1 **Adatbiztonság, adatvédelem**

Szteganográfia

2 **Mi is az a Szteganográfia?**

- Görög eredetű szó.
- Jelentése: Leplezni, rejtjelezni
- Krisztus előtt nagyjából 440 környékén már használták.
- Célja: kommunikáció tényének elrejtése egy harmadik fél elől.
- Egyfajta művészet is.

3 **Mi is az a Szteganográfia?**

- A szteganográfia szót Johannes Trithemius német szerzetes használta legelőször 1499-ben írt Steganographia c. könyvében.
- A könyvet csak 1606-ban adták ki. 1609-ben pedig felkerült az egyház tiltott könyveinek listájára.
- A listáról a könyv csak 1900-ban került le.

4 **Mi is az a Szteganográfia?**

- A könyv 3 kötetes
- A tiltás oka az volt, hogy sokáig nem értették, miről szól a könyv, mivel szteganográfiai adatrejtéssel íródott.
- A 3. kötetet nemrég sikerült csak megfejteni.
- Az első két kötettel nem volt gond, mivel a titkosító kulcs ismert volt hozzá.

5 **Sztenográfia**

- Nem azonos a szteganográfiával
- A rövidített írásmód „művészete”
- Főként jegyzetelők alkalmazzák, lényege az, hogy hosszú szavakat cserélünk rövidebb, kódolt szavakra.
- Egyfajta papír alapú tömörítés
- Akár 200 szó leírása is lehetséges percenként
- A szabályokat a jegyzetkészítő találja ki

6 **Sztenográfia**

- Éppen ezért a jegyzet eredeti készítője tökéletesen tudja olvasni a szöveget, viszont más, aki nem ismeri a szabályokat nem, vagy nagyon sok munka befektetésével képes csak megfejteni azokat.
- Gyakran keverik a szteganográfia fogalommal.

7 **Szteganográfia**

- Két fő csoportra bontható
  - Technikai adatrejtés
    - Tudományos módszerek adatrejtésre. Az eredmény analóg és digitális formában is megjeleníthető marad
  - Nyelvészeti adatrejtés
    - Pl.: zsargon kód.

8 **Technikai adatrejtés analóg módon**

- Tetoválás
- Láthatatlan tinták

- Nyomdai megoldások:
  - Betűk közötti térköz módosításával
  - Betűtípusok közötti módosítással

9 **Nyomdai megoldások**

- Leginkább a nyomtatás elterjedésének időszakára tehetőek ezen megoldások
- Akkoriban a legtöbb könyv kevert betűkkel lett nyomtatva, mivel a nyomdászoknak sokszor nem állt rendelkezésre ugyanazon betűből sok, így ki kellett pótolniuk más betűtípusból.
- A betűtípusok variálásával lehet rejteni információt.

10 **Példa**

- *R*óka *e*jtett *J*ánosra *t*etemes *e*lázott *t*ölgy *t*önköket.
- A mondatnak önmagában nem sok értelme van. Két következtetést vonhatunk le:
  - Ez művészet ☺
  - Dőlt betűket összeolvasva azt kapom: Rejtett

11 **Nyomdai megoldások**

- Ugyanez a hatás térköz módosításokkal elérhető
- Ha pedig nem adunk semmi támpontot, és csak szabályokra hagyatkozunk (pl. minden szó első betűje), akkor azt kapjuk, hogy ez nyelvészeti megoldás.

12 **Láthatatlan tinta házilag**

- Üzenet felírása a papírra citromlé segítségével (igazi citromból!)
- Hagyni papírt önmagától megszáradni
- Papírt normál módon használni
- Üzenet „előcsalogatása”: melegíteni kell a papírt, míg láthatóvá nem válik az üzenet. (barna színű lesz)
- Másik megoldás: UV tinta, kereskedelmi forgalomban is kapható.

13 **UV Tinta**

14 **Technikai adatrejtés digitális módon**

- Kép
- Videó
- Hang
- Szöveg és gyakorlatilag bármilyen fájl.

15 **Adatrejtés szövegbe és egyéb fájlokba**

- Mikor fog működni:
  - Akkor, ha a hordozó formátumom egy komplex formátum. Pl: DOC, EXE
  - Alapelv: Eredeti fájl végére írom a titkos információmat. A formátumhoz tartozó natív program úgyis csak az eredeti fájltartalmat olvassa be a formátum felépítése miatt.
  - N+1 alkalmazás az interneten ilyen célra.

16 **Gyakorlati példa**

- Fogjuk a titkosítandó fájljainkat. Becsomagoljuk őket valami tömörítőprogrammal. Pl: ZIP, RAR, 7z, stb…
- Kész tömör fájlt hozzáfűzzük a hordozó formátumunkhoz.
- Erre saját program is használható lenne, de alapból adott Windows rendszeren is az eszköz a célra ☺

17 **Gyakorlati példa**
- Hozzáfűzés:
  - Copy /b hordozo.exe + rejtett.zip kimenet.exe

18 **Gyakorlati példa**
- Kibontás:
  - Egyszerű átnevezéssel. Igen, ennyire egyszerű. ☺
- Pro:
  - Egyszerű, mint a faék.
  - Archívum fájlok titkosíthatóak is, méghozzá egész jól.
- Problem:
  - Digitális aláírt EXE-vel nem működik.
  - Viszonylag egyszerű detektálni némi szakértelemmel

19 **Demo ☺**

20 **Digitális aláírás problémája**
- Azonosítja, hogy a fájl sérült – e, vagy sem.
- Valamilyen Hash algoritmussal azonosít, ezért itt bukik az egész ☹
- Nem aláírt programok esetén minden rendben van. Programok nagy része nem aláírt, mivel drága mulatság.
- Vállalati környezetben adminisztrátor kikényszerítheti az aláírás meglétét AD segítségével

21 **Egy másik exe-be rejtési mód ☺**
- Resource információk módosításával
- EXE/DLL = program kód + resource fájlok
- Resource: például képek, hangok, ikonok, amiket a program használ működése során
- N+1 program ezen információk olvasására és módosítására.
  - Pl: Resource Edit, http://www.resedit.net/
-

22 **Gondok a módszerrel**
- Nem minden EXE/DLL esetén működik szintén, mivel ahogy a fordító programok fejlődnek, az előállított EXE formátuma is fejlődik.
- Ezért az eszköznek ezt követnie kell, mert a kimeneti fájl használhatatlanná válhat.
- Valamint tömörített EXE fájlokkal (UPX és egyéb kereskedelmi) nem működik a módszer. Ezeket először ki kell csomagolni, majd újra csomagolni.

23 **Adatrejtés képekbe**
- Igazi művészet ☺
- Több megközelítés és módszer lehetséges
  - Mintakereséssel
  - Színek módosításával

24 **Minta kereséssel és módosítással**

25 **Gondok**
- Vannak képek, amelyek jobban alkalmasabbak a többieknél.
- Ennél fogva képenként változik a rejthető adat mennyisége
- Komplex analízist igényel, amely időigényes lehet. -> GPU használata segít

26 **Színinformációk módosítása**

- Mielőtt belemennénk, egy kis elmélet (és némi matek ☺) szükséges a számítógépes színábrázolásról.

27 **Színábrázolási módszerek**

- Fekete - fehér – 1 bit/pixel
- Rögzített/adaptív palettás képek – 16 … 256 különböző szín 2,3 byte-tal ábrázolva.
- 16 bit színmélység
  - 5r 6g 5b
  - 5r 5g 5b 1x
- 24 bit színmélység
  - 8r 8g 8b
- 32 bit színmélység
  - 8a 8r 8g 8b
-

28 **Színábrázolási módszerek**

- Legalkalmasabb erre a célra a 24 bit / 32 bit színmélység
- Rögzített / adaptív palettás képek nem igazán alkalmasak a célra, mivel könnyen észrevehető a módosítás.
-
-

29 **A módszer lényege**

- Minden színkomponens utolsó 1 (LSB) bitjét módosítjuk a rejteni kívánt adatunknak megfelelően.
- 1 pixel így 3 bit rejtett információt hordoz.
- Amennyiben 2 bitet módosítunk és mondjuk 32 bites képet használunk, akkor 2 pixel 1 byte-ot tárol.
- Ebben az esetben egy 800x600-as képbe 234 Kb információ rejthető el, szinte észrevehetetlenül.

30 **Amiért nem vesszük észre**

- Emberi szem nem képes 16,7 millió színt megkülönböztetni.
- Ha 2pixel egymás mellett eltérő színű, azt sem tudjuk megkülönböztetni, ha a kép elég nagy.
- Ezen az elven alapul a JPEG tömörítés is.
- JPEG nem a legjobb választás szteganográfiai hordozónak, de megoldható.
- BMP, TIFF túl nagy fájlméret. Legjobb választás: PNG

31 **Példa #1**

- H betű rejtése 1db 32 bites pixelen
- A H betű 8 bites ASCII kódja: 48 hex -> binárisan: 0100 1000
- Hordozó pixel színe:
  - FF 31 72 D4
- 2 legkisebb helyiértékű bit módosítások után komponensenként a színkód:
  - FD 30 72 D4
  -

-

32 **Példa #1**

33 **Példa #1 továbbgondolva**

- Egy 1024x768 pixeles kép 786 432 pixelt tartalmaz.
- 32 bites pixelek esetén, ha 2 bitet rejtünk minden pixelbe, akkor 196 608 byte rejtése lehetséges, ami pont 192Kb.
- Ebbe azért elég sok szöveg belefér .txt formában.
- De ha csak 1 bit/pixel módosítással dolgozunk, akkor is 96 Kb információ belefér.

34 **Példa #2**

1. Eredeti Kép
2. Rejtendő kép

35 **Példa #2**

1. Kimenet
2. Előállítása:
3.
   - Steghide programmal:
     - Steghide embed –cf eredeti.bmp –ef rejtett.png
   - Kibontás:
     - Steghide extract –sf kimenet.bmp
   - http://steghide.sourceforge.net

36 **Hanganyagok digitális tárolása**

- A tömörítetlen formátumoknak két fontos jellemzője van:
  - Bitmélység
  - Mintavételezési frekvencia.
- A bitmélység a hang intenzitását (hangerejét) határozza meg. Általában 16 vagy 24 bit

37 **Hanganyagok digitális tárolása**

- A mintavételezési frekvencia azt határozza meg, hogy mennyi mintát tároljunk egy másodperc hanghoz.
- Shannon-Nyquist tétel alapján, ha maximum 22 500Hz-ig akarok mintavételezni (emberi fül hallásküszöbe), akkor dupla akkora mintavételezés kell a hullámjelenségek miatt*

38 **Hanganyagok digitális tárolása**

- Gyakorlatban azonban a mintavételezési frekvencia növelése egy pont felett feleslegessé válik.
- Ez nagyjából 48 KHz környékén van.

39 **Rejtés Hanganyagban**

- Elv hasonló a képeknél alkalmazott megoldáshoz
- Emberi fül „hibáját” használja ki.
  - 20Hz -> ~22KHz hangot tudunk észlelni, de a nagy átlag „csak” 16KHz környékéig hall.
  - 2 hangot akkor tudunk csak megkülönböztetni, ha azok között valamekkora idő eltelt.
  - Ezen elveket használják ki a veszteséges hangtömörítő algoritmusok is. Pl: MP3, MP4, OGG, WMA

40 **Gondok szintén**

- Mivel a veszteséges tömörítés is ezeken az elveken dolgozik, szintén nem szerencsés választás az MP3, MP4, stb.. formátum
  - MP3 is alkalmas, de nem sok adat tárolásra
- PCM és lossless algoritmusok azonban tökéletesek a célra ☺
  - Sok adat belefér így rejtetten.
  -

41 **Példa**

- 19 mintán Sinus 0-180 fokig, 16 bites intenzitással ábrázolva.
- Az ábrázolási lépésköz: 5 fok
  - 0: 0, 1: 32768, -1: -32768
- Minden hangminta egy rejtett karaktert tárol
- Üzenet: ezegy rejtett pelda

42 **Példa kimenete ábrázolva**

43 **Példa kimenete ábrázolva 2,6x nagyítás**

44 **Példa tovább gondolva**

- A CD hanganyag 44 100 minta/másodperc
- Egy Audio CD hossza 80 perc
- Amiben 80*60*44 100 minta fér
  - Ez 211 680 000 mintát jelent.
- Ha minden minta 1 byte rejtett adatot tárol, akkor:
  - 211 680 000 byte extra adat rejtése lehetséges
  - Ez 201,87Mb adat!
  -

45 **Példa továbbgondolva**

- Ha csak 4 bitet rejtünk mintánként, akkor is ~100Mb bőven belefér.
- Ebben az esetben a detektálhatóság esélye tovább csökken.

46 **Példa MP3Stego program segítségével**

- ~5 perces hanganyagba 2kb-ot tud beágyazni észrevétlenül
-

47 **Problémák**

- A képek és a hangok esetén bemutatott példák túlzottan ideális esetek
- Feltételezzük, hogy az átjuttatás során az információ nem sérül meg.
- Pedig nagyon is sérül.

48 **Problémák**

- Képmegosztó szolgáltatások esetén bevett gyakorlat, hogy a feltöltött képeket veszteségesen újratömörítik kisebb méretre
- Hangmegosztó szolgáltatásoknál is fennáll ez a probléma.
- E-mail küldés esetén a fájloknak van feltöltési korlátja.

49 **Szteganográfia detektálhatósága**

- Ideális esetben lehetetlen, mivel a 3. fél nem tud a kommunikáció tényéről, és a használt módszerről.
- Statisztikai elemzéssel lehet detektálni, de nagyon nehéz. Általában erős gyanú fennállása

esetén folyamodnak ehhez.

50 **Szteganográfia a mindennapokban**

- Nem csak titkosítási célokra alkalmazott.
- Újabb színes lézer nyomtatók sárga pontokból álló mátrixot helyeznek el minden lapon.
- A mátrix tartalmazza kódolva a nyomtató sorszámát és a nyomtatás idejét is.

51 **Szteganográfia a mindennapokban**

- Vízjelek
- Digitális technikában kétfajtát különböztetünk meg:
  - Törékeny: A legkisebb módosítások is megváltoztatják, könnyen felfedezhetővé teszi a hamisítványokat.
  - Robosztus: A legtöbb módosítást túléli, így bizonyítja a hordozó eredetét.

52 **Vízjelek (kiegészítés)**

- A gyakorlatban kevesen tudják alkalmazni rendesen. Lásd: 9gag-szerű „egymástól lopunk" és a lájkokból élő oldalak.
- Egy vízjel akkor jó, ha egyszerűen nem távolítható el, vagyis robosztus.
- Legjobb módszer: halványan látható logó bevágása a kép közepére.

53 **Robosztus vízjel példa**

- Stock photograpy oldalakon lehet vele találkozni, mint a dreamstime.com

54 **Robosztus vízjel nyomtatás tiltására**

- Az ipari nyomtatók 4 színnel dolgoznak, míg a számítógépes színábrázolás 3 színnel.
- A kettő között vannak olyan színátfedések, amik képernyőn nem látszanak, de a nyomtató színkeverésében már igen.
- Az ilyen képek nyomtatás ellen védettek.

55 **Robosztus vízjel nyomtatás tiltására**

- Háromszínű nyomtatók sem megoldás, mivel a számítógépes modellben, ha minden színt összekeverünk, akkor fehéret kapunk.
- Nyomtatáskor azonban a színek hiánya felel meg a fehérnek, az összes szín keverése pedig feketének.
- Spórolási szempontból a színes nyomtatók nyomtatáskor a fekete patront is alkalmazzák

56 **Pontszerzési lehetőség**

- Készítsen egy olyan programot, ami egy 32 bites PNG képbe képes rejteni szöveget és a rejtett szöveget vissza is tudja olvasni.
- Nyelv: szabad választás, de a program működését és használatát dokumentálni kell.
- Csak az első helyes beküldőt díjazom, Lehet csoportmunka is (max. 3 fő)
- Prezentálni kell előadáson a programot.
- Díjazás: 40 pont

57 **Köszönöm a figyelmet**