CS2107 Introduction to Information Security

AY23/24 Sem 2, github.com/gerteck

0. Introduction

CS2107: Introductory module, illustrates fundamentals of how systems fail due to malicious activities and how they can be protected.

Lecture Notes notation

- **Textbook:** Security in Computing (5th ed). Prentice Hall. Reference [PFx.y] refer to chapter x section y of this book.
- Links with "read": Part of lecture, required.
- Links with "see": Optional good to browse references.

Internet Security Threat Report Exc. Summary

Persistent threats are threats that often operate within the shadows, outside of attention focused.

- Highly targeted, often use combination of social engineering and software vulnerabilities to establish footholds within the targeted enterprise.
- Network protection not enough to migitgate threats, comprehensive monitoring program required to scan all internal and external network traffic. Identifying, securing data is also key to protecting assets.
- Recent attack types: Formjacking, Ransomware, Living off the Land (LotL),

Introduction to Computer/Info Security

Systems may fail due to operator mistakes, hardware failures, poor implementation etc. Many systems robust against typical noise. Security is about intentional failures inflicted by deliberate human actions.

Security Definitions: C-I-A triad

- **Confidentiality**: Prevention of unauthorized disclosure of information.
- **Integrity**: Prevention of unauthorized modification of info / processes.
- **Availability**: Prevention of unauthorized withholding of info / resources.
- Other requirements may include (Confidentiality: anonymity, covert channel), (Integrity: Non-repudiation (digital signature)), (Other: Accountability, traitor-tracing (printout w hidden watermark)) etc.
- Importance of understanding security requirements before adopting mechanisms. Do not adopt mismatched protection mechanisms.

Difficulties in achieving Security

- **Security not considered** (in early design stage).
  Lack of adverarial thinking in design / trade-off in security with ease-of-use, performance / cost.
- Difficult to formulate requirements /design / implementation bugs.
- Difficult to operate/manage (Human error).
- **Known Vulnerabilities: CVE (Common vulnerabilities & Exposures**. Repository of discovered vulnerabilities. Significant portion considered "implementation bugs".
- **Zero-day Vulnerabilities**: Unpublished vulnerabilities. If attackers deploy attacks on zero-day v., victims have "zero-day" to react. Not easy to get.

1. Encryption

Key Summary & Takeaways

- **Encryption designed for confidentiality.** (Not necessarily integrity).
- Formulate attack scenario by defining attacks it can prevent.
- Notions of "Oracle": Encryption, Decryption, Padding Oracle.
- **Key strength**: Quantifying security by equivalence of best-known attack to exhaustive search.
- No known efficent attacks on modern schemes under "original" threat models, but there are pitfalls. Such as implementation error (wrong mode, wrong random source, mishandle of IV), side channel attack, implicity require integrity, padding oracle attack.
- Design of various symmetric key encryption schemes:
  - One-time pad, stream cipher (xor'ing with "pseudo-random" string)
  - Block cipher (mode of operations: CBC, ECB, CTR, GCM)
- **Crucial role of IV**. (Need randomness for indistinguishability). Make encryption probabilistic, how it is deployed, why it is important.

Definitions

- **Symmetric Key Encryption Scheme**: Two algorithms: encryption and decryption. Meets correctness property, and must be secure.
- **Correctness:** $(D_k(E_k(x)) = x)$. **Security**: Informally, "difficult" to derive useful info of the key k, and plaintext x. Ciphertext resemble random sequence of bytes.
- **Cryptography**: Study of techniques in securing communication in present of adversaries with access. Encryption just a primitive (other includes crypto hash, digital signature etc. Common placeholders: Alice, Bob, Eve ("eavesdroppper"), Mallory (malicious, modify message), etc.

Attack Model

Aka: Threat / Adversary / Security Model, Attack Scenario.
Measuring **security of a system**: Through attack classes it can prevent. Secure w.r.t these classes of attacks. Attack models are application-dependent. Attack models described by:

- Attacker's knowledge (info / service exposed) and computing resources
- Attacker's goal

Types of information we assume access to: (Access to info can be formulate to accesses to an **"Oracle"**).

- **Ciphertext only attack:** Adversary given collection of ciphertext c. May know some properties of the plaintext, for e.g. the plaintext is an English sentence. (adv. can't choose plaintext).
- **Known plaintext attack:** Adversary given collection of plaintext m and corresponding ciphertext c. (adv. can't choose plaintext.)
- **Chosen plaintext attack (CPA):** Access to blackbox (i.e. the oracle). Can choose, feed any plaintext m, obtain corresponding ciphertext c (all encrypt with the same key), reasonable large number of times. Can see ciphertext and choose next input. *(black-box called encryption oracle).*
- **Chosen ciphertext attack (CCA2)**: Same as chosen plaintext attack, but adv. chooses ciphertext and blackbox outputs plaintext. *black-box called decryption oracle*.
  Note: (strange to assume attacker has power to decrypt, but good reason is that in some practical scenario, attacker does have some but not full decryption capability, e.g. *padding oracle*. To cater for all scenarios, in formulation and design of encryption, we consider oracle with full decryption capability.)

Adversary's Goals

- **Total Break**: Attacker wants to find the key.
- **Partial Break**: Few definitions, e.g. decrypt ciphertext, determine coarse information about plaintext, etc.
- **Indistinguishability (IND)**:) Attacker may satisfy with distinguishability of ciphertext: with some "non-negligible" probability more than ½, the attacker is able to distinguish the ciphertexts of a given plaintext (say, "Y") from the ciphertext of another given plaintext (say, "N"). (Equivalently, unable to distinguish the ciphertext and a randomly sequence).
- **Note**: total break most difficult goal, since achieves all. Distinguishability weakest goal, design cryptosystem preventing weakest goal.

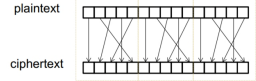
Classicial Ciphers (Symmetric)

1. Substitution Cipher

- **Plaintext, ciphertext**: String over a set of symbols U.
- **Key**: Subsitution table S, 1-1 onto function from U to U.
- **Key space**: Set of all possible keys (e.g. 27). **Space Size** is total number of possible keys (factorial, 27!). **Key Size** is number of bits required to represent a key. (log 2 (27!), since 27! unique)
- **Exhaustive Search (Brute-force): Can be infeasible in worst case**
- **Known-plaintext-attack**: Access to pairs of ciphertext and corresponding plaintexts: *Sub. cipher "not secure / totally broken under known plaintext attack"*. (Possible in practice, e.g. certain words in header of email "From, Subject", protocols bytes fixed header information etc.)
- **Ciphertext only attack**: Vulnerable to frequency analysis attack, when plaintexts English sentences.

2. Permutation Cipher (aka transposition cipher)

- Groups plaintext into blocks of t characters, applies secret permutation (1-1 onto function). Fails miserably on known-plaintext attack.



- S & P cipher not secure, performing substitution multiple times no use. However, by interlacing them (S&P), attacks become more difficult. Many modern encryption scheme (e.g. AES) designed using rounds of S & P.

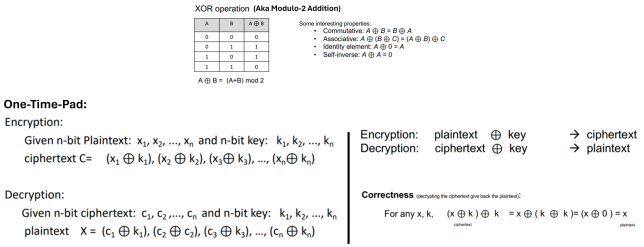
Terminology

- **Cryptosystem**: A system for encryption and decryption.
- **Plaintext**: Original form of message.
- **Ciphertext**: Encrypted form of message.
- **Perfect Secrecy**: A cryptosystem has perfect secrecy if for any distribution X, for all x, y:
  $$Pr(X = x|Y = y) = Pr(X = x)$$
- For any ciphertext y and plaintext x, the chances attacker correctly predicts x before knowing y, and after knowing y, are the same.
- **Work Factor**: Difficulty of breaking an encryption (Amount of effort necessary).
  - (E.g. determine time it would take to test single password, multiply by total possible passwords).

3. One Time Pad

One-time pad (OTP) is an encryption technique that requires the use of a single-use pre-shared key that is larger than or equal to the size of the message being sent.

- Plaintext is paired with a random secret key (known as one-time pad).
- Each bit or character of the plaintext encrypted by combining it with corresponding bit/character from pad using modular addition.[1]



- **Requirement**: Key cannot be re-used, used only once. Hence, 1GB plaintext would need 1GB key to encrypt.
- From pair of ciphertext & plaintext, key can be derived. However, key useless, as only used once.
- **Security**: OTP leaks no information of plaintext, sometimes called "unbreakable". There is a exhaustive key for any English sentence. (Perfect Secrecy)

Modern Ciphers (Symmetric)

Generally refers to schemes that use computer to encrypt / decrypt. E.g:
- DES [Data Encryption Standard 1977]
- RC4 [Rivest's Cipher 4 1987]
- A5/1 [used in GSM 1987]
- AES [Adv. Encrypt. Std.], RSA).

- Designs take into consideration of known-plaintext-attack, freq. analysis and other known attacks.
- Supposed to be secure, so any successful attack does not perform noticeably better than exhaustive search.

Key Length, Exhaustive Search DES, AES

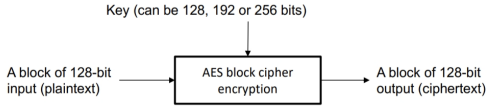
- **Security of Encryption Scheme**: Quantified by **length of key**, w.r.t. exhaustive search.
- Given a key length of 32 bits, there are 2^{32} possible keys. Hence, exhaustive search needs to "loop" 2^{32} in worst case.
- **No. of bits to be considered "secure"**: 128, 192, 256 bits,
  - (NIST reccomendation for AES).

1. DES, AES and Exhaustive Search

- **Exhaustive Search on DES**: Key length of DES is 56 bits. Previously, seemed infeasible, but has since become easily broken.
- **AES**: New standard for block cipher in 2000, AES block length is 127, key length can be 128, 192 or 256 bits.
- Currently, no known attacks on AES. NSA classifies AES as "Suite B Cryptography".

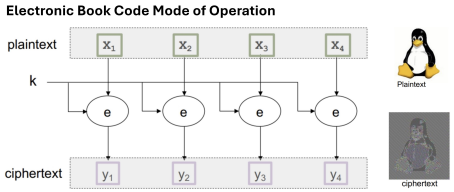
2. Block Cipher & Mode-of-Operations

- **Block Cipher**: DES & AES known as "Block Cipher". Block cipher designed for some fixed size input/output.
  - E.g. AES designed for 128 bits input/output.
- **Block Cipher Mode Of Operation**: Describes how to repeatedly apply cipher's single block operation to securely transform amounts of data larger than a block. (Extending encryptino from single block to multiple blocks).



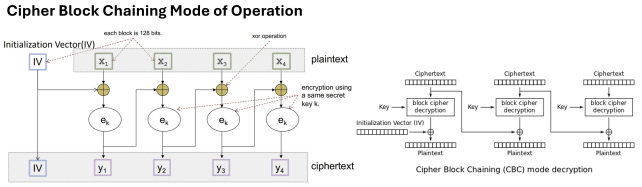
Mode-of-Operation: ECB Mode on AES

- **Electronic Book Code**: Divide plaintext into blocks, apply block cipher to each block, with the same key.
- ECB leaks information! AES encryption determinsitic without randomly chosen IV.
- **Deterministic**: Produces same ciphertext for given plaintext and key over separate executions.



Mode-of-Operation: CBC Mode on AES

- **Cipher Block Chaining on AES**: Using an IV, uses chaining process that causes decryption of block of ciphertext to depend on all preceding ciphertext blocks.
- **Initialization Vector (IV)**: Arbitrary number of certain length used with secret key, to provide the initial state, for data encryption.
- **Encryption**: Each plaintext block is XOR-ed with previous ciphertext block, and then encrypted. Process repeats untill all plaintext is ciphertext blocks.
- **Decryption**: Reverse encryption process. Note, process does not need to start with final block, can happen simultaneously as all inputs present.



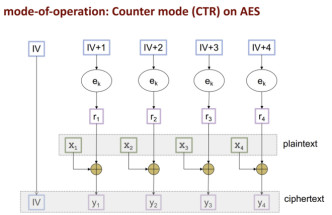
- The Initialization Vector (IV) is an arbitrary value chosen during encryption. So, it is different in different encryptions of the same plaintext. Depending on implementation, IV can be randomly chosen, obtain from a counter, or from other info.

$y_0 = IV. \quad y_i = E_k(x_i \oplus y_{i-1}) \quad for\ i>0$

Note: In the above figure, we treat IV as part of the final ciphertext. The terminology is not consistent in the literature. Some documents may state that "the final message to be sent are the IV and the ciphertext" (i.e. IV is not included in the "ciphertext"). In this module, when it is crucial, we will explicitly state whether IV is included or excluded. (e.g. "Ciphertext together with an IV").

Mode-of-Operation: Counter Mode (CTR) on AES

- **Counter Mode**: Turns block cipher into stream cipher, generates next keystream block by encrypting successive values of a "counter". Counter can be any function producing sequence, incld. simple increment by one counter.



Mode-of-Operation: GCM Mode (Galois/Counter)

- **GCM**: Combines Counter mode (CTR) with Galois authenication. Construction of mode more complicated, is "Authenticated-Encryption".
- Ciphertext consists of extra tag for authentication, secure in presence of decryption oracle.

Python Programming: CBC

- Python.
- (package PyCryptodome


```
>>> from Crypto.Cipher import AES
>>>
>>> key = b'Sixteen-byte key'
>>> iv  = b'Sixteen-byte  IV'
>>> cipher = AES.new(key, AES.MODE_CBC, iv)
>>> c=iv+cipher.encrypt(b'Plaintext of length with multiple of 16    bytes')
```

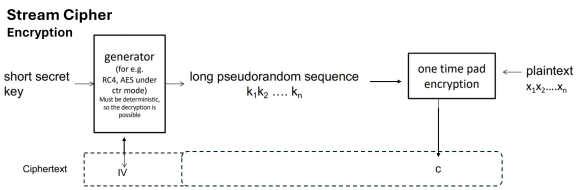
In Python, to display a byte sequence, we can use…

```
>>> from base64 import *
>>> b16encode(c)
b'536978746565624E2D627974652020495 6B186083256CACCBD1638AF4877FBF2AAFBECB66FE13C403D7CE8EA04D028E66CA6AE1294
FF51C2F363CCC8953137A6A3'
```

3. Stream Cipher and IVs

Stream Cipher

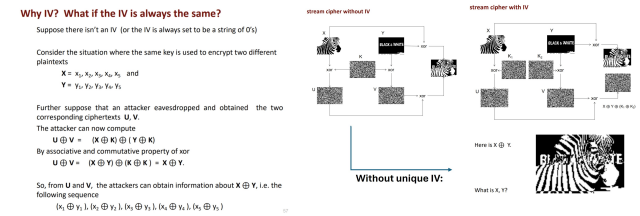
- **Stream Cipher**: A symmetric key cipher where plaintext combined with **pseudorandom cipher digit stream (keystream)**.
- "Inspired" by one-time-pad, generate some cyrptographically secure pseudorandom sequence.
- w/o knowing secret key, computationally diff. to distingush from truly random sequence. Similarly diff. to get short secret key from sequence, or predict part of sequence from another part.
- **IV**: Most ciphers have Initialization Vector, randomly chosen or from counter.



Decryption: Given the key and the ciphertext with the IV.
Step 1: Extracts the IV from the ciphertext
Step 2: From the key and IV, generates the long sequence.
Step 3: Performs xor to get the plaintext.

Role of Unique IV

- Recall, IV appended to front of c to form ciphertext. This IV must be different for every message. IV need not be secret.
- Sequence is derived from IV and secret key. IV needs to be unique for every message! Otherwise, leaks information.
- **Unique IV**: If IV different, two pseudorandom sequences will be different. Two ciphertexts of same plaintext will be different. Can just randomly choose the IV for each message.
- Hence, xor'ing two ciphertexts will not cancel out pseudomsequences.
- **IV makes encryption "probabilistic.**



- IV also needed in **CBC mode**, to make encryption non-deterministic. If encyrption deterministic (without IV), it will leak information on whether plaintext of two ciphertext are the same, if $C_1 == C_2$. Could be crucial piece of information.

Types of Attacks

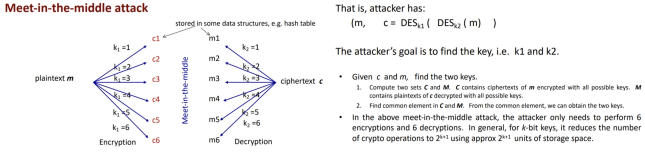
Meet-In-The-Middle Attack (Double / Triple DES)

Double / Triple DES

- **Double / Triple DES**: To improve DES security, (DES weak as key length of 56 bits short), do multiple repeated encryptions using different keys. A reason for this may be to utilize already existing suitable hardware to encrypt.
- DES does not form a group ($E_{k1}(E_{k2}(x)) \neq E_{k2}(x)$) for some $k3$, so makes sense to use multiple encryptions.
- **Double DES**: Consider double encryption, we expect key-strength to be 112. (2^{56+56}). However, attacker, with storage space, can reduce key strength to below 57.

Meet-In-The-Middle Attack

- **MITM**: Meet-in-the-middle Attack, a well-known plaintext attack, is a generic space-time tradeoff cryptographic attack against encryption schemes that perfrom multiple encryption operations in sequence.
- Breaking two-part encryption from both sides simultaneously.
- Primary reason why **Double DES not used**, and why **Triple DES key** (168-bit) can be brute forced by attacker with 2^{56} space and 2^{122} operations.
- **Mechanism**: Assume attack has a **pair** (m, c) of plaintext and corresponding ciphertext.
- **Remedy**: Use triple encryption, but with 2 keys. (E.g. 3DES, TDES, 3TDES etc.)



Padding Oracle Attack

Padding Format

- For fixed size blocks, e.g. block size of AES is 128bits (16 bytes), padding is needed to fit plaintext into last block.
- Many ways to fill values, but important piece of information encoded: **number of padded bits**. If information missing, receiver will not know lenght of plaintext.
- E.g. **PKCS#7 Padding Standard.**

PKCS#7 Padding Standard

- PKCS#7 is a padding standard.

Read https://en.wikipedia.org/wiki/Padding_(cryptography)#PKCS7

- The following example is self-explanatory.

Suppose the block size is 8 bytes, and the last block has 5 bytes (thus 3 extra bytes required), padding is done as follow:

DD DD DD DD DD DD DD DD | DD DD DD DD DD **03 03 03**

- In general, the paddings are:

01
02 02
03 03 03
04 04 04 04 etc.

- If the last block is full, i.e. it has 8 bytes, an extra block of all zeros is added.

Oracle

- **Security Analysis**: Need to know **1. information attackers have**, 2. attacker's goals.
- **Oracle**: Query-answer system. Attacker can send in multiple queries, **Oracle** will output the answer. E.g.
  - **Encryption Oracle**: Output ciphertext for given plaintext, of s. key k.
  - **Decryption Oracle:** Output plaintext given ciphertext, of s. key k.
- **Padding Oracle**: Attacker can query a ciphertext (encrypted using some secret key k, padding oracle knows k). Oracle outputs yes/no if plaintext is in correct "padding" format.
- Padding Oracle can come in many forms, e.g. query response behaviour, response time, subtle differences.
- **Padding Oracle Attack Model**:
  - Attacker has ciphertext including iv: (iv, c)
  - Attack's goal: plaintext of (iv, c).

Padding Oracle Attack (on AES CBC Mode)

- AES CBC mode not secure against padding oracle attack. (In particular, when done with PKCS#7).
- Easily extend algorithm to find all plaintext. Attack is practical as there are protocols between client and server which performs this. *Now, if attack obtained ciphertext, attack can interact with server to get plaintext.*
- **Prevention of Padding Oracle Attack**:
  - Deny access to such Oracle. Might not be feasible all the time.
  - Change padding standard to mitigate attack. However, may be smarter way to attack new padding.
  - Using CTR mode might avoid padding. In practice, bit strings have to be padded in one way or another.
  - Padding Oracle weaker form of Decryption oracle. If scheme secure in presence of decryption oracle, scheme also secure against padding oracle attack. GCM believed to be IND-CCA2 secure.

Padding Oracle Attack Algorithm

- **Algorithm**:
  - Force last X bytes to be padding
  - Do this by modifying byte of previous ciphertext/IV (Loop till YES)
  - Here, by XOR'ing padding and input, we get intermediate byte.
  - By XOR'ing this int. byte with original IV / ciphertext, find plaintext!
- **Easily extend algorithm** to find all plaintext, by using increasing length of padding, decrypt from end to start. (Right to Left, repeat process).