling: a report to Congress, 2000. június.

[20] A. Singhal: Relevance meets the real-time web. 2009. december 7. http://googleblog.blogspot.com/2009/12/relevance-meets-real-time-web.html

[21] A. Narayanan, V. Shmatikov: De-anonymizing Social Networks, Proc. 30th IEEE Symposium on Security and Privacy, 2009. március, 173–187. o.