Chapter 2

Mathematical Background

In this chapter, we mainly discuss those mathematical preliminaries, which are useful in designing and analyzing our proposed schemes in Chapters 4–6. We first discuss the basic preliminaries of symmetric cryptosystem and public key cryptosystem, and their properties. We then discuss the properties of a one-way collision-resistant cryptographic hash function. We also discuss the message authentication code and hashed message authentication code.

2.1 Basics of cryptography

Let $A$ be a finite set called the alphabet of definition. For example, $A$ can be a binary alphabet set, that is, $A = \{0, 1\}$ consisting of 0 and 1. Let $M$ denote a set consisting of strings of symbols from an alphabet set of definition $A$. $M$ is called the message space. An element of $M$ is called a plaintext message or simply a plaintext. We define another set $C$, called the ciphertext space, which consists of strings of symbols from an alphabet of definition, which may differ from the alphabet of definition. An element of $C$ is known as a ciphertext.

Let $K$ denote a set called the key space. An element of $K$ is called a key. Each element $e \in K$ uniquely determines a bijection function from $M$ to $C$, denoted and defined by

$$E_e : M \rightarrow C,$$

such that $E_e(m) = c$, where $m \in M$ and $c \in C$ are called the plaintext and its corresponding ciphertext, respectively [95]. The function $E_e$ is known an encryption
function or encryption transformation and $e$ is called an encryption key. For each $d \in K$, a decryption function or decryption transformation is denoted and defined by

$$D_d : C \rightarrow M,$$

such that $D_d(c) = m$, where $c \in C$ and $m \in M$ are called the ciphertext and its corresponding plaintext, respectively [95], and $d \in K$ is called the decryption key corresponding to the encryption key $e \in K$. An encryption scheme consists of a set $\{E_e : e \in K\}$ of encryption functions and a corresponding set $\{D_d : d \in K\}$ of encryption functions with the property that for each $e \in K$ there is a unique $d \in K$ such that $D_d = E_e^{-1}$.

### 2.1.1 Symmetric key cryptography

The encryption scheme is said to be $S$-key or symmetric-key, if for each associated encryption/decryption key pair $(e, d)$, it is computationally “easy” to determine $d$ from $e$ and to determine $e$ from $d$ [122]. In most practical symmetric-key encryption schemes, $e = d$. Other terms used are single-key, one-key, private-key and conventional encryption.

Consider the model of symmetric encryption shown in Figure 2.1. Before the secure communication takes place, both the sender, say $S$ and the receiver, say $R$ share the same secret key $k$. If $S$ wants to communicate securely with $R$, $S$ can encrypt a plaintext, say $X$ with the key $k$ using the encryption function $E$ to produce the corresponding ciphertext $Y$ as $Y = E_k(X)$, and then can send the ciphertext $C$ to $R$ over a public (insecure channel). On the other hand, having the ciphertext $C$, $R$ can recover the original plaintext $P$ by decrypting $C$ using the same secret key $k$ as $X = D_k(Y)$. Since the channel is public, an adversary $A$ can eavesdrop, modify or delete the messages from the channel. In this model, $A$ can try to derive the secret key and the plaintext with the help of the eavesdropped ciphertext $Y$. This kind of attack is known as ciphertext-only attack.

Some of the well-known examples of symmetric key encryption include the following [122]:
• **Data Encryption Standard (DES)**

  - It is the most widely used encryption adopted in 1977 by the National Institute of Standards and Technology (NIST), USA.
  - For DES, data are encrypted in 64-bit blocks using an 56-bit key.
  - The encryption algorithm transforms 64-bit input (plaintext block) in a series of steps into 64-bit output, called the ciphertext block.
  - The same steps, with the same key, are used to reverse the encryption (decryption).
  - DES was finally and definitely proved insecure in July 1998, when the Electronics Frontier Foundation (EFF) announced that it had broken a DES encryption using a special-purpose “DES cracker” machine that
was built for less than 250,000 USD. The attack took less than three days [122].

The alternatives to the DES are given below.

- **Double DES (2DES)**
  - It uses two 56-bit keys $K_1$ and $K_2$, and 64-bit plaintext block.
  - It then produces 64-bit ciphertext block.
  - Known-plaintext attack (meet-in-the-middle attack) is possible against 2DES to derive two keys $K_1$ and $K_2$, which has a key size of 112 bits and with an effort on the order of $2^{56}$ [122].

- **Triple DES with three keys (3DES with three keys)**
  - It uses three 56-bit keys $K_1$, $K_2$ and $K_3$, and 64-bit plaintext block.
  - It produces 64-bit ciphertext block.
  - There is no practical attack so far on this cipher. Hence, it is considered as a secure symmetric cipher.

- **Modes of operation of DES**
  The following are various modes of DES:
  - Electronic Codebook Mode (ECB)
  - Cipher Block Chaining Mode (CBC)
  - Cipher Feedback Mode (CFB)
  - Output Feedback Mode (OFB)
  - Counter Mode (CTR)

- **Advanced Encryption Standard (AES)**
  - AES-128 takes 128-bit key and 128-bit plaintext block as input.
  - AES-128 produces 128-bit ciphertext block.
  - AES is very efficient as compared to DES, 2DES, 3DES and various modes of operation of DES.

Detailed descriptions of these symmetric encryption schemes can be found in [122].
2.1.2 Public key cryptography

The encryption scheme is said to be public key or asymmetric-key, if for each associated encryption/decryption key pair \((e, d)\), it is computationally infeasible to determine \(d\) from \(e\) and also to determine \(e\) from \(d\) [122].

Consider the model of public key encryption shown in Figure 2.2. In this model, each entity (party), say \(X\) in the network generates a pair of private key \(K_{Rx}\) and public key \(K_{Rx}\). Suppose the sender (Bob) has a key ring consisting of other parties, say Joy, Mike, Alice and Ted. Suppose Bob (\(B\)) wants to communicate securely to Alice (\(A\)). Then, \(B\) will use the public key \(K_{Ra}\) of Alice to encrypt a plaintext message \(P\) to produce the corresponding ciphertext \(C\) as \(C = E_{K_{Ra}}(P)\), where \(E(\cdot)\) represents the public key encryption, and then send the ciphertext \(C\) to Alice \(A\) via a public channel. Upon receiving the ciphertext \(C\), \(A\) will decrypt \(C\) to recover the original plaintext message \(P\) using her own private key \(K_{Ra}\) as \(P = D_{K_{Ra}}(C) = D_{K_{Ra}}(E_{K_{Ra}}(P))\), where \(D(\cdot)\) represents the public key decryption.

Some of the well-known public key ciphers are RSA, elliptic curve cryptography (ECC), ElGamal encryption and Rabin cryptosystem [122]. It is well-known that
160-bit ECC provides comparable security to 1024-bit RSA and 224-bit ECC provides comparable security of 2048-bit RSA, and ECC works on relatively smaller key size as compared to RSA [129]. Hence, ECC is computationally much efficient as compared to RSA. Furthermore, public key encryption is generally expensive in terms of computation as compared to symmetric encryption. For example, an RSA decryption is around 1000 times slower than the same process with the DES [28]. Considering the resource limitations of the sensor nodes in a wireless sensor network, the symmetric encryption is much viable option as compared to the public key encryption.

2.2 One-way hash function

A cryptographic one-way hash function accepts a variable length block of data as input and then outputs a fixed-size bit string, known as the hash value or message digest. The hash function can be applied to a large set of inputs which can produce outputs that are evenly distributed, and apparently random. The hash function is also used to provide data integrity in order to check whether the message has been modified in between the communication by an adversary. It is noted that a change in any bit or bits in input data results, with high probability, in a change to the hash digest. It is computationally infeasible (hard) to find either a data object that maps to a pre-specified hash result (the one-way property) or two data objects that map to the same hash result (the collision-free property).

Mathematically, a one-way hash function $h: \{0, 1\}^* \rightarrow \{0, 1\}^l$ takes an arbitrary-length input $x \in \{0, 1\}^*$, and produces a fixed-length (say, $l$-bits) output $h(x) \in \{0, 1\}^l$, called the message digest or hash value. The hash function may be the fingerprint of a file, a message, or other data blocks, and has the following properties [122].

- $h$ can be applied to a data block of all sizes.
- For any given input $x$, the message digest $h(x)$ is easy to operate, enabling easy implementation in software and hardware.
- The output length of the message digest $h(x)$ is fixed.
- Deriving the input $x$ from the given message digest $y = h(x)$ and the given
2.2 One-way hash function

hash function \( h(\cdot) \) is computationally infeasible. This property is called the one-way property. It is also called the preimage-resistance property [123].

- For any given input \( x \), finding any other input \( y \neq x \) so that \( h(y) = h(x) \) is computationally infeasible. This property is known as weak-collision resistant property. It is also called the second-preimage-resistance property [123].

- Finding a pair of inputs \((x, y)\), with \( x \neq y \), so that \( h(x) = h(y) \) is computationally infeasible. This property is referred to as strong-collision resistant property. It is also known as the collision-resistance property [123].

The formal definition of a one-way hash function \( h(\cdot) \) can be defined as follows [116], [124].

**Definition 2.1** (One-way collision-resistant hash function). A one-way collision-resistant hash function \( h : \{0, 1\}^* \rightarrow \{0, 1\}^l \) is a deterministic algorithm that takes an input as an arbitrary length binary string \( x \in \{0, 1\}^* \) and produces a fixed-length binary string, say \( l \) bits string, \( h(x) \in \{0, 1\}^l \). The formalization of an adversary \( \mathcal{A} \)'s advantage \( \text{Adv}^{\text{HASH}}_{\mathcal{A}}(t) \) in finding collision with the execution time \( t \) is given below.

\[
\text{Adv}^{\text{HASH}}_{\mathcal{A}}(t) = Pr[(x, x') \leftarrow_R \mathcal{A} : x \neq x' \text{ and } h(x) = h(x')],
\]

where \( Pr[E] \) denotes the probability of an event \( E \), and \((x, x') \leftarrow_R \mathcal{A} \) denotes the pair \((x, x')\) is randomly selected by \( \mathcal{A} \). In this case, \( \mathcal{A} \) is also allowed to be probabilistic and the probability in the advantage is computed over the random choices made by \( \mathcal{A} \) with the execution time \( t \). By an \((\epsilon, t)\)-adversary \( \mathcal{A} \) attacking the collision resistance of \( h(\cdot) \), we mean that the runtime of \( \mathcal{A} \) is at most \( t \) and that \( \text{Adv}^{\text{HASH}}_{\mathcal{A}}(t) \leq \epsilon \).

Some examples of one-way hash function include the following.

- **MD5 message digest algorithm:**
  
  - It is developed by Ronald Linn Rivest at MIT, USA [122].
  - It takes input as a message of arbitrary length.
  - It produces 128 bits hash output (message digest).
• **Secure Hash Algorithm (SHA):**
  
  - It is developed by National Institute of Standards and Technology (NIST), USA and it has several variants [3].
  - SHA-1: It produces 160 bits message digest.
  - SHA-256: It produces 256 bits message digest.
  - SHA-384: It produces 384 bits message digest.
  - SHA-512: It produces 512 bits message digest.

• **RIPEMD-160:**
  
  - It is developed under the European RACE Integrity Primitives Evaluation (RIPE) project [122].
  - It takes input as a message of arbitrary length.
  - It produces 160 bits hash output (message digest).

There are numerous applications of the hash functions, for example, in the field of cryptology and information security, notably in digital signatures, message authentication codes (MACs), and other forms of authentication. A hash function then becomes the basis of many cryptographic protocols. One fundamental property of a hash function is that its outputs are very sensitive to small perturbations in its inputs. For example, SHA-1 is a secure hash algorithm [3]. Also, the cryptographic hash functions are computationally very efficient.

### 2.3 Message authentication code

A Message Authentication Code (MAC) is a public function of the message and a secret key that produces a fixed-length value that serves as an authenticator [122]. Mathematically, it is defined by $C_K: \{0,1\}^* \rightarrow \{0,1\}^n$, which takes a secret key $K$ and an arbitrary-length string $x \in \{0,1\}^*$ and then outputs a fixed length output, say $n$ bits, as $y = C_K(x)$.

An authentication scheme using the $MAC$ is given in Figure 2.3. In this method, the sender $A$ has a message $M$. $A$ computes $MAC = C_K(M)$ using the shared secret key $K$ with the receiver $B$ and forms the message $M || MAC$, and sends it to the
receiver $B$. Upon receiving the message $M||MAC$, $B$ first separates the plaintext message $M$ from the MAC. $B$ then re-computes $MAC' = C_K(M)$ on the retrieved plaintext message $M$ using the shared key $K$ with the sender $A$. If the condition $MAC' = MAC$ holds, $B$ ensures that $M$ has not altered by an adversary and $M$ is considered authentic. Since the attacker does not know the secret key $K$, he/she does not have any ability to alter the MAC.

![Figure 2.3: Message authentication using MAC](image)

### 2.4 Keyed-hash message authentication code

Mechanisms that provide data integrity checks based on a secret key are usually known as message authentication codes (MACs). Typically, MACs are used between two parties that share a secret key in order to authenticate information transmitted between these parties. When a MAC uses a cryptographic hash function in conjunction with a secret key is called HMAC [79]. HMAC uses a secret key for the calculation and verification of the MACs. The main goals behind the HMAC construction are as follows [79]:

- To use available hash functions without modifications; in particular, hash functions that perform well in software, and for which code is freely and widely available.

- To preserve the original performance of the hash function without incurring a significant degradation.
• To use and handle keys in a simple way.

• To have a well-understood cryptographic analysis of the strength of the authentication mechanism based on reasonable assumptions on the underlying hash function.

• To allow for easy replaceability of the underlying hash function in the event that faster or more secure hash functions are later available.

2.5 Summary

In this chapter, we have reviewed the basic cryptographic primitives. In particular, we have discussed those cryptographic techniques which are useful for describing and analyzing the intrusion detection and prevention mechanisms proposed for hierarchal wireless sensor networks in the thesis, such as symmetric encryption versus public key encryption, one-way hash function, message authentication code and its variant keyed-hash message authentication code.