Cryptography Tool

Cryptology

field of cryptography and cryptoanalysis

cryptography

study of encryption/decryption principle/methods

> both **Encryption Decrption** are based on `key`

Encryption

> **Enciphering** or **Encryption**

- **transformation** of **intelligible/understandable** information
- into
- **unintelligible form** to `disguis` it's meaning

Decryption

> **Deciphering** or **Decryption**

- The **inverse transformation** of encrpyed information `into` intelligible form.

cryptoanalysis

> Codebreaking

- analyzing encrypted info with **intent of recovering** orginal plain text.
- without knowing the key

`alt def:` deciphering ciphertext **without knowing** the key

components of a crypto system

1. `plain text` original message pre-encrption
2. `cipher text` the encrypted text [unintelligible form]
3. `encryption algo` algo used to **tranform** plain text *into* cipher text
4. `encryption key` key used by the **encryption algo**
5. `Decryption algo` algo used to transform **cipher text** ***into*** **plain text**
6. `Decryption key` key used by **decryption algo**

cryptographic mechanisms

1. Confidentiality[`privacy and secrecy`]
2. Integrity[`no modification`]
3. Authencity[`verfied entity`]

4. Identity[`specific inividual behind entity`]
5. Non-repudiation[`can't deny`]

cryptography characterize by:

1. **Type of encryption operations use**
    - `Substitution`/ `Transposition` / `Product` / `Bit Manipulation`
2. **Number of keys used**
    - `Single-key[secret]` / `two-key[public]`
3. **Way in which plaintext is processed**
    - `Block` / `Stream`

Shannon's priciple of Confusion and Diffusion

Confusion

each binary digit(bit) of ciphertext depends on **several parts of the key**

Defussion

- if we change a single bit of the plaintext one/two[half] bits of ciphertext should change

- if we change a single bit of the ciphertext then approximately one half of the plaintext bits should change.

Number of keys used

1. symmetric[single-key/private-key]

the same key is use for **encryption** and **decryption**

> Used in **DES** [Data Encryption Standard] **AES**

2. asymmetric[two-key/public-key]

two **mathematically related keys** are used.

- one is the **public key** to encrypt
- the other is the **private key** to decrypt.

> Used in **RSA** or **Al Gamal**, **DSA**

Processing way

Block cipher

- breaks the plaintext into **equal-sized** blocks
- usually 64/128 bits
- encrypts each block separately.
- one block at a time.

Stream cipher

- the input element are processed **individually/Continuously**, producing output as one element at a time.

Encryption Scheme Security

1. **Unconditionally Secure**:
    - no matter how much time an opponent has,it impossible to decrypt
2. **Computationally Secure**:
    - cost of breaking exceeds the value of info.
    - time required to break exceeds the useful lifetime of the info.

Triple-DES

- repeating **DES algo** three times using either `two` or `three` unique keys
- key size of 112 or 168 bits
- pros:
    - 168-bit key length overcomes the vulnerability to brute-force attack of DES
    - Underlying encyption algo is the same as DES
- cons:
    - software is laggy (secure but much slower).
    - uses a 64-bit block size

Practical Security Issues

- Typically symmetric encryption is applied to a unit of data larger than a single 64-bit or 128-bit block

- `Electronic codebook (ECB)` mode is the ***simplest*** approach to multiple-block encryption

    - each block is encrypted with the same key.
    - Cryptoanalysts may be able to exploit regularities in plaintext.

- `Cipher-block chaining (CBC)` incease security of symmetric clock ecnrption for large sequences

- there are two basic approachs to block encryption:

    - encrypt each clock independently
    - encrypt each block so that it's output ciphertext is dependent on the output of the **pervious block**

Electronic CodeBook (ECB)

- Same key used on each block
- the encryption of each block is completely independent
- **Draw backs of ECB:**
    - Two similar blocks of plaintext will result in similar blocks of ciphertext
    - ECB isn't practical when data involves long repetitive strings

Cipher-Block Chaining (CBC)

- A depenedent encryption approach
- `XOR` process is used to combine the **ciphertext output** with **plaintext input** of the next block.
- the encrption of each block is dependent on the previous one
- An encrption of identical input blocks will `have different results`
- `initialization vector`
    - is an input to the first block
    - pseudo-random binary sequence
    - is used to `XOR` the **First block ONLY**
- **Drawback of CBS**
    - single encryption **Error** is **cascaded** through the following *blocks*
    - decryption relies on knowledge of previous block.

Block cipher VS Stream cipher

block cipher	Stream cipher
one block at a time	one byte Continuously
can reuse key	Unpredictable without the Knowledge of the input key

- `Stream cipher` users a **keystream** combined with one byte[from plaintext] at a time.