SJSU SAN JOSÉ STATE UNIVERSITY

Lesson 4 – Block Ciphers 2

Yan Chen

CS166 Fall 2024

- Block ciphers uses the idea of “codebook”
  - Each block of plaintext has a corresponding block of ciphertext
  - Key is used to generate codebooks
  - Change key = switch the codebook
  - “Electronic” version of codebook (“book” not fixed)
- Round function F: encrypt the plaintext to ciphertext
  - For each round, input is key and output of previous round
- Notations
  - P = plaintext block, C = ciphertext block, K = key
  - Encrypt: C = E(P, K); decrypt: P = D(C, K)

- Feistel cipher: general block cipher design principle
  - A type of block cipher, not a specific block cipher
- Feistel cipher encryption
  - Split plaintext block into left and right halves: $P = (L_0, R_0)$
  - For each round $i = 1, 2, \ldots, n$, compute

    $L_i = R_{i-1}$; $R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$, where K_i is called "subkey"

    After n rounds, ciphertext block $C = (L_n, R_n)$
- To decrypt, "inverse" the encryption
- Round function does not need to be invertible
  - Since swap and $\oplus$ both are invertible

	DES	AES	TEA
Full Name	Data Encryption Standard	Advanced Encryption Standard	Tiny Encryption Algorithm
Block Size	64 bits (Split into 32 & 32)	128 bits (4 * 4 byte matrix)	64 bits (Split into 32 & 32)
Key Length	56 bits (48 for subkeys)	128/192/256 bits	128 bits
# Of Rounds	16	10 - 14	>=32
Feistel?	Yes	No	Almost (use +/- instead of XOR)
Round Function Complexity	Relatively simple	Complex (highly mathematical)	Lightweight & simple
Other Notes	8 S-boxes map 6 bits to 4 bits	3 layers & 4 functions	Impressive performance
Satisfied Kerckhoffs' Principle?	Kind of not (NSA secretly involved)	Yes	Yes
Status Of Lifecycle	Alive with improved version 3DES	Alive (mandatory for U.S. gov. apps)	Alive with several improved versions

- Security of DES
  - Depends heavily on S-boxes (everything else is linear)
  - 35+ years of intense analysis has revealed no back door
  - All known attacks were essentially exhaustive key search
  - But 56 bit DES key is too small today –using 3DES instead
- 3 Layers & 4 functions on 4 * 4 byte matrix in AES
  - Nonlinear layer: ByteSub – substitute each byte
  - Linear mixing layer: MixColumn – * a constant 4 * 4 matrix;

    ShiftRow – cyclic shift in each row
  - Key addition layer : AddRoundKey – XOR with subkey

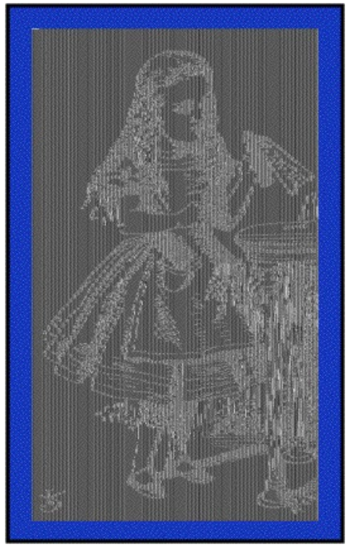
- DES, AES, etc. are different ways to encrypt 1 block
- How to encrypt multiple blocks?
  - Encrypt each block using a new key (not practical)
  - Encrypt each block independently using the same key
  - Encrypt each block dependently using the same key
- Encryption modes: ways to encrypt multiple blocks
  - NIST defined 5 different encryption modes
  - (Cover in class) ECB, CBC, CTR
  - (Will not cover) Cipher Feedback (CFB), Output Feedback
  - These modes are for all block ciphers in general

- Electronic Codebook (ECB): encrypt/decrypt each block independently with the same key
  - ➢ Simplest mode
  - ➢ "Electronic" codebook without additive
- Given plaintext blocks $P_0, P_1, P_2, \ldots, P_n$

Encrypt	Decrypt
$C_0 = E(P_0, K)$	$P_0 = D(C_0, K)$
$C_1 = E(P_1, K)$	$P_1 = D(C_1, K)$
$C_2 = E(P_2, K)$	$P_2 = D(C_2, K)$
...	...
$C_n = E(P_n, K)$	$P_n = D(C_n, K)$

- ECB weakness: same plaintext yields same ciphertext
  - If $P_i = P_j$, then $C_i = C_j$ and if $C_i = C_j$, then $P_i = P_j$
  - If the ciphertext is large enough, Trudy can get statistics
  - E.g.: Alice’s uncompressed image, and ECB encrypted

Encrypt under ECB mode



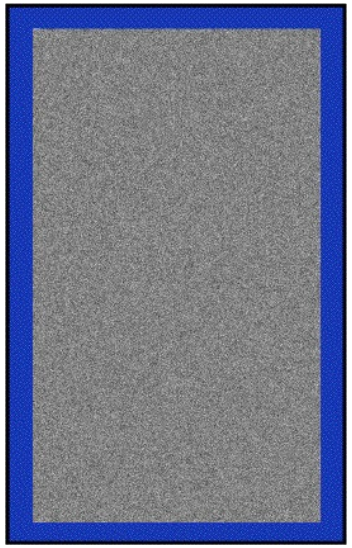
- So, ECB only good for small number of blocks
  - Example application: encrypt keys

- Cipher Block Chaining (CBC): blocks are "chained"
  - Encrypt/decrypt each block dependently with the same key
  - "Electronic" codebook with additive
- Given plaintext blocks $P_0, P_1, P_2, \ldots, P_n$
  - And a random and PUBLIC initialization vector (IV)

Encrypt	Decrypt
$C_0 = E(IV \oplus P_0, K)$	$P_0 = IV \oplus D(C_0, K)$
$C_1 = E(C_0 \oplus P_1, K)$	$P_1 = C_0 \oplus D(C_1, K)$
$C_2 = E(C_1 \oplus P_2, K)$	$P_2 = C_1 \oplus D(C_2, K)$
…	…
$C_n = E(C_{n-1} \oplus P_n, K)$	$P_n = C_{n-1} \oplus D(C_n, K)$

- Using CBC, same plaintext yields different ciphertext
  - ➢ E.g.: Alice’s uncompressed image, and CBC encrypted

Encrypt under CBC mode →



- What if 1 ciphertext block is wrong?
  - ➢ Will automatically recover from errors (Why?)
  - What if we used ECB instead? Can it be auto-recovered also?

- Counter (CTR): increment the random IV
  - Encrypt/decrypt each block independently with the same key
  - Popular for random access
  - Use block cipher like a stream cipher
- Given plaintext blocks P0, P1, ..., Pn
  - And a random and PUBLIC initialization vector (IV)

Encrypt	Decrypt
$C_0 = P_0 \oplus E(IV, K)$	$P_0 = C_0 \oplus E(IV, K)$
$C_1 = P_1 \oplus E(IV + 1, K)$	$P_1 = C_1 \oplus E(IV + 1, K)$
...	...
$C_n = P_n \oplus E(IV + n, K)$	$P_n = C_n \oplus E(IV + n, K)$

- Recall: confidentiality vs. integrity
  - ➢ Confidentiality: prevent unauthorized reading of information
  - ➢ Integrity: detect unauthorized writing of information
- Does encryption provide confidentiality?
  - ➢ Yes, that's what encryption is for –hide the information
- Does encryption provide integrity?
  - ➢ NOT if we only use it alone...
  - ➢ e.g.: Trudy can give a wrong key for one-time pad;

    Trudy can change the order of blocks in ECB, etc.
- Need to use MAC for integrity

- Message Authentication Code (MAC)
  - Used for data integrity, not confidentiality
  - Serves as a cryptographic checksum for data
- MAC is computed as CBC residue
  - The calculation is the same as CBC encryption
  - But only save the last cipher block, and call it "MAC"
  - $C_0 = E(IV \oplus P_0, K)$

    $C_1 = E(C_0 \oplus P_1, K)$

    ...

    $C_{N-1} = E(C_{N-2} \oplus P_{N-1}, K) = MAC$

- Alice send Bob the IV, the P’s and the MAC
- Bob do the same computation to verify
- Example: suppose Alice has 4 plaintext blocks P_0 to P_3
  - She computes MAC: $C_0 = E(IV \oplus P_0, K)$, $C_1 = E(C_0 \oplus P_1, K)$, $C_2 = E(C_1 \oplus P_2, K)$, $C_3 = E(C_2 \oplus P_3, K) = MAC$
  - Alice sends IV, P_0, P_1, P_2, P_3, and MAC to Bob
  - Suppose Trudy change P_1 to X
  - When Bob computes MAC, he would get a wrong MAC:

    $C_0 = E(IV \oplus P_0, K)$, $C_1' = E(C_0 \oplus X, K)$,

    $C_2' = E(C_1' \oplus P_2, K)$, $C_3' = E(C_2' \oplus P_3, K) = MAC' \neq MAC$!

- MAC works since error propagates into MAC
  - ➢ Trudy cannot make the same MAC without knowing K
  - ➢ Note that error does NOT propagate when decrypt in CBC
- Encryption is for confidentiality, MAC is for integrity
- Can we achieve both confidentiality AND integrity?
  - ➢ NOT with the same work –need extra work!
  - ➢ Example: Encrypt with one key, MAC with another key
  - Why not use the same key?
  - ➢ Confidentiality and integrity with same work as one encryption is a research topic

Stream Cipher vs. Block Cipher

- Symmetric key: use same key to encrypt and decrypt
  - Encryption algorithm needs to be invertible: the encrypted message can be decrypted using the same key
  - Usually, $\oplus$ does the magic: if $C = P \oplus K$ then $P = C \oplus K$

	Stream Ciphers	Block Ciphers
Idea from	One-time pad	Codebook
Crucial Algorithm	Keystream generation	Round function
Examples	A5/1, RC4, etc.	DES, AES, TEA, etc.
Encryption/Decryption	$C = P \oplus K$ $P = C \oplus K$	Based on the cipher & mode (ECB, CBC, CTR)
Application	Hardware	Software
Status Of Lifecycle	Alive but not popular anymore	Alive and become popular

Uses of Symmetric Key Ciphers

- Symmetric key crypto mainly used for confidentiality
  - ➢ Transmitting data over insecure channel
  - ➢ Secure storage on insecure media
- MAC used for integrity
- Other usages to be covered
  - ➢ Anything you can do with a hash function (~Lesson 8)
  - ➢ Authentication protocols (~Lesson 19 or 20)

- Public key ciphers
  - Introduction
  - Math preliminaries – prime & mod
  - RSA

- Block Cipher Modes
  - ECB
  - CBC
  - CTR
- MAC

- Draw diagrams to illustrate encryption/decryption in CBC mode
- Suppose instead of $C_n = P_n \oplus E(IV + n, K)$ for CTR, we used

$$C_n = P_n \oplus E(K, IV + n)$$

  - Is this secure? Why or why not?
- How to do random access on data encrypted in CBC mode?
  - Compared to CTR mode, is it better or not? If better, explain the advantages; if not, explain the disadvantages
- Given a MAC value X and the key K, but not the original message, how can you construct a message M that also has its MAC equal to X?

References

- Stallings, William, “Cryptography and Network Security, Principles and Practice, 6th ed.,” Pearson, USA, 2014
- Paar, Christof, “Understanding Cryptography,” Faller, Berlin, Germany, 2010
- Stamp, Mark and Low, Richard M., "Applied Cryptanalysis: breaking ciphers in the real world," John Wiley & Sons, Inc., New Jersey, USA, 2007