EE 6886: Topics in Signal Processing -- Multimedia Security System

Lecture 3: Multimedia Encryption

Ching-Yung Lin
IBM T. J. Watson Research Center
Hawthorne, NY 10532, USA

Outline -- Introduction

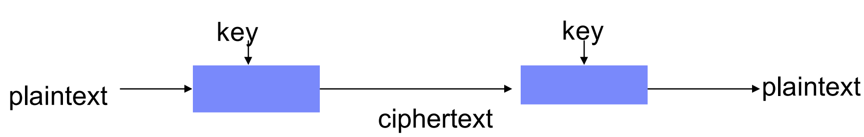
- Multimedia Security :
  - Multimedia Standards – Ubiquitous MM
  - Encryption and Key Management – Confidential MM
  - Watermarking – Uninfringible MM
  - Authentication – Trustworthy MM
- Security Applications of Multimedia:
  - Audio-Visual Person Identification – Access Control, Identifying Suspects
  - Surveillance Applications – Abnormality Detection
  - Media Sensor Networks – Event Understanding, Information Aggregation

Security Services (X.800)

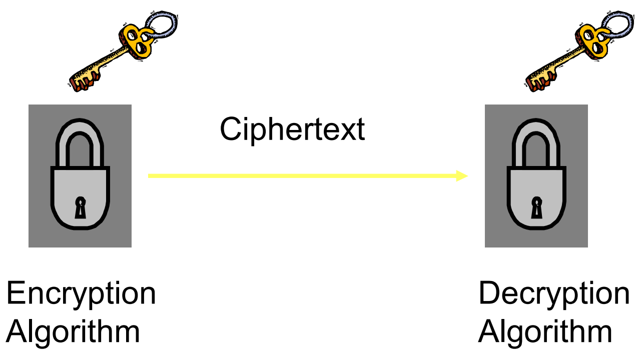
- Person Authentication
  - Assurance that communicating entity is the one claimed
- Access Control
  - Prevention of unauthorized use of a resource
- Data Confidentiality
  - Protection of data from unauthorized disclosure
- Data Integrity
  - Assurance that data received is as sent
- Non-Repudiation
  - Protection against denial by the parties in a communication

Conventional Encryption Algorithms

- Data Encryption Standard (DES)
  - The most widely used encryption scheme
  - DES is a block cipher – the plaintext is processed in 64-bit blocks
  - The key is 56-bits in length
  - Based on Feistel Cipher Structure
- Triple DES
  - Effective key length of 112/168 bits
- Advanced Encryption Standard (AES)
  - 128-bit data, 128/192/256-bit keys
  - Stronger & faster than Triple-DES
- Public key encryption: asymmetric key



Symmetric Encryption



Example: One-Time Pad

Message	0 1 0 0 0 1 0
Key	1 0 0 1 0 1 1
Encrypted Message = Message ⊕ Key	1 1 0 1 0 0 1

Basic Transformations on Block Data

- Substitution – (look-up table):
  - Specifies, for each of the 2^k possible values of the input, the k-bit output.
  - Suitable for short blocks of length, e.g., 8 bits.
- Permutation – (bit shuffle):
  - Specifies, for each of the k input bits, the output position to which it goes.
  - Needs k 2^k bits to specifies k-bit permutation.

Substitution and Permutation

- Substitution
- Message
- = 01001000110101
- Substitute 10 for 01

Permutation
- Message
  = 01001000110101

0 1 0 0 1 0 0 0 1 1 0 1 0 1

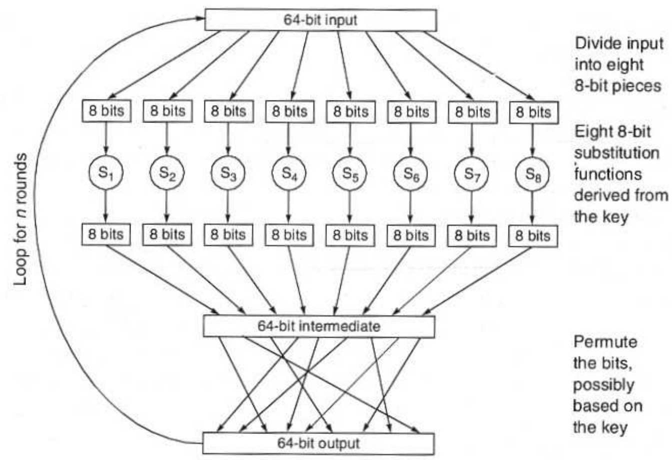
1 0 0 0 1 0 0 0 1 1 1 0 1 0

0 1 0 0 1 0 0 0 1 1 0 1 0 1

0 0 1 1 1 0 0 0 0 0 0 1 1 1

Block Cipher Example

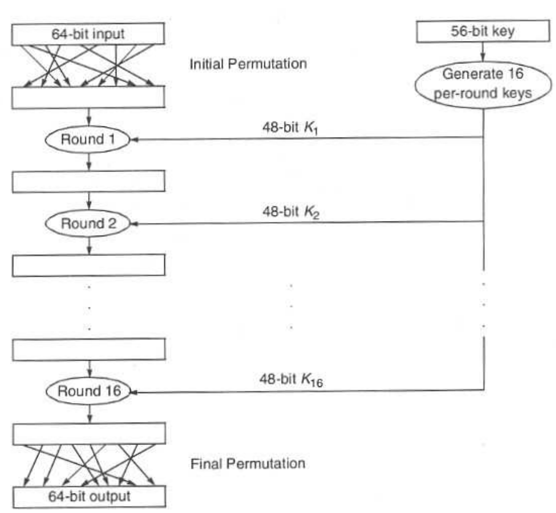
- Block cipher



E 6886 Topics in Signal Processing: Multimedia Security Systems

DES Structure

- DES structure



DES – Initial Permutation and Final Permutation

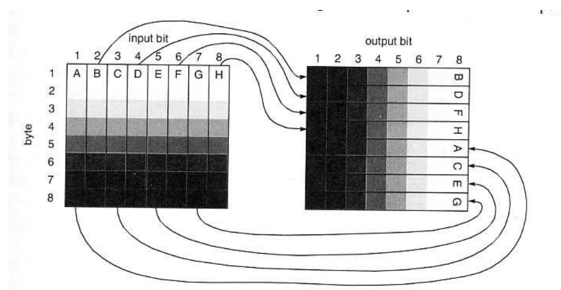
- Objectives: Make the output data more random
- Do not increase security level

Initial Permutation (IP)

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Final Permutation (IP^{-1})

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25



Key generation for each round

- DES keys

C_0

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36

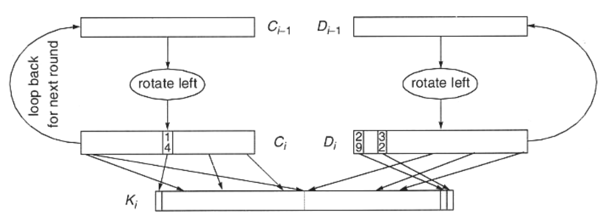
D_0

63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

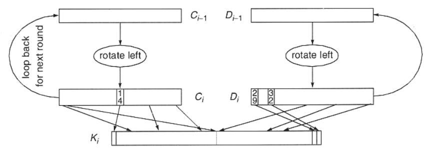
1	2	3	4	5	6	7
9	10	11	12	13	14	15
17	18	19	20	21	22	23
25	26	27	28	29	30	31
33	34	35	36	37	38	39
41	42	43	44	45	46	47
49	50	51	52	53	54	55
57	58	59	60	61	62	63

⇒

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4



Generating keys



- In rounds 1, 2, 9 and 16, it is a single-bit rotate left.
- In the other rounds, it is a two-bit rotate left
- The permutation of C_i that produces the left half of K_i is the folloing. Note that bits 9, 18, 22, and 25 are discarded.

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2

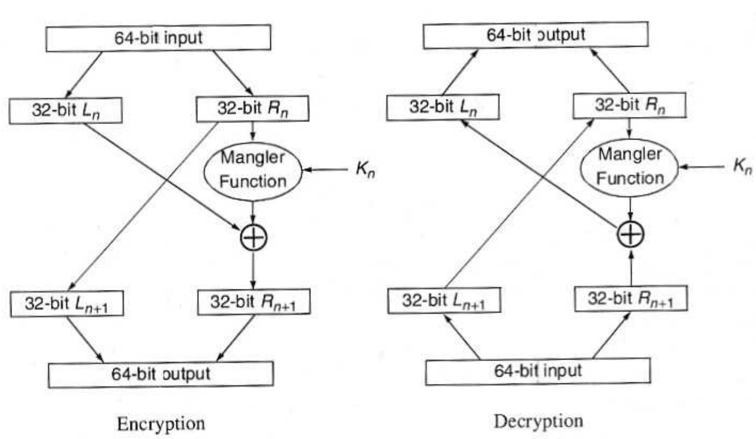
- The permutation of the rotated D_{i-1} that produces the right half of K_i is as follows (where the bits of the rotated D_{i-1} are numbered 29, 30, ..56, and bits 35, 38, 43, and 54 are discarded.

41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

- Each of the halves of K_i is 24 bits, so K_i is 48 bits long.

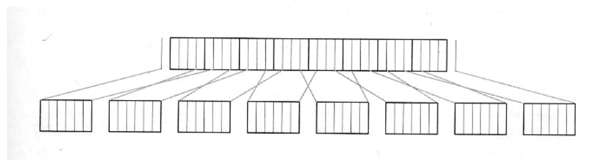
Encryption and Decryption in A DES Round

- Round

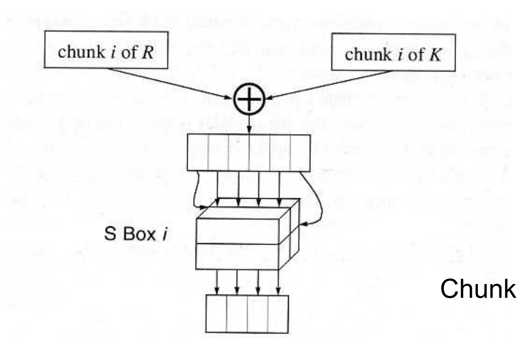


The Mangler Function and Chunk Transformation

- Main functions in DES



Mangler Function



Chunk Transformation

S Box Tables

- S Box Tables

Input bits 1 and 6 — Input bits 2 thru 5

↓	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	1110	0100	1101	0001	0010	1111	1011	1000	0011	1010	0110	1100	0101	1001	0000	0111
01	0000	1111	0111	0100	1110	0010	1101	0001	1010	0110	1100	1011	1001	0101	0011	1000
10	0100	0001	1110	1000	1101	0110	0010	1011	1111	1100	1001	0111	0011	1010	0101	0000
11	1111	1100	1000	0010	0100	1001	0001	0111	0101	1011	0011	1110	1010	0000	0110	1101

Input bits 7 and 12 — Input bits 8 thru 11

↓	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	1111	0001	1000	1110	0110	1011	0011	0100	1001	0111	0010	1101	1100	0000	0101	1010
01	0011	1101	0100	0111	1111	0010	1000	1110	1100	0000	0001	1010	0110	1001	1011	0101
10	0000	1110	0111	1011	1010	0100	1101	0001	0101	1000	1100	0110	1001	0011	0010	1111
11	1101	1000	1010	0001	0011	1111	0100	0010	1011	0110	0111	1100	0000	0101	1110	1001

Input bits 13 and 18 — Input bits 14 thru 17

↓	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	1010	0000	1001	1110	0110	0011	1111	0101	0001	1101	1100	0111	1011	0100	0010	1000
01	1101	0111	0000	1001	0011	0100	0110	1010	0010	1000	0101	1110	1100	1011	1111	0001
10	1101	0110	0100	1001	1000	1111	0011	0000	1011	0001	0010	1100	0101	1010	1110	0111
11	0001	1010	1101	0000	0110	1001	1000	0111	0100	1111	1110	0011	1011	0101	0010	1100

Input bits 19 and 24 — Input bits 20 thru 23

↓	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	0111	1101	1110	0011	0000	0110	1001	1010	0001	0010	1000	0101	1011	1100	0100	1111
01	1101	1000	1011	0101	0110	1111	0000	0011	0100	0111	0010	1100	0001	1010	1110	1001
10	1010	0110	1001	0000	1100	1011	0111	1101	1111	0001	0011	1110	0101	0010	1000	0100
11	0011	1111	0000	0110	1010	0001	1101	1000	1001	0100	0101	1011	1100	0111	0010	1110

S Box Tables --II

- S Box Tables

Input bits 25 and 30 — Input bits 26 thru 29

↓	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	0010	1100	0100	0001	0111	1010	1011	0110	1000	0101	0011	1111	1101	0000	1110	1001
01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1010	0011	1001	1000	0110
10	0100	0010	0001	1011	1010	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000	1110
11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1010	0100	0101	0011

Input bits 31 and 36 — Input bits 32 thru 35

↓	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	1100	0001	1010	1111	1001	0010	0110	1000	0000	1101	0011	0100	1110	0111	0101	1011
01	1010	1111	0100	0010	0111	1100	1001	0101	0110	0001	1101	1110	0000	1011	0011	1000
10	1001	1110	1111	0101	0010	1000	1100	0011	0111	0000	0100	1010	0001	1101	1011	0110
11	0100	0011	0010	1100	1001	0101	1111	1010	1011	1110	0001	0111	0110	0000	1000	1101

Input bits 37 and 42 — Input bits 38 thru 41

↓	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	0100	1011	0010	1110	1111	0000	1000	1101	0011	1100	1001	0111	0101	1010	0110	0001
01	1101	0000	1011	0111	0100	1001	0001	1010	1110	0011	0101	1100	0010	1111	1000	0110
10	0001	0100	1011	1101	1100	0011	0111	1110	1010	1111	0110	1000	0000	0101	1001	0010
11	0110	1011	1101	1000	0001	0100	1010	0111	1001	0101	0000	1111	1110	0010	0011	1100

Input bits 43 and 48 — Input bits 44 thru 47

↓	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	1101	0010	1000	0100	0110	1111	1011	0001	1010	1001	0011	1110	0101	0000	1100	0111
01	0001	1111	1101	1000	1010	0011	0111	0100	1100	0101	0110	1011	0000	1110	1001	0010
10	0111	1011	0100	0001	1001	1100	1110	0010	0000	0110	1010	1101	1111	0011	0101	1000
11	0010	0001	1110	0111	0100	1010	1000	1101	1111	1100	1001	0000	0011	0101	0110	1011

Permutation

16	7	20	21	29	12	28	17	1	15	23	26	5	18	31	10	2	8	24	14	32	27	3	9	19	13	30	6	22	11	4	25

More Secure Techniques

- Triple DES

$$C = E_{K3} (D_{K2} (E_{K1} (P)))$$

  - Effective key length of 112/168 bits

- Advanced Encryption Standard (AES)
  - 128-bit data, 128/192/256-bit keys
  - Stronger & faster than Triple-DES

Encrypt A Large Message

- Electronic Code Book (ECB)
- Cipher Block Chaining (CBC)
- k-Bit Cipher Feedback mode (CFB)
- k-Bit Output Feedback mode (OFB)

Electronic Code Book Encryption/Decryption

- ECB

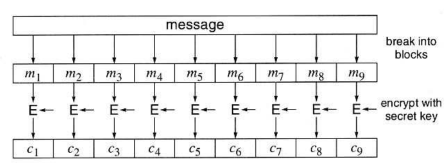


Figure 3-23. Electronic Code Book Encryption

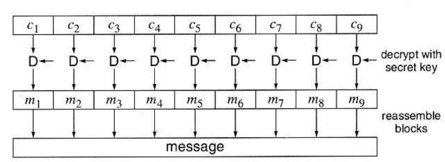


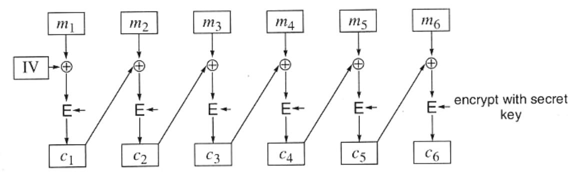
Figure 3-24. Electronic Code Book Decryption

Name	Position	Salary
Adams, John	President	78,964.31
Bush, Neil	Accounting Clerk	623,321.16
Hoover, J. Edgar	Wardrobe Consultant	34,445.22
Stern, Howard	Affirmative Action Officer	38,206.51
Woods, Rosemary	Audiovisual Supervisor	21,489.15

Block boundaries

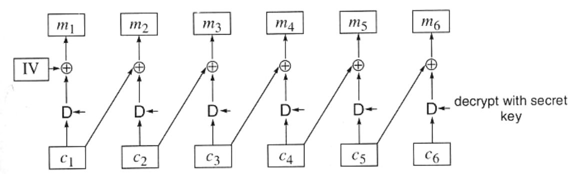
Cipher Block Chaining (CBC)

- CBC



Cipher Block Chaining Encryption

Decryption is simple because ⊕ is its own inverse.



Cipher Block Chaining Decryption

Attack of CBC

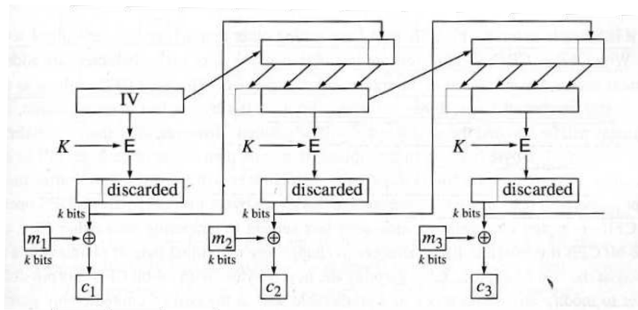
- Modifying Ciphertext Block

```
Tacker, Jo A          System Security Officer        54,122.10
```

```
Tacker, Jo A          System Security Of#f8Ts9(*     74,122.10
```

Output Feedback Mode (OFB)

- Output feedback mode acts like a pseudorandom number generator. A message is encrypted by XOR it with the pseudirandom stream generated by OFB.
- One-Time Pad: a long random string used to encrypt a message with a simple XOR operation.

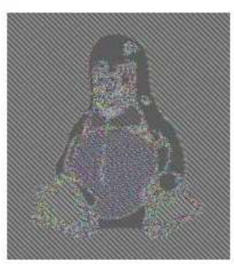


k-bit OFB

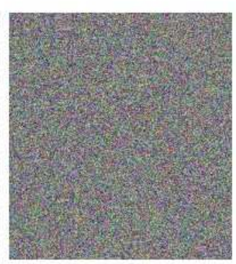
Example of Image Encryption

- Image Encryption

Original



Encrypted using ECB mode



Encrypted securely

Streaming media encryption

- Streaming digital media –
  - digital media being transmitted through a network, from a server to a client, in a streaming or continuous fashion. The streamed data is transmitted by a server application and received and displayed in real-time by client applications.
  - Applications start displaying video or playing back audio as soon as enough data has been received.

  - In a packet switch network, the server breaks the media into packets that are routed and delivered over the network. At the receiving end, a series of time-stamped packets, called stream, is reassembled by the client and the media is played as it comes in.

Challenges of streaming media encryption

- Real time constraint
- Potential cost increase
- Potential bit rate increase
- The rate variation challenge
- Dynamic network conditions
- Transcoding challenge
- Other challenges

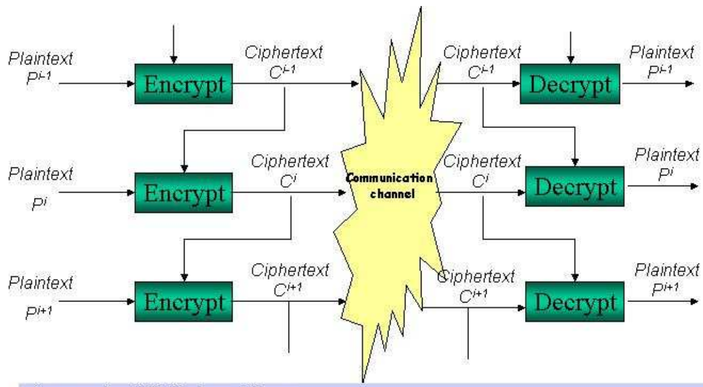
Summary of system requirement

- should be secure but low cost in implementation
- should not reduce the playback quality of the streaming media
- should sustain current and new heterogeneous environment
- should be extendable from PCs to mobile devices
- should be easily renewable
- should be able to preserve entertainment like experience

Block cipher and stream cipher

- SC (CBC)
  - convert plaintext to ciphertext 1 bit (1 block) at a time, offer means to decrypt earlier portion of the ciphertext without the availability of the later portion of it.
- Provide scalability - intuitively
  - prioritize data, bit-by-bit or block-by-block, and encrypt the bitstream using SC or CBC

CBC



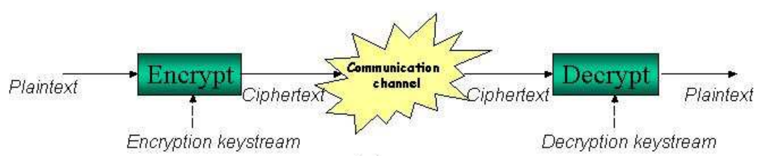
A sample CBC algorithm

Encryption: $C^i = \mathrm{Enc}(P^i \oplus C^{i-1}, K)$

Decryption: $P^i = C^{i-1} \oplus \mathrm{Dec}(C^i, K)$

Cipher Block Chaining

SC



(a)

A sample SC algorithm

Encryption: *For* $i=[1,N]$

$C_i = P_i \oplus K_i$

Decryption: *For* $i=[1,N]$

$P_i = C_i \oplus K_i$

$= P_i \oplus K_i \oplus K_i$

$= P_i$

(b)

Stream Cipher

A scalable streaming media encryption scheme that enables transcoding without decryption

Wee, S. J., Apostolopoulos, J.G., Secure scalable streaming enabling transcoding without decryption, in *Proc. IEEE Int. Conf. Image Processing*, Oct. 2001

- utilizes CBC or SC to achieve progressive decryption ability
  1. the input video frame is segmented into regions.
  2. each region is coded into scalable video data and header data
  3. the scalable video data is encrypted with CBC or SC based progressive encryption, which allows truncation of the scalable video data stream and quality adaptation in accordance
  4. secure scalable packets are created by combining the unencrypted header data with the progressively encrypted scalable video data

Format compliant encryption

- For mobile applications
  - Recall: mobile device – limited resource
    - Selective encryption – computational complexity
      - Security level → target application
    - Encryption algorithm selection – computation power
  - Compression: wireless communication – limited bandwidth; multimedia data stream – large
    - If selective encryption is not done smartly

    → Compression + encryption = bitrate increase

Encryption and Compression

Content → Encryption → Compression → Secured Content

High complexity/ poor compression

Content → Compression → Encryption → Secured Content

Naïve approach/ High complexity

Content → Transform → Joint Encryption Compression → Secured Content

Low complexity/ compromised compression

Example: MPEG I frame intra block shuffling

24	20	18	17	10	8	4	1
21	16	13	9	6	3	0	0
15	10	4	2	0	0	0	0
10	7	3	0	0	0	0	0
6	2	0	0	0	0	0	0
4	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

shuffle →

transpositions

7	0	1	0	4	8	0	10
0	6	0	0	3	0	0	4
0	0	4	2	0	15	0	0
3	0	10	0	13	0	0	0
6	2	0	0	0	0	21	0
9	0	10	0	0	24	0	0
0	16	0	18	0	0	0	0
0	0	0	0	17	0	0	20

0s – clustered together → not any more

bitrate ↑

Selective Encryption

- To reduce the amount of processing overhead/ delay
  - E.g., only I frame/blocks encrypted (Maples & Spanos ’95, Mayer & Gadegast ’95)
  - Sign bits, Motion Vectors (Shi & Bhargava ’98, Zeng & Lei ’99, Wen et. al. ’01)
  - Privacy /security low due to information leakage
    - Useful for applications that focus on introducing quality degradation.

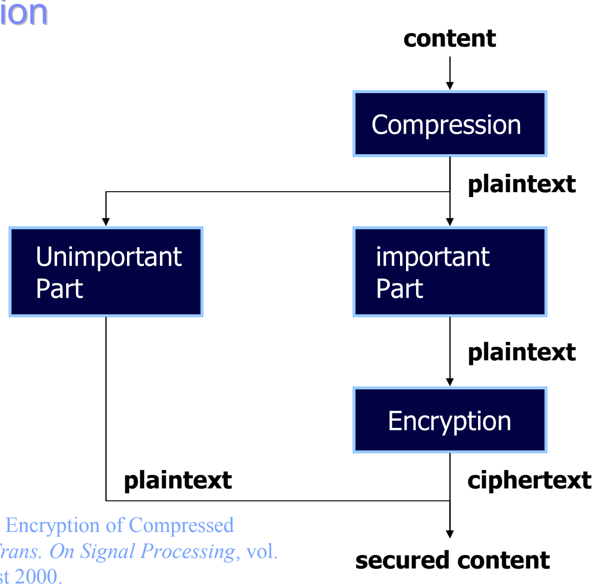
Joint Encryption and Compression

- To achieve improved overall performance:
  - E.g., Zigzag-Permutation (Tang ’96)
    - Simple, but significantly lower compression ratio.
    - Local scrambling -> spatial energy distribution unchanged -> less effective scrambling
  - Spatially shuffle coefficients/ MVs (Zeng & Lei ‘099)
    - Coefficient block shuffling, block rotation, and coefficient shuffling within a dubband segment.
    - Local statistics largely unchanged -> good coding efficiency
    - Global spatial configuration changed -> good security
- In both schemes, the resultant encrypted bitstream conforms to the compression format ➔ Decodable

Partial Encryption

Cheng and Li (2000)

- Quadtree compression
- Wavelet compression based on zero trees



H. Cheng and X. Li, "Partial Encryption of Compressed Images and Video," *IEEE. Trans. On Signal Processing*, vol. 48(8), pp. 2439-2451, August 2000.

Format compliant encryption

Wen, J., Severa, M., Zeng, W., Luttrell, M., Jin, W., "A Format-compliant Configurable Encryption Framework for Access Control of Video", *IEEE Trans. Circuits & Systems for Video Technology,* 2002

The questions to ask when design MM encryption

- Not only 'faster?'
- But also 'bitrate increase?' – format compliant

- Algorithm
  - Bitstream partition, extract bits that are important.
  - Concatenate extracted bits.
  - Choose a public key or a private key encryption algorithm, such as DES or AES.
  - Encrypt the concatenated bits. For VLC coded bitstreams, encrypt the indices of codewords from the code table instead, and then map it back to codewords in code table.
  - Put the encrypted bits back into their original positions.

Example: MPEG I frame intra block shuffling

-- format compliant

24	20	18	17	10	8	4	1
21	16	13	9	6	3	0	0
15	10	4	2	0	0	0	0
10	7	3	0	0	0	0	0
6	2	0	0	0	0	0	0
4	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

shuffle

7	0	1	0	4	8	0	10
0	6	0	0	3	0	0	4
0	0	4	2	0	15	0	0
3	0	10	0	13	0	0	0
6	2	0	0	0	0	21	0
9	0	10	0	0	24	0	0
0	16	0	18	0	0	0	0
0	0	0	0	17	0	0	20

shuffle, FC

24	20	18	17	10	8	4	1
21	16	13	9	6	3	0	0
15	10	4	2	0	0	0	0
10	7	3	0	0	0	0	0
6	2	0	0	0	0	0	0
4	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Example: MPEG I frame intra block shuffling

-- format compliant

24	20	18	17	10	8	4	1
21	16	13	9	6	3	0	0
15	10	4	2	0	0	0	0
10	7	3	0	0	0	0	0
6	2	0	0	0	0	0	0
4	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

shuffle

7	0	1	0	4	8	0	10
0	6	0	0	3	0	0	4
0	0	4	2	0	15	0	0
3	0	10	0	13	0	0	0
6	2	0	0	0	0	21	0
9	0	10	0	0	24	0	0
0	16	0	18	0	0	0	0
0	0	0	0	17	0	0	20

shuffle, FC

24	21	18	17	10	8	4	1
20	16	10	7	2	0	0	0
15	13	4	2	0	0	0	0
10	9	3	0	0	0	0	0
6	6	0	0	0	0	0	0
4	3	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Format compliant encryption

Wen, J., Severa, M., Zeng, W., Luttrell, M., Jin, W., "A Format-compliant Configurable Encryption Framework for Access Control of Video", *IEEE Trans. Circuits & Systems for Video Technolgy*, 2002

The questions to ask when design MM encryption

- Not only 'faster?'
- But also 'bitrate increase?' – format compliant

- Algorithm
  - Bitstream partition, extract bits that are important.
  - Concatenate extracted bits.
  - Choose a public key or a private key encryption algorithm, such as DES or AES.
  - Encrypt the concatenated bits. For VLC coded bitstreams, encrypt the indices of codewords from the code table instead, and then map it back to codewords in code table.
  - Put the encrypted bits back into their original positions.

E 6886 Topics in Signal Processing: Multimedia Security Systems

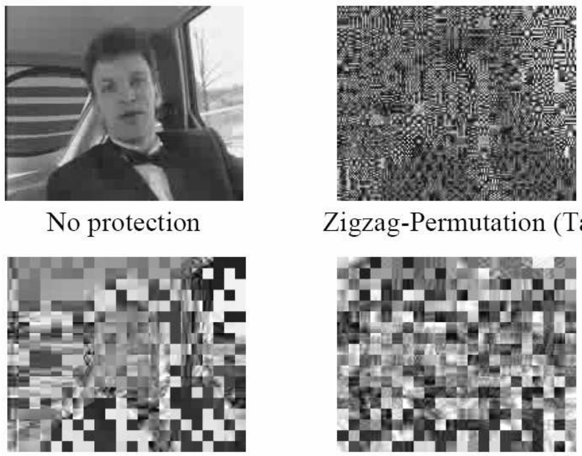
Example, VLC:

- Assume codeword table
  - 0 10 110 111
  - A two-codeword concatenation: 010
  - Encrypt 001 → not in the codeword table
  - $2^n=N$ (N codeword, n-bit index)

	010110111	0 10 110 111	→	C
①	00011011	00 01 10 11	→	S
②	10011100	10 01 11 00	→	S'
③	110101110	110 10 111 0	→	C'

④ Put back to the content bitstream in place of C

Encryption example



No protection — Zigzag-Permutation (Tang'96)

Sign encryption — Spatially shuffle coefficients (Zeng & Lei'99)

Error resilient capability

- Wireless – error prone channels
- Encryption – traditionally error prone
- Streaming, real time media – time limitation, infeasible to retransmit
- Highly correlated fields: e.g., motion vector – high correlation in neighboring fields

Good News: use to design an error resilient enc. System

Bad News: error concealment attacks – leakage of non-encrypted fields used as prior knowledge and attempt to obtain certain information from the encrypted video content

Tradeoffs – design considerations

- Security (error concealment attack)
  ↔ error resilient
- Security
  ↔ system complexity
- Security
  ↔ user convenience
- Governed by the targeted application and the security requirement and system capability

References

- C. Kaufman, R. Perlman, M. Speciner, “Network Security: Private Communication in a Public World,” Prentice-Hall
- Wee, S. J., Apostolopoulos, J.G., Secure scalable streaming enabling transcoding without decryption, in *Proc. IEEE Int. Conf. Image Processing*, Oct. 2001
- Wen, J., Severa, M., Zeng, W., Luttrell, M., Jin, W., “A Format-compliant Configurable Encryption Framework for Access Control of Video”, *IEEE Trans. Circuits & Systems for Video Technology,* 2002
- H. Cheng and X. Li, “Partial Encryption of Compressed Images and Video,” *IEEE. Trans. On Signal Processing*, vol. 48(8), pp. 2439-2451, August 2000.

Acknowledgements: Thanks to
Dr. Heather Yu (Panasonic), Prof. Wenjun Zhen (Univ. of Missouri) and
Prof. Deepa Kundur (Texas A& M Univ.) for courtesy of slides used in this lecture.

Homework Assignment #1 (due Feb. 22)

1. [30 pts] Generate an image format converter which reads raw images (e.g., PPM format) and writes them as BMP format. The generated BMP file has to be readable by general image viewer.

2. [40 pts] Perform 8x8 DCT transform on the images. Try to following operations. And, then perform inverse DCT transform. Compare the differences (e.g., show PSNR values).
   1. Leave only the DC values. Delete all AC values.
   2. Leave the DC values and 5 AC values (in the zigzag order).
   3. Quantize the DCT coefficients using the standard JPEG quantization table.

3. [30 pts] Perform encryption on the DCT coefficients of images by the following shuffling methods and then perform inverse DCT on them. Observe the results. You can use any encryption method (e.g., randomly generate a look-up table for shuffling)
   1. Encrypting the DC coefficients of blocks. Don't change the AC coefficients.
   2. Encrypting the AC coefficients within each block. Any shuffling method can be used.
   3. Encrypting the AC coefficients based on the format compliant encryption as described in the slides (Wen et. al. 2002)

4. [Extra 20 pts] Implement the DES encryption method and perform ECB and CBC encryption