Basic Cryptography

(Lecture 3 and 4 slides)

Table of Contents:

Background Noise: Youtube

Block Ciphers

They are mainly used for security & authentication services. Also for *symmetric key* encryption.

> Why? FAST! and easy to compute (for computers not us lol)

Stream Ciphers

These ciphers generate a sequence of random bit values (KEY) then use those bits in a XOR operation with the plaintext to produce the ciphertext.

> Why? EVEN FASTER!

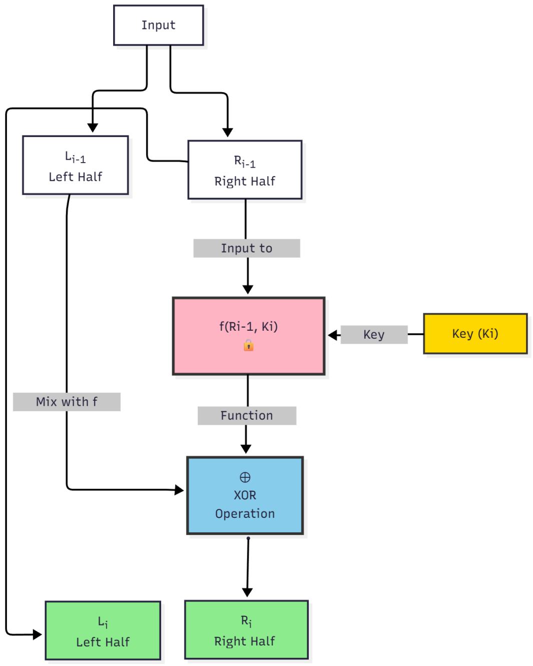
Why stream cipher suck?

1. Every plaintext must have a unique KEY stream (*Keystream reuse attack*).
2. Integrity issues, if the cipher is modified then you will get the wrong plaintext (*Ciphertext modification attack*).

   Ok now forget about stream ciphers they are never mentioned again.

Block Cipher Principles

Feistel Cipher Structure: the structure used by most of *symmetric* block ciphers. It's when you divide the input into right and left. This implements the **Diffusion & Confusion** mechanism.



DES (Data Encryption Standard)

```
You can skip this table it has specs only
```

Feature	DES	2DES	3DES
Key Size	56 bits	112 bits (2×56)	168 bits (3×56) or 112 bits (2-key)
Block Size	64 bits	64 bits	64 bits
Encryption Process	Single DES	Two DES encryptions	Encrypt-Decrypt-Encrypt
Rounds	16	32 (16×2)	48 (16×3)
Security Level	Broken	Vulnerable to MITM	Deprecated but relatively secure
Speed	Fast	2× slower	3× slower
Effective Security	56 bits	~57 bits	~112 bits
Number of keys	1	2	3 (could be 2 if E-D-E is used)

What is the significance of using D round in the middle for 3DES (E-D-E) ??

- No Mathematical Reason - Pure Compatibility ;D

Possible Attacks

Attack Type	DES	2DES	3DES
Brute Force	Feasible in hours	Impractical	Impossible
Meet-in-the-Middle	N/A	Effective (~2^57 ops)	Requires gazillion years
Differential Cryptanalysis	Theoretical weakness	Inherited from DES	Inherited from DES
Linear Cryptanalysis	Theoretical weakness	Inherited from DES	Inherited from DES

```
2DES: Meet-in-the-Middle attack reduces security to ~57 bits instead of 112 bits. Never
adopted as standard.
```

AES (Advanced Encryption Standard)

128-bit blocks, 128/192/256-bit keys.

> Strength of 3-DES, EFFICIENCY much higher

AES perks?

1. resistance against known attacks
2. SPEED
3. SIMPLICTY

   I didnt write how the cipher works got bored tbh.

Modes of Operation

> Electronic Code-Book (ECB) ≣ Cipher Block Chaining (CBC) ≣ Counter (CTR)

These modes of operation usually use *padding*, padding, however, introduces security risks!

1. **Padding Oracle Attack:** If a system reveals whether padding is valid/invalid (through error messages or timing), attackers can decrypt messages byte-by-byte without the key.
2. **Information Leakage:** Padding reveals plaintext length and can expose patterns.

(A) Electronic Codebook Book (ECB)
Each block is encrypted independently with their own key (*deterministic*, bad for large data).

(B) Cipher Block Chaining (CBC)
All the blocks are dependent on one another (chain).
Disadvantages:
- **Need Initial Value (IV)** known to sender & receiver
- **Padding** might be an issue

How CBC does NOT protect against block modification, reordering or deletion?
For example, If you know byte 5 in a block is "0" (no/false), you can flip the corresponding bit in the previous ciphertext block to change it to "1" (yes/true). And for Deletion you can delete the last block without issues. Reordering works too! you can reorder the text sure it will be scrambled but who cares no one is checking.

```
CBC only provides confidentiality (encryption), NOT integrity (authentication).
```

MAC (Message Authentication Code)

MAC is used for data *AUTHENTICATION*. It uses *SYMMETRIC* key.

Properties of MAC:

1. Easy to Compute.
2. H(x) is fixed length, but x can be any length.
3. Forgery Resistance: every pair of (x, MAC(x)) is unique.

Message Authentication (MA) Scheme

- **G**: Key generation algorithm
- **T**: Tag generation algorithm (MAC), inputs are K and Message
- **V**: Verification Algorithm, inputs are T and Message

HMAC

```
HMAC_K = h[(K XOR a) || h[(K XOR b) || m)]]
# where b and a are paddings
```

- Birthday attack secure.
- Can only crack if attacker knows K.

SSL- Authenticate then Encrypt

Not generally secure, authentication cannot discover if the ciphertext has changed or not.

SSH – Encrypt and Authenticate

Not generally secure, authentication tag can leak information on the plaintext. Can be fixed by *rekeying*.

IPSec – Encrypt then Authenticate

This is secure,.