COMP 8920: Cryptography Winter 2025

Lecture 1 — Jan 7, 2025

Prof. Curtis Bright *Scribe: Aaron Barnoff*

1 Overview

The course textbook is Cryptography: Theory and Practice, 4th edition, by Stinson and Paterson.

This lecture covers material from Chapter 1: Introduction to Cryptography.

2 Secret Key Cryptography

Scenario: Alice wants to communicate with Bob over a secure channel.

- First, they agree beforehand on a key k that only they know.
- Alice encrypts the original *plaintext* message P with an encryption function e_k to produce a *ciphertext* $C = e_k(P)$.
  - The ciphertext is a scrambled message that is unintelligible to anyone who does not possess the key.
- Bob recieves the ciphertext and uses the key to transform (decrypt) it back into the original plaintext message.

Several key assumptions are made about the security of the cryptosystem:

- An eavesdropper, Eve, can listen to the communication, and therefore knows C.
- Eve also knows the encryption function e_k, but not the key k.
- There must be many possible values for k, otherwise Eve could use brute force to find it.
  - In practice, k should be ≥ 128 bits (2^{128} possible k's).

Problem 1: Key Exchange Alice and Bob need to manually and securely exchange the secret key k beforehand.

- (i) They could do it in person.
- (ii) They could also use a key-agreement protocol, e.g., Diffie–Hellman key exchange.

Problem 2: One-Way Communication If Bob cannot respond back, how can Alice communicate securely?

- This method won't work because a secret key cannot be established.
- They must use another system, such as public key cryptography.

3 Public Key Cryptography

Public-key cryptography was introduced in the 1970s by Diffie and Hellman.

In public-key Cryptography, every party has both a public key and a private key:

- Alice uses Bob's public key to encrypt the message; anyone can see Bob's public key.
- Bob uses his own private key to decrypt it; only Bob has the private key.

4 Block and Stream Ciphers

There are two main types of cryptosystems: block ciphers and stream ciphers.

Block Cipher The plaintext is divided into blocks, which are fixed-sized chunks. Each block is specified to be a bitstring (0's and 1's) of fixed length, such as 64 or 128bit. Block ciphers encrypt/decrypt one block at a time.

Stream Cipher The key is used to construct a keystream. The keystream has the same length as the plaintext, which itself is an arbitrary-length bitstring. Encryption and decryption can be accomplished with an XOR of the keystream with the plaintext and ciphertext, respectively.

Public-key cryptosystems use block ciphers, while secret-key can use either.

5 Hybrid Cryptography

In contrast to secret-key cryptosystems, which use simple bitwise operations in their encryption and decryption functions, public-key cryptosystems rely on computationally expensive operations like modular exponentiation. As a result, public-key cryptosystem are much slower than private key systems, and are generally infeasible for encrypting long messages.

However, the hybrid method achieves the benefits of both systems, with a small trade off in performance:

- Alice chooses a random short secret key k, and uses it to encrypt the long plaintext message using a (fast) secret-key cryptosystem.
- Alice then encrypts k using a (slow) public-key cryptosystem and sends the ciphertext and the encrypted key to Bob.
- Bob uses his private key to decrypt the secret key, then uses the secret key to decrypt the ciphertext.

As a result, the hybrid cryptography achieves nearly the same efficiency of secret-key cryptography, but does not require a predetermined secret key.

6 Message Integrity

Cryptosystems ensure secrecy and confidentiality against eavesdropping adversaries. However, it is equally important to verify the authenticity of received messages, ensuring they have not been forged or manipulated by an adversary attempting to impersonate the sender.

For example, an adversary can flip the n'th bit of an encrypted stream cipher, which flips the corresponding n'th bit of the message. In this bit-flip attack, the adversary can modify the plaintext in a predictable way without knowing what the bits are.

6.1 Message Authentication Code (MAC)

MAC relies on Alice and Bob sharing a secret key:

- When Alice sends her message M, she appends a tag $t = \mathrm{MAC}_k(M)$.
- Bob also computes $\mathrm{MAC}_k(M)$ and verifies that it matches the given tag.

MAC is useful for "off-the-record" communication, where plausible deniability is important. It can't be proven that Alice and Bob sent the messages because only they can compute the tag.

6.2 Message Signature Scheme

Alice has a private key known only to her, and publishes a public key:

- When sending a message, Alice computes a signature and appends it to the message.
- The signature is a function of her private key and the message.
- Anyone who knows Alice's public key (and message) can verify the signature is hers.

Signature schemes are less efficient than MACs, and in practice only a **hash** of the message is signed.

6.3 Hashes

A hash function accepts arbitrary input and produces a fixed-size output (e.g., 128-bit).

A cryptographic hash should be infeasible to invert, meaning that one should not be able to find a preimage (input) which hashes to a given hash.

Also, it should be computationally infeasible to find collisions, in which two messages M and M', with $M \neq M'$, have the same hash $h(M) = h(M')$. Note that there must be infinitely many collisions since the input has arbitrary size and the hash is fixed-size (pigeonhole principle).

7 Certificates

If buying something online, how do you know if you have Amazon's public key?

Certificate: A message contains the public key signed by a trusted authority.

- The certifying authority signs with their private key.
- A browser has the trusted authority's public key built-in to verify the certificates.

8 Security

The goal of an attacker is to find the key (full break), or to determine something about the plaintext of a given ciphertext (partial break). We want to prove our protocols are secure against attack.

The four attack models are:

1. Known ciphertext attack
2. Known plaintext attack
3. Chosen plaintext attack
4. Chosen ciphertext attack

Chosen plaintext and ciphertext attacks give the adversary more information, which makes the job easier than simply knowing some ciphertext messages.

A **one-time pad** is cryptographically secure. The key is as long as the plaintext, and is never used again. The attacker cannot get any partial information from the ciphertext, other than message length.