Copyright Protection

Lecture 1: Watermarking and Fingerprinting (passive copyright protection)

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

Copyright Protection

- Digital objects are very easy to copy:
  Music, Movies, Text, Executables, … , e-money.
- How to protect digital copyrighted content?
  - Main topic of this lecture.
- Should content be protected? (not our main topic)
  - 40 billion $$$ a year in foreign trade for the US.
  - Should not conflict with "fair-use" doctrine.
- Can content be protected?
  - Persistent pirate will always succeed in copying.
  - Technology can potentially prevent small scale copying: "keeping honest people honest"

Method 1: copyright crawlers

- From here on, always use music as an example.
- Suppose we had a "content-aware" hash function:
  H: {music} $\rightarrow$ {short strings}

  satisfying:
  1. If M_1 and M_2 are two music clips (e.g. MP3 files) that play the "same" song then $H(M_1) = H(M_2)$
  2. Given a clip M a pirate cannot create an "acceptable" clip M' such that $H(M) \neq H(M')$
- Hash function must resist all signal processing tricks…
- Do such hash functions exist ??
  - Unknown. (some claim to have them)

Using these hash functions

- Write a copyright crawler as follows:
  - Crawler has DB of hashes of all copyrighted content.
  - Crawler constantly scans all web sites, KaZaA network, Napster, etc.
  - For every music file found, compute hash of music file and compare to DB.
  - If match is found, call the lawyers.
- Problems:
  - Hash functions unlikely to exist for music.
  - Does not protect against anonymous postings: publius
  - Very high workload.

Examples

- DigiMarc MarcSpider. Scans for pirated images.
- SCAM: N. Shivakumar, Stanford.
  - crawls the web looking for academic plagiarism.
  - Several success stories.
  - http://www-db.stanford.edu/~shiva/SCAM/scamInfo.html

Slight improvement: watermarking

- "content-aware" hash functions may not exist.
- Idea: at the recording studio embed a hidden watermark in the music file:
  - Embed(M, I): outputs a watermarked version of music M with the information I embedded in it.
  - Retrieve(M'): takes a watermarked music file M' and outputs the embedded watermark I.
- Properties:
  - Watermark must be inaudible.
  - Watermark should be robust: M_1 = Embed(M,I)
    Given M' pirate cannot create an "acceptable" M_2 such that Retrieve(M_2) $\neq I$.
  - Note: watermark must resist all signal processing tricks.
    Resampling, cropping, low-pass filtering, …

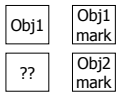
Issues

- Copyright crawler uses "Retrieve" algorithm.
- Benefits:
  - Copyright crawler does not need to maintain DB of all copyrighted material.
  - No need for content aware 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, Embed & Retrieve algs are kept secret.
- Do robust watermakring systems exist?
  - Unknown.
  - StirMark: generic tool for removing image watermarks. Oblivious of watermarking scheme.
  - SDMI challenge:
    - Broken: Felten, 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 (>200 papers)

- Aris Tech (MusicCode):
  - Rate: 100 bits/sec of music
- Solana (E-DNA)
  - Used by LiquidAudio.

Merged to form Verance Used by SDMI

- Argent:
  - Embed full text information.
  - FrameBased: 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 CD 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

Trace & Revoke schemes

13

Content protection via encryption

- Basic idea:
  - Content distributor encrypts content before releasing it. Release: $C = E_K[\text{content}]$
  - Decryption key embedded in all players.
  - Player 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. Propagates in cleartext copies of content.
- Problem: what if one pirate uses reverse engineering to expose global key k ??

14

Example: CSS

- CSS: Content Scrambling System
  - Used to protect DVD movies.
- Each DVD player manufacturer i has key K_i, e.g. K_{sony}
  - Embed same key K_{sony} in all players from sony.
  - Every DVD movie M is encrypted as follows:
    1. enc-content = $E_K[M]$; K – a random key.
    2. $E_{Ksony}[k]$, $E_{Kphilips}[K]$, …
  - About 220 manufacturer keys.

15

Problems with CSS

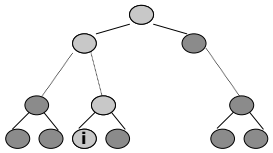
- DeCSS:
  - Extracted manufacturer key from Xing software player.
  - Could then decrypt any DVD movie that could be played on the Xing player.
  - MPAA revoked Xing key: disabled all Xing players!
- Bigger problem:
  - Encryption algorithm in CSS is based on LFSR's
  - Very fast: video rate decryption on weak DVD player.
  - Very weak: given one manuf. key can get all keys.

16

Better revocation technique

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

Players:

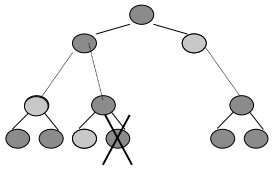


- Every node v has an associated key K_v.
- Every player corresponds to leaf node.
- Key for player i: all keys on path from root to leaf i.

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, MPAA knows which keys to revoke.
- What if pirate sells pirated players?
  - How can MPAA tell which keys embedded in player?
- Solution: Tracing systems can interact with player and determine how to revoke that player.
  - How? Homework.

19