SPCS Cryptography Class Lecture 4

June 25, 2015

Introduction to Public Key Cryptography

- We have seen a few examples of ciphers now: substitution cipher, Vigenere cipher, autokey cipher etc.
- One thing in common: the same key is used for encryption and decryption. Such ciphers are called *symmetric ciphers*, and such encryption/decryption algorithm is called *symmetric-key algorithm.*

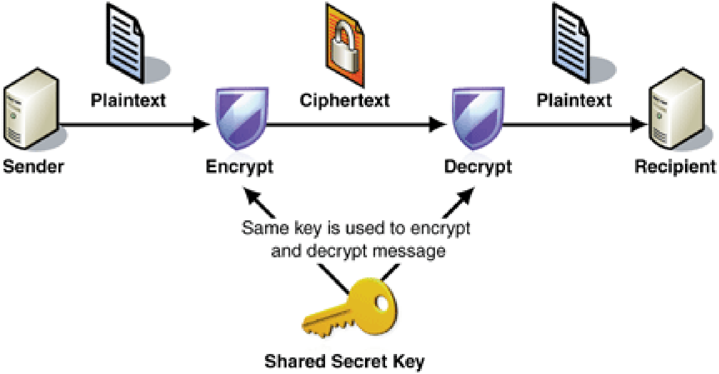

Image from `https://msdn.microsoft.com/en-us/library/ff650720.aspx`

Introduction to Public Key Cryptography

- A toy model of the situation is this: Alice and Bob have a safe and both has a key to the safe. When Alice wants to send Bob a message, she puts the message in the safe by opening the box (Encryption), close it, give it to Bob, who opens the box to retrieve the message. (Decryption)
- This requires Alice and Bob to meet in private beforehand to share the same key. What if they don't have such a chance?

Introduction to Public Key Cryptography

- William Jovens has an idea to solve this problem (Back in 1874!), but it was not implemented successfully until 1970s.
- The idea is simple: Alice and Bob does not need to share a safe - Bob would make a safe, as well as two keys for the safe: public key, and secret key.
- The safe is designed in a way that
    - If you have the public key, you can lock it.
    - Even if you have the public key, you CANNOT unlock it, whatever you do. (Look at it, STARE at it, poke around with it,...)
    - Bob would put the safe and the public key out, and keep the private key to himself.
- If Alice wants to send Bob a message, she only needs to put it in Bob's safe, using the public key. (Encryption)
- If Bob wants to read the message, he opens the safe using his secret key (Decryption) and read the message.

Introduction to Public Key Cryptography

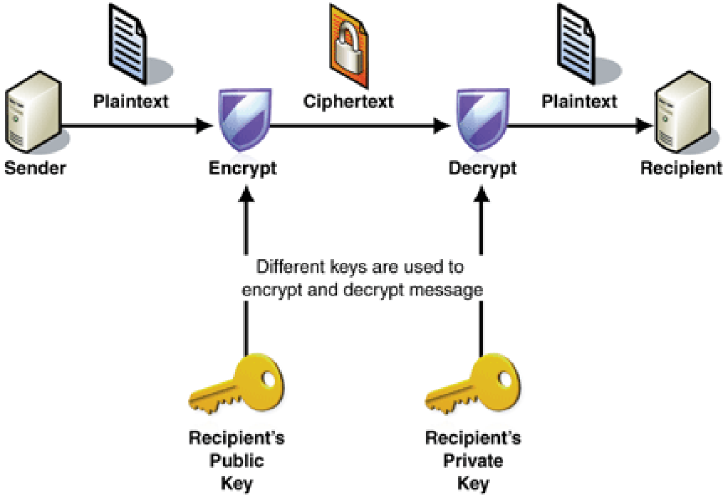

Image from `https://msdn.microsoft.com/en-us/library/ff650720.aspx`

Introduction to Public Key Cryptography

- Only problem is: how can Bob make such a safe?
- Also, if this method can be widely used, Kerckhoff's principle should be satisfied: every one knows how to make such a safe, but without the private key, only Bob can open it.
- One (philosophical) method: Use hard math problems.
- It turns out that there are math problems that are impossible to solve, without knowing the answer or just key to the answer. This is the basic idea of public-key cryptography.
- We will talk about Diffie-Hellman Problem and RSA Problem, the hard problems behind ElGamal Cryptosystem and the RSA system.

Why such things are possible?

A thought experiment:

- Imagine Bob has a big, thick phone book, ordered according to names. This is his *public key*.

Image from `http://www.dreamstime.com/`

- Bob also has the same big, thick phone book, but ordered according to phone numbers. This is his *private key*.

Why such things are possible?

A thought experiment:

- If Alice wants to send Bob a message, say `COOL`, she would use the public phone book, and replace each letter by the phone number of someone with name starting with that letter.
- For example, she finds in the phone book the following entries,
  - "Cory" with phone number 6501234567.
  - "O'neal" with phone number 2025550162.
  - "O'Connor" with phone number 8035550140.
  - "Larry" with phone number 4021323603.

  She can put those together, so that COOL gets encoded as

  6501234567202555016280355501404021323603

  Is this safe?

What happens in real life?

- Symmetric-key ciphers are still very important! They are easier to implement, and much faster to compute.
- Only problem - how to decide the shared key in the first place?
- We will talk about Diffie-Hellman Key Exchange. This is a process to let Alice and Bob get a shared key safely, even if they communicate in public channel.
- In real life, people use *Public Key Cryptography* to decide a shared secret key, then use *symmetric cryptography* to actually encode the message. (Data Encryption Standard, Advanced Encryption Standard)