بسم الله الرحمن الرحیم

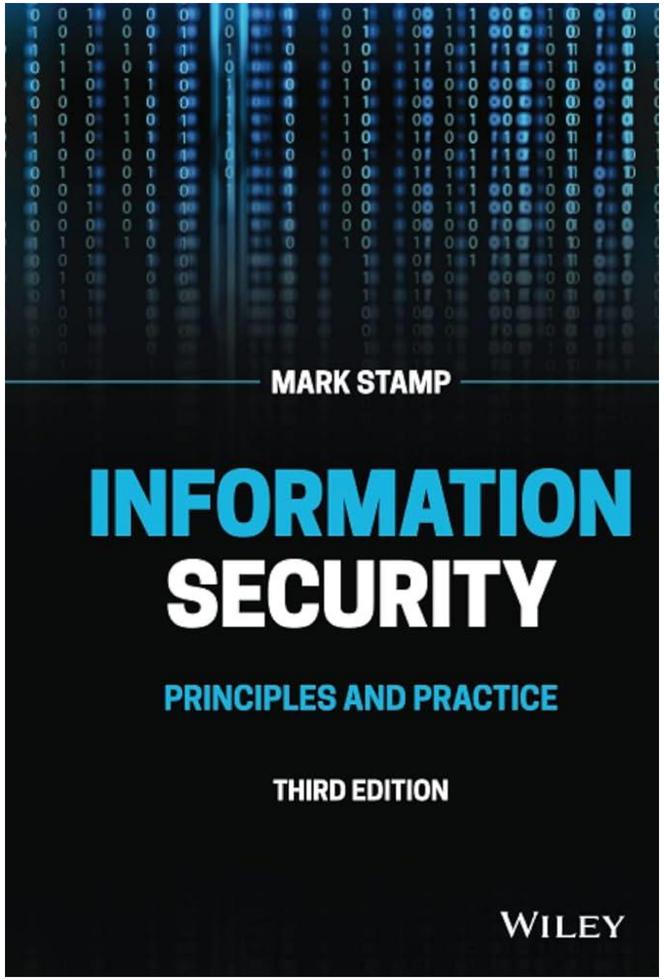
مبانی رایانش امن

جلسه ۶

مجتبی خلیلی
دانشکده برق و کامپیوتر
دانشگاه صنعتی اصفهان

IUT-ECE

◄ فصل سوم کتاب



CBC Mode

- Blocks are "chained" together
- A random initialization vector, or IV, is required to initialize CBC mode
- IV is random, but not secret

Encryption

$C_0 = E(IV \oplus P_0, K),$
$C_1 = E(C_0 \oplus P_1, K),$
$C_2 = E(C_1 \oplus P_2, K), \ldots$

Decryption

$P_0 = IV \oplus D(C_0, K),$
$P_1 = C_0 \oplus D(C_1, K),$
$P_2 = C_1 \oplus D(C_2, K), \ldots$

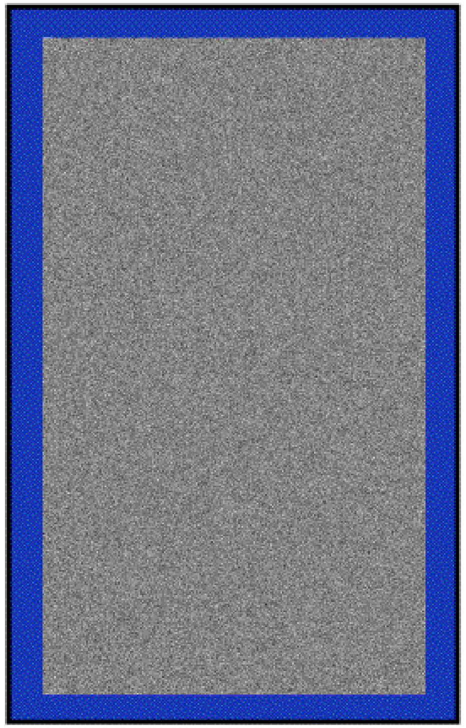
CBC Mode

- Identical plaintext blocks yield different ciphertext blocks — this is very good!
- But what about errors in transmission?
  - If C_1 is garbled to, say, G then

    $P_1 \neq C_0 \oplus D(G, K)$, $P_2 \neq G \oplus D(C_2, K)$
  - But $P_3 = C_2 \oplus D(C_3, K)$, $P_4 = C_3 \oplus D(C_4, K)$, …
  - Automatically recovers from errors!

Alice Likes CBC Mode

- Alice's uncompressed image, Alice CBC encrypted

- Why does this happen?
- Same plaintext yields different ciphertext!

Counter Mode (CTR)

- CTR is popular for random access
- Use block cipher like a stream cipher

Encryption

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

$C_1 = P_1 \oplus E(IV+1, K)$,

$C_2 = P_2 \oplus E(IV+2, K)$,…

Decryption

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

$P_1 = C_1 \oplus E(IV+1, K)$,

$P_2 = C_2 \oplus E(IV+2, K)$,…

- Note: CBC also works for random access

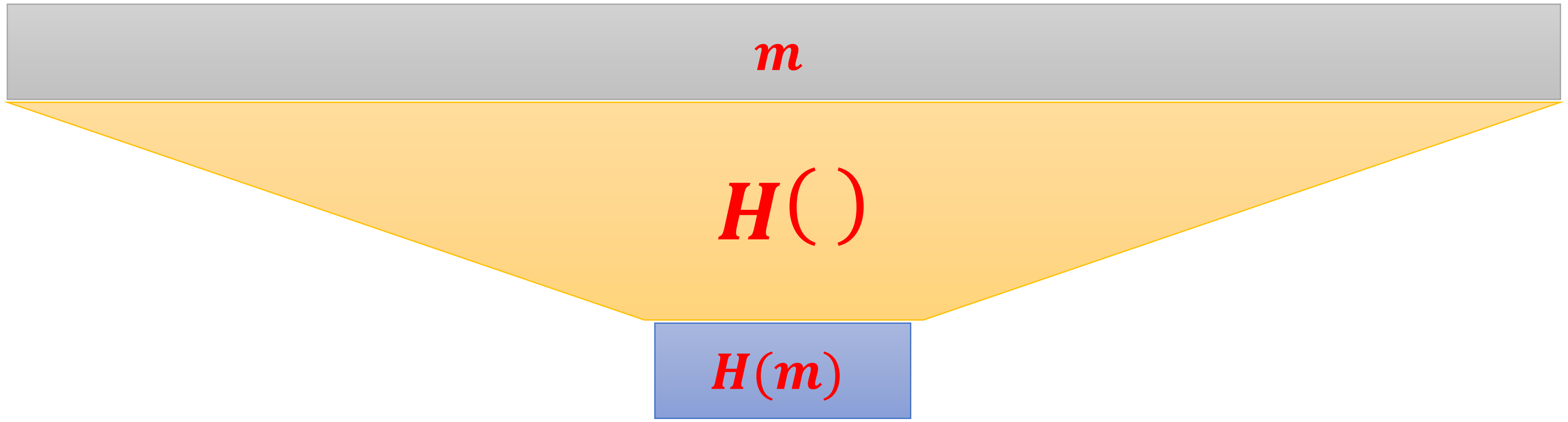
Modes of Operation

- Many modes — we discuss 3 most popular
- Electronic Codebook (**ECB**) mode
  - Encrypt each block independently
  - Most obvious approach, but a **bad** idea
- Cipher Block Chaining (**CBC**) mode
  - "Chain" the blocks together
  - More secure than ECB, virtually no extra work
- Counter Mode (**CTR**) mode
  - Block ciphers acts like a stream cipher
  - Popular for random access

Data Integrity

- **Integrity** — detect unauthorized writing (i.e., detect unauthorized mod of data)
- Example: Inter-bank fund transfers
  - Confidentiality may be nice, integrity is ***critical***
- Encryption provides **confidentiality** (prevents unauthorized disclosure)
- Encryption alone does **not** provide integrity
  - One-time pad, ECB cut-and-paste, etc., etc.

Crypto Hash Function



Crypto Hash Function

- Crypto hash function h(x) must provide
  - **Compression** — output length is small
  - **Efficiency** — h(x) easy to compute for any x
  - **One-way** — given a value y it is infeasible to find an x such that h(x) = y
  - **Weak collision resistance** — given x and h(x), infeasible to find y ≠ x such that h(y) = h(x)
  - **Strong collision resistance** — infeasible to find ***any*** x and y, with x ≠ y such that h(x) = h(y)
- Lots of collisions exist, but hard to find ***any***

Popular Crypto Hashes

- **MD5** — invented by Rivest (of course…)
  - 128 bit output
  - MD5 collisions easy to find, so it's broken
- **SHA-1** — A U.S. government standard, inner workings similar to MD5
  - 160 bit output
- Many other hashes, but MD5 and SHA-1 are the most widely used
- Hashes work by hashing message in blocks

Crypto Hash Design

- Desired property: avalanche effect
  - Change to 1 bit of input should affect about half of output bits
- Crypto hash functions consist of some number of rounds
- Want security and speed
  - "Avalanche effect" after few rounds
  - But simple rounds
- Analogous to design of block ciphers

MD5 and SHA-1

- MD5 invented by Rivest, SHA-1 a U.S. government standard
- Most popular crypto hash algorithms were (are?) MD5 and SHA-1
  - 128 bit output MD5, 160 for SHA-1
  - Both of these are considered ***broken***
- Collision attack on MD5 in 2004
- Collision attack on SHA-1 recently
- Both MD5 and SHA-1 use similar algorithm

SHA-3

- Secure Hash Algorithm 3
  - Why 3? SHA-1 was a previous standard and SHA-2 is a family of algorithms
- SHA-3 developed via open competition
  - Very similar to AES competition
- Released by NIST in 2015
  - Internally, SHA-3 is completely different from MD5 and SHA-1