Copyright Protection

Lecture 1: Watermarking and Fingerprinting (passive copyright protection)

Lecture 2: Tracing and Revoking pirates. (copyright protection via encryption)

1

Copyright Protection

- □igita□o□□ects are very easy to copy:
  □ usic□ □ ovies□ Te□t□ □□ecuta□□es□□ □ e□money.
- □ o□ to protect digita□copyrighted content□
  - □ ain topic o□ this □ecture.
- □ hou□d content □e protected□ (not our main topic)
  - □□ □i□□on □ □ □ a year in □oreign trade □or the □ □.
  - □ hou□d not con□□ct □ ith □□air□use□ doctrine.
- Can content □e protected□
  - Persistent pirate □ i□□a□□ays succeed in copying.
  - Techno□ogy can potentia□□y prevent sma□□sca□e copying: □keeping honest peop□e honest□

2

□ ethod 1: copyright cra□ □ers

- From here on□a□□ays use music as an e□amp□e.
- □ uppose □ e had a □content□a□ are□ hash □unction:
  □ : □music□ $\rightarrow$ □short strings□
  satis□ying:
  1. □□ $□_1$ and $□_2$ are t□o music c□ps (e.g. □ P□ □i□es) that p□ay the □same□ song then $□(□_1) □ □(□_2)$
  2. □ iven a c□p □ a pirate cannot create an □accepta□□e□ c□p □ □ such that $□(□) \neq □(□ □)$
- □ ash □unction must resist a□□signa□processing tricks□
- □ o such hash □unctions e□ist □□
  - □ nkno□ n. (some c□aim to have them)

3

□ sing these hash □unctions

- Write a copyright cra□ □er as □o□□o□ s:
  - Cra□ □er has □ □ o□ hashes o□ a□□copyrighted content.
  - Cra□ □er constant□y scans a□□ □ e□ sites□ □a□ a□ net□ ork□□ apster□etc.
  - For every music □i□e □ound□compute hash o□ music □i□e and compare to □ □.
  - □□ match is □ound□ca□□the □a□ yers.
- Pro□□ems:
  - □ ash □unctions un□ke□y to e□ist □or music.
  - □ oes not protect against anonymous postings: pu□□us
  - □ ery high □ ork□oad.

4

□ □ amp□es

- □ igi□ arc □ arc□ pider. □ cans □or pirated images.
- □ C□ □ : □ . □ hivakumar□□ tan□ord.
  - cra□□s the □ e□ □ooking □or academic p□agiarism.
  - □ evera□success stories.
  - http:□□□ □ □ □d□.stan□ord.edu□□shiva□□ C□ □ □scam□n□o.htm□

5

□ □ght improvement: □ atermarking

- □content□a□ are□ hash □unctions may not e□ist.
- □dea: at the recording studio em□ed a hidden □ atermark in the music □i□e:
  - □ m□ed(□ □□): outputs a □ atermarked version o□ music □ □ ith the in□ormation □ em□edded in it.
  - Retrieve(□ □): takes a □ atermarked music □i□e □ □and outputs the em□edded □ atermark □.
- Properties:
  - Watermark must □e inaudi□□e.
  - Watermark shou□d □e ro□ust: $□_1$ □ □m□ed(□ □□)
    □ iven □ □ pirate cannot create an □accepta□□e□ $□_2$ such that Retrieve($□_2$) $\neq$ □ .
  - □ ote: □ atermark must resist a□□signa□processing tricks. Resamp□ng□cropping□□o□ □pass □i□tering□□

6

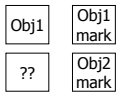
Issues

- Copyright crawler uses Retrieve algorithm.
- Benefits:
  - Copyright crawler does not need to maintain DB of all copyrighted material.
  - DB need store only hash.
    - Watermarking music seems to be an easier problem.
- Same problems as before:
  - Does not defend against anonymous postings.
  - High workload.

7

Robust Watermarks

- Note: typically, almond Retrieve algs are kept secret.
- No robust watermakring systems exist.
  - Unknown.
  - Stirmark: generic tool for removing image watermarks.
    - Obvious of watermarking scheme.
  - Fridrich challenge:
    - Broken: Fridrich et al.



8

Watermarking Images (>200 papers)

- DigiMarc: embeds creator's serial number.
  - Add or subtract small random quantities from each pixel. Embedded signal kept secret.
- Signafy (NEC).
  - Add small modifications to random frequencies of entire Fourier Spectrum.
  - Embedded signal kept secret.
- Caronni: Embed geometric shapes in background.
- SigNum Tech. (SureSign).

9

Watermarking Music (few papers)

- Aris Tech (MusicCode):
  - Rate: few bits/sec of music
- Solana (EMDNA)
  - Used by LiquidAudio.

Merged to form Verance
Used by SDMI

- Argent:
  - Embed full text information.
  - Frame-based: info. inserted at random areas of signal
  - Secret key determines random areas.

10

Method 2: policy watermark

- No copyright crawlers.
- Embed usage policy as watermark in music file.
- Every music player in the world works as follows:
  - Use Retrieve algorithm to check if watermark exists.
  - If so, play music only if policy is satisfied (e.g. payment, authorized player, etc.).
- Big big problems with this:
  - How to upgrade all music players? Why would consumers agree?
  - Retrieve algorithm is in the public domain.
    - Makes watermarking an even harder problem.
  - Open source players will ignore embedded policy.
- Seems to be the approach preferred by RIAA.

11

Method 3: Fingerprinting

- No copyright crawlers. No big brother DB players.
- Completely passive.
- Basis idea:
  - embed a unique user ID into each sold copy.
  - If user posts copy to web or Napster, embedded user ID identifies user.
- Problem:
  - Need ability to create distinct and indistinguishable versions of object.
  - Collusion: two users can compare their objects to find parts of the fingerprint.

12

□race □ Revoke schemes

13

□ontent protection via encryption

- Basic idea:
  - □ontent distributor encrypts content before releasing it. Release: □ □ $E_{\square}$(content)
  - □ecryption key embedded in all players.
  - □layer will only decrypt if policy is satisfied.
- Note: cannot prevent copying after decryption.
  - User can probe bus to sound card.
  - Unlike watermarking: watermark is embedded in content. □ropagates in cleartext copies of content.
- □roblem: what if one pirate uses reverse engineering to expose global key k ??

14

Example: □SS

- □SS: □ontent Scrambling System
  - Used to protect □□□ movies.
- Each □□□ player manufacturer i has key $\square_i$, e.g. $\square_{sony}$
  - Embed same key $\square_{sony}$ in all players from sony.
  - Every □□□ movie M is encrypted as follows:
    1. enc(content □ $E_{\square}$(M□ □ □ – a random key.
    2. $E_{\square_{sony}}$(k□ , $E_{\square_{philips}}$(□□ , □
  - About □□□ manufacturer keys.

15

□roblems with □SS

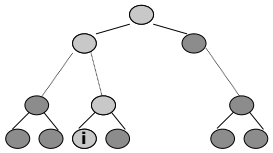
- □e□SS:
  - Extracted manufacturer key from □ing software player.
  - □ould then decrypt any □□□ movie that could be played on the □ing player.
  - M□AA revoked □ing key: disabled all □ing players□
- Bigger problem:
  - Encryption algorithm in □SS is based on □□SR□s
  - □ery fast: video rate decryption on weak □□□ player.
  - □ery weak: given one manuf. key can get all keys.

16

Better revocation techni□ue

- Basic idea: embed a distinct key in every player.

Players:

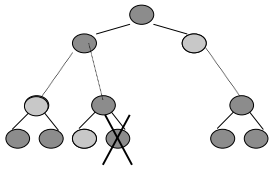


- Every node v has an associated key $\square_v$.
- Every player corresponds to leaf node.
- □ey for player i: all keys on path from root to leaf i.

17

Revocation

- Initially: encrypt all content with key at root.
  - Any player can decrypt content.
- When player i is revoked encrypt content-key so that all players can decrypt other than player i.



18

How to tell which player to revoke

- When pirate publishes single key on Internet, AACS knows which keys to revoke.
- What if pirate sells pirated players?
  - How can AACS tell which keys embedded in player?
- Solution: tracing systems can interact with player and determine how to revoke that player.
  - How? Not obvious how to do it.

19