Communication-Friendly Encryption of Multimedia

Min Wu and Yinian Mao

Department of Electrical and Computer Engineering

University of Maryland, College Park

www.ece.umd.edu/~minwu/research.html

Motivation

- Access control as an important element in many multimedia applications
  - For securing information sharing and E-marketplace of MM
    - *Archive and remote access of surveillance video and commercial digital library*
    - *Multimedia conferencing and on-demand services*
- Can generic encryption fit all needs?
  - Syntax and structure of multimedia stream will disappear after generic encryption
  - Multimedia-specific issues such as intermediate processing may not be addressed sufficiently by generic encryption

Special Needs with Encrypted Multimedia

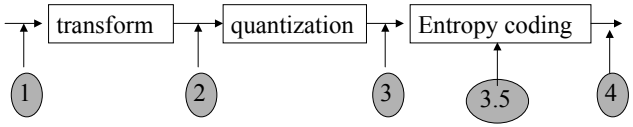
- Desirable intermediate processing in multimedia transmission
  - Unequal error protection
  - Scalability by transcoding or scalable coding
    - *heterogeneous environment: various bandwidth condition, computing capability, and power limit*
- Quality, Time, and Value of multimedia
  - High quality version need to be protected with much bigger effort
  - Commercial value of movie usually decrease rapidly with time

=> Selective encryption
  - To allow encrypting only desired portion for applications' needs

=> "Encryption-domain" processing
  - Transcoding, content-based search in lower-level features, etc.

Multimedia Encryption

- A general transform coding framework



- *Several possible locations for encryption*
  - (1): sample domain encryption
  - (3) and (3.5): encrypt intermediate symbols, e.g. (run,size)
  - (4): encrypt the compressed bitstream
- *Prior work*
  - *spatial-domain shuffling (Zeng-Lei), codeword index encryption (Wen et al.), encryption using multiple entropy codebook (Wu-Kuo)*

Multimedia Encryption: Two "Philosophies"

- Treating multimedia as generic data
  - Examples: (1) encrypt the whole multimedia bit-stream using AES, (2)encrypt the sub-bitstream such as I frames using DES/AES.
  - Pro: simple, "provable" security of the data as a whole
  - Con: limited applicable intermediate processing, and often need difficult to decrypt a considerable portion of the bit-stream
- Encryption with MMSP philosophy: Joint compress. & encrypt.
  - Maintain the multimedia format: possible to be "standard-compliant"
  - Easy to identify the structure in encrypted stream
    - *video frame, layer, block, …*
  - Facilitate many intermediate processings on encrypted data
    - *"Communication-friendly":* frame dropping, bit-plane dropping
    - *Partial decryption for extraction of region-of-interest, DC-images, and other useful content search features*

Proposed Multimedia Encryption Operations

- Generalized index mapping with controlled overhead
  - Can be applied to symbols taken value from a finite set
  - Control overhead by set partitioning
- Intra bit-plane shuffling compatible to FGS
  - Compatible to Fine Granularity Scalable coding (FGS)
  - Compression overhead smaller than index mapping approach

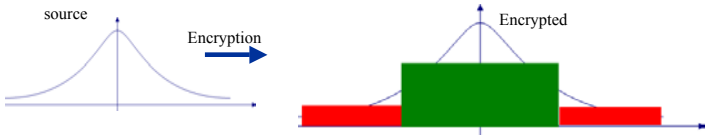
Generalized Index Mapping (G.I.M.)

- Generalized upon codeword index encryption by Wen et al. '01
  - Can encrypt symbols taken values from a finite set *S*
- Basic idea of index encryption
  - Equiv. to location-varying mapping
  - Map one symbol to an encrypted symbol in *S* by encrypting concatenated indices & inversely mapping from indices to symbol

$$\begin{array}{lcl} X_1 & & X_{Enc(1)} \\ X_2 & ===> & X_{Enc(2)} \\ : & & : \\ X_S & & X_{End(S)} \end{array}$$

- Example: to encrypt a symbol sequence "ABBDC"
  - Assign fixed-length index: A [00], B [01], C [10], D [11]
  - Convert symbol sequence to index sequence: "00 01 01 11 10"
  - Encrypt index sequence (by block/stream cipher): "01 01 11 00 00"
  - Convert encrypted index seq. to symbol sequence: "BBDAA"

Bitrate Overhead Control of G.I.M.



- Overhead control through set partitioning
  - Assume standard entropy codebook is used
  - Partition symbols into subsets & restrict mapping within subsets
- Analytic result of overhead
  - Based on Kullback-Leibler divergence *(see MMSP'02 paper for details)*

$$\delta L = \Sigma (q_i - p_i)\, l_i = D(p||q) + D(q||r) - D(p||r)$$

Experimental Results of G.I.M.

- Apply G.I.M. to DC prediction residue of JPEG stream

  - Before encryption: 5.78 bits per DC coeff.
  - Apply G.I.M over [-63, 64]: +2.82 bits per coeff. overhead
  - Apply 2-subset overhead control: +1.53 bits per coeff. overhead

- Experimental results on MPEG-4 video: *overhead from 2%-30% depending on how many elements (MV, DC, AC coeff.) are encrypted*

Intra Bit Plane Shuffling

- Fine Granularity Scalability (FGS) achieved by bit-plane coding:
  - MPEG-4 FGS enhancement layer, EZW, …
- Proposed joint encryption and compression compatible with FGS
  - For a selected unit (such as a macroblock), shuffle bits within each of the significant bitplanes:
    - *e.g. "0 1 0 0 1 0 0 0 0" have 45 different shuffled versions*
  - # of FGS symbols (run-EOP) within each bitplane is preserved
- Analysis of expected compression overhead
  - Smaller than the overhead of index mapping approach

$$E(N_d|n_1) = \left(\prod_{k=0}^{d-1}(1-\frac{n_1}{n-k})\right)\frac{{n_1}^2}{n-d}$$

Expected # of occurrences that run of zeros before a "1" equals to d with total# of "1" being n_1

Results of Intra Bit Plane Shuffling

- Unencrypted base layer only 28.8 dB
- Unencrypted base + FGS enhancement layers 55 dB
- Base + encrypted FGS layers 26 dB
- Overhead: 7% for intra shuffling 3 bitplanes, 14.3% for G.I.M

Conclusions

- Identified unique issues with multimedia encryption
  - Identified a set of domains along the representation and communication process of multimedia where encryption can be applied
- Proposed and analyzed three encryption operations with a focus on communication and compression issues
  - Bit-stream approach has little bitrate overhead but limited flexibility
  - Moving encryption domain to upper levels:
    - *preserves standard compliance and allows more sophisticated intermediate processing be applied directly on the encrypted data*
  - Generalized index mapping with controlled overhead: can be applied to any scalar or vector symbols with a finite value range
  - Intra-bitplane shuffling: compatible with FGS enhancement layer coding