Multimedia Security

MPEG Security

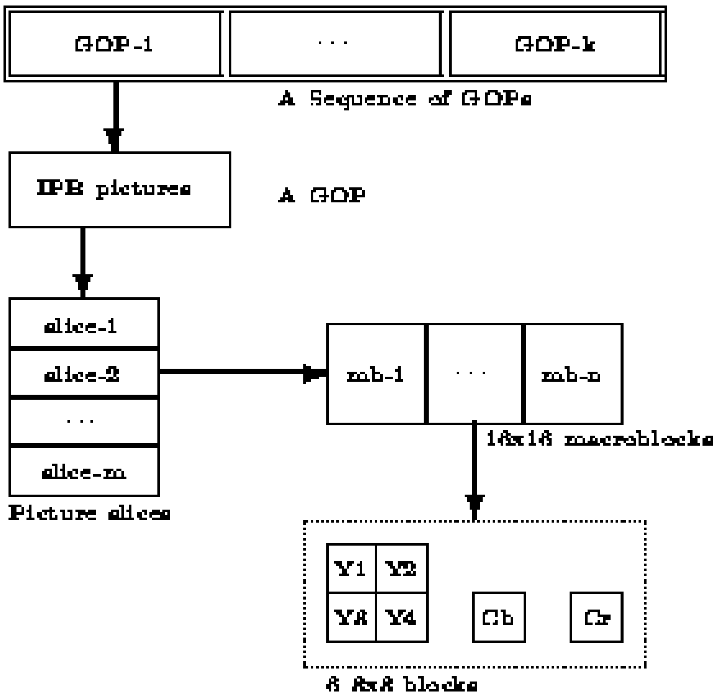
Ronald Blaschke
9620411
rblasch@cosy.sbg.ac.at
http://www.cosy.sbg.ac.at/~rblasch/

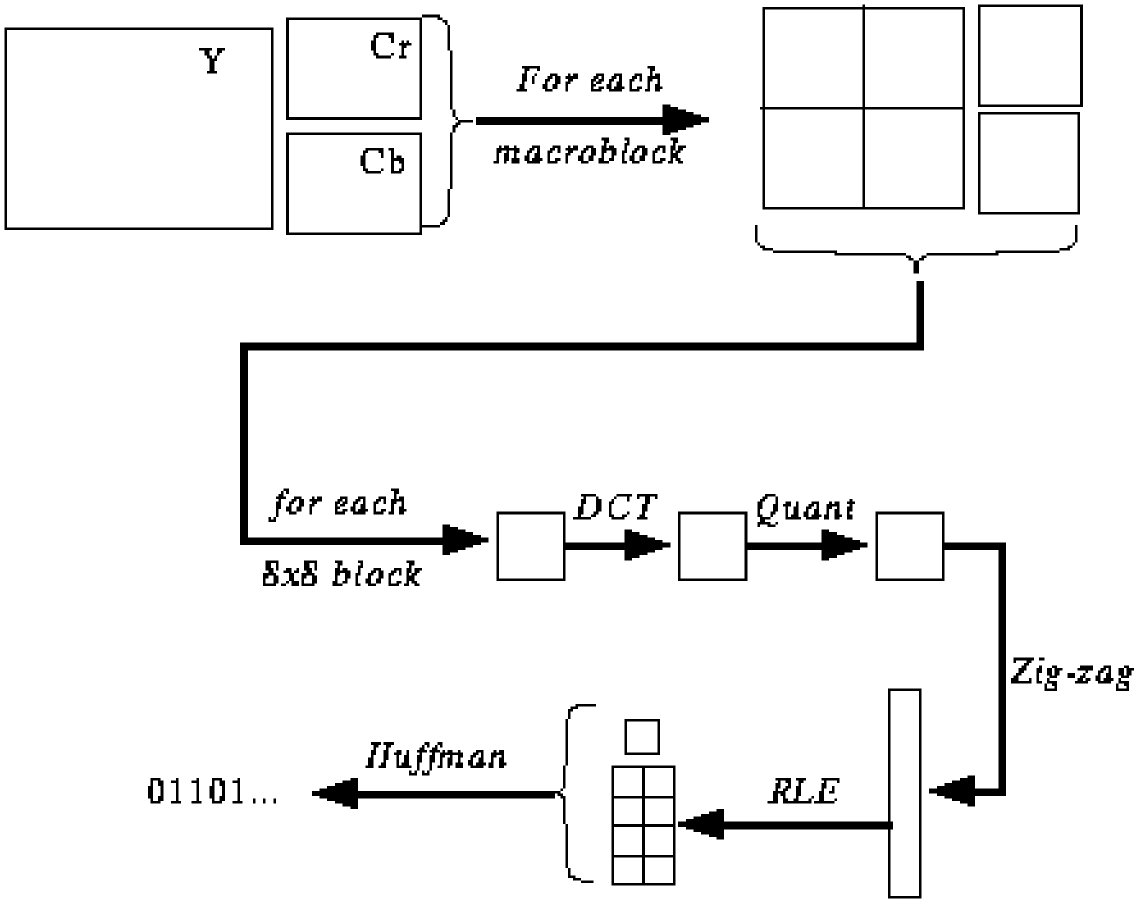
16. Jänner 1999

- MPEG Überblick
- Algorithmen: Naive Algorithm, Pure Permutation Algorithm, Zig-Zag Permutation Algorithm, Selective Algorithm und Video Encryption Algorithm
- Einsatzgebiete
- Quellen

MPEG Überblick

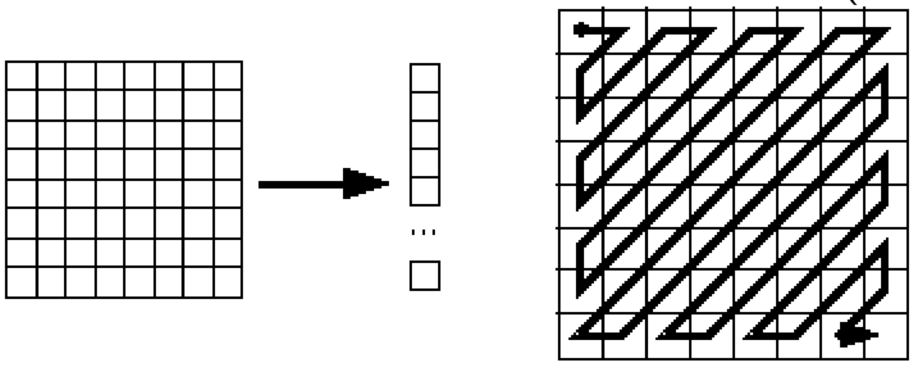
- Ein MPEG-Video besteht aus vielen Gruppen von Bildern (group of pictures, GOP).
- Eine GOP besteht aus einer Reihe von I, P und B Frames.
- Ein I-frame ist ein vollständiges Bild, ein P-frame (predictive-coded) stellt eine Bewegung im letzten Frame dar und ein B-frame (bidirectional predictive-coded) ist ein P-frame, der sich auch auf zukünftige Frames bezieht (jedoch in einer GOP bleibt).
- Ein Frame besteht aus 16x16 Pixel großen Macroblöcken: 4 Luminanz-Planes (Grauwerte, Y) und 2 Chrominanz (Farbwerte, Cr und Cb).
- Diese wiederum bestehen aus 8x8 Pixel großen Blöcken.





- Discrete Cosine Transform (DCT) – gute Approximation von Zeilen, mit wenig Koeffizienten. Liefert 8x8 Matrix. Werte werden DC und A_1 bis A_{63} genannt. Reihenfolge durch Zig-zag Muster (später mehr).
- Quantization (Quant) – Weniger Bits für Darstellung, jedoch Rundungsfehler (Hauptquelle für Qualitätsverluste).

- Zig-zag Scan (Zig-zag) – Mapping von 8x8 auf 1x64 Vector. Grund für Muster: Große Werte links oben, Rest sind kleine Werte (viele Nullen).



- Run Length Encode (RLE) – Viele Nullen in Vector, daher Darstellung als (skip, value) Paare; *skip* = Anzahl der folgenden Nullen, *value* = nächster Wert ($\neq 0$).
- Huffman Codierung.

Folgende typische Eigenschaften sind für die Verschlüsselung nützlich.

- Gleichmäßige Frequenzverteilung von Byte-Werten, d. h. jeder Bytewert tritt in einem Datenstrom ca. gleich oft auf.
- Jede Nachbarschaft von Byte-Werten (256x256 Werte) ist möglich.
- Perioden tretten kaum auf; zumindest kann dies für $1/16$ eines Frames ausgeschlossen werden.

Und folgende für das Brechen.

- Nach *Quant* befinden sich alle großen Werte in der linken oberen Ecke, der größte oft ganz links oben.

Algorithmen

- Naive Algorithm: Bekannte Verschlüsselungsalgorithmen. Sehr sicher, langsam, keine Größenänderung.
- Pure Permutation Algorithm: Ergebnis-Datenstrom wird mittels Permutationsschlüssel permutiert. Anfällig für known-plaintext Attacke, man muß nur Plaintext mit Ciphertext vergleichen; 1 I-Frame sollte zum Brechen genügen. Wenig sicher, sehr schnell, keine Größenänderung.
- Zig-Zag Permutation Algorithm: Bekanntes Zig-zag Muster wird durch Permutation ersetzt. Sehr anfällig für known-plaintext Attacke.
  Erweiterungen: Binary coin flipping sequence + 2 Permutationsschlüssel. „Münze“ entscheidet welcher Schlüssel für einen 8x8 Block verwendet wird. Oder: DC-Werte aus 8 Blöcken sammeln, mit DES verschlüsseln und getrennt übertragen.
  Anfällig für known-plaintext Attacke und ciphertext-only Attacke. Known-plaintext: einfacher Vergleich und finden der Permutation; 2 Perm. Schlüssel hilft nicht, da „große Werte links oben“ auf die richtige Liste deutet. 2 Alternative entspricht etwa SECMPEG 2nd level (später mehr).
  Sehr unsicher, sehr schnell, viel Größenzuwachs (>30%); Grund: Verlust von vielen großen Nuller-Blöcken behindert RLE-Komprimierung.

- Selective Algorithm: Alle Algorithmen, die sich der MPEG-Layer Struktur bedienen, um gewisse ausgewählte Teile zu verschlüsseln. Basisalgorithmus: Verschlüssle I-Frames, Rest (P,B-Frames) ist dann unbrauchbar. Tatsächlich: Auch ohne Entschlüsselung können Teile sichtbar sein. Sicherheit steigt mit I-Frames Anteil, da dann mehr verschlüsselt wird.
  SECMPEG nutzt und erweitert Basis: Verwendung von DES und RSA für Verschlüsselung, 4 Sicherheitsebenen:
  1. Header verschlüsseln
  2. Header + DC + ersten ACs
  3. alle I-Frames, auch in B,P-Frames eingebettete
  4. alles verschlüsseln

  Nachteil von SECMPEG: Nicht kompatibel zu MPEG.
  Relativ sicher, schnell, keine Größenänderung.
- Video Encryption Algorithm (VEA): Grundidee: Ein Teil eines Frames sieht so aus: $a_1 a_2 a_3 \ldots\ a_{2n-1} a_{2n}$. Aufteilen in 2 Listen: Odd List, Even List. Dann

$$\begin{array}{r cccc}
 & a_1 & a_3 & \cdots & a_{2n-1} \\
\text{Xor} & a_2 & a_4 & \cdots & a_{2n} \\
\hline
 & c_1 & c_2 & \cdots & c_n
\end{array}$$

  Wähle Verschlüsselungsfunktion E (z. B. DES), um $a_2 a_4 \ldots\ a_{2n}$ zu verschlüsseln.
  Ciphertext: $c_1 c_2 \ldots\ c_n E(a_2 a_4 \ldots\ a_{2n})$. D. h. man verwendet den halben Strom als one-time pad. Möglich da keine wiederholenden Muster auftreten.

- VEA Sicherheitserweiterungen:
  - *KeyM* Einführung eines zufälligen 128-bit Schlüssels, der die Odd-Even-List Zuordnung permutiert (1 = Byte in Odd List, 0 = Byte in Even List)..
  - Key_i Nur $1/16$ eines Frames Muster-wiederholungsfrei. Ausdehnung auf $1/2$ Frame durch $Key_1, \ldots, Key_8$. Key_i ist eine zufällige Permutation von (1, . . . , 32). Jeweils 32 Byte der Even List werden der Reihe nach mit Key_i permutiert, die nächsten 32 mit Key_{i+1}, usw.
  - *KeyF* 64-bit Schlüssel, enthält eine zufällige Permutation von (1, . . . , 16), wobei je 4 Bit eine Zahl darstellen. Damit wird KeyM und Key_i bei jedem Frame verändert.
  - *KeyE* Schlüssel für Funktion E (kann auch für die Verschlüsselung der obigen Schlüssel verwendet werden).

  Sehr sicher, schnell, keine Größenänderung.

Einsatzgebiete

Selective Algorithms kann man gut bei pay-per-view, video-on-demand, etc. einsetzten. Überall wo SSMC (Single Server Multiple Client) im Spiel ist. Man kann 90% unverschlüsselt (P, B-Frames) übertragen, die restlichen 10% (I-Frames) müssen pro Zuseher verschlüsselt werden, weil

1. jeder Zuseher einen eigenen Schlüssel hat.
2. jeder Zuseher ein Video mit eigenem Watermark erhält, um anonyme Raubkopien zu erschweren.

Daher: Unverschlüsseltes multicasten, Verschlüsseltes singlecasten.
Dieses Einsatzgebiet muß nicht unbedingt hochsicher sein, dafür die Video Durchsatzraten hoch.

Die anderen sicheren Algorithmen eignen sich für Videokonferenzen, etc. Dort überwiegt Sicherheit, Geschwindigkeit ist durch wenige Personen kaum ein Problem.

Quellen

Literatur

[C&G22] *COMPUTER & GRAPHICS*, Volume 22, November 4, July/August 1998.

[M2TUT] *MPEG-2: the basics* Einführung in MPEG-2. http://www.tmo.hp.com/tmo/pia/component_test/PIAApp/Notes/English/MPEGtutorial1.html

[JSAC1] *Run-Time Performance Evaluation of A Secure MPEG system Supporting both Selective Watermarking and Encryption.* S.F. Wu, T.L. Wu http://shang.csc.ncsu.edu/papers/jsacA4.ps.gz

[JSAC2] *NCSU SHANG: Secure MPEG*, Implementierung von [JSAC1]. http://shang.csc.ncsu.edu/smpeg.html

[SHANG1] *Selective Encryption and Watermarking of MPEG Video.* T.L. Wu, S.F. Wu http://shang.csc.ncsu.edu/papers/smpeg.ps.gz

[ISOC96] *An Empirical Study of Secure MPEG Video Transmissions.* Iskender Agi and Li Gon http://www.csl.sri.com/~iskender/papers/isoc_sndss96.html

[VEA98] *ACM Multimedia 98 - Electronic Proceedings, A Fast MPEG Video Encryption Algorithm.* Changgui Shi & Bharat Bhargava http://www.acm.org/sigmm/MM98/electronic_proceedings/shi/index.html

[MPG] *MPEG . ORG - MPEG Pointers and Resources* http://www.mpeg.org/

[SECMPG] *SECMPEG – Secure MPEG* http://www.powerweb.de/phade/work.html