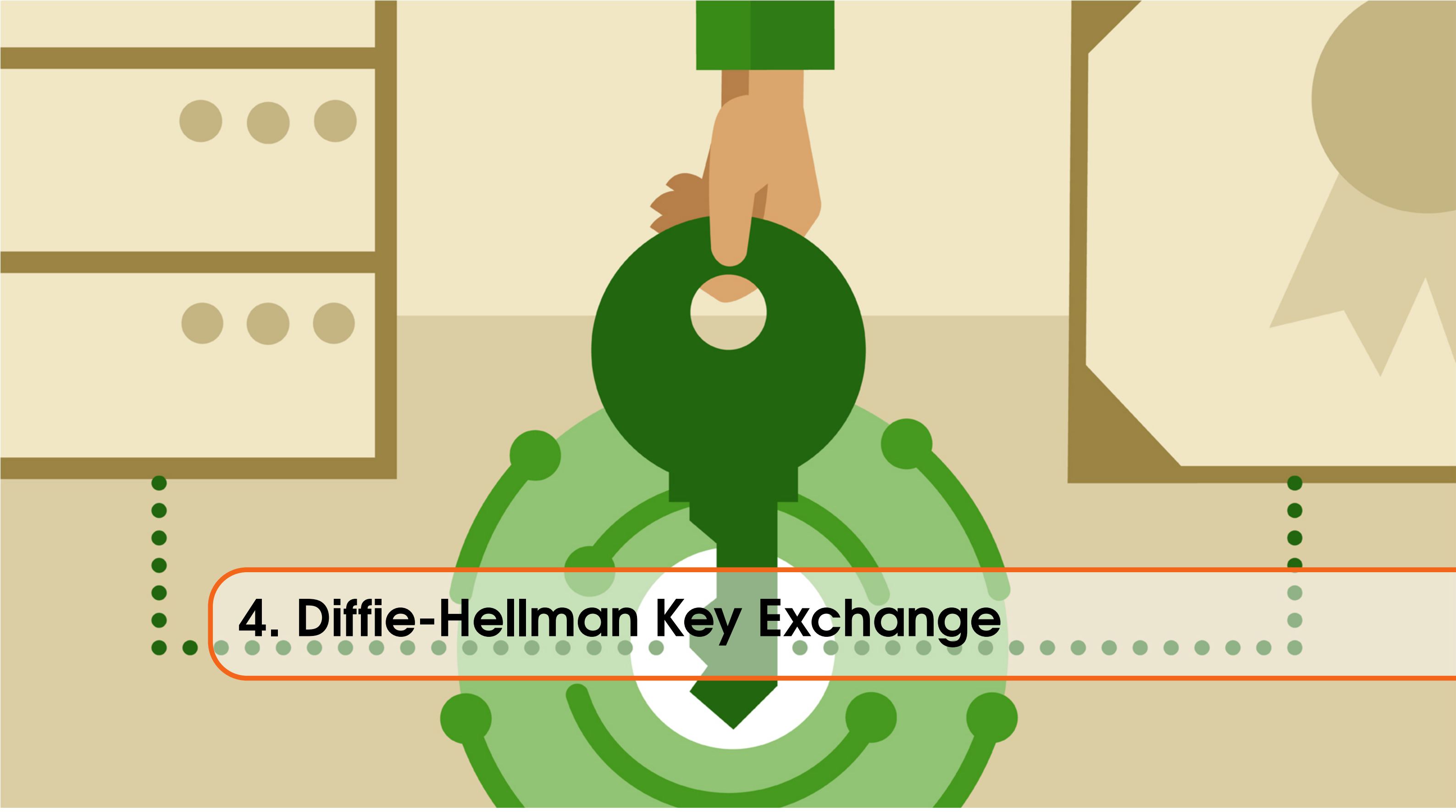

4. Diffie-Hellman Key Exchange

4.1 OVERVIEW

4.1.1 Introduction and learning objective

The Diffie-Hellman (DH) key exchange is a method of securely exchanging cryptographic keys (can be used for symmetric ciphers) over a public channel and was one of the earliest and simplest Public Key Cryptography Standards (PKCS). DH algorithm depends for its effectiveness on the computing **discrete logarithms**.

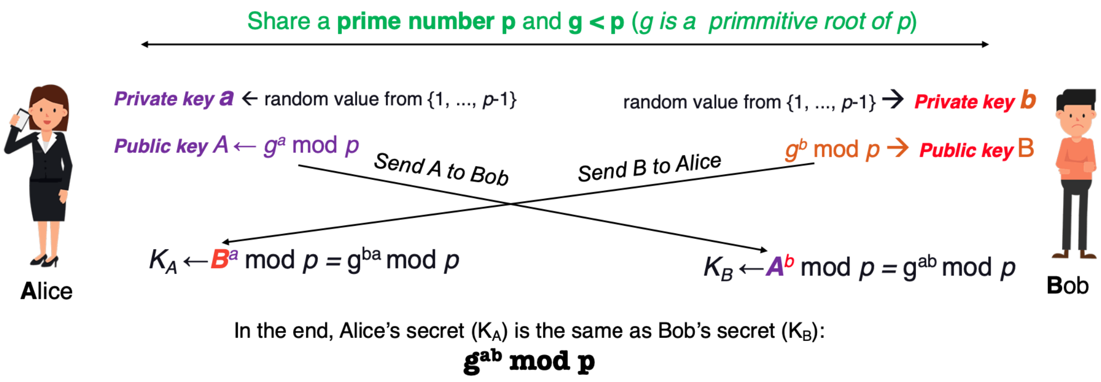
Share a **prime number p** and **g < p** (*g is a primmitive root of p*)
Private key a ← random value from {1, ..., p-1}
Public key A ← g^a mod p
random value from {1, ..., p-1} → *Private key b*
g^b mod p → *Public key* B
Send A to Bob
Send B to Alice
Alice
Bob
$K_A \leftarrow B^a \bmod p = g^{ba} \bmod p$
$K_B \leftarrow A^b \bmod p = g^{ab} \bmod p$
In the end, Alice's secret (K_A) is the same as Bob's secret (K_B):
g^{ab} **mod p**

Figure 4.1: The Diffie–Hellman Key Exchange

The DH key exchange can be briefly described as follows:

1. Firstly, Alice and Bob share a prime number **p** and an integer **g** such that **g<p** and **g** is a **primitive root** of **p**.
   *Note that **p** and **g** are publicly known and don't need to be kept secret.*

2. Subsequently, Alice selects a random integer **a** such that **a < p** and keep it secret. Bob also independently selects a random integer **b** such that **b < p** and keep it secret (**a** and **b** are called private keys)
3. Then, Alice computes $A = g^a mod p$ and send to Bob. Similarly, Bob computes a $B = g^b mod p$ and send to Alice. Thus a and A is Alice's corresponding public key, and similarly for Bob.
4. Finally, each side compute the shared secret K. Alice computes:
   $K_A = B^a mod\, p = (g^b)^a mod\, p = g^{ba} mod\, p$
   Bob computes:
   $K_B = A^b mod\, p = (g^a)^b mod\, p = g^{ab} mod\, p$
   These two calculations produce identical results: $K = g^{ab} mod\, p$. The result is that the two sides have exchanged a secret value. Typically, this secret value is used as shared symmetric secret key.

The learning objective of this lab is for students to gain hands-on experiences on the Diffie-Hellman key exchange. From lectures, students should have learned the theoretic part of the DH algorithm, so they know mathematically how Alice and Bob share the secret key through an insecure channel. Essentially, students will be implementing the DH algorithm to secure their chat application (which was developed in Network Programming course).

4.1.2 Background

To complete this lab well, you are expected to gain knowledge about:
- The Diffie-Hellman key exchange
If you are not familiar with this algorithm, you should find out in more detail in:
Chapter 10: Other Public-key Cryptosystems *W. Stallings, CS book - Cryptography and network security: Principles and practice, 7th ed. Boston, MA, United States: Prentice Hall, 2017*
- Symmetric ciphers (Lab 1 - 2)
- Socket programming (Network Programming course).

4.1.3 Lab environment and Tools

1. **Operating system:** Windows, Linux (Ubuntu), MacOS
2. **Programing languages and IDE:** Flexible, you are free to choose any programming language you wish (Python, Golang are highly recommended)

4.2 LAB TASKS

Task 4.1 Your task is to write a client/server application named **"Secure Chat"**.

- First, both sides share a secret key using Diffie-Hellman algorithm. This process needs to be illustrated step-by-step, follows the description in section 4.1.1.
- Then, client and server select an symmetric cipher (at least 3 ciphers) to encrypt the conversation (text messages).

Prove your application is probably "secure" by using an sniffing tool like *Wireshark* to capture the traffic between two sides, then analyze and find your encrypted messages.
■

Tips 4.1. *To implement client-server chat application, you can use* **Socket** *(Perhaps you knew this technique from Network Programming course this semester). You can also find out the corresponding Socket guideline for your programming language (C#, Java, Python,..). The completed application includes two part: Client and Server, which are able to run on same computer or different computers in the same network. For instance, you can watch a demonstration at `https://www.youtube.com/watch?v=B9AN3PDoAB8`*

Advanced Task 4.1 Upgrading your application in task 4.1 by allowing each side to transmit encrypted images or files.

Task 4.2 In Diffie-Hellman (DH), if an adversary knows the following ingredients $g, p, g^a mod\, p, g^b mod\, p$, then will he able to determine the secret key $g^{ab} mod\, p$? Why? Is DH totally secure? Describe the attacks against DH and the countermeasure *(if any)*. ■

Tips 4.2. *The following diagram shows an example of the attack against Diffie-Hellman key exchange protocol:*

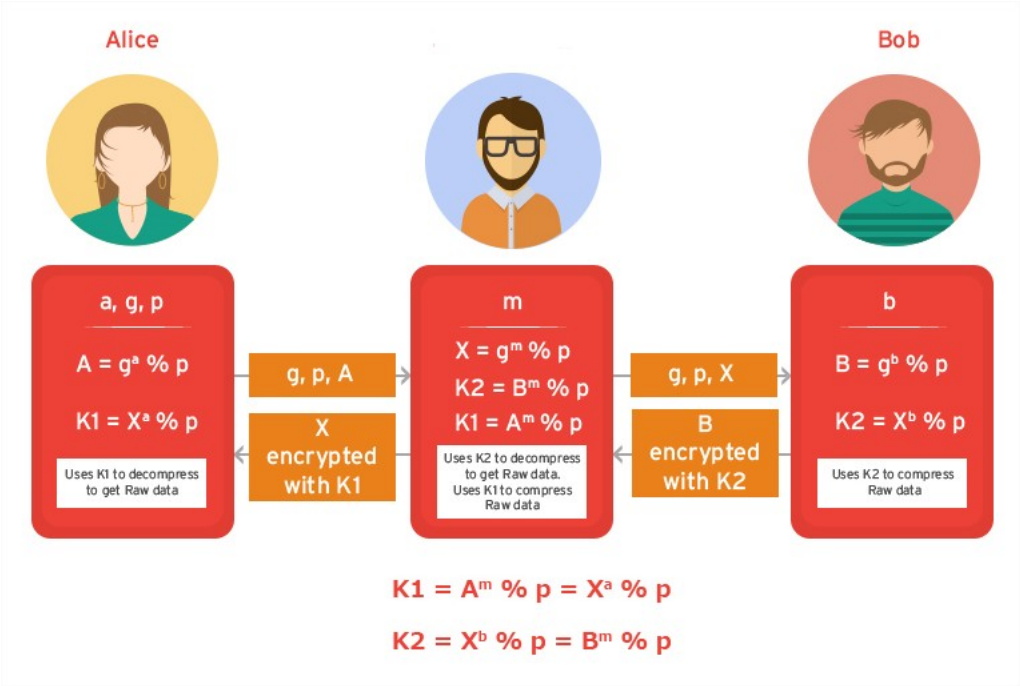

Figure 4.2: An attack against Diffie-Hellman key exchange

Advanced Task 4.2 Extending your application in task 4.1 to illustrate the attack in figure 4.2

4.3 REQUIREMENTS - EVALUATION

4.3.1 Requirements

You are expected to complete all tasks in section **Lab Tasks**, advanced tasks are optional and you could get bonus points for completing those tasks. You can either practice individually or work in team (2 members/team). If you prefer to work in team, please register in the first class with instructor and keep working in your team afterwards.

Your submission must meet the following requirements:

- You need to submit a **detailed lab report in .PDF** format, **using report template** that was provided on the courses website. You need to provide screenshots, to describe what you have done and what you have observed and explanation to the observations that are interesting or surprising.
- **Both of Vietnamese and English report are accepted**, that's up to you. Students in High Quality and Honor program are expected to write lab report in English.
- **Students in Honor program are expected to finish all advanced tasks.**
- When it comes to **programming tasks**, please attach all source-code and executable files (if any) in your submission. Please also list the important code snippets followed by explanation and screenshots when running your application. Simply attaching code without any explanation will not receive points.
- **Submit work you are proud of - don't be sloppy and lazy!**
  Your submissions must be your own work. You are free to discuss with other classmates to find the solution. However, report-copy is prohibited. Both of report owner and copier are received *a special gift - Zero point*. Please remember to clearly cite any source of material (website, book,...) that influences your solution.

Notice 4.3.1 Combine your lab report and all related files into a single ZIP file (.zip), name it as follow:

StudentID1_StudentID2_ReportLabX

For example: 19520123_19520234_ReportLab4.zip. Maximum file size: 10MB.
If it is larger than 10 MB, then you should upload to Google Drive and submit the link to your report *(remember to share view permission with instructor)*

4.3.2 Evaluation

- Well complete all basic tasks: 70% *or 50% with students in Honor program (.ANTN)*
- Well complete the advanced tasks: 10-100% or bonus points to the next lab.
- In-class activities: + up to 10 points
- Work individually: **+ 2 point** *(updated)*
- Report written in English: + up to 2 points

Notice 4.3.2 Assignments are expected to be completed by due date. For every day the assignment is late after the deadline, 10% will be deducted from the lab score. No assignments will be accepted once they are 7 or more days late.

Any part presented in your report may be randomly examine at the next class to verify your work. Absence without any rational reason could result in 30% deduction (or more) of your team's score.

4.4 REFERENCES

[1] William Stallings, *Cryptography and network security: Principles and practice, 7th ed*, Pearson Education, 2017. *Chapter 10: Other Public-key Cryptosystems*

[2] Wenliang Du (Syracuse University), *SEED Cryptography Labs* `https://seedsecuritylabs.org/Labs_16.04/Crypto/`.
[3] Bernhard Esslinger et al., *The CrypTool Book: Learning and Experiencing Cryptography with CrypTool and SageMath, 12th ed, 2018*. Available: `https://www.cryptool.org/en/ctp-documentation`.

Training platforms and related materials

- ASecuritySite - `https://asecuritysite.com`
- Cryptopals - `https://cryptopals.com`

Attention: *Don't share any materials (slides, readings, assignments, labs,...) out of our class without my permission!*