INFORMATION SECURITY

Spring 2025: CS-3610

Bhumika Mittal

Contents

1 Introduction

In this lecture, we will explore the fundamental concepts of cryptography and security. The topics include:

- Understanding security and its importance.
- Differentiating between **security** and **privacy**.
- Analyzing whether cryptographic methods ensure security.
- Evaluating the role of software, keys, and protocols in security.

1.1 Key Questions

- How do we know that a protocol is secure?
- How do we analyze security?
- Does cryptography guarantee security?
  - Is it dependent on the *software*?
  - Is it dependent on the *key*?
  - Is it dependent on the *protocol*?

Discussion: Recall the discussion from the lecture and address the following:

1. How would you attack the attendance QR system on AMS?
2. How can such attacks be prevented?
3. Can the IT admin attack the entire class? What about an individual student?

1.2 Security vs Privacy

Secrecy is not equivalent to privacy. **Security** is a *necessary* but not a *sufficient* condition for privacy.

Question: What is a protocol? How do we evaluate if a protocol is secure?

2 Cryptography Basics

The security of communications and commerce in the digital age depends on the modern adaptation of the ancient art of codes and ciphers.

2.1 Symmetric-Key Cryptography

For thousands of years, codes and ciphers were based on the premise that the communicating parties, **Bob** and **Alice**, shared a secret key unknown to the adversary, **Eve**.

Example: The Caesar Cipher As Julius Caesar surveys the unfolding battle, a courier delivers a cryptic message:

j s j r d k f q q n s l g f h p g w j f p y m w t z l m n r r n s j s y q z h n z x

The message is decrypted by shifting each letter five places back in the alphabet. For example:

$$\text{Ciphertext letter: } j \quad \Rightarrow \quad \text{Plaintext letter: } e$$

This technique, known as the Caesar cipher (or shift cipher), replaces each letter in the plaintext with a fixed substitute.

Limitations: The Caesar cipher is vulnerable to brute force since there are only 26 possible shifts.

Enhancement: Bob can use a substitution cipher, replacing each letter with a unique counterpart. For instance:

$$a \leftrightarrow z, \quad b \leftrightarrow y, \quad c \leftrightarrow x, \ldots, \quad m \leftrightarrow n$$

This type of cipher, called a **simple substitution cipher**, maps each plaintext letter uniquely to a ciphertext letter. Mathematically, the substitution function must be **injective** (one-to-one) for decryption to be possible. **Exercise:** Analyze the security of a substitution cipher. Hint: The keyspace is 26!. Why is brute force impractical? What other cryptanalytic methods could Eve use?

2.1.1 Mathematical Formalism of Symmetric Ciphers

To communicate securely, Bob and Alice share a secret key k and perform encryption and decryption as follows:

- **Encryption:** Use the key k to transform a plaintext message m into a ciphertext c.
- **Decryption:** Use the same key k to recover m from c.

Encryption and Decryption Functions:

$$e : \mathcal{K} \times \mathcal{M} \to \mathcal{C} \quad \text{(Encryption function)}$$
$$d : \mathcal{K} \times \mathcal{C} \to \mathcal{M} \quad \text{(Decryption function)}$$

Here:

- $\mathcal{K}$: Set of all possible keys.
- $\mathcal{M}$: Set of all possible plaintext messages.
- $\mathcal{C}$: Set of all possible ciphertexts.

The encryption and decryption functions satisfy:

$$d(k, e(k, m)) = m \quad \forall\, k \in \mathcal{K},\, m \in \mathcal{M}$$

Alternatively, we can express the functions with subscripts:

$$e_k : \mathcal{M} \to \mathcal{C}$$
$$d_k : \mathcal{C} \to \mathcal{M}$$

These satisfy:

$$d_k(e_k(m)) = m \quad \forall\, m \in \mathcal{M}$$

Symmetry: Since both encryption and decryption depend on the same key k, symmetric ciphers require equal knowledge of k for both Bob and Alice.

2.2 Public-Key Cryptography

Let us begin with a brief history lesson (only because I am taking two history classes this sem so why not!).

In 1976, Whitfield Diffie and Martin Hellman published their groundbreaking paper entitled *"New Directions in Cryptography."* In this paper, they introduced the concept of a **public-key encryption system**, which revolutionized the field of cryptography. Interestingly, their work was not entirely isolated. Around the same time, Ralph Merkle independently formulated one of the fundamental problems of public-key cryptography and devised a public-key construction as part of an undergraduate project in a computer science course at Berkeley. His work, titled *"Secure Communication over Insecure Channels,"* was published in 1982, but it initially received little attention.

However, it turns out that the concept of public key encryption was originally discovered by James Ellis while working at the British Government Communications Headquarters (GCHQ). Ellis's discoveries in 1969 were classified as secret material by the British government and were not declassified and released until 1997, after his death. It is now known that two other researchers at GCHQ, Malcolm Williamson and Clifford Cocks, had independently discovered the Diffie–Hellman key exchange algorithm and the RSA public-key encryption system, respectively. These discoveries preceded the rediscovery and public dissemination by Diffie, Hellman, Rivest, Shamir, and Adleman. All five of these pioneers were later awarded the Turing Award for their seminal contributions to cryptography.

Moral of the story: Avoid conducting confidential research if you aim to win a Turing Award — someone else might beat you to public recognition!

2.2.1 Public-Key Encryption

Recall from the last section that the major disadvantage of symmetric-key cryptography is the need for Alice and Bob to share a pre-established secret key. This assumes the existence of a secure channel, even if only once, to agree on the key before communicating over an insecure channel. However, such an assumption is often impractical in real-world scenarios. To address this limitation, we study **public-key cryptosystems**.

For this course, it is sufficient to understand the basic intuition behind *public-key encryption* and *digital signatures*.

Definition: A **public-key encryption scheme** is a tuple of three probabilistic polynomial-time (PPT) algorithms:

$$\mathcal{E} = (\mathrm{KeyGen}, \mathrm{Enc}, \mathrm{Dec}),$$

which satisfies the following:

1. **KeyGen:**

- Input: A security parameter 1^n (unary representation of n).
- Output: A pair of keys (pk, sk), where:

  - pk is the public key (made public).
  - sk is the secret key (kept private).

2. **Enc:**

- Input: A message $m \in \mathcal{M}$ (where $\mathcal{M}$ is the message space specified by the scheme) and the public key pk.
- Output: A ciphertext $c = \text{Enc}(m, \text{pk})$.

3. **Dec:**

- Input: A ciphertext c and the secret key sk.
- Output: The original message $m = \text{Dec}(c, \text{sk})$.

Correctness: For every security parameter $n \in \mathbb{N}$, every key pair $(\text{pk}, \text{sk}) \leftarrow \text{KeyGen}(1^n)$, and every message $m \in \mathcal{M}$, the following holds:

$$\text{Dec}(\text{Enc}(m, \text{pk}), \text{sk}) = m.$$

2.2.2 Public-Key Signature

A **public-key signature scheme** is a tuple of three PPT (probalistic polynomail time) algorithms:

$$\Sigma = (\text{KeyGen}, \text{Sign}, \text{Verify}),$$

which satisfies the following:

1. **KeyGen:**

- Input: A security parameter λ (often denoted as 1^λ).
- Output: A pair of keys (pk, sk), where:

  - pk is the public key (made public).
  - sk is the secret key (kept private).

2. **Sign:**

- Input: A message $m \in \mathcal{M}$ and the secret key sk.
- Output: A signature $\sigma = \text{Sign}(m, \text{sk})$.

3. **Verify:**

- Input: A message-signature pair (m, σ) and the public key pk.
- Output: A bit $b \in \{0, 1\}$, where:

  - $b = 1$ indicates that the signature is valid.
  - $b = 0$ indicates that the signature is invalid.

Correctness: For every security parameter $\lambda \in \mathbb{N}$, every key pair $(\mathrm{pk}, \mathrm{sk}) \leftarrow \mathrm{KeyGen}(1^\lambda)$, and every message $m \in \mathcal{M}$, the following holds:

$$\mathrm{Verify}(m, \sigma = \mathrm{Sign}(m, \mathrm{sk}), \mathrm{pk}) = 1.$$