Removing Watermarks

Nick Feamster
`feamster@lcs.mit.edu`

Applied Security Reading Group

May 14, 2001

Background

- **SDMI:** Secure Digital Music Initiative
  - Consortium of music-industry companies. They want to give publishers of music more control over what we can do with the music we buy.
  - *September 2000:* 3-week public challenge inviting members of the public to attempt to break some of SDMI's data-encoding technologies
- Edward Felten's research group took part, refused to sign confidentiality agreement to retain their right to publish findings. (hmm...)

Challenge Overview

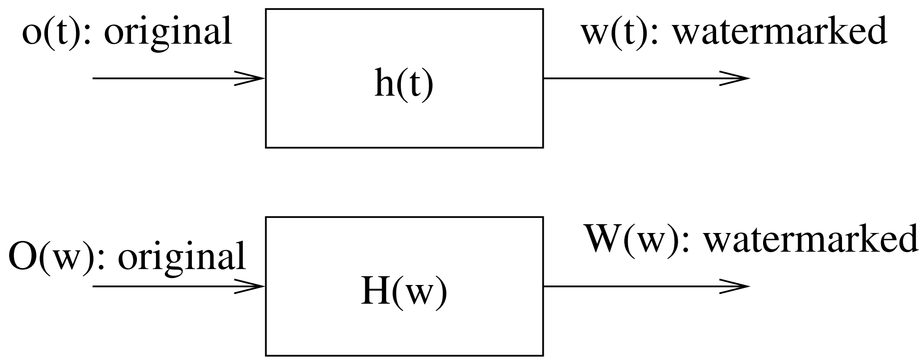
- **A, B, C, and F:** Watermarking technologies, challenge provided 3 files:
  - unwatermarked song
  - same song witha watermark
  - a different watermark song
- Goal was to remove watermark from 3rd file.
- **D and E:** Encodings trying to prevent “ripped” CDs.
- *I’ll give an overview of how Felten’s group defeated the watermarking technologies.*

Signal Processing Review

- For Linear, Time-Invariant Systems:

$$
\begin{aligned}
y(t) &= x(t) * h(t) \\
Y(\omega) &= X(\omega)H(\omega)
\end{aligned}
$$

- Use of FFT with a short window to determine the impulse response of the filter.



Echo Hiding Overview

- Human auditory system cannot preceive short echoes (e.g., 1 ms), introduces an echo with amplitude α, and delay τ.

$$\begin{aligned} y(t) &= x(t) * h(t) \\ h(t) &= \delta(t) * \alpha \cdot \delta(t - \tau) \\ y(t) &= x(t) + \alpha x(t - \tau) \end{aligned}$$

- Given knowledge about α and τ, can remove the echo. But this requires some signal processing magic...

Challenge A: Echo Hiding

- Step 1: look at $H(\omega) = W(\omega)/O(\omega)$
- Ripples/Sinusoids in frequency $\rightarrow$ Echoes in time!
- Step 2: look for frame sizes where echo parameters were the same $\rightarrow$ 1 frame $\approx$ 20 ms
- Step 3: Determine "delay hopping pattern" (i.e., the "secret key") by looking at sequences of frames. Delay hopping refers to how the echo parameters change over each frame.
  - Each frame has two echoes (an *echo pair*).
  - Echo pairs repeat every 16 frames
  - Echo gain is not the same each time

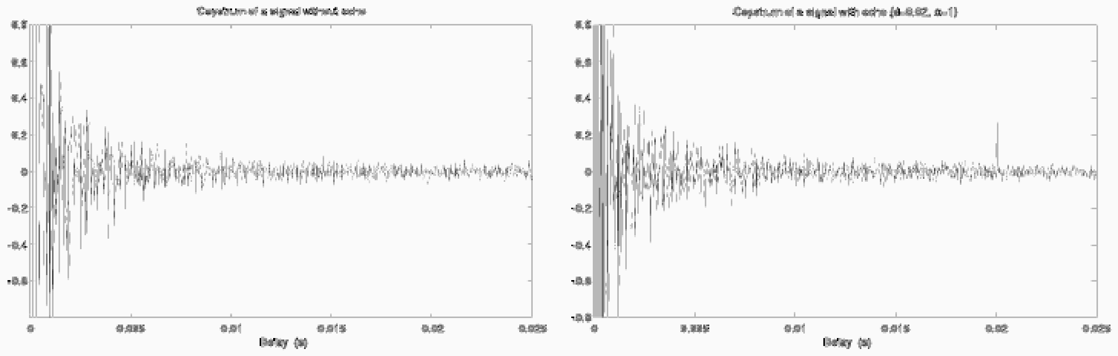
Determining the Encoding Parameters

- Since echo parameters are known to repeat every 16×20 milliseconds, can use *cepstral alanysis* of *liftered log spectra* —

$$
\begin{aligned}
X(e^{j\omega}) &= \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n} \\
\hat{X}(e^{j\omega}) &= \log\left[X(e^{j\omega})\right] \\
w(t) &= o(t) * h(t) \\
\hat{w}(t) &= \hat{o}(t) + \hat{h}(t)
\end{aligned}
$$

add up lots of $\hat{w}(t)$ believed to have the same echo $\rightarrow$ a spike corresponding to the echo imposed by $h(t)$.

Looking at the Cepstrum



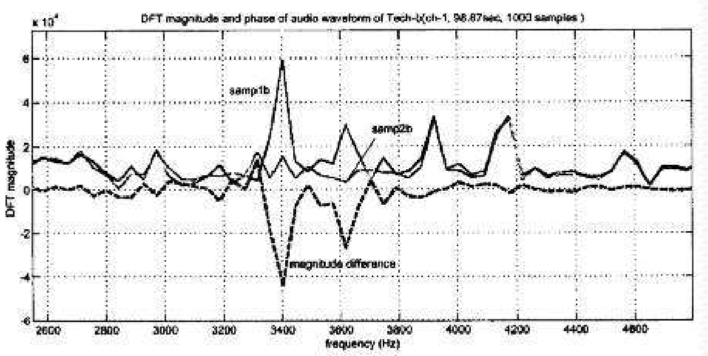
(a) (b)

Fig. 5. Graph (a) represents the cepstrum of a signal without echo. Graph (b) is the cepstrum of the same signal with a 20 ms echo which is emphasised by the very clear peak at 0.02 s.

$$\begin{aligned} \hat{w}(t) &= \hat{o}(t) + \hat{h}(t) \\ h(t) &= \delta(t) * \alpha \cdot \delta(t - \tau) \end{aligned}$$

Challenge B: Frequency Spikes

- Used “short-time FFT”: take a short window in time of the signal, and compute the FFT for that window.
- Frequency spikes at 2.8 KHz and 3.5 KHz
- Attack simply consists of smoothing out these spikes



Challenge C: Narrowband Bursts

- Analysis in time revealed short bursts of narrowband bursts centered around 1.35 KHz
- Two attacks:
  - bandstop filter at 1.35 KHz
  - frequency shift (narrowband no longer centered at 1.35 KHz)