SSH - Secure Shell

Thang Huu Nguyen

Ho Chi Minh City University of Technology
Falcuty of Computer Science & Engineering

Outline

What is SSH?

```
ssh {user}@{host}
```

- SSH (also referred to as Secure Shell or Secure Socket Shell) is a cryptographic network protocol that use encryption to secure the connection between a client and a server.
- All user authentication, commands, output and file transfers are encrypted to against attacks in the networks.

Why should we use SSH?

- Providing secure access for users and automated processes
- Interactive and automated file transfers
- Issuing remote commands
- Managing network infrastructure and other mission-critical system components

Encryption Techniques

There are three different encryption technologies used by SSH:

- Symmetric Encryption
- Asymmetric Encryption
- Hashing

Encryption Techniques

Symmetric Encryption

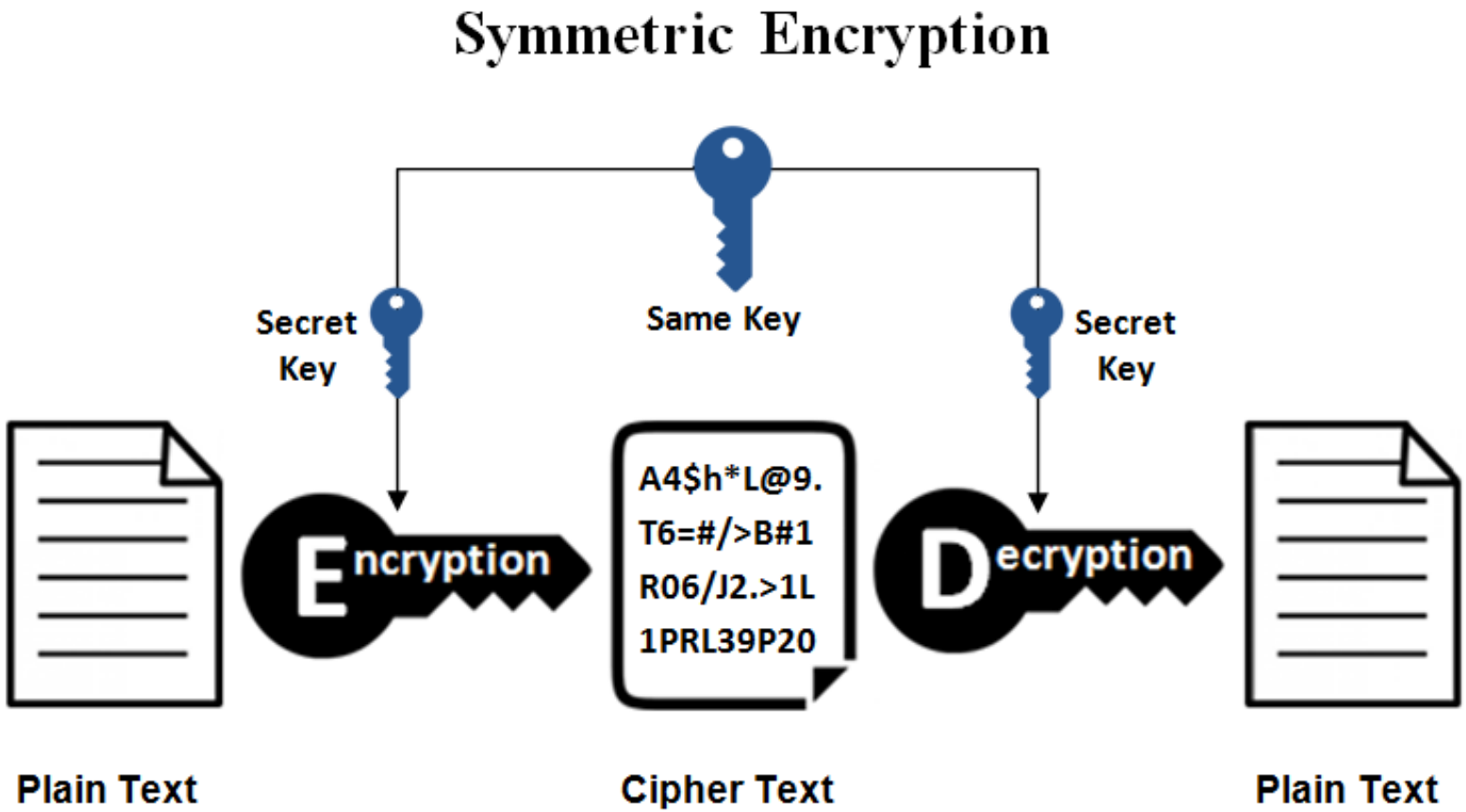

Figure 1: Symmetric Encryption

Encryption Techniques

Asymmetric Encryption

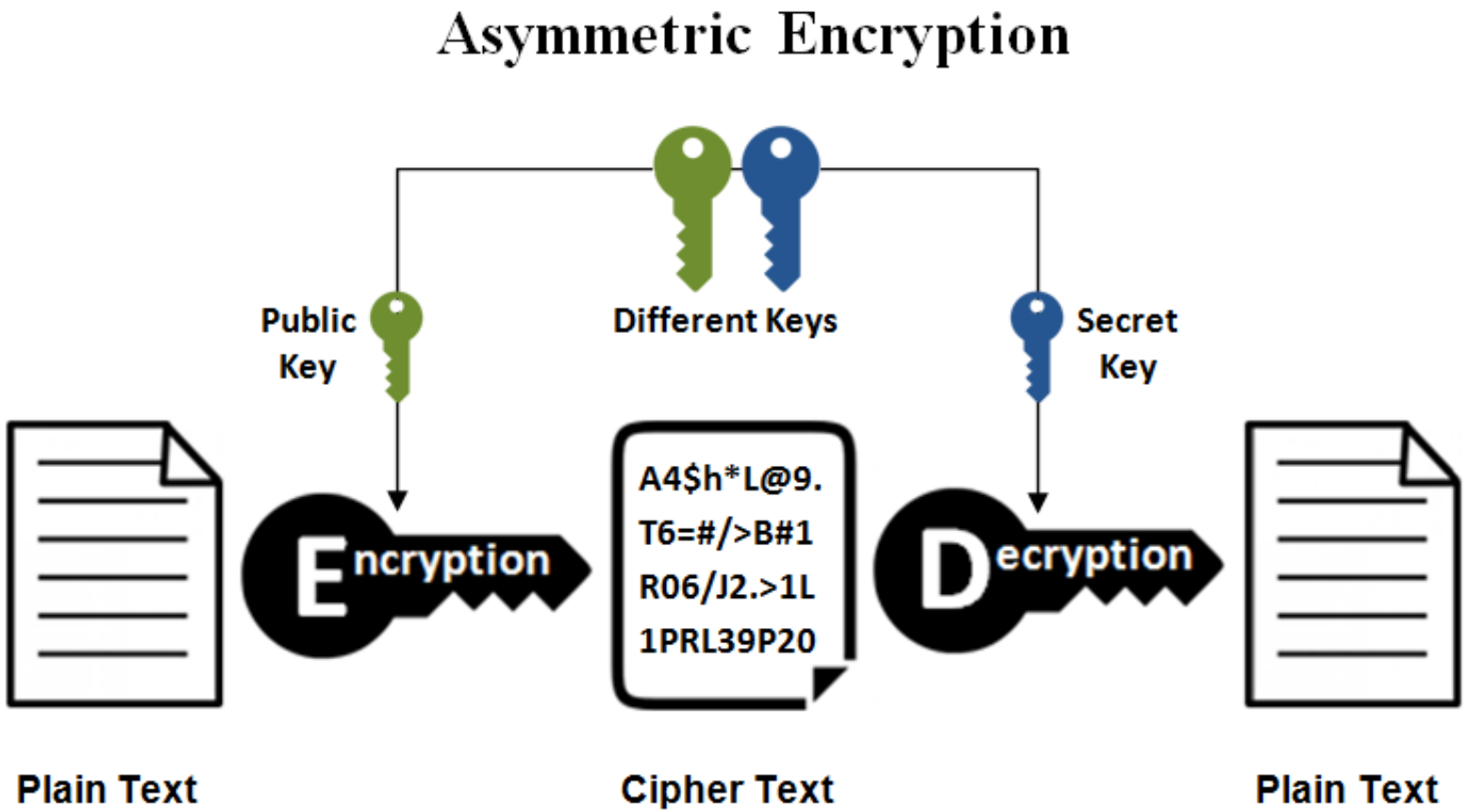

Figure 2: Asymmetric Encryption

Encryption Techniques

Hashing

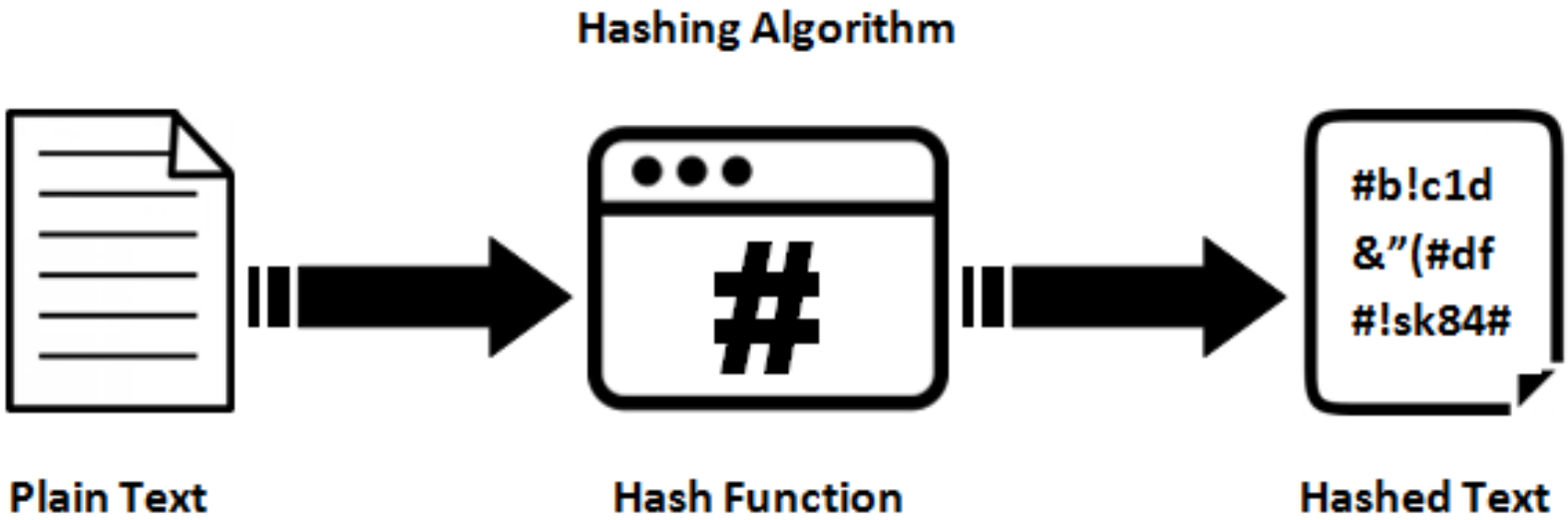

Figure 3: Hashing

How to the SSH works?

The SSH connection between the client and the server happens in three stages:

- Verification of the server by the client
- Generation of a session key to encrypt all the communication
- Authentication of the client

How to the SSH works?

Verification of the server by the client

- If the client is accessing the server for first time, client is asked to authenticate server manually by verifying public key of server
- If the client is not accessing the server for the first time, the server's identity is matched with previously recorded information in **known_hosts** file for verification

```
thang@thang:~/.ssh$ cat known_hosts
|1|0/3tjQOrOCEuhaJlq1xpE2jUNn8=|7+BkBuxHi/xPEqYFc1Qa349wOLw= ecdsa-sha2-nistp256
 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBHZzIR5bwmilZiJv1aobrqyxXFa
Fv+Ks/cggv/e4XQTUDNamTXMVtS55gP8/VRNUCda3J10qWxFMfhhuzoIY1Zc=
|1|CDeh1t2HbDkIcikFZn8RtLX1alY=|zuC5dKTrNNxhLB+ihGH5uH6uO80= ecdsa-sha2-nistp256
 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBJ7D1lPVYo4N1f0AHRJKWJoKkGj
tGWMxtr9Qs3C9n5phjjmLSktqiubK/Km9/U4GYUTLEQY4j7vhZ11C0AXAqA4=
|1|rRMA9qnfiP4Cjcc9lgQxzolSSl0=|+GI2niIOYaFrP2se4iP8xo05F/8= ecdsa-sha2-nistp256
 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBMaKF233fVtM6SOtxJxHxuiip9E
```

Figure 4: Content of known_hosts file

How to the SSH works

Generation of a session key

After the server is verified, both the parties negotiate a session key using a version of something called the Diffie-Hellman algorithm.

This algorithm is designed in such a way that both the parties contribute equally in generation of session key

This session key is used for encryption and decryption.

How to the SSH works

Authentication of the client

After symmetric encryption has been established, the authentication of the client happens as follows:

1. The client begins by sending an ID for the key pair it would like to authenticate with to the server
2. The server checks the **authorized_keys** file of the account that the client is attempting to log into for the key ID
3. If a public key with matching ID is found in the file, the server generates a random number and uses the public key to encrypt the number and sends this encrypted message
4. If the client has the correct private key, it will decrypt the message to obtain the random number that was generated by the server

How to the SSH works

Authentication of the client

5. The client combines the obtained random number with the shared session key and calculates the MD5 hash of this value
6. The client then sends this MD5 hash back to the server as an answer to the encrypted number message
7. Server calculates MD5 value on its own and compare with value which client sent back.

Some popular SSH implementation

- Tectia SSH client & server for Windows, Unix, Linux - with 24x7 support
- Tectia SSH for IBM z/OS client & server for IBM z/OS mainframes - with 24x7 support
- **PuTTY** client for Windows and Linux
- **OpenSSH** server for Unix, Linux
- CyberDuck client for Mac

THANKS FOR YOUR ATTENTION