Revision notes - CS2107

Ma Hongqiang

February 23, 2018

Contents

0 Overview

Security concerns with **deliberate** human actions that are designed to cause failure.

Definition 0.1 (Security)**.**
Security is defined by the C-I-A triad.

- Confidentiality: Prevention of unauthorised disclosure of information
- Integrity: Prevention of unauthorised modification of information or processes
- Availability: Prevention of unauthorised withholding of information or resources.

It is difficult to achieve security due to

- Security is not considered in design.
- Security requirements are difficult to be determined.
- There are many constraints in design the security features.
- There may exist implementation flaws.

Clearly, there is a trade-off between security and ease-of-use, performance and cost.

1 Encryption

1.1 Definitions

Definition 1.1 (Encryption Scheme)**.**
An encryption scheme, or cipher, consists of two algorithms: **encryption** E_k and **decryption** D_k, where k is the key.
Suppose there is a plaintext x, $c := E_k(x)$ is defined as ciphertext.

Theorem 1.1 (Properties of Encryption Scheme)**.**
An encryption scheme must meet the **correctness property**, that is

$$\text{for any plaintext } x \text{ and key } k, D_k(E_k(x)) = x$$

Equivalently, $D_k \circ E_k = I$ the identity map.
It also needs to satisfy the security property, i.e., from the ciphertexts, it is difficult to derive useful information of the key k and the plaintext x. Specifically, the ciphertexts should resemble sequences of random bytes.

Cryptography is the study of techniques in securin communication in the present of adversaries who have access to the communication. Some common placeholders used in cryptography are

- Alice, usually the originator of message
- Bob, usually the recipient
- Eve, eavesdropper who can only listen
- Mallory, a malicious party who can modify messages

1.2 Classical Ciphers

Remark: Classical ciphers are not secure in the computer era.

1.2.1 Substitution Ciphers

Definition 1.2 (Substitution Cipher)**.**
In substitution cipher,

- Plaintext: string over a set of symbols U.
- Ciphertext: string over a set of symbols U.
- Key: a substitution table S, which is an isomorphic function $S : U \to U$.[1]

[1]This guarantees the existence of the inverse function S^{-1}.

Encryption: Given a plaintext string $X = x_1x_2\cdots x_n$ and key S, output the ciphertext

$$E_S(X) = S(x_1)S(x_2)\cdots S(x_n)$$

Decryption: Given a ciphertext string $C = c_1c_2\cdots c_n$ and key S, output the plaintext

$$D_S(C) = S^{-1}(c_1)S^{-1}(c_2)\cdots S^{-1}(c_n)$$

The concept of key space is a general concept of encryption scheme.

Definition 1.3 (Key space)**.**
Key space $\mathbb{S}$ is the set of all possible keys k. The **key space size** is the total number of possible keys $|S|$.
The **key size**(or **key length**) is the number of bits required to represent a key. It is given as $\lceil \log_2 |S| \rceil$.

For the substitution cipher from the set of alphabets and space character to itself, the key space size is 27! and key size is 94.
In general, attacks have purpose to find the key or to obtain some information of the plaintext. Attacks requires access to some information beforehand. A simple and general attack method is **exhaustive search**.

Definition 1.4 (Exhaustive Search)**.**
Exhasutive search examines all possible keys one by one. In the case of substitution cipher, given a ciphertext C encrypted from a plaintext X,

---

for each S in key space $\mathbb{S}$ **do** Compute $X' = D_S(C)$.
  if X'==X **then** break;
Display S

The running time of exhaustive search depends on teh size of key space $\mathbb{S}$. When key space size is large, exhaustive search becomes ineffective.
An efficient attack on substitution cipher is **known-plaintext-attack**.

Definition 1.5 (Known-Plaintext-Attack)**.**
An attack is a known plaintext attack if adversary has access to pairs of ciphertext and their **corresponding** plaintexts, and tries to get the key.

For substitution cipher, given a plaintext X and ciphertext C, one can ontain directly entries in the table. Therefore, the key can be found given sufficiently long ciphertext, and substitution cipher is deemed not secure under known plaintext attack.
Even sometimes plaintexts are not readily available, they can be guessed.
Also, substitution cipher is vulnerable to **frequency analysis**, if the plaintexts is sentences of any language.

1.2.2 Permutation Cipher

Definition 1.6 (Permutation Cipher)**.**
Permutation cipher is also known as transposition cipher. **Encryption**: The plaintext is grouped into blocks of t characters, and then applied a secret permutation to each block by shuffling the characters. **Key**: the isomorphic function

$$\begin{aligned} e : \{c_1, c_2, \ldots, c_t\} &\to \{c_1, c_2, \ldots, c_t\} \\ c_i &\mapsto p_i := e(c_i) \end{aligned}$$

We denote this permutation p by

$$p = (p_1, p_2, \ldots, p_t)$$

The size t is also part of the key and is kept secret.
Decryption: Apply e^{-1} to C to get P.

Remark: Permutation cipher also fails miserably under known-plaintext attack and is easily broken if plaintext is English text.

1.2.3 One Time Pad

Definition 1.7 (One Time Pad)**.**
Key: n-bit $k_1 k_2 \cdots k_n$.
Encryption: Given n bit plaintext, apply key to get ciphertext

$$C := (x_1 \oplus k_1)(x_2 \oplus k_2) \cdots (x_n \oplus k_n)$$

where $\oplus$ is the binary `XOR` operator. **Decyption**: Given n bit ciphertext, apply key to restore plaintext

$$X = (c_1 \oplus k_1)(c_2 \oplus k_2) \cdots (c_n \oplus k_n)$$

This encryption scheme works because $\forall x, k, (x \oplus k) \oplus k = x$. **Remark**: Although one-time-pad is unbreakable, the length of key is same as plaintext, which is useless in many applications.

1.3 Modern Ciphers

Modern ciphers generally refer to schemes that use computer to encrypt/decrypt, and their design take into considerations of plaintext attack and frequency analysis.
They are supposedly secure in a sense that any successful attack does not perform noticeably better than exhaustive search.
The security of an encryption scheme can be quantified by the **length of the key**, or more precisely $\lceil \log_2 N \rceil$ bits where N is the number of searches required to break the cipher.

1.3.1 Data Encryption Scheme

Data Encryption Scheme has a key length of 56 bits. It can be broken using distributed computing or specialised chip.

1.3.2 Stream Cipher and Initial Values

Definition 1.8 (Stream Cipher)**.**
Suppose plaintext has M bits and secret key has n bits. Stream cipher generates a M-bit sequence from the key, which is used as the secret key in the one-time-pad for encryption and decryption.

Remark: Generator must be carefully designed so that it gives **cryptographically secure pseudorandom sequence**.

Definition 1.9 (Stream Cipher with Initial Value)**.**
Most ciphers, like stream ciphers, have an initial value(IV). The initial value can either be randomly chosen or from a counter.
In stream cipher with IV, the M bit pseudorandom sequence is generated from the secret key *together* with IV. Final ciphertext contains the IV, *followed by* output of the one-time-pad encryption.
For decryption, first extract IV. Generate pseudorandom sequence with key and extracted IV and decrypt the ciphertext.

Remark: In this case, initial value is known to everyone.
The purpose of IV here is to prevent leaking of `XOR`ed plaintexts, as without IVs, keys to encrypt the two plaintexts are the same, which causes `XOR`ed ciphertexts to be equal to `XOR`ed plaintext.

1.3.3 Block cipher & Mode-of-Operations

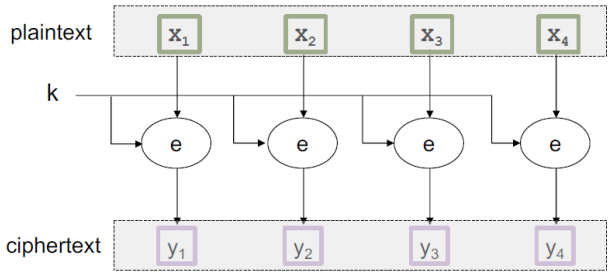
Block ciphers are designed for some **fixed size** input/output. For example, AES is designed for 123 bits input/output.
For large plaintext, it is

1. divided into blocks first, and
2. have block cipher applied

The method of extension of encryption from a single block to multiple blocks is called **mode-of-operation**.

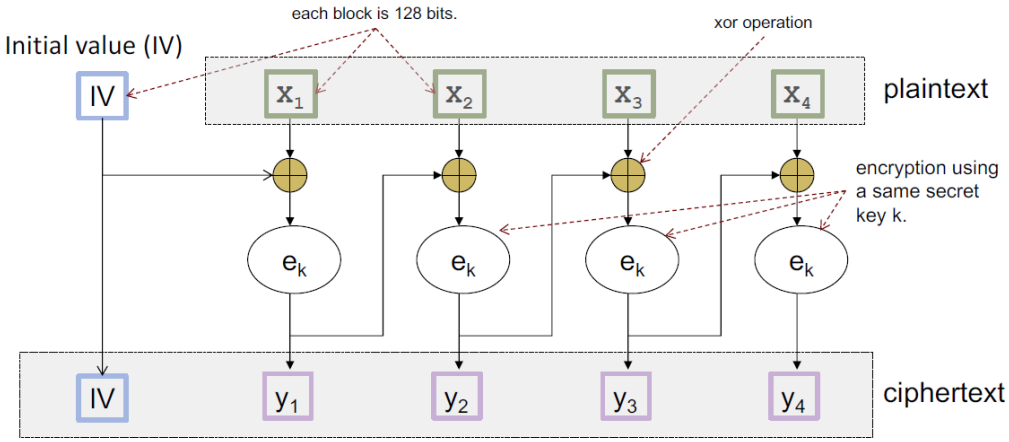
Definition 1.10 (Electronic Code Block(ECB) Mode)**.**
Electronic Code Book is the most simple mode where it

1. Divide the plaintext into blocks, and
2. applies block cipher to each block, **with the same key**.



Remark: ECB might leak information on the plaintext, if a deterministic encryption scheme is used.
Cipher Block Chaining has additional mechanisms so that two blocks with the same content gives different ciphertext.

Definition 1.11 (Cipher Block Chaining(CBC) mode)**.**
In cipher block chaining, the initial value will be used to generate first ciphertext block, which will be used as initial value of the next block.



1.4 Cryptography Pitfalls

1.4.1 Reusing IV/one-time-pad in Encryption

Initial values is not secure if it is predictable.

1.4.2 Keys Generation

Numbers used to generate key must be pseudorandom for a computational perspective, like `java.security.SecureRandom`.

1.4.3 Designing your own cipher

The basic rule is to use existing scheme, and do not design own cipher.

1.4.4 Reliance on Obscurity

Generally, we need to adhere to **Kerckhnoffs' Principle**:

Definition 1.12 (Kerckhoffs' Principle)**.**
A system should be secure even if everything about the system, except the secret key, is public knowledge.

Security through obscurity, that is to hide the design of the system in order to achieve security, should be used to build a layered defense and should not be solely relied on.
In this module, we always assume that the attackers know the algorithms.

2 Authentication

Definition 2.1 (Authentication)**.**
Authentication is the process of assuring that the communicating entity, or origin of a piece of information, is the one that it claims to be.

For this course, authenticity implies integrity.
Authentication concerns communication channel, data, etc.

2.1 Password

In any password system, there are two stages.

1. **Bootstrapping**: server and user establish a common password. Server keeps a file recording the *identity* and *password*.
   This is done by
   - Server/user chooses a password and sends it to the user/server through another communication channel.
   - Default password
2. **Authentication**: server authenticates an entity. It the entity gives the correct password coressponding to the claimed identity, the entity will be deemed authentic.
   This can be done either with or without interactions.

Remark: Password is secret and only authentic user and server knows, whereas identity is not necessarily secret. In other words, the fact that an entity knows the password impliesthat the entity is either the server or the authentic user.

Definition 2.2 (Weak Authentication)**.**
A weak authentication is one that subjected to the **replay attack**: information sniffed from the communicated channel can be used to impersonate the user.

Password system is classified as a weak authentication.

2.1.1 Attacks on Password System

Password system may be compromised via

- Attack on bootstrapping: Password can be intercepted during bootstrapping; attacker uses default password.
- Searching for password:
  - Guessing password from social information
  - Dictionary attacks
  - Stealing password: shoulder surfing, sniffing, viruses, keylogger, login spoofing, phishing, spear phishing, etc.
- Cache of shared workstation
- Insider attack

2.1.2 Preventive measures

Preventive measures to protect password include

- Use strong password that is either **random**, mix of special characters.
- Password policies, which prevents weak passwords during bootstrapping and prevents dictionary attack/guessing by locking account after some failed attempts
- Layered protection on password files through **hashing**

2.1.3 Security Questions

Security question is viewed as a mechanism for **fallback authentication**, or a **self-services password reset**.
Choice of security questions need to be **memorable**, **consistent**, **nearly universal** and **safe**.
Scurity questions enhance "usability" at a cost of security.

2.2 Biometric

Biometric uses unique physical characteristics of a person for authentication. It consists of two stages:

1. **Enrolment**: During enrolment, a template of an user's biometric data is captured and stored.
2. **Verification**: During verification, biometric data of the person-in-question is captured and compared with the template using a *matching algorithm*. The algorithm decides whether to accept or reject.

Unlike password, there are possibilities of error, introducing the two rates below.

$$\text{False Match Rate, FMR} = \frac{\text{number of successful false matches}}{\text{number of attempted false matches}}$$

$$\text{False Non Match Rate, FNMR} = \frac{\text{number of rejected genuine matches}}{\text{number of attempted genuine matches}}$$

The *matching algorithm* typically makes decision based on some adjustable threshold. Usually, there is a tradeoff between FMR and FNMR.
There are other types of errors:

- Equal error rate: the threshold when FMR = FNMR.
- False-to-enroll rate: Some users' biometric data can't be captured during enrolment.
- Failure-to-capture rate: Biometric data may fail to be captured during authentication.

Biometric system is secure if the scanner and the communication channel to matching algorithm is secure. Additional protection may include liveness detection.
The following table contains some differences between biometrics and password.

Password	Biometric
Can be changed	Cannot
Need to remember	Don't have to
Zero non-match rate	Probability of error
Users can pass the password to others	Not possible

2.3 n-factor authentication

n-factor authentication requires at least two different authentication factors, such as

- Something you know, e.g., password
- Something you have, e.g., security token, mobile phone
- Who you are, e.g., Biometric

2.3.1 One Time Password Token

One Time Password Token is a hardware that generates one time password. Each token and the server share some secret keys used to generate the OTP by

1. either **Time-based**: Based on the shared secret and current time interval, a password K is generated which is known to both server and the user
2. or **Sequence-based**: An event triggers the change of the password

Examples of 2FA, like Password + OTP token, are given in the lecture notes and will be omitted here.
n-FA is secure if we assume that the reader, server and the channel of communication are all secure.

3 Authenticity, MAC and Signature

3.1 Type of Cryptography

Definition 3.1 (Symmetric Key Cryptography)**.**
A symmetric key encryption scheme uses the same key for encryption and decryption, i.e.

$$c := E_k(m)$$
$$m = D_k(c)$$

A symmetric key cryptography is secure if it is difficult to get the plaintext from the ciphertext without the key.

Definition 3.2 (Public Key Cryptography)**.**
A public key encryption scheme uses **public key** for encryption and **private key** for decryption, i.e.

$$c := E(k_{\text{public}}, m)$$
$$m = D(k_{\text{private}}, c)$$

A public key cryptography(PKC) is secure if it is difficult to get either the plaintext or the private key from the ciphertext and public key.
One benefit of public key cryptography allows reduction of number of keys used. It also allows establishment of secure channel if one party publishes its public key.

Popular PKC schemes include RSA, which has a key size of 2048 bits, and ElGamal.
Detailed implementation of classroom RSA is covered in `GEH1036.pdf`.
Classroom RSA is *not* secure; the standard for RSA is PKCS#1.

Even for the standard RSA, it has the following problem:

- RSA is much less time efficient compared to AES.
- RSA is not necessarily secure since
    1. It is not known whether getting plaintext from ciphertext and public key is as difficult as getting the private key by prime factorisation.
    2. Homomorphic property of RSA can used to exploit RSA itself.

The performance issue can be resolved by

1. Use PKC to establish a symmetric key cryptography key k.
2. Use k and Symmetric Key Cryptography to encrypt the file.

4 Hash

4.1 Hash with No Secret

Definition 4.1 (Hash)**.**
A cryptographic hash is a function that takes an arbitrarily large message as input and outputs a fixed size **digest**, and observe the following security properties:

- Collision-resistant: It is difficult to find two different messages m_1 and m_2 that hash to the same digest, i.e. $h(m_1) = h(m_2)$.
- One-way: Given a digest d, it is difficult to find a message m, such that $h(m) = d$.

In fact, (1) $\Rightarrow$ (2).

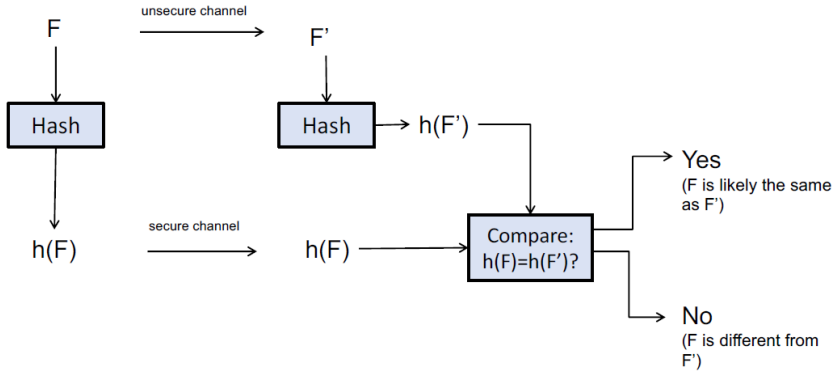
Definition 4.2 (Keyed-Hash(aka MAC))**.**
A keyed-hash is a function that takes an arbitrarily large message **and a secret key** as input, and outputs a fixed size **mac**(message authentication code), and observes the following security property:

- Without knowing the key, it is difficult to forge the mac.

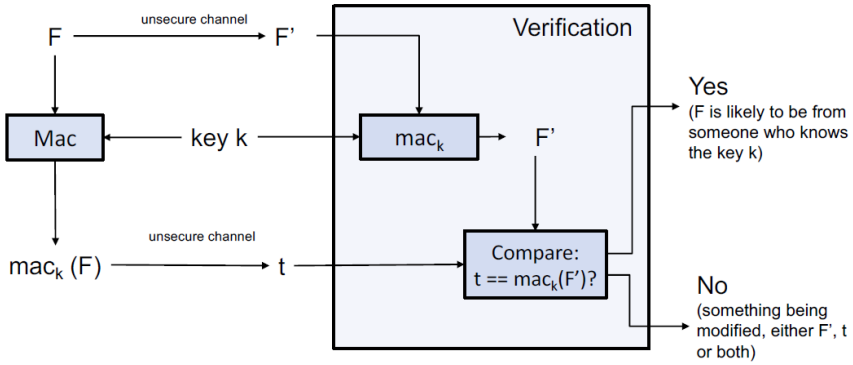
General hashes may not work well for cryptographic purposes. One example is CRC checksum.
Popular and secure hashes include SHA-2, SHA-3.
Popular and secure keyed-hashes include CBC-MAC and HMAC.

4.2 Data Integrity/Authenticity without secret keys

With (unkeyed) hash, the integrity of the file can be checked by the algorithm in the diagram below. **Remark**: The assumption of secure channel for the **distribution of digest** is a



must.

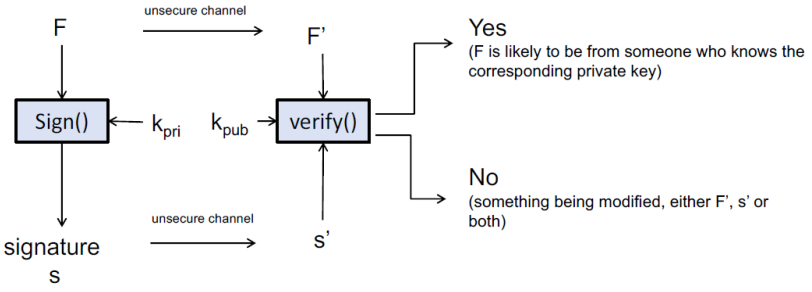


4.3 Data Origin Authenticity with mac and Signature

Data origin can be checked using mac following the algorithm in the next diagram. **Remark**:

- We separate the concern of confidentiality from authenticity.
- Usually mac is appended to F, and then transmitted. As such, mac is also called **authentication tag**.
- The key must be sent to the verifier in a secure channel.

Data origin can alternatively be checked using digital signature, where symmetric key k is replaced by a pair of public and private keys. Signature is better in achieving authenticity



compared to mac since it achieves **non-repudiation**, which means that, no one, except the authentic signer, can forge the signature.
Popular signature schemes include RSASSA-PSS and DSA(Data Signature Algorithm).

4.4 Some Pitfalls

Still, due to the reduced size of digest, it is subject to birthday attack, where it is quite likely to generate a collision from a pool of (file, hash) pairs.
Therefore, we require the length of the digest to be sufficiently large.
NIST reccomend key length for symmetric-key to be 112, and the recommended length of digest to be **at least** 224.

5 Public Key Distribution and Channel Security

5.1 Public Key Distribution

For public key encryption scheme, we need a **secure channel** to distribute the secret key. There are conventionally 3 ways to distribute these keys:

- Public Announcement
- Publishing publicly available directory
- Publick Key Infrastructure

5.1.1 Public Announcement

The owner of publick key can broadcasts her public key via website, etc.
Limitations: not standardised and therefore there is not a systematic way to find/verify the public key when needed.

5.1.2 Publicly Available Directory

A server can be used to store and maintain the public keys of everyone.
Limitations: It is not easy to have a *secure* public directory.

5.1.3 Public Key Infrastructure and Certificate

Public Key Infrastrcture(PKI) is a standardised system that distribute public keys. PKI is centered around two important components:

1. Certificate Authority
2. Certificate

Public Key Infrastructure also provides a mechanism for "trust" to be extended, starting from the "root".

5.2 Public Key Infrastructure

Certificate Authority(CA) keeps a directory of public keys. CA also has its own public-private key pair. We assume that the CA's public key has been securely distributed to all entities involved.

5.2.1 Certificate

Definition 5.1 (Certificate)**.**
A **certificate** is a digital document that contains at least the following 4 main itemz:

1. The identity of an owner, for example, `alice@yahoo.com`

2. The public key of the owner
3. The time window that this certificate is valid
4. The signature of the CA

It also has additional information like the purpose of the public key.

Certificate enables Bob to obtain Alice's public key and verifies its authenticity even without communicating with CA.

5.2.2 Certificate Authority & Trust Relationship

Apart from **issuing certificate**, CA is also responsible to **verify that the information is correct**. This may involve manual checking.

CA also enstablishes a certificate chain:

- Most OS, browsers already have a few CA's public key pre-loaded. These are the "root CA".
- Suppose Alice's certificate is issued by $CA1$, but Bob doesn't have public key of $CA1$.
- In the first place, Alice who anticipates this, can send, along with her certificate, the $CA1$ certificate to Bob.
- Bob then
    1. verify $CA1$'s certificate using root CA's public key.
    2. verify Alice's certificate using $CA1$'s public key.
    3. verify Alice's email using Alice's public key.
- If Alice doesnt attached $CA1$'s certificate, then Bob has to obtain it from other sources.

Above working example illustrates the hierarchy of trust.

5.3 Limitations and Attacks on PKI

5.3.1 Implementation Bugs

One well-known implementation bugs is that some browsers ignore substrings in the name field after the null character when *displaying* it but include them when *verifying* the certificate.

5.3.2 Abuse by CA

A rogue CA can practically forge any certificate.

5.3.3 Social Engineering

Typo-squatting by changing some characters to some others which looks alike.

5.4 Strong Authentication

A weak authentication system, like *password*, is subject to **replay attack**. Strong authentication can prevent replay attack by applying the **challenge-response** scheme.

5.4.1 Strong Authentication under Symmetric Key Cryptography

Suppose Alice and Bob have a shared key k, and have agreed on a message authentication code. The strong authentication can be carried out by

1. Alice identifies herself by sending hello message to Bob
2. Bob sends out **challenge**, which is a *randomly* chosen plaintext m.
3. Alice **responds** Bob with $t = \text{MAC}_k(m)$.
4. Bob verifies t is indeed the MAC of m.

The secret key k ensures the authenticity of Alice. The randomness of m prevents the replay attack.
Such scheme where only Bob verifies Alice, but not vice versa is called **unilateral authentication**.

5.4.2 Strong Authentication under Public Key Cryptography

Suppose Alice wants to authenticate Bob,

1. Alice **challenges** Bob with a random number r.
2. Bob **responds** to the challenge by using his private key to sign r. Bob also attaches his certificate, in case Alice does not know his public key.
3. Alice verifies Bob's certificate, extracts Bob's public key from the certificate, and verifies that the signature is correct.

Here, the value r is also known as the **crypotgraphic nonce**.

5.5 Key Exchange and Authenticated Key Exchange

Strong authentication prevents the attack from eavesdropper, but does not prevent the malicious attacker. With authenticated key exchange, we can verify the authenticity as well as **establish a common key** between two parties.

5.5.1 Key Exchange

Under key exchange, we *assume* authenticity and hope to achieve key establishment secure in the sense that eavesdropper fails.
One scheme is Diffie-Hellman key exchange, which achieve this purpose and *forward secrecy*. Following is a key exchange based on Public Key Cryptography.

1. Alice generates a pair of private/public key.
2. Alice sends the public key k_e to Bob.
3. Bob
   (a) Randomly chooses a secret k,
   (b) Encrypts k using k_e,
   (c) Sends the ciphertext c to Alice.
4. Alice uses her private key k_d to decrypt and obtain k.

However, the above scheme fails when mallory presents, who can establishes 2 different key with Alice and Bob as the **man-in-the-middle**.

Therefore, to account for authenticity, we require **authenticated key exchange**. One such scheme is **station-to-station protocol**.

5.6 Securing Communication Channel

Suppose a communication channel is subjected to sniffing and spoofing, the typical design in TLS to secure it is to

- Use **long-term keys** for handshake protocol, in order to establish **session keys**
- Subsequent communication is protected by session keys

Unilateral handshake is achieved in the following steps. Here, Alice wants to authenticate Bob.

1. Alice obtains Bob's public key through PKI.
2. Alice and Bob carry out **unilateral authenticated key exchange** protocol with Bob's public/private key.
   After authentication, Alice and Bob know **two** *randomly* selected **session keys**: t and k. t is the secret key of a MAC, whereas k is the secret key of a symmetric-key encryption like AES.
3. Subsequent communication between Alice and Bob will be protected by t, k and a `sequenceNumber`.
   Suppose $m_1, \ldots, m_n$ are the sequence of messages exchanged, the actual data to be sent for m_i will be
   $$E_k(i \oplus m) \oplus \text{mac}_t(E_k(i \oplus m))$$

Here, the session keys t and k ensures freshness to prevent replay attack. `SequenceNumber` ensures integrity, preventing reordering and dropping. E_k ensures confidentiality and $textmac_t$ of the encrypted message ensures authenticity.

6 Terminologies

Tutorial 1:

- Cryptology: the study of codes, or the art of writing and solving them.
- Cryptanalysis: the study of analyzing information systems in order to study the hidden aspects of the systems.
- Cryptography: the practice and study of techniques for secure communication in the presence of third parties called adversaries.
- NSA: National Security Agency
- NIST: National Institute of Standards and Technology
- cryptograrphy backdoor: a mechanism whereby one can alter a specific algorithm, providing the perpetrator with a means to break the resulting cipher in significantly less time than would normally be required.
- Key Escrow: an arrangement in which the keys needed to decrypt encrypted data are held in escrow so that, under certain circumstances, an authorized third party may gain access to those keys.
- Decryption order: legislation that requires individuals to surrender cryptographic keys to law enforcement.
- Whitfield Diffie: an American cryptographer and one of the pioneers of public-key cryptography.
- Ron Rivest: a cryptographer and an Institute Professor at MIT.
- Alice, Bob, Eve, Mallory, Trent
  - Alice and Bob: The original, generic characters. Generally, Alice and Bob want to exchange a message or cryptographic key.
  - Eve: Evesdropper, who is usually a passive attacker. While she can listen in on messages between Alice and Bob, she cannot modify them.
  - Mallory: A malicious attacker who is active and who can modify messages, substitute messages, or replay old messages.
  - Trent: A trusted arbitrator, who acts as a neutral third party.

Tutorial 2:

- Graphical Passwords: an authentication system that works by having the user select from images, in a specific order, presented in a graphical user interface (GUI).

- Covert channel: a type of computer security attack that creates a capability to transfer information objects between processes that are not supposed to be allowed to communicate by the computer security policy.
- Side channel attack: any attack based on information gained from the physical implementation of a computer system, rather than weaknesses in the implemented algorithm itself (e.g. cryptanalysis and software bugs)
- End-to-end encryption: a system of communication where only the communicating users can read the messages. The systems are designed to defeat any attempts at surveillance or tampering because no third parties can decipher the data being communicated or stored.

Tutorial 3:

- Single Sign-On(SSO): A user logs in with a single ID and password to gain access to a connected system or systems without using different usernames or passwords, or in some configurations seamlessly sign on at each system.
- hardware random number generator: (true random number generator, TRNG) is a device that generates random numbers from a physical process, rather than a computer program. Such devices are often based on microscopic phenomena that generate low-level, statistically random "noise" signals, such as thermal noise, the photoelectric effect, involving a beam splitter, and other quantum phenomena.
- Quantum random Number Generator: a hardware random number generator that uses quantum phenomena.
- Authenticated encryption (AE) and authenticated encryption with associated data (AEAD, variant of AE) is a form of encryption which simultaneously provides confidentiality, integrity, and authenticity assurances on the data.
- Retinal vs. Iris scan