Topic 0

CIA

1. Confidentiality
   - Prevention of unauthorized disclosure of information
2. Integrity
   - Prevention of unauthorized modification of information or processes
   - Non-repudiation
   - Authentication
3. Availability
   - Prevention of unauthorized withholding of information or resources

Threat model

- The attacker's goals
- The attacker's capabilities

Trade-off in security

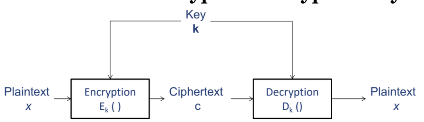
- Ease-of-use
- Performance
- Cost

Threat-Vulnerability-Control

- **Threat**: A set of circumstances that has the potential to cause harm (e.g. an attacker with control of the workstation in the LT could maliciously gather sensitive info like passwords)
- **Vulnerability**: A weakness in the system (e.g. anyone can reboot the workstation from USB or Disk to gain control)
- **Control**: A control, countermeasure, security mechanism is a mean to counter threats (e.g. restrict physical access to the workstation, disable USB booting)
- **A threat is blocked by control of a vulnerability**

Topic 1: Encryption

1.1 Definition: Encryption/decryption/keys



- A symmetric-key encryption scheme consists of encryption and decryption
- A cipher must be correct and secure
  - **Correctness**: For any plaintext x and key k, $D_k(E_k(x)) = x$
  - **Security**: Definition depends on the threat models. Informally, from the ciphertext, the eavesdropper is unable to derive useful information of the key k or the plaintext x, even if the eavesdropper can probe the system.
- Probabilistic encryption: for the same x, there could be different c's. But they all can be decrypted to the same x.

1.2 Security Model and Requirement

Threat model

- Attacker's goal
  - Total break (most difficult goal)
    - ∗ Attacker wants to find the key
  - Partial break
    - ∗ Attacker may want to decrypt a ciphertext but not interested in knowing the key
    - ∗ Attacker may simply want to extract some info abt the plaintext (e.g. if it is a jpg or excel file)
  - Distinguishability (weakest goal)
    - ∗ With some non-negligible probability of $> 1/2$, the attacker acan correctly distinguish the ciphertexts of a given plaintext from the ciphertext of another given plaintext
- Attacker's capability
  - Ciphertext only attack
    - ∗ Attacker is given a collection of ciphertext c. The attacker may know some properties of the plaintext (e.g. the plaintext is an English sentence)
  - Known plaintext attack
    - ∗ The attacker is given a collection of plaintext m and their corresponding ciphertext c
    - ∗ Attacker might get this as they know the header or part of the plaintext
  - Chosen plaintext attack (CPA)
    - ∗ The attacker has access to an oracle. The attacker can choose and feed any plaintext m to the oracle and obtain the corresponding ciphertext c (all encrypted with the same key). The attacker can access the oracle many times, as long as within the attacker's compute power. He can see the ciphertext and then choose the next input. This black-box is an **encryption oracle**.
    - ∗ e.g. attacker has access to a smartcard
    - ∗ e.g. attacker can eavesdrop
  - Chosen ciphertext attack (CCA2)
    - ∗ Same as CPA but the attacker chooses the ciphertext and the black-box outputs the plaintext. The black-box is a **decryption oracle**.
    - ∗ Padding oracle is a weaker form of a decryption oracle.

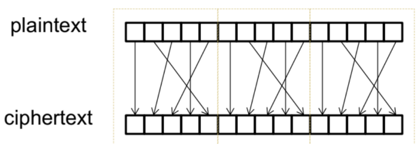
From defender's POV, want a cipher that can protect against the attacker with the highest capability. Cipher is secure against CCA2 (decryption oracle) $\implies$ secure against CPA (encryption oracle)

1.3 Classical ciphers + illustration of attacks

1.3.1 Substitution cipher

- Plaintext and ciphertext are both strings over a set of symbols U.
- The key is a 1-1 onto func from U to U
- Key space: set of all possible keys
- Key space size: total number of possible keys
- Key size/length: number of bits required to represent a key
- Attacks
  1. Exhaustive search (examine all possible keys 1 by 1)
     - Running time depends on size of key space
     - If the table size is 27, the key can be represented by a sequence of 27 symbols. The size of key space is 27!. Exhaustive search eneds to carry out 27! loops, which is infeasible using current compute power.
  2. Known plaintext attack
     - Given sufficiently long ciphertext, the full table can be found
     - Substitution cipher is not secure under known plaintext attack.
  3. Ciphertext only attack
     - Given that the attacker knows that the plaintext is an English sentence, he can do frequency analysis attack. The frequency of letters used in English is not uniform. Given a sufficiently long ciphertext, attacker may correctly guess the plaintext by mapping frequent characters in the ciphertext to the frequent character in English.

1.3.2 Permutation cipher



- AKA transposition cipher
- First group the plaintext into blocks of t characters, then apply a secret permutation to each block by shuffling the characters
- The key is the secret permutation, which is a 1-1 onto func e from $\{1, 2, \ldots, t\}$ to $\{1, 2, \ldots, t\}$. t can also be part of the key.
- Attack
  - Fails under known-plaintext attack
  - Easily broken under ciphertext only attack if the plaintext is English text

1.3.3 One Time Pad

Properties of xor:

- Commutative: $A \oplus B = B \oplus A$
- Associative: $A \oplus (B \oplus C) = (A \oplus B) \oplus C$
- Identity element: $A \oplus 0 = A$
- Self-inverse: $A \oplus A = 0$

One Time Pad

- Encryption: plaintext xor key bit by bit
- Decryption: ciphertext xor key bit by bit
- Key is only used once, so 1GB of plaintext would need a 1GB key to encrypt
- Security
  - From a pair of ciphertext and plaintext, attacker can derive the key but useless bc key won't be used anymore

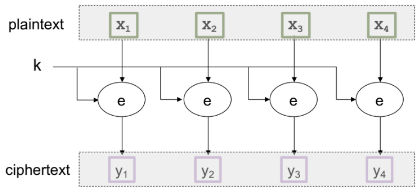
1.4 Modern ciphers + recommended key length

1.4.2 Block cipher & mode of operations

DES/AES are known as block ciphers. Block ciphers have a fixed size of input/output. AES: 128 bits (16 bytes).

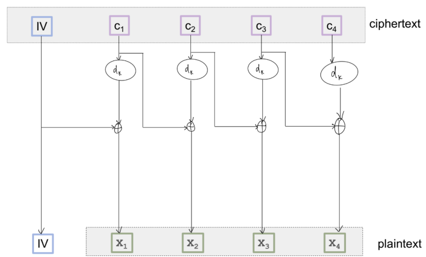
Large plaintext is divided into blocks before applying the block cipher.

ECB (electronic code book) mode

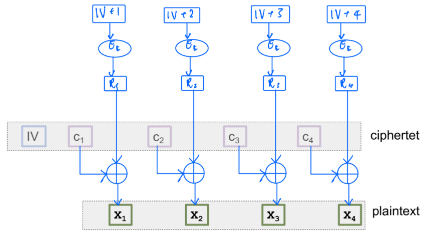


CBC (cipher block chaining) mode on AES

- Initialization vector (IV) is an arbitrary value chosen during encryption, must be different in different encryptions.
- In CBC mode, IV must be unpredictable, else it is susceptable to BEAST attack.
- If IV is randomly chosen, it is unpredictable



CTR (counter) mode

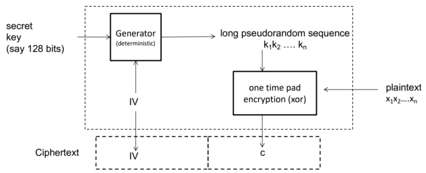


GCM mode (Galois/counter)

Authenticated encryption, ciphertext consists of extra tag for authentication. Secure in the presence of decryption oracle.

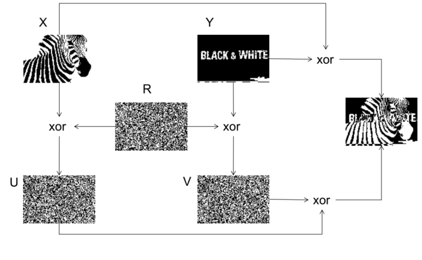
1.4.3 Stream cipher and IVs

Stream cipher is bit by bit. CTR mode is a "stream cipher" but it is not bit by bit.

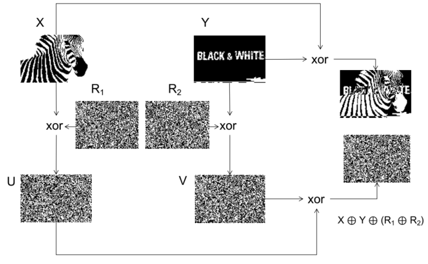


- Need IV and no two IVs can be the same

Stream cipher without IV



Stream cipher with IV

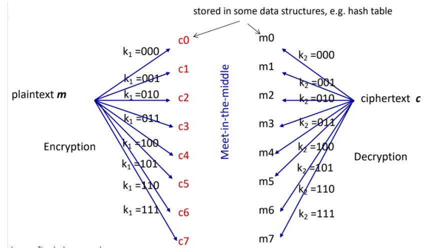


- IV makes an encryption probabilistic

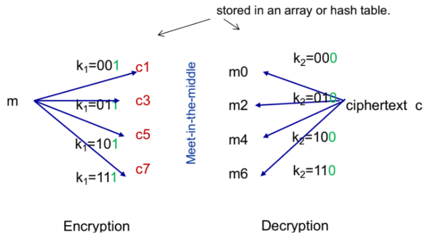
1.5 Examples of attacks on crypto

1.5.1 Meet-in-the-middle

- DES is not secure $\rightarrow$ improve by encrypting multiple times using different keys
- Consider double encryption under known plaintext attack. Attacker has m and c and wants to know k_1, k_2.
- Using exhaustive search, amount of DES encryption/decryption would be 2^{56+56}
- Hence use meet-in-the-middle attack.
- for k-bit keys, this reduces the number of crypto operations to 2^{k+1}



Tradeoff with time and space



- Last bit of k_1 fixed to 1, last bit of k_2 fixed to 0
- Perform meet-in-the-middle on the first 2 bits of k_1 and k_2

1.5.2 Padding Oracle

Plaintext needs to be padded to split into blocks

- PKCS#7 is a padding standard

01

02 02

03 03 03

04 04 04 04

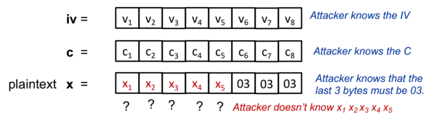
....

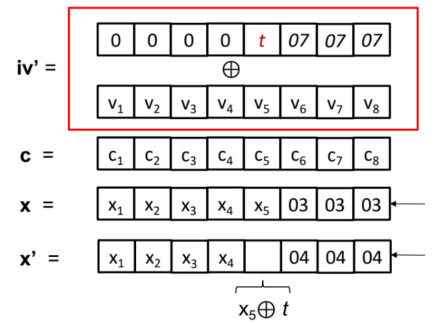
08 08 08 08 08 08 08 08

Padding oracle attack

Padding oracle attack on AES CBC mode

Attacker knows:





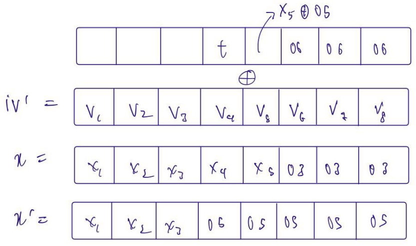
$$iv \oplus d(c) = 03$$

$$iv' \oplus d(c) = 04$$

xor the 2 tgt to get $iv' = 07 \oplus iv$

$$iv' = iv \oplus 00\,00 \ldots t\,07\,07\,07$$
$$d(C_5) \oplus t \oplus V_5 = 04$$
$$d(C_5) \oplus V_5 \oplus t = 04$$
$$d(C_5) \oplus V_5 = x_5$$
$$x_5 \oplus t = 04$$
$$x_5 = 04 \oplus t$$

Keep guessing t until padding oracle outputs YES, then we know x_5
To get next byte:



1.6 Pitfalls in usages and implementations

1. Wrong choice of IV / reusing one-time pad
2. Randomness is predictable
3. Modify existing or design your own encryption scheme
4. Reliance on obscurity: Kerkchoff's principle
   - Kerkchoff's principle: A system should be secure ven if everything about the system, except the secret key, is public knowledge

Topic 2: Authentication Credential

Authentication

Authentication is the process of assuring that the communicating identity, or origin of a piece of information, is the one that it claims to be.

- Authentication implies integrity.

Data-origin authentication: is a piece of data generated by an authentic entity?

- Signature or MAC (message authentication code)

Commnication authentication: is the enitity interacting with the verifier an authentic entity?

- Authentication protocol

2.2 Password

Password vs key

Passwords are generated by human and can be remembered by human. Keys are binary sequences that are infeasible to be remembered by humans.

Password system

1. Bootstrapping
   - User and server establish a common password, server keeps a password file keeping the identity and the corresponding password
   - Password established during boostrapping either by a default password or by the server/user choosing a password and sending it to the user/server through another communication channel
2. Authentication
   - Server authenticates an entity. An entity who can convince the server that it knows the password is deemed to be authentic.
3. Password reset
   - Need to authenticate the entity before allowing the entity to change password.
   - Need another credential (other than the old password) to authenticate bc ppl might want to reset password when they forget
   - Can be done through OTP, security question (not secure as entropy of answers is lower than the password)

Attack on passwords

2.2.1 Attack on Bootstrapping

- Attacker intercepts the password during bootstrapping, e.g. if password is sent through postal mail, an attacker steals the mail to get the password
- Attacker uses the "default" passwords
  - Mitigation: require the user to change password after first login
- Example: zoom flaw allowed account hijacking

2.2.2 Attack on Password Reset

- Mechanism of security questions weakens the password system, but it is less common now
- Social engineering + password reset

2.2.3 Searching for the password

Dictionary attacks

- Test passwords using a dictionary that could contain words from English dict, known compromised passwords, etc.
- Also test combinations of words in the dictionary. e.g. combinations of 2 words, all possible capitalizations of letters in each word, substituting 'a' with '@', etc.
- **Online dictionary attack**
  - To test a password, attacker must interact with the authentication system
- **Offline dictionary attack**
  - Attacker first obtains some information D about the password, possibly by sniffing the login session of an authentic user, or by interacting with the server. (e.g. attacker obtains the hashed password)
  - Next, the user carries out dictionary attack using D without interacting with the system (e.g. attacker compares the hashed password with the hashed words in dictionary)
- Guessing password from social information

2.2.4 Stealing the password

1. Sniffing
   - Shoulder surfing: look-over-the-shoulder attack
   - Sniffing the communication: Some systems simply send the password over the public network in clear (i.e. not encrypted), e.g. FTP, Telnet, HTTP
   - Sniff wireless keyboards that employ insecure encryption method
   - Using sound made by keyboard
   - Viruses, key-logger
     - Key-logger captures keystrokes and sends the info back to the attacker.
     - Can be in the form of software (viruses) or hardware.
2. Phishing
   - Victim is tricked into voluntarily sending the password to the attacker
   - Asks for password under false pretense
3. Spear Phishing
   - Phishing that is targeted to a particular small group of users, e.g. NUS staff

Phishing Prevention

- User training
- Blacklisting, e.g. phishtank.com
- Visually spot by ensuring that there is a padlock in the address bar and that the domain name in the url is correct

2.2.5 Password strength

- We quantify the key-strength by the size of the key if best-known attack is exhaustive search.
- If best-known attack is faster, then we quantify it by its equivalent in exhaustive search.

Using strong password

- Truly random password: password is chosen randomly among all possible keys using an automated passsword generator. High entropy but difficult to remember.
- User selection:
  - Mnemonic method
  - Altered passphrases
  - Combining and altering word
- Usability:
  - Strong passwords are difficult to remember
  - It is difficult to enter alphanumeric passwords into mobile devices. There are alternatives, e.g. graphical or gesture-based

Password entropy

Suppose a set P contains N unique passwords. A password is chosen by randomly picking a password from P. Entropy of password is

$$-\sum_{i=1}^{N} p_i \log_2 p_i$$

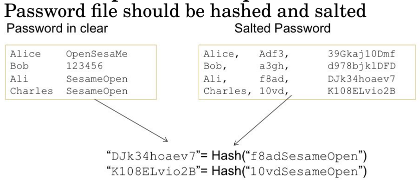
where p_i is the probability that the i-th password is picked.
If the password is chosen uniformly, each password in P has probability of $1/N$ of being chosen. The entropy of the password is

$$log_2 N \text{ bits}$$

For the entropy to be highest for a set of N items, p_i must be $1/N$

Additional protection to password files

Password file should be hashed and salted



Make it harder for rainbow table attack

2.3 Biometric

Biometric data is the password

$$\text{FMR (false positive)} = \frac{\text{no. of successful false matches (B)}}{\text{no. of attempted false matches (B + D)}}$$

$$\text{FNMR (false negative)} = \frac{\text{no. of rejected genuine matches (C)}}{\text{no. of attempted genuine matches (A + C)}}$$

	accept	reject
genuine attempt	A	C
false attempt	B	D

Threshold: FNMR/FMR. Lower threshold more relax, higher threshold more stringent

Attack on biometric system

Biometric data can be spoofed, use liveness detection e.g. temperature sensor in fingerprint scanner

2.4 n-factor Authentication and Multi-Step Verification

n-factor Authentication

Requires at least 2 different authentication "factors"

1. Something you know: password, PIN
2. Something you have: Security token, smart card, phone, ATM card
3. Who you are: Biometric

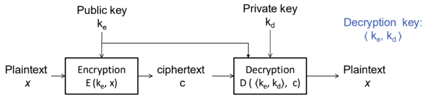
Multi-Step Verification

If both are the same category of factors (2 passwords, both are something you know) then it is 2-step verification

Topic 3: Authenticity (data origin)

3.1 PKC

- With multiple identitites, many pairs of symmetric keys are required.
- Symmetric key requires both entities to know each other before the actual communication session. Hence use public key encryption.

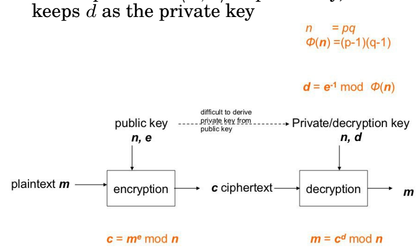


Popular PKC schemes

- RSA
- ElGamal
- Paillier
- Post-quantum cryptography

3.1.1 RSA

1. Owner randomly chooses 2 large primes p, q and computes $n = pq$
2. Owner randomly chooses an encryption exponent e s.t. $\gcd(e, (p-1)(q-1)) = 1$
3. Owner finds decryption exponent d where $de \mod (p-1)(q-1) = 1$
4. Owner publishes $\langle n, e \rangle$ as public key, and safekeeps d as the private key



Got algo to find d given e, p, q. For faster speed, choose small e. Common value is 65537. e is not a secret in such cases

3.1.2 Security of RSA

Getting RSA private key from public key is as difficult as factorizing n.

Padding of RSA

- Some forms of IV is required so that encryption of the same plaintext at different times would give different ciphertexts. Additional padding required for security.

3.1.3 Efficiency

RSA encryption/decryption is significantly slower than AES. Can use PKC to encrypt a symmetric key then use AES for encryption

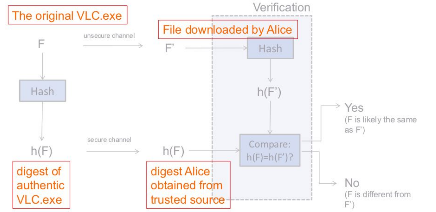
3.2 Data Authenticity

Security requirement of hash

- Collision-resistant
  - Collision: Find 2 different messages m_1, m_2 s.t. $h(m_1) = h(m_2)$
- 2nd pre-image resistant
  - 2nd pre-image: Given m_1, find m_2 s.t. $h(m_1) = h(m_2)$
- One-way
  - Pre-image: Given y, find m s.t. $h(m) = y$

Application of unkeyed hash

When downloading something from a website, match the hash of the file with the checksum displayed in the browser.



If not 2nd pre-image resistant, can be attacked

3.3 Data Origin Authenticity (mac), keyed

Keyed-hash is a function whose input is an arbitrary large message and a secret key, output is a fixed-size mac (message authentication code)

- Security requirement (forgery): Even if attacker sees multiple valid pairs of messages and their corresponding mac, it is difficult for the attacker to forge the mac of a message not seen before
- CBC-MAC: based on AES operated under CBC mode
- HMAC: Hashed-based MAC based on SHA

Application of mac

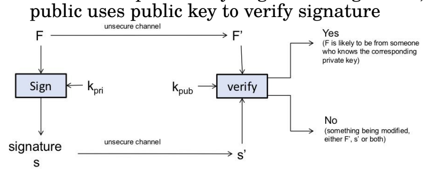
Same situation as before but dh secure channel to deliver digest. Protect the digest with the help of some secrets.

- In symmetric key setting, called mac
- In public key setting, called digital signature
- mac typically appended to file, also called authentication tag or authentication code

3.4 Data Origin Authenticity (Signature)

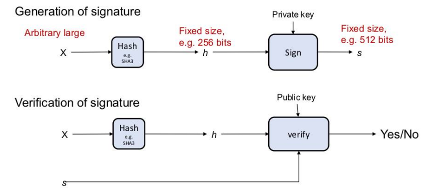
Public key version of MAC is called signature

- Owner uses private key to generate signature, public uses public key to verify signature



Signature is appended to the file F

- Signature scheme achieves **non-repudiation**



hash and sign / hash and encrypt

3.5.1 Birthday Attacks

Birthday attack is used to find collision.
Suppose we have M messages, and each message is tagged with a value randomly chosen from $\{1, 2, 3, \ldots, T\}$. Then the probability that there is a pair of messages tagged with the same value is approx

$$1 - e^{\left(-\frac{M^2}{2T}\right)}$$

Let S be a set of k distinct elements where each element is an n-bits binary string. Let us independently and randomly select m n-bit binary strings and put them in the set T. The probability that S has non-empty intersection with T is more than

$$1 - 2.7^{-km2^{-n}}$$

4. PKI + Channel Security

4.1 Distribution of public keys

- PKC requires a secure broadcast channel to distribute public keys: PKI
- If no secure way to distribute public key, attacker can impersonate by giving his own public key
- Certificate: A piece of document that binds a "name" to a "public key" & certified by a CA
- Certificate contains:
  - Name, public key, expiry date
  - Meta info: usages, type of crypto, name of CA, etc
  - CA's signature

4.2 PKI

4.2.1 Certificate & CA

- Certificates are used to distribute public keys. A CA issues certificates.
- CA: trusted authority that manages a directory of public keys
- CA has its own public-private key, some CA's public keys have been distributed securely through other means.
- OSes and browsers have pre-loaded CA's public keys, these CAs are known as root CAs.
- Other CA's public keys added through chain-of-trust
- A **certificate** is a digital document containing at least the following:
  - Name (e.g. alice@yahoo.com / bbc.com / *.bbc.com)
  - Public key of the owner
  - Time window that this cert is valid
  - Signature of CA

4.2.2 CA's chain-of-trust

4.2.3 Revocation

4.3 Limitations / attacks on PKI

4.4 Protocol 1: Authentication

4.5 Protocol 2: Key Exchange

4.6 Protocol 3: Authenticated Key Exchange

4.7 Securing Communication Channel