Digitale Wasserzeichen

Verschlüsselung vs. Wasserzeichen

- Konzept
- Angriffe
- Beispiel Memorystick

Markierung von Audiodateien:

- EQ, Filter
- LSB-Verfahren
- Korrelations-Verfahren, Echo-Verfahren
- SDMI
- HackSDMI-Wettbewerb

Das Problem

- Riesenmarkt: USA sales 1997: CDs 9.915 M$, CCs 1.523 M$ (IBM)

Analogtechnik:

- Masterbänder altern
- jede Kopie schlechter als die Vorlage
- schlechte Qualität bei Consumertechniken (VHS, CC, ...)

Digitaltechnik:

- alle Kopien identisch mit Vorlage
- Alterung durch Kopieren kompensierbar
- bisheriger Kopierschutz sinnlos (S/PDIF copy-bit)
- auch billige Geräte/Recorder bieten 1:1 Kopien
- extrem gutes Preis/Leistungsverhältnis

=> (Raub-) Kopieren" nimmt zu (Napster, DivX, ...)

Literatur

- Proceedings of the IEEE, special issue on "identification and protection of multimedia information", 07/1999
- IEEE Trans. Signal Processing, spec. issue, "digital watermarking", 09/2000
- "Information hiding", Lecture notes in computer science, LNCS 1174, Cambridge 1996, K-INF-23262
- Cambridge security group, www.cl.cam.ac.uk
- J. Boeuf, J.P.Stern: An analysis of one of the SDMI candidates www.julienstern.org
- www.research.ibm.com/journal/sj/mit/sectiona/bender.html
- diverse Firmen und Organisationen, u.a.:
  www.sdmi.org www.hacksdmi.org
  www.musicode.com www.watermarkingworld.org
  www.audiotrack.com www.4centity.com
  ftp://ftp.cryptocd.com/pub/cryptocd/

Die Hoffnung

- Riesenmarkt "Musik"
- digitale Daten perfekt kopierbar

Verfahren zum Kopierschutz notwendig:

1) Verschlüsselung der Daten
   Zugriffskontrolle: Abspielen, Exportieren, ...
2) Markierung von Daten mit Urheber-Informationen
   Erkennung von Raubkopien
3) Personalisierung der Daten
   Zurückverfolgen von Raubkopien

- einige Verfahren bereits am Markt:
  LiquidAudio, WindowsMedia, SDMI, ...

Begriffe

Kryptographie

- Nachricht komplett verschlüsseln
- sichere Verfahren bekannt

Steganographie

- geheime Nachricht in offener Nachricht verbergen
- Morsecode mit {i,j} {f,t}
- Formatierung, z.B. Binärcode mit space, tab

Wasserzeichen ("watermark")

- Sichern des Urheberrechts, Angaben des Käufers, etc.
- offen oder versteckt

Verschlüsselung

- symmetrisch: DES, IDEA, ...
- asymmetrisch (public key): RSA, ...

- viele Verfahren gelten als sicher
- abhängig von Schlüssellänge und -"qualität"
- obwohl die Algorithmen bekannt sind

für Audiodateien:

- gängige Algorithmen eignen sich auch für Audio
- zunehmend verwendet, z.B. in WindowsMedia, LiquidAudio, ...
- auch in Hardware: Sony MagicGate MemoryStick

- aber: einmal entschlüsselte Daten können (raub)kopiert werden

MemoryStick

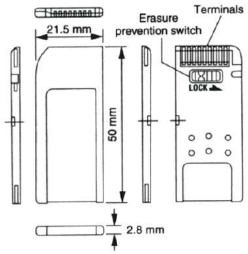


Figure 2. Memory Stick dimensions (mm).

Table 1. Memory Stick specifications.

Item	Description
Capacity (Mbytes)	4, 8, 16, 32, and 64 (currently); 128 (future)
No. of connector pins	10
Interface type	Serial
Serial clock	20 MHz (maximum)
Write speed	1.5 Mbytes/s (maximum)
Read speed	2.45 Mbytes/s (maximum)
Power source voltage	2.7 V to 3.6 V
Operating current	45 mA (average); 130 μA (standby)
Dimensions	21.5 mm wide × 50 mm long × 2.8 mm thick
Weight	~4 grams

Sony memorystick (1998):

- Flash-RAM basiertes Speichermedium
- kompakte Abmessungen, robustes Gehäuse
- als Konkurrenz zu SMC/MMC Speicherkarten
- "MagicGate"-Erweiterung: mit on-chip Verschlüsselung

[IEEE Micro 7/8-2000, 40]

MemoryStick: Konzept

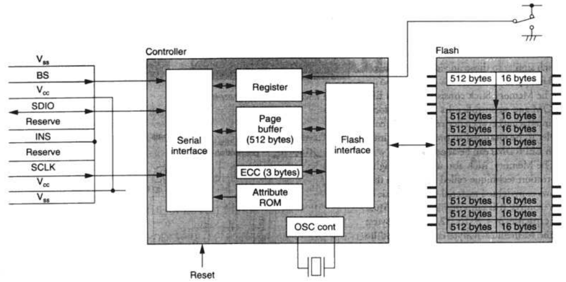


Figure 3. System configuration.

- für Consumergeräte: klein, robust, wenig Pins

- einheitliches Dateisystem, basiert auf FAT-16
- eingebauter Controller zur Ansteuerung des FLASH-RAM
- für Multimedia: Blockgröße 8 .. 16 KB

MemoryStick: serielles Protokoll

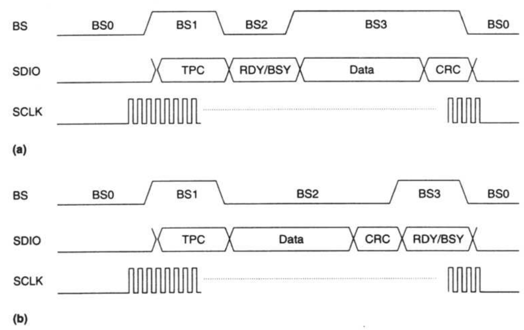


Figure 4. Timing example: read protocol (a) and write protocol (b).

- serielle Datenübertragung, SCLK bis 20 MHz
- nur drei Leitungen BS, SDIO, SCLK
- Lesen bis 1.5 MB/s, Schreiben bis 2.45 MB/s

MemoryStick: Vergleich



bisher jeder Hersteller mit eigenem Format:

- ○ Memorystick
- ○ Smart Media Cards
- ○ Compact Flash Cards
- ○ SIM-Cards
- ○ PCMCIA usw.

- jedes Format hat Vor- und Nachteile
- z.B. Frage controller / raw memory
- zukünftige Marktentwicklung unklar
- Übersicht: siehe Digitalkamera-Test in ct 11/2000

MemoryStick: Anwendungen

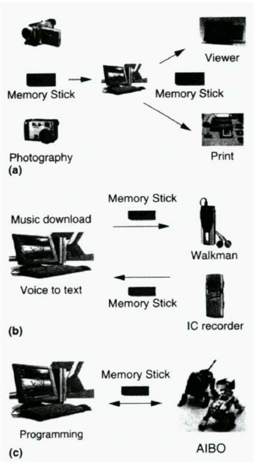
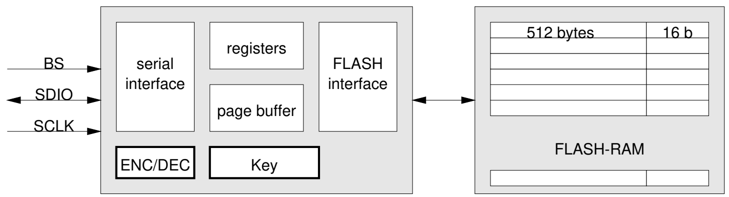


Figure 9. The worlds of pictures (a), audio/voice (b), and the AIBO robot (c).

- Datentransfer
- Datenspeicherung:
  - ○ Bildformate: JPEG, ...
  - ○ Audio: WAV, MP3, ...
  - ○ Sprache: G.726, ...
  - ○ Steuerdaten: Rohdaten
  - ○ ...
- Kameras (Photo, Video)
- Drucker
- Audioplayer (WAV, MP3, AAC, ...)
- Spielzeug (Aibo, ...)
- PCs
- usw.

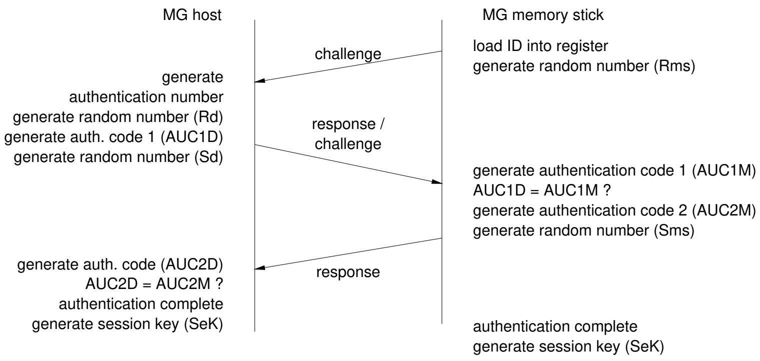
MemoryStick: MagicGate



MagicGate := erweiterter MemoryStick mit Verschlüsselung, Ende 1999

- eindeutige Seriennummer in jedem MG-MemoryStick
- erlaubt Identifikation des Mediums und der Daten
- Host übernimmt die Ver-/Entschlüsselung
- geringer Hardwareaufwand im Memorystick-Controller
- dadurch geringe Kosten

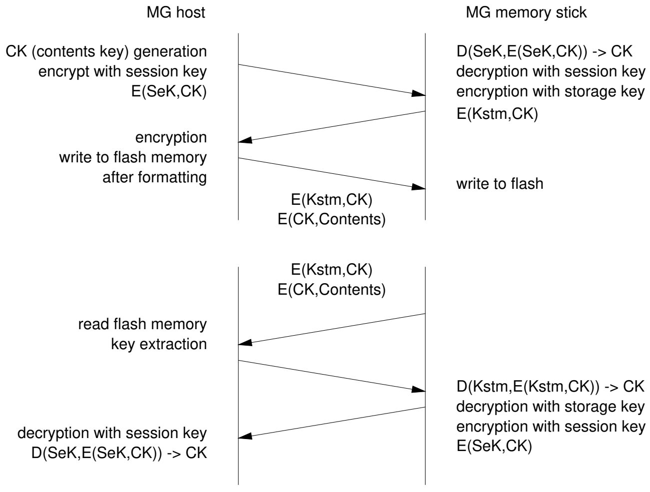
MemoryStick: Authentifizierung



- basiert auf der (eindeutigen) ID des MG-Memorystick
- erzeugt "session key" für die Ver-/Entschlüsselung

Leerseite

MG-MemoryStick: Read/Write

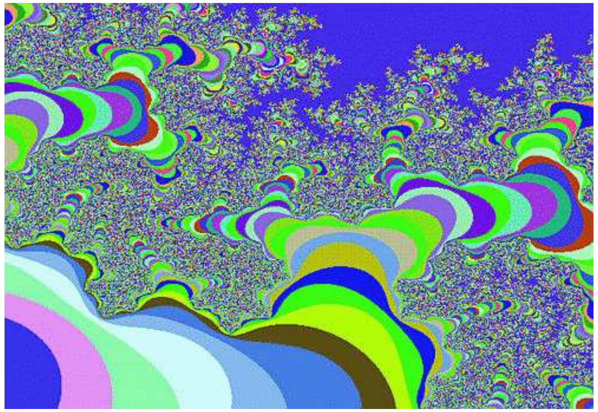


Steganographie

"information hiding" :

- geheime Nachricht in offener Nachricht verbergen
- bei Bedarf: geheime/offene Nachricht zusätzlich verschlüsseln
- z.B. Umgehen von Kryptographie-Exportverboten
- diverse Algorithmen und Tools erhältlich
- aber: keine sicheren Verfahren bekannt
- nur sicher, solange:
  - keine geheime Nachricht vermutet wird
  - der eingesetzte Algorithmus geheim bleibt
- viel schwieriger als Kryptographie:
  weil das Original "unverändert" aussehen soll

Steganographie: Mandelsteg

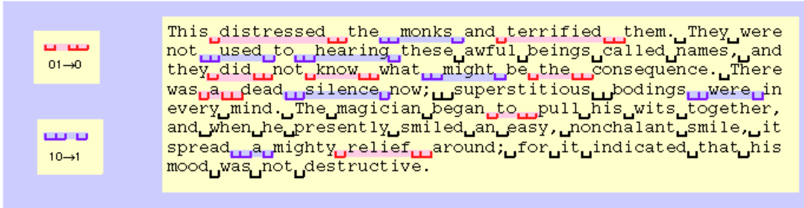


- Versteckten von Daten in veränderten (Mandelbrot-) Fraktalen

ftp://ftp.csua.berkeley.edu/pub/cypherpunks/steganography/

Steganographie: Textdateien

Figure 30 Data hidden through justification (text from *A Connecticut Yankee in King Arthur's Court* by Mark Twain)



[Bender 98]

Textdateien sind besonders schwer zu sichern:

- hinzugefügte Zeichen zerstören die Nutzinformation
- Wasserzeichen nur über Formatierung
- z.B. Tab/Space-Verfahren

Steganographie: Videokonferenz

- Übertragen von "Nebenabreden" ...

www.inf.tu-dresden.de/~hf2/publ/1997/FFWW_97ITSiStego.pdf

Steganographie: Textdateien

beim Papierausdruck mehr Möglichkeiten:

- Zeilenabstände oder Wortabstände modulieren (z.B. ~1/300)
- modifizierte Fonts
- fällt normalerweise nicht auf

- übersteht Vergrösserung, mehrfaches Kopieren
- übersteht Drucken, Scannen, OCR
- aber ASCII-Export zerstört die Info

Wasserzeichen

Wasserzeichen:

- Monogramm/Logo auf/in jedem Blatt
- seit etwa 1500
- als Qualitätsnachweis des Papiers

- oder zur Authentifizierung (Banknoten, Ausweis, Fahrkarten, etc)
- Fälschung stark erschwert
- Entfernung praktisch unmöglich

digitale Wasserzeichen:

- sichtbare oder unsichtbare Markierung von Daten
- bisherige Verfahren noch wenig robust

Wasserzeichen: Sony SACD

- Verfahren nicht dokumentiert

Wasserzeichen: Papier (um 1550)

Fig. 6. Monograms figuring TGE RG (Thomas Goodrich Eliensis – Bishop of Ely, England – and Remy/Remigius Guedon, the paper-maker). One of the oldest watermarks found in the Cambridge area (c.1550). At that time, watermarks were mainly used to identify the mill producing the paper; a means of guaranteeing quality. Courtesy of Dr E. Leedham-Green, Cambridge University Archives. Reproduction technique: beta radiography.

Anforderungen an Wasserzeichen

Wasserzeichen:

- ausreichende Datenrate für Kennzeichnung
- copy prohibit: 1 bit, ISBN: 10 Dezimalstellen
- Personalisierung: > 128 bit notwendig
- keine oder geringe Beeinträchtigung des Nutz- (Audio-) Signals

Robustheit:

- gegen elementare Signalverarbeitung
- gegen psychoakustische Signalverarbeitung
- gegen möglichst viele "Angriffe"
- Entfernung nur bei gleichzeitiger Verschlechterung des Nutzsignals
- gegen Fälschung

Wasserzeichen: Audio

Möglichkeiten bei Audiodateien:

- typ. Datenrate 64 kbps (MP3) bis 1.5 Mbps (CDDA)
- Wasserzeichen: 100+ bits, ca. alle 10 Sekunden wiederholen

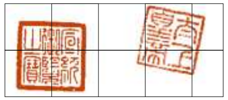
- "externe" Markierung (Chunk-Dateiformate)
- Filterung der Daten (Notch-Filter)
- LSB-Techniken
- Phasenverschiebungen
- Frequenzverschiebungen
- Spread-Spectrum
- Echo-Markierung

- Kombinationen dieser Verfahren

Mosaic-Attack

"Mosaic-Attack":

- Wasserzeichen erfordert Mindestlänge der Nutzdaten
- lange Dateien in viele kurze zerlegen
- Wasserzeichen wird verstümmelt



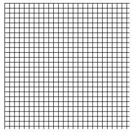
- für Bilddaten bereits Tools verfügbar
- Verfahren eignet sich auch für Audio

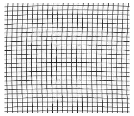
- z.B. als Verfahren gegen Web-Robots
  auf der Suche nach geklauten Abbildungen

typische Angriffe

einfache Angriffe:

- einfache digitale Filter
- Mischen mehrerer Signale oder mit Rauschen
- Lautstärkeänderung, Dynamikänderung
- sample-rate conversion
- A/D-D/A Konvertierung
- Tempoänderung, pitch-shifting
- MP3-Kodierung usw.
- Kombination mehrerer Verfahren (vgl. StirMARK)





gezielte Angriffe möglich, sobald Algorithmus bekannt:

- Angreifer kann sehr viel Rechenzeit investieren
- praktisch nicht zu verhindern

Interpretation-Attack

"Interpretation-Attack":

- die meisten Algorithmen sind "additiv"
- auch verschiedene Verfahren kombinierbar
- Angreifer fügt eigenes Wasserzeichen hinzu

Original:	d
Original + Wasserzeichen:	d + w
Pirat verbreitet:	d + w + x
Pirat behauptet:	d + w ist das Origial

=> Priorität der Urheberschaft?!

=> Reihenfolge der Wasserzeichen beweisbar ?!

Collusion-Attack

Überlagerung:

- Angriff gegen "personalisierte" Dateien
- Sammeln von vielen Varianten für eine Datei
- Mittelung all dieser Dateien

Original:	d
personalisierte Dateien:	d1=d+w1, d2=d+w2, ..., dn=d+wn
Mittelung:	D= 1/n *(d+d+...+d + w1+w2+...+wn)

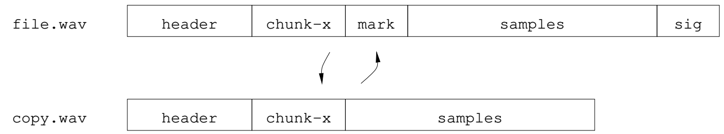
Nutzdaten bleiben erhalten, Wasserzeichen "mitteln sich raus"

=> Nachweis aller einzelnen Wasserzeichen ?!
=> Robustheit und Skalierung für großes n ?

externe Marken

Wasserzeichen im Header/in Kommentaren:

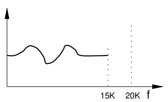
- einfachste Realisierung, etwa zustäzlicher Chunk im WAV-Format
- Nutzdaten werden nicht gestört
- trivial entfernbar (s. S/PDIF Copy-Bit) und fälschbar
- aber Kombination mit digitaler Signatur möglich



Radio: mit Kopierschutz und Marken . . .

auch analoge Medien sind geschützt:

Radiosender jederzeit identifizierbar:



- Jingles
- besondere EQ-Einstellungen
- extreme Dynamikkompression

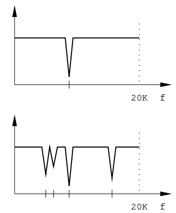
trotz zweifelhafter Qualität:

Mitschneiden unmöglich:

- Moderator spricht in Anfang und Ende jedes Songs
- Titel werden nicht ausgespielt
- Titel werden überblendet
- usw.

Notch-Filter

Signal mit Kerbfiltern (notchfilter) bearbeiten



- sehr schmale Notchfilter (z.B. 1/100 Oktave)
- kaum hörbar
- aber mit FFT sofort erkennbar
- evtl. mehrere Bänder sperren

- skaliert nicht auf hohe Anzahl verschiedener Wasserzeichen
- leicht entfernbar
- sehr leicht fälschbar

LSB-Techniken

Wasserzeichen im LSB der Nutzdaten kodieren:

```
sample[t] = sample[t] & 0xfffe
          + mark[t] & 0x0001;
```

- sehr einfach zu realisieren
- fällt in (Pop-) Musik nicht auf
- in leisen Passagen evtl. hörbar
- sehr hohe Bitrate für das Wasserzeichen möglich

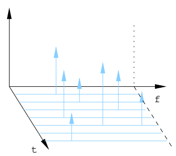
- Konflikt mit Dithering-Verfahren
- anfällig gegen Rauschen (z.B. durch DA-AD Wandlung)
- sehr leicht entfernbar, evtl. fälschbar
- bei 24-bit (DVD-Audio) auch mehrere Bits nutzbar

Spread-Spectrum

schmalbandiges Signal in breitbandigem Signal verstecken

- Nutzsignal wird auf mehrere Frequenzbänder aufmoduliert
- Frequenzbänder werden ständig gewechselt
- Auswahl der Frequenzbänder pseudozufällig
- Sender und Dekoder verwenden gleiche Zufallszahlen

- seit WW2 militärisch genutzt
- GSM, DECT Mobiltelefone, GPS, usw.



- ohne Kenntnis der Zufallszahlen nicht detektierbar
- Nutzsignal detektierbar, auch wenn << Rauschen
- unempfindlich gegen einfache Angriffe

- sehr empfindlich gegen Timing-Veränderungen

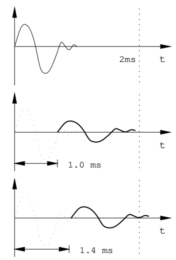
Echo-Marking

Wasserzeichen als Echo im Signal verstecken:

- kurze Echos sind kaum wahrnehmbar
- eignen sich damit als Kennzeichnung:

  Bit 0: 1.0 msec Echo

  Bit 1: 1.4 msec Echo



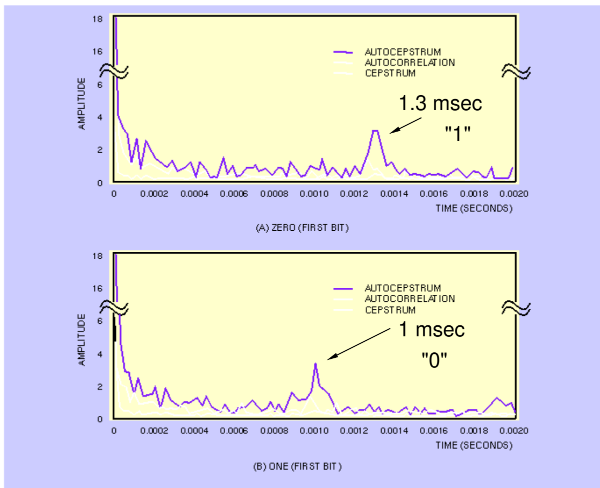
- Detektion erfordert Analyse der Echos
- sehr robust gegen alle einfachen Angriffe

- Entfernung des Echos sehr aufwendig
- aber machbar ...

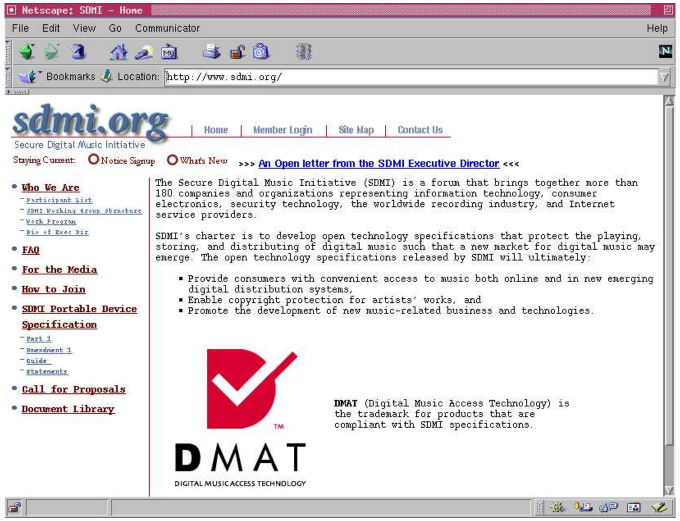
`http://www.research.ibm.com/journal/sj/mit/sectiona/bender.html`

Echo-Marking

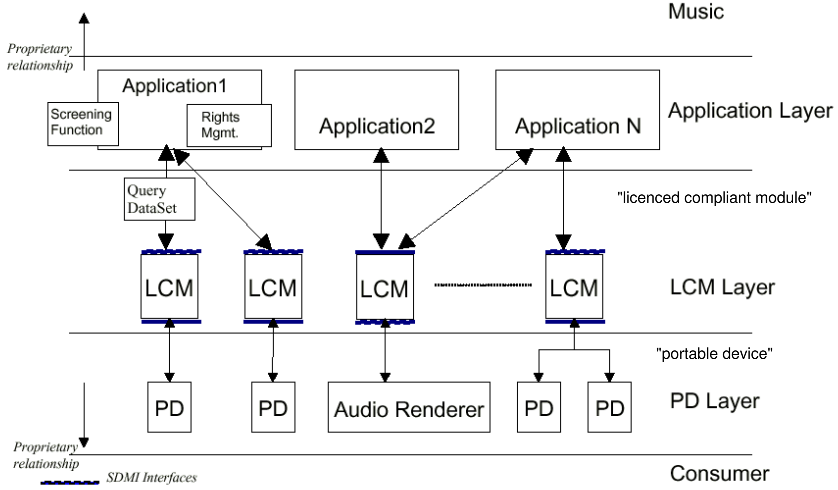
Figure 28 Result of autocepstrum and autocorrelation for (A) "zero" and (B) "one" bits



SDMI: homepage



SDMI: layers



SDMI: Konzept

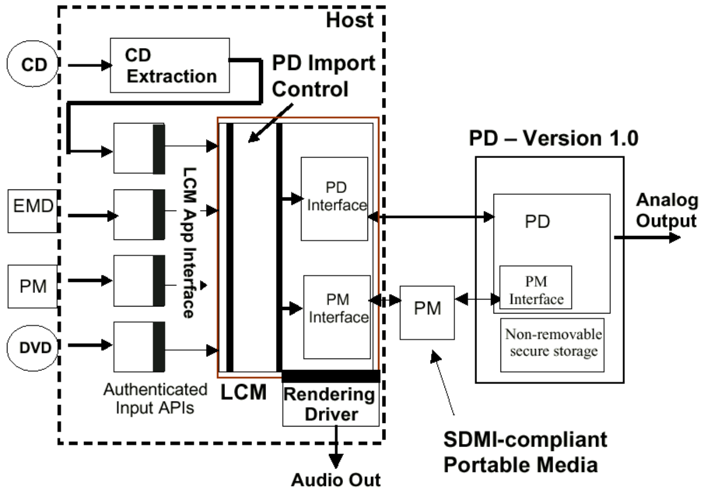
SDMI := "Secure Digital Music Initiative"

- Kopierschutz für digitale Audiodaten:
  gegen Raubkopien bzw. das Abspielen von Raubkopien
- Entwicklung entsprechender Algorithmen und Geräte
- Kooperation von ca. 200 Firmen

- Trennung von digitalem (sicheren) und analogem Bereich
- Erkennen von "compressed data" (d.h. insb. MP3)
- Umstieg "legacy" auf SDMI in mehreren Phasen

- Einsatz von Verschlüsselung
- Erkennung von Raubkopien über digitale Wasserzeichen
- Test der ersten Algorithmen Ende 2000

SDMI: devices



SDMI: Matrix

Capabilities	Today (Non-SDMI Devices)	SDMI (Devices in Holiday 99)	SDMI (Future)
Download and Play current digital music tracks including MP3s	✓	✓	✓
Download and play SDMI digital music tracks		✓	✓+
Transfer personal CD collection to a PC	✓	✓	✓
Transfer current digital music tracks from PC to a portable device	✓	✓	✓
Transfer SDMI digital music tracks from PC to portable device		✓	✓
Share current digital music tracks	✓	✓	✓
Enable sharing of SDMI digital music tracks		✓	✓+
Enable independent artists, church choirs, etc. to create and distribute digital music	✓	✓	✓
Explicitly supports copyright / rights management for digital music distribution		✓	✓+

- Abspielen und Kopieren von "legacy" Medien erlaubt
- aber "neue" Medien (ab. ca. 2000) geschützt

SDMI: Phasen

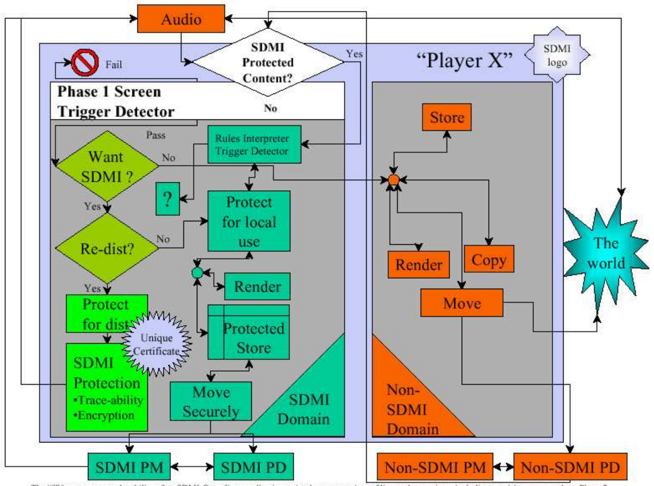
Phase 1:

- erste Gerätegeneration, mit Option zum Upgrade auf Phase 2
- unmarkierte Dateien können abgespielt werden
- markierte Dateien werden zurückgewiesen
- dann Upgrade auf Phase 2 notwendig

Phase 2:

- spielt unmarkierte und "heile" markierte Dateien
- erkennt komprimierte markierte Dateien, spielt diese nicht
- bisher nicht voll spezifiziert

SDMI: domains



The "?" box represents the ability of an SDMI-Compliant application to implement a variety of licensed operations, including requiring an upgrade to Phase 2.

Leerseite

SDMI: HackSDMI

Test der SDMI Algorithmen (Kandidaten) notwendig

- öffentlicher Wettbewerb, www.hacksdmi.org
- Preisgeld von $10.000 für das "Knacken" der Algorithmen

- für jedes der vorgestellten Verfahren je drei Wav-Dateien:
  - Song A   Original, ohne Wasserzeichen
  - Song AW   mit Wasserzeichen
  - Song BW   mit Wasserzeichen

- und ein Orakel:
  - Upload "geknackter" Versionen B´ von BW
  - Erkennung des Wasserzeichens
  - Bewertung der Audioqualität

HackSDMI: Analyse und Angriff

- gegeben: Demodateien A, AW, BW
- und: Wasserzeichen W ist für alle Dateien gleich (!)

Angriffsprinzip:

- Wasserzeichen extrahieren, W = (A - AW)
- anschließend analysieren: Autokorrelation usw.
- Spezifikation fordert: Periode < 15 sec.

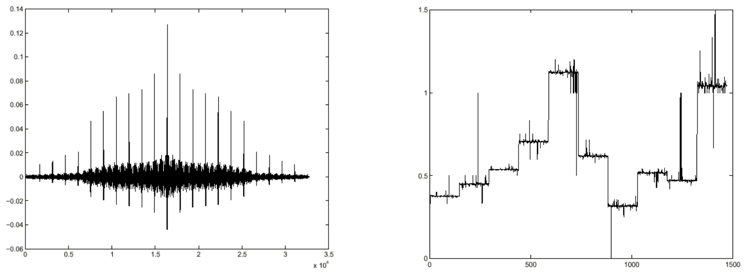
anschließend:

- Prinzip des Algorithmus erkennen
- passenden Encoder/Decoder schreiben
- Wasserzeichen gezielt angreifen ("surgical attack")

- notfalls "random attack": bis Orakel(Player) die Datei akzeptieren

HackSDMI: homepage

HackSDMI: Autokorrelation



- links: Autokorrelation des Wasserzeichens (Differenz AW-A)
- offensichtliche Korrelation alle 1470 Samples, => Periode 1470

- rechts: Korrelation des um 1470 Samples verzögerten W-Signals
- alle 147 Samples mit veränderter Amplitude
- vermutlich W proportional zum Nutzsignal, ||W(j)|| ~ ||S(j)||

HackSDMI: vermutete Algorithmen

Algorithm 1 Marking algorithm: inputs: $w \in [-1,1]^{1470}$, $s \in [-1,1]^m$

Output and skip *start* samples from the original song
while The song is not over **do**
  $s \leftarrow$ the next 1470 samples of the song
  for $j = 1$ to 10 **do**
    $s[j] \leftarrow s[j] + \beta||s[j]||w[j]$
  end for
  Output s
end while

Algorithm 2 Detection algorithm inputs: $w \in [-1,1]^{1470}$, $s' \in [-1,1]^m, p, \delta$

Skip *start* samples (possibly resynchronize by correlation)
while The song is not over **do**
  $sum \leftarrow 0$
  Get the next p chunks of 1470 samples
  for Each of these chunks **do**
    $s \leftarrow$ the next 1470 chunk
    for $j = 1$ to 10 **do**
      $s[j] \leftarrow s[j]/\beta||s[j]||w[j]$
    end for
    $sum \leftarrow sum + s$
  end for
  $Q = sum.w$
  if $Q > \delta$ **then**
    Outputs "mark found"
  end if
end while

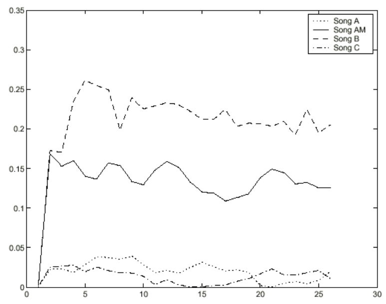
HackSDMI: Status

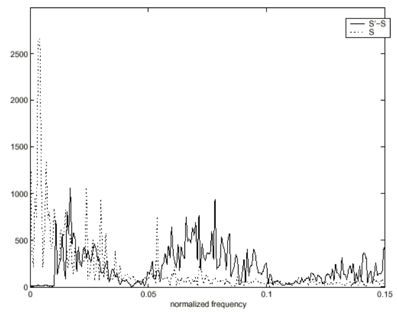
- hacksdmi.org Website derzeit nicht mehr erreichbar
- keine eigenen Experimente mehr möglich

- Preisgeld für zwei Angriffe ausgezahlt
- angeblich alle Verfahren "geknackt":
  http://www.cs.princeton.edu/sip/sdmi/
  http://www.julienstern.org/sdmi/

- derzeit keine "sicheren" Verfahren bekannt
- einige Sicherheit nur bei "geheimem" Algorithmus
- auch dann trügerisch (vgl. DeCSS)

HackSDMI: Analyse des Angriffs





- Analyse der Signale A, AW, BW, und des rekonstruierten B
- links: Ausgangssignal des Detektionsalgorithmus
- rechts: Spektrum des Wasserzeichen-Signals

=> Wasserzeichen ohne Qualitätsverlust entfernt

Feedback

- Hinweise / Ergänzungen / Berichtigungen
- Wünsche / Anmerkungen
- neue Themenvorschläge

- hörenswerte neue CDs / DVDs / MP3s / MIDIs
- interessante neue Software, neue Geräte

- Interesse an Studien/Diplomarbeiten !?
- Vorschläge für Projekte !? (auch für das Baccalaureat)

=> bitte an: hendrich@informatik.uni-hamburg.de

- vielen Dank für das Interesse!