COMP5631 Review

Yilun Jin

yilun.jin@connect.ust.hk

December 2020

Outline

Table of Contents

Contents of this course

- Security Backgrounds
  - Security Services
- Cryptography
  - Mathematical Backgrounds
  - One-key cipher
  - Public key cipher
  - Key management
  - Hash functions
  - ...
- Applications
  - Distributed Systems
  - Digital Signature
  - E-mail
  - Network Security: IPSec, SSL, VPN, Firewall

Table of Contents

Security Services

What are the security services covered in the course?

- Confidentiality: Outsiders do not know what is transferred.
- Authentication: Alice is indeed Alice, Bob is indeed Bob.
- Integrity: The message is authentic and not tampered with.
- Non-repudiation: Alice cannot deny her sending out a certain message.
- Anti-replay: You cannot intercept a message at 2 p.m. and resend it at 5 p.m.

Security Services

What are the security services covered in the course?

- Confidentiality: Outsiders do not know what is transferred.
- Authentication: Alice is indeed Alice, Bob is indeed Bob.
- Integrity: The message is authentic and not tampered with.
- Non-repudiation: Alice cannot deny her sending out a certain message.
- Anti-replay: You cannot intercept a message at 2 p.m. and resend it at 5 p.m.

They are key concepts in this course, and you should understand them, including when they should be provided and how to achieve them.

Attacks

What are the attack models to a security system mentioned in this course?

Attacks

What are the attack models to a security system mentioned in this course?

- Passive Attack: the attacker can only see but not modify the communication.
- Active Attack: the attacker can control the communication channel and modify contents.
- Also for one-key and public-key ciphers, we have known-ciphertext attacks, known (m, c) pair attack, and known public key attack.

Comments: Please be careful about the assumptions when we talk about security, as security must come with certain conditions.

Table of Contents

Theory: Cryptography

This part mainly answers a question: how to provide these security services listed above?

- Confidentiality
- Authentication
- Integrity
- Non-repudiation
- Anti-replay

Mathematical Background

Mathematics (discrete maths) are building blocks of cryptography:

- Sets
- Functions
- Greatest Common Divisor (GCD)
- Modulo arithmetic
- Euclidean algorithm
- Multiplicative inverse
- Finite field Z_p or $Z_p[x]$ where p is a prime.

One-key Ciphers

- $(\mathcal{M}, \mathcal{C}, \mathcal{K}, E_k, D_k)$.
- Security of one-key ciphers: we face two attacks, knowing ciphertext-only, and knowing plaintext-ciphertext.
- Simple examples: transposition and simple substitution ciphers.
  - Some of you got the concepts wrong in Assignment 2 regarding transposition and substitution.

One-key Ciphers

- Common one-key ciphers: AES, DES, ...
  - **A general suggestion**: Real world cryptographic applications are generally very complex for security reasons. Therefore, it is neither necessary nor possible to remember them.
    - You won't be asked to compute, e.g. an AES encryption.
    - However, you need to know some principles and design ideas. e.g. What is diffusion/confusion.
- Modes: ECB, CBC. Similarly, you do not need to remember, but you should be able to analyze.
  - e.g. what are the advantages and disadvantages (maybe from the perspectives of error-prone, efficiency and security levels).

Key Distribution

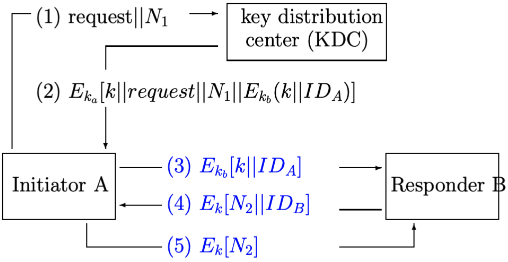
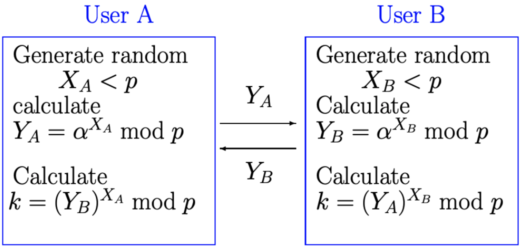


Figure 1: One Key Distribution Protocol Mentioned

- Key distribution, by itself, requires security services.
- For each protocol, it is good for you to understand why these transmissions are necessary. e.g. why nonce, why $E_{k_b}(k\|ID_A)$, what services can it provide.

Key Distribution



- Diffie-Hellman: Passive and active attacks, discrete log problem.

One-key cipher

What security services can one-key ciphers provide?

One-key cipher

What security services can one-key ciphers provide?
In general:

- Confidentiality. Note that this is achieved with adequate key distribution.
- Authentication. If A and B share a key k, then a nonce can be used to authenticate each other (also known as challenge-response).

Public-key ciphers

- Definition: $(\mathcal{M}, \mathcal{C}, \mathcal{K}_e, \mathcal{K}_d, E_k, D_k)$
- Security: Given public key (a.k.a. encryption key), cannot derive private key (a.k.a. decryption key).
  - The security guarantee may not be straightforward. Recall Assignment 2, question about RSA.
- One appealing property: key distribution becomes not necessary.
- Applications:

Public-key ciphers

- Definition: $(\mathcal{M}, \mathcal{C}, \mathcal{K}_e, \mathcal{K}_d, E_k, D_k)$
- Security: Given public key (a.k.a. encryption key), cannot derive private key (a.k.a. decryption key).
  - The security guarantee may not be straightforward. Recall Assignment 2, question about RSA.
- One appealing property: key distribution becomes not necessary.
- Applications:
  - Confidentiality: A encrypt a message m with B's public key k_B^e, and send to B. (Note: In practice, it is generally avoided. Why?)
  - Key distribution: A generates a session key k, and send $E_{k_B^e}(k)$ to B.
  - Non-repudiation: e.g. A sending out $m \| D_{k_A^d}(h(m))$.

Public-key ciphers

RSA, ElGamal and related concepts (Euler functions, etc.)

- You need to understand questions like: why this is secure, what if some rules are broken, etc. (Recall Assignment 2 about the common factor.)

Public key infrastructure and Digital certificate

- Hierarchical CA structure. Why?

Public-key ciphers

RSA, ElGamal and related concepts (Euler functions, etc.)

- You need to understand questions like: why this is secure, what if some rules are broken, etc. (Recall Assignment 2 about the common factor.)

Public key infrastructure and Digital certificate

- Hierarchical CA structure. Why?
- Scalability requires distributed CAs.
- For example, for a course with 300 students, we generally need 2 lecturers, with each lecturer coordinating approx. 3 TAs.

Public-key ciphers

- What security services can public-key ciphers provide?

Public-key ciphers

- What security services can public-key ciphers provide?
- Generally,
  - Confidentiality
    - e.g. Public-key cryptography is commonly used for key distribution.
  - Authentication
    - For example, you may have used the command `ssh-keygen` that allows you to ssh to a server without password.
  - Non-repudiation (by digital signature, detailed later)

Hash Functions

- Definition: $h : \mathcal{A} \to \mathcal{A}'$, where $\mathcal{A}'$ denotes some **fixed length** strings.
- Property: one-way (Given x, hard to find m, s.t. $h(m) = x$); weak collision resistance (Given x, hard to find y, s.t. $h(x) = h(y)$).
  - You can think about, what will happen if they are violated.
  - Note the difference between the following:
    - Given x, it is hard to find y, s.t. $h(x) = h(y)$.
    - It is hard to find x, y, s.t. $h(x) = h(y)$.
    - Which implies the other? Google "the birthday paradox" for a good explanation.
  - Also note, in general, hash functions are always publicly known. This makes it more important to design non-colliding hash functions (need to withstand attacks from everyone).

Hash Functions

- Instances: HMAC, SHA1.
  - Again you do not need to remember the details (because the details are very complex due to security reasons).
- One **common** application: Digital Signature.
- Security Services: Generally speaking, hash functions can provide integrity.
  - For example, when we download some large files, we may want to do a MD5 verification.
  - The file owner provides file x, and a precomputed MD5 value v.
  - The user downloads x' and verifies $MD5(x') = v$.

Table of Contents

Security Protocols

Lecture 13 introduces various protocols that provide various security services.

- Again, you do not need to remember (as one can create a new protocol very easily), but you should understand, why a protocol provides certain services, and what can be the vulnerabilities.
- e.g. $m \| h(m), m \| D_{k_A}[h(m)], m \| E_k(h(m))$

A general notice on real-world applications

The lecture notes may introduce many details, which you don't need to remember all. However, it is always important that you know what are the **security services needed** for each application, and why. Also, what the **tools** are, and what the high level **ideas** are.

- e.g. Digital Signature - What is it designed for? - design ideas.
- PGP - email security - What do we need for emails - tools - design ideas.
- Kerberos - Distributed system access control - tools - why so designed.

Digital Signature

- The primary question: what is a digital signature?
  - Analogous, but **not completely the same** to handwritten signatures - authentication, integrity and non-repudiation.
  - Then, what are the design requirements?

Digital Signature

- The primary question: what is a digital signature?
  - Analogous, but **not completely the same** to handwritten signatures - authentication, integrity and non-repudiation.
  - Then, what are the design requirements?
  - e.g. easy to verify, vary according to contents, etc.
- Currently used one: DSS, RSA.

Secret Sharing

- (Personally I think this topic should appear earlier, in the "Theory" part.)
- Shamir (t, n)-thresholding scheme involves an order $t-1$ polynomial.

$$a(x) = \left(s + \sum_{i=1}^{n} a_i x^i\right) \mod p$$

  - There are many symbols here (s, n, x, p). You need to be careful.
  - Security proof: via Linear Algebra.

E-mail Security

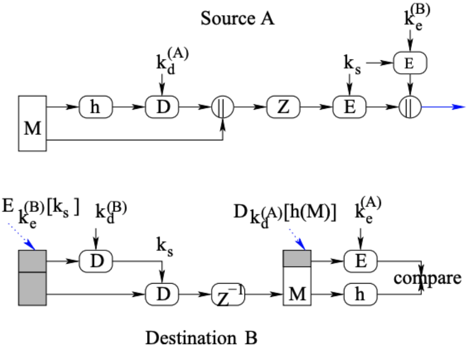
- What are the services that are needed?

E-mail Security

- What are the services that are needed? Non-repudiation, integrity, confidentiality, authentication.
- What tools to use?

E-mail Security

- What are the services that are needed? Non-repudiation, integrity, confidentiality, authentication.
- What tools to use? Digital signature, public-key cipher, one-key cipher (for confidentiality), zip, etc.
- Why so designed? You'd better understand a little, e.g. why zipping before encryption.



Distributed System Security

- What kind of attacks is a distributed system vulnerable to, and consequently, what security services should we provide? What are the challenges?
  - Basically, distributed authentication is needed.
- Kerberos is designed to solve **authentication** for distributed systems.
  - Authentication Server (AS), Ticket Granting Server (TGS).
  - The procedure can be connected with some real-world examples.
    - For example, I send an email to my advisor, to ask whether I can gain access to CYT3007. My advisor replies yes. I then send the email to the CYT3007 manager, who will grant me access.
  - More specifically, what protocols are involved, how to authenticate. (You have a question in Assignment 3 on this.)

Network Security: IP Security

- IP is not robust. (As your networking course may tell)
- What does IPSec provide, and how?
  - Authentication, anti-replay (important for network services), integrity.
  - But **not necessarily** confidentiality.
- Two protocols, ESP (encapsulating **secure** payloads) and AH (**authentication** header), what are the differences?
  - The names tell some of the difference.
  - You do not need to remember the detailed formats.
- Transport mode VS tunnel mode.
  - Analogy: A friend in the US wants to send a package to your home in mainland China, but he does not know Chinese.

Web Security: SSL

SSL:

- Remember that it is built upon TCP, which should be **connection-oriented** and **reliable**.
- Concepts: Session, connection and states, and their relationships.
  - A session is shared for multiple connections.
  - There are both connection and session states.
  - What are in the states, and which of them change between sessions/connections?
  - What is a pending state? (Recall the "change cipher spec".)
- Protocols: Handshake, alert, record.
  - What information is exchanged during handshake?
  - The handshake is more complex than the TCP handshake, why?

Firewall

- Purpose: Basically access control.
- Different types of firewalls:
  - Packet filtering, Session filtering.
  - Circuit gateways, application gateways...
  - This lecture basically involves little cryptographic protocols and is relatively easy to understand.

VPN

VPN: **Virtual Private Network**

- What does it provide? Recall "Virtual Private"

VPN

VPN: **Virtual Private Network**

- What does it provide? Recall "Virtual Private"
- Authentication, confidentiality (virtual), and integrity (required by all network services)
- Key technique: Tunnel, which supports "virtual".
  - The data is encapsulated with a header which contains routing information.
  - Encapsulated frames are transported over the Internet like ordinary frames.
  - Recall tunnel mode in IPSec.
- PPTP, L2TP. No details required, but understanding of security services is required.

Secure Shell

Establish a secure channel for two computers for remote control, file transfer, etc.

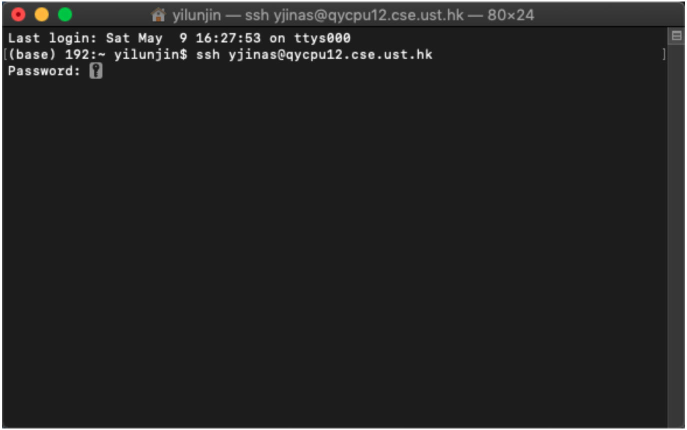


Figure 2: My Secure Shell to connect with lab machines.

Secure Shell

- Three layer: Transport, User Authentication, Connection
- Transport Layer does the following:
  - Parameter Negotiation:
  - Key exchange: Exchange master key, and each end derives private key. (This is similar to SSL.)
  - Server authentication: Each server has a public-private key pair. On my computer there is a file `known_hosts`, that saves a list of known hosts.
- User Authentication: The server authenticates the user by e.g. public key (Maybe you have used the `ssh-keygen`), or password.

Table of Contents

Remarks on final exam

The final exam will be on December 10th, open-book, take-home, and will only consist of several (maybe 10 or 20) multiple choice questions.

- However, do not think them as easy questions. They will be hard, and you won't expect to find answers on textbooks or slides.
- As I said before, open-book means that remembering details is useless. Rather you should understand the relationship between techniques/protocols and security services, and why is the relationship.
  - e.g. what services can a tool provide, and how?
  - What problems are considered as 'hard' problems in cryptography?
  - Given a protocol, what can it provide?
  - Given an application, what security property should it have?

Questions?

Thanks!

- TA: Yilun Jin
- yilun.jin@connect.ust.hk
- Feel free to ask questions related to the course via email.