SE6003 CRYPTOGRAPHY

WEEK 1

Introduction

The branch of cryptology dealing with the design of algorithms for encryption and decryption, intended to ensure the secrecy and/or authenticity of messages.

[CRYPTOLOGY] = [CRYPTOGRAPHY] + [CRYPTANALYSIS]

- [CRYPTOLOGY]: The study of secure communications, which encompasses both cryptography and cryptanalysis.
- [CRYPTANALYSIS]: The branch of cryptology dealing with the breaking of a cipher to recover information, or forging encrypted information that will be accepted as authentic.

Cryptographic schemes can be categorized by:

① types of keys

↳ symmetric (most normos assume this is what cryptography is about)

* encryption and decryption methods share one key.
* all cryptography from ancient times to 1976 was exclusively based on symmetric methods
* still widely used especially for data encryption and integrity check of messages

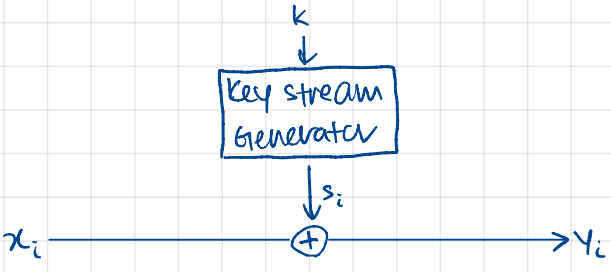
↳ asymmetric (public key cryptography, PKC)

* both sender and recipient have 2 keys — 1 public and 1 private
* scheme proposed by Whitfield Diffie, Martin Hellman and Ralph Merkle in 1976
* used for applications such as digital signatures and key establishment, as well as classical data encryption

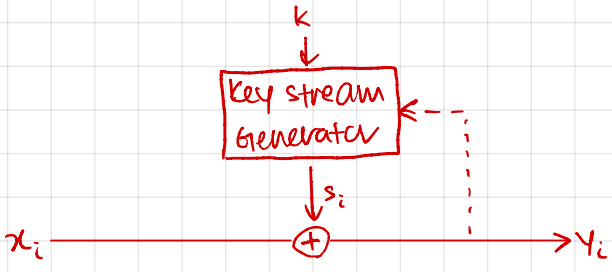
② processing mechanisms

↳ stream cipher: message (plaintext) is processed as a stream of bit strings

* encrypts bits individually
  → achieved by adding a bit from a key stream to a plaintext bit
* 2 types:
  → synchronous stream ciphers: output depends only on the key
  → asynchronous stream ciphers: output depends on both key and ciphertext



Synchronous Stream Cipher



Asynchronous Stream Cipher

* most practical stream ciphers are synchronous ones

↳ block ciphers
* encrypts an entire block of plaintext bits at a time with the same key.
  ⇒ encryption of any plaintext bit in a given block depends on every other plaintext bit in the same block
* vast majority of block ciphers either have block length of 128 bits (16 bytes) such as the Advanced Encryption Standard (AES), or block length of 64 bits (8 bytes) such as the Data Encryption Standard (DES) or triple DES (3DES) algorithm

Notes about Stream & Block Ciphers:

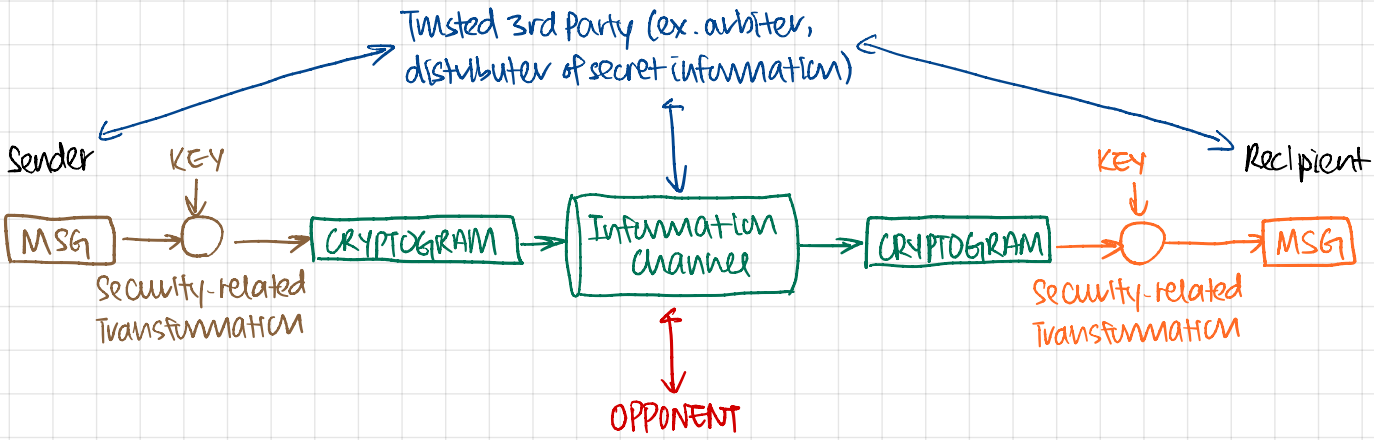
(1) In practice, in particular for encrypting computer communication on the Internet, block ciphers are used more often than stream ciphers.

(2) Because stream ciphers tend to be small and fast, they are particularly relevant for applications with little computational resources, ex., for cell phones or other small embedded devices.
A prominent example for a stream cipher is the A5\1 cipher, which is part of the GSM mobile phone standard and is used for voice encryption.
However, stream ciphers are sometimes also used for encrypting Internet traffic, especially the stream cipher RC4.

(3) Traditionally, it is assumed that stream ciphers tended to encrypt more efficiently than block ciphers.
Efficient for
→ software-optimized stream ciphers means that they need fewer processor instructions (or processor cycles) to encrypt one bit of plaintext.
→ hardware-optimized stream ciphers means that they need fewer gates (or smaller chip area) than a block cipher for encrypting at the same data rate.

However, modern block ciphers such as AES are also very efficient in software. Moreover, for hardware, there are also highly efficient block ciphers, such as PRESENT, which are as efficient as very compact stream ciphers.

The Abstract Communication Model:



6 High-Level Security Goals:

① Confidentiality: property that information is not made available or disclosed to unauthorized individuals, entities or processes.

② Integrity: property of protecting the accuracy and completeness of assets (ie., anything that has value to the organization)

③ Availability: property of being accessible and usable upon demand by an authorized entity

④ Authenticity: property that an entity is what it claims to be.
↳ Authorization: provision of assurance that a claimed characteristic of an entity is correct

⑤ Authorization: the decision to permit or deny a subject access to system objects (network, data, application, service, etc.)

⑥ Accountability: assignment of actions and decisions to an entity

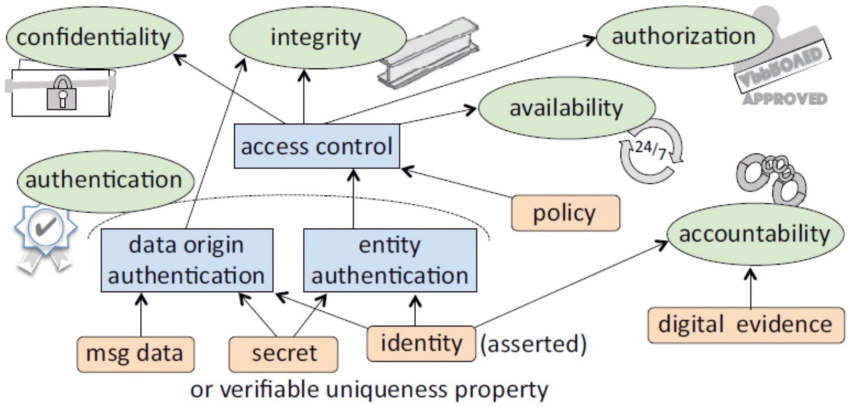
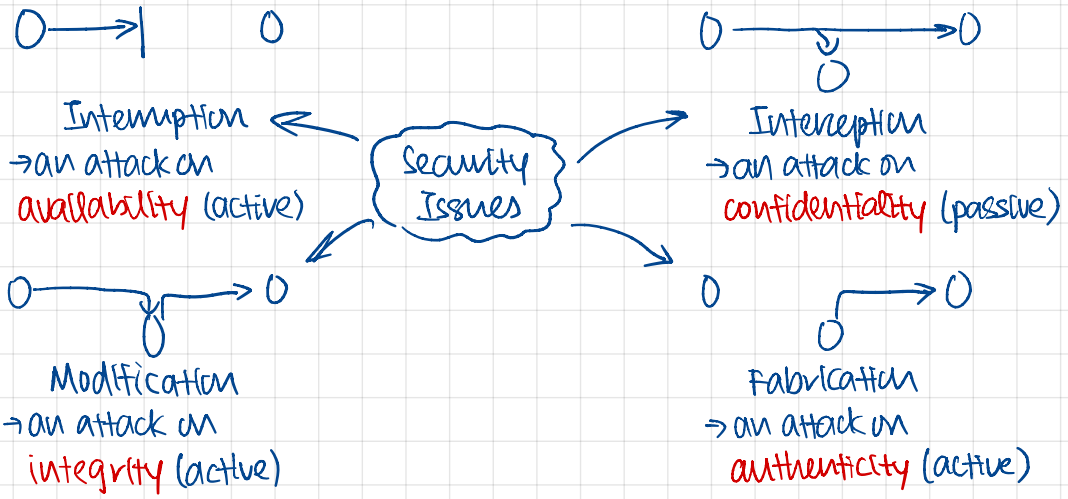


Figure 1.1: Six high-level computer security goals (properties delivered as a service). Icons denote end-goals. Important supporting mechanisms are shown in rectangles.

Refer: Paul C. van Oorschot, Computer Security and The Internet: Tools and Jewels, Springer 2020.



Active Attack: an attempt to alter the system resources or affect their operation
↳ involves some modification of the data stream or the creation of a false stream
↳ goal of defense: detect attacks and to recover from any disruption or delays caused by them
↳ difficult to prevent ∵ wide variety of potential physical, software & network vulnerabilities
↳ examples → masquerade: takes place when one entity pretends to be a different entity; usually includes one of the other forms of attack
→ replay: involves the passive capture of a data unit and its subsequent retransmission to produce an unauthorized effect
→ modification of messages: some portion of a legitimate message is altered, or messages are delayed or reordered to produce an unauthorized effect
→ denial of service: prevents or inhibits the normal use or management of communications facilities

Passive Attack: an attempt to learn or make use of information from the system that does not affect system resources
↳ goal of attacker: obtain information that is being transmitted
↳ 2 types → eavesdropping communications and releasing of messages
↳ traffic analysis on the identities, locations, frequency, etc. of communications
↳ difficult to detect ∵ do not involve any alteration of data

Protocols: rules or standards that are agreed upon to enable connection and interaction between parties
↳ can specify — data formats
└ rules of exchange (ie., who does what when?)
└ specify termination or error rules or handling conditions

Cryptography development is closely related to:
① computing devices (cipher should be computed easily)
↳ primitive methods allow only simple & normally weak cipher implementations

Paper & Pencil → Electromechanical computing device → Electronic computing device

Paper & Pencil: simple & normally weak ciphers

Electromechanical computing device: rotor machines from 1920s - 1960s

Electronic computing device: modern ciphers (ex. DES, AES, RSA, ...)

② Communication techniques
↳ radio telegraph (wireless communication)
→ message interception is easy ⇒ strong ciphers needed
↳ computer network
→ How can 2 computers communicate secretly if the two computers do not share any secret key before the communication starts?
⇒ led to public-key cryptography in 1970s (aka-revolution)

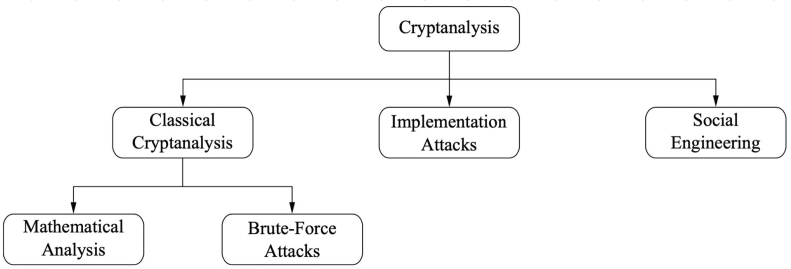
Significance & Limitation of Cryptography:
↳ cryptography is the foundation of cybersecurity
(weak ciphers ⇒ weak information system)
↳ However, using strong ciphers does not guarantee the security of an information system.

Modern cryptology
↳ aims to protect privacy and integrity
↳ is complexity-based,
using computational assumptions.
→ All participants are computationally-bounded algorithms.
→ There are computational problems that cannot be solved by bounded algorithms.
↳ in its abstract model expresses
→ objects as information bits
→ actions as (digital) communications

Cryptology forms a great example of the unreasonable effectiveness of mathematics.
↳ most research in cryptology is initially driven by curiosity

There is no mathematical proof of security for any practical cipher.
↳ The only way to have assurance that a cipher is secure is to try to break it (and fail)!
Hence, the need for cryptanalysis!

Overview of cryptanalysis:



Classical cryptanalysis: the science of recovering plaintext message or key from ciphertext\cryptogram
↳ can be divided into:
* mathematical analysis
ex. letter frequency analysis (analyzing statistical properties of encrypted plaintext)
* brute-force attacks: attempt decryption of ciphertext with all possible keys to match a known plaintext (ex. header of an encrypted file)
→ adversary obtains plaintext & ciphertext via eavesdropping on channel
⇒ can be more complicated ∵ incorrect keys can give false positive results
→ brute-force attack against symmetric ciphers is always possible in-principle; feasibility in practice depends on number of possible keys that exist for the cipher
↳ if testing all keys on many modern computers take too much time, cipher is computationally secure against a brute-force attack.

Def^{n} (Basic Exhaustive Key Search \ Brute-Force Attack)
↳ Let (x, y) denote the pair of plaintext and ciphertext, and let $K = \{k_1, k_2, \dots\}$ be the key space of all possible keys k_i.
A brute-force attack now checks for every $k_i \in K$ if $D_{k_i}(y) \stackrel{?}{=} x$.
If the equality holds, a possible correct key is found; if not, proceed with the next key.

* How many keys do we need?

Key length	key space	Security Lifetime Estimation (assuming brute-force attacks)
64 bits	2^{64}	short term (few days or less)
128 bits	2^{128}	Long term (several decades in the absence of quantum computers)
256 bits	2^{256}	Long term (also resistant against quantum computers)

Note: An adversary only needs to succeed with just one attack. Thus, a long key space does not help if other attacks (ex. social engineering) are possible.

Important:
① The discussion for key lengths for symmetric crypto algorithms is only relevant if a brute-force attack is the best known attack.
② The key lengths for symmetric and asymmetric algorithms are dramatically different. For instance, an 80 bit symmetric key provides roughly the same security as a 1024-bit RSA key. (RSA is a popular asymmetric algorithm.)

Implementation Attacks: → extract information from a noisy signal
↳ ex. [side-channel analysis] to obtain a secret key
→ measuring electrical power consumption of a processor which operates on the secret key
→ power trace can then be used to recover the key by applying signal processing techniques
→ in addition, electromagnetic radiation or runtime behavior of algorithms can give info abt secret key.
↳ mostly relevant against cryptosystems to which adversary has physical access, such as smart cards
↳ usually not a concern in most Internet-based attacks against remote systems.

Social Engineering
↳ a manipulation technique that exploits human error to gain private information, access or valuables
ex. bribing, blackmailing, tricking or classical espionage can be used to obtain a secret key by involving humans

Other possible attacks:
↳ buffer overflow attacks
↳ malware

An attacker always looks for the weakest link in your cryptosystem. That means we have to choose strong algorithms and we have to make sure that social engineering and implementation attacks are not practical.

Solid cryptosystems should adhere to Kerckhoff's principle.

Kerckhoff's Principle ("Desiderata de la Cryptographie Militaire", 1883):

> "It must not require secrecy and it can, without disadvantage, fall into the hands of the enemy."

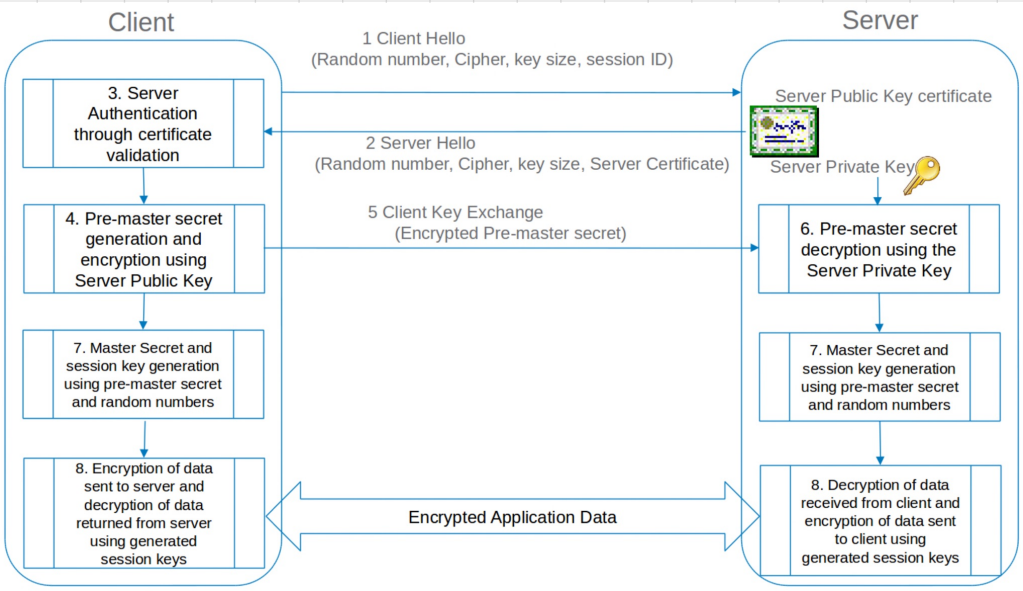
In order to achieve Kerckhoff's principle in practice,
→ Only use widely-known ciphers that have been cryptanalyzed for several years by good cryptographers!

Remark: It is tempting to assume that a cipher is "more secure" if its details are kept secret. However, history has shown time and again that secret ciphers can almost always be broken once they have been reverse engineered.
ex. Content Scrambling System (CSS) for DVD content protection

Shannon Maxim ("A Mathematical Theory of Cryptography", Sept. 1945):

> "The enemy knows the system"

A Schematic Picture of Transport Layer Security:



Some New and Emerging Topics in Cryptography:
→ Distributed ledgers
→ Privacy-Preserving Cryptography (ex. Confidential computing in the cloud)
→ Quantum-Secure Cryptography
→ Threshold cryptography
⋮