CSA E0 235: Cryptography (12-August-2019)

Scribe for Lecture 2

Instructor: Arpita Patra *Submitted by: (Shravani Mahesh Patil)*

1 Introduction

Up until the late 20th century, cryptography was viewed upon as an art of developing codes for secure communication. It relied heavily on the *design-break-patch* model of development wherein a cryptographic scheme for secure communication was considered secure until it was broken. Once a break for found for a scheme, it was modified to patch the break. This period of cryptography up till 1980, is referred to as classical cryptography. In 1980, the beginning of modern cryptography was marked, from where cryptography is now viewed as an applied science rather than art. Modern cryptography not only includes secure communication but also encompasses various other aspects such as authenticated message communication, e-cash, e-auction, secure outsourcing to cloud, secret sharing, etc. Moreover, modern cryptography does not rely on the *design-break-patch* mechanism and follows a more scientific approach towards design of cryptographic primitives.

This scientific approach of modern cryptography can be outlined as a 3-step process as follows:

1. Provide a formal definition of security which captures the requirements of the cryptographic task to be solved.
2. Identify precise and well studied assumptions on which the security of the designed construction relies.
3. Provide a rigorous mathematical proof of security of the construction under the formal security definition based on the stated assumptions.

1.1 Importance of Formal Definition of Security

The first step in the scientific approach of modern cryptography focuses on capturing the requirements of a cryptographic task by establishing formal definitions of security. This is primarily important due to the following reasons:

- First, if we do not know what to achieve, then how can we know when and if we have achieved it?
- Secondly, more often than not, constructing a cryptographic scheme without a security definition results in a scheme which either satisfies lesser requirements than required or more requirements than necessary, at the cost of affecting the efficiency of construction.
- Definitions provide a way to analyze and evaluate a construction.

- Moreover, definitions also enable the comparison of various schemes. This is mainly due to the fact that there can exist multiple definitions for the same security goal. Thus, precisely defining security allows for identifying equivalent classes of cryptographic algorithms. Moreover, it also enables modularity with respect to using the equivalent algorithms interchangeably in a larger construction.

It is crucial to note here that, proof of security of a construction is relative to the security definition provided. Hence, the security definition requires to capture the requirements of a cryptographic task in the real-world scenario. Failing to do this, results in an insecure cryptographic scheme in a real-world scenario, irrespective of the security proof. Moreover, establishing security definitions is not a one-time task. Development of new application scenarios often leads to a requirement of new security definitions. Hence, a security definition should be ensured to be capturing all the requirements laid down by an application in a real-world scenario.

2 Secure Communication in Private Key Setting

As noted, both the classical cryptography as well as modern cryptography deal with secure communication. Secure communication in a private key setting involves two parties, a sender and a receiver, communicating over an untrusted channel, using a pre-shared secret key which is possessed by both the parties involved in communication. Typically, the sender requires to communicate a message 'm' referred to as the plaintext over the untrusted channel (accessible to an adversary) by scrambling (encrypting) it using the secret key 'k'. The scrambled message 'c', referred to as the ciphertext, is communicated over the untrusted channel to the receiver and can be unscrambled (decrypted) using the same secret key 'k' to obtain the plaintext 'm'. Notice that, the same key 'k' is used for encryption as well as decryption, thus contributing to the symmetry of the scheme. Consequently, either of the parties can act as a sender as well as receiver at any point in the scheme. Such a cryptographic setting which enables secure communication using a private/secret/symmetric key is referred to as private/secret/symmetric key encryption. Figure 1 illustrates the entities and procedures involved in symmetric key encryption scheme.

Typically, symmetric key encryption is applicable in two settings.

1. When two different parties (typically separated geographically) require to communicate securely over an untrusted channel. In this case, the communicating parties require to exchange a key secretly before the communication. For the time-being, we assume that the communicating parties already possess a shared secret key.

2. When a party requires to communicate with itself at different instances of time. For example, consider the scenario of secure file storage on a hard drive. In this case, the party stores encrypted data on the hard drive, which might be accessible to the adversary. The party may access the data on the hard drive at any later point of time by decrypting it using the same key that it used for encryption. In this setting, the sharing of the secret key does not require any special mechanism, since, the a single party performs the role of the sender as well as the receiver.

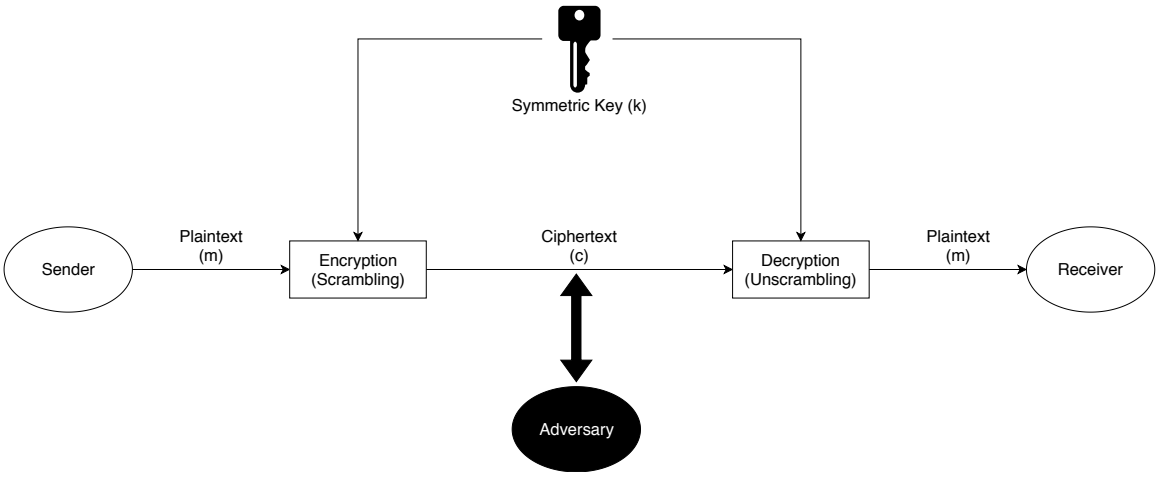

Figure 1: Symmetric Key Encryption

2.1 Syntax of Symmetric Key Encryption

A symmetric key encryption scheme comprises of three algorithms namely, the key generation algorithm Gen, the encryption algorithm Enc and the decryption algorithm Dec. A symmetric encryption scheme can thus be defined as a tuple (Gen, Enc, Dec) of algorithms. Each of these algorithms is described as follows.

1. Key Generation Algorithm [$\mathsf{Gen}()$]
   - This algorithm outputs a key k chosen according to some probabilistic distribution. The key generation algorithm must be a randomized algorithm to ensure secrecy of the key.
   - Syntax: $k \leftarrow \mathsf{Gen}()$
2. Encryption Algorithm [$\mathsf{Enc}()$]
   - This algorithm outputs a ciphertext c of the input plaintext m encrypted under key k. Encryption algorithm may be deterministic or randomized.
   - Syntax: $c := \mathsf{Enc}_k(m)$ or $c \leftarrow \mathsf{Enc}_k(m)$
3. Decryption Algorithm [$\mathsf{Dec}()$]
   - This algorithm outputs the plaintext m underlying the input ciphertext c decrypted using key k. Decryption algorithm is typically deterministic.
   - Syntax: $m := \mathsf{Dec}_k(c)$

Note: The notation $\leftarrow$ represents assignment from the output of a randomized algorithm and the notation $:=$ represents the assignment from the output of a deterministic algorithm.

Each symmetric key encryption scheme is associated with three spaces: key space ($\mathcal{K}$), plaintext space ($\mathcal{M}$) and ciphertext space ($\mathcal{C}$). The algorithm Gen defines the key space $\mathcal{K}$ which is the set of all the possible keys which are given as output by Gen. The plaintext space $\mathcal{M}$ defines the set of all the possible plaintext messages. Similarly, set of all possible ciphertexts which can be obtained as output from the Enc algorithm comprise the ciphertext space $\mathcal{C}$.

A symmetric key encryption scheme is said to be correct if and only if for every key $k \in \mathcal{K}$ and every $m \in \mathcal{M}$, the following equality holds.

$$\mathsf{Dec}_k(\mathsf{Enc}_k(m)) = m$$

Moreover, in a symmetric key encryption scheme, each of the spaces $\mathcal{K}, \mathcal{M}$ and $\mathcal{C}$ defined above has a random variable associated with. Specifically, K is the random variable associated with the key space $\mathcal{K}$, M is the random variable associated with the plaintext space $\mathcal{M}$ and C is the random variable associated with ciphertext space $\mathcal{C}$. The random variable K, has a probability distribution which is induced by the Gen algorithm. The probability distribution of M is independent of the outputs of any algorithm. Also, the random variables K and M are independent of each other. However, the probability distribution of the random variable C associated with the ciphertext space $\mathcal{C}$ is dependent on the random variable K of the key space as well as the random variable M of the plaintext space. It is important to note that, the probability distributions associated with K, M and C are publicly known in a secure communication setting in private key environment.

2.2 Kerckhoffs' Principle

One of the earliest encryption method developed by Julius Caesar, relied on the secrecy of the encryption algorithm in order to ensure secure communication. However, this encryption method was susceptible to letter frequency attack. Moreover, once the encryption algorithm is known, the scheme can no longer be used for secure communication. In the 19th century, Auguste Kerckhoffs claimed that the security of a symmetric key encryption scheme should solely rely on the secret key shared between the two parties involved in communication and not on the secrecy of the encryption algorithm [1]. That is, even when the algorithms Gen, Enc and Dec are publicly known, a symmetric key encryption scheme should remain secure as long as the secret key is secure. Formally [1, 2],

> *The cipher method must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience.*

Some concrete arguments which favor Kerckhoffs' claims are as follows:

- Maintaining privacy of a short key is much easier than maintaining privacy of a large algorithm.
- Replacing a compromised key is much easier than replacing a compromised algorithm. This is due to the fact that designing algorithms requires a large amount of effort to ensure secrecy and correctness.

- In a large group of parties (such as an organization), it is very difficult for every pair of parties to maintain a secret algorithm. In a group of n parties where each one is required to communicate securely with every other party, the number of algorithms held by each party would be $n-1$. Since the storage space required by each algorithm is much larger than that required to store keys, storing $n-1$ algorithms is infeasible. Moreover, the total number of distinct algorithms required to enable secure communication among each pair of parties would be $\frac{n(n-1)}{2}$, which is quadratic in the number of parties involved. Designing such a large number of secure algorithms is highly infeasible.
- If the algorithms are made public then they are available for public scrutiny and their flaws are likely to be found rapidly. This contributes towards strong constructions.

In the following section, we enumerate some of the concepts of probability required to establish a formal security definition for symmetric key encryption.

3 Probability Background

- **Sample Space:** Set of all the possible outcomes of a random experiment. Specifically, we consider a finite set S.
- **Event:** Any subset E of the sample space S is called as an event, i.e. $E \subseteq S$.
- **Probability Distribution:** A function $Pr : S \to [0, 1]$ which specifies the probabilities of occurrence of the elements of S such that,

$$\sum_{x \in S} Pr(x) = 1$$

- **Uniform Probability Distribution:** A function $Pr : S \to [0, 1]$ such that,

$$Pr(x) = \frac{1}{|S|}, \forall x \in S$$

- **Union Bound:** For two events E_1 and E_2,

$$Pr[E_1 \cup E_2] \leq Pr[E_1] + Pr[E_2]$$

In general, for n events $E_1, E_2, \ldots, E_n$,

$$Pr\left[\bigcup_{1 \leq i \leq n} E_i\right] \leq \sum_{1 \leq i \leq n} Pr[E_i]$$

- **Conditional Probability:** The conditional probability of an event E_1 given that event E_2 has occurred is,

$$Pr[E_1|E_2] = \frac{Pr[E_1 \cap E_2]}{Pr[E_2]}$$

- **Law of Total Probability:** If $E_1, E_2, \ldots, E_n$ are n exhaustive and pairwise mutually exclusive events, then for any event E,

$$Pr[E] = \sum_{1 \leq i \leq n} Pr[E \cup E_i] = \sum_{1 \leq i \leq n} Pr[E|E_i] \cdot Pr[E_i]$$

- **Bayes' Theorem:** For two events E_1 and E_2, if $Pr[E_2] \neq 0$, then,

$$Pr[E_1|E_2] = \frac{Pr[E_2|E_1] \cdot Pr[E_1]}{Pr[E_2]}$$

- **Random Variable:** Variable which takes discrete values from a finite set with certain probabilities.

- **Probability Distribution of a Random Variable:** A function which specifies the probabilities with which the random variable takes each possible value of a finite set.

4 The 3-Step Scientific Approach of Modern Cryptography for Symmetric Key Encryption

In this Section, we outline the 3-step process of modern cryptography for symmetric key encryption. Precisely, establishing the notion of security, identifying assumptions and providing mathematical proof of security.

4.1 Formal Definition of Security for Symmetric Key Encryption

The formal definition of security of a symmetric encryption scheme consists of two components: *Threat Model* and *Break Model* [3].

1. **Threat Model**
   It identifies the capability of the adversary. These capabilities of the adversary are determined by various parameters such as computational power, capability with respect to attacking a channel, deterministic or randomized, etc. Threat model basically deals with identifying what we are trying to protect our scheme against. Thinking like an adversary allows us to better determine the capabilities of the adversary.

2. **Break Model**
   It identifies what constitutes a successful attack on the part of the adversary. It concretely establishes what we are trying to secure in the scheme.

Following the above definition of security, we consider one such Threat Model-Break Model for symmetric key encryption.

1. Threat Model

   - Computational power: We consider the strongest adversary which does not assume any bound on the computational power, i.e. an unbounded powerful adversary which can solve the hardest problem in a reasonable time.
   - Capability with respect to attacking secure channel: We consider a passive adversary which can merely eavesdrop on the channel to tap the ciphertext during communication. Such an attack by an adversary which enables it an access to only the ciphertexts being communicated is referred to as Ciphertext-Only-Attack (COA).
   - Deterministic or Randomized: We consider a randomized adversary which is more powerful than a deterministic adversary. Since randomization is a necessary component of cryptography and the communicating parties are assumed to have access to it, we also assume the randomization power for an adversary.

2. Break Model
   We enumerate various flawed attempts to define a break model before finally providing the right notion.

   - **Attempt 1:** A symmetric encryption scheme is secure if the ciphertext does not leak any information regarding the secret key.
     Counterexample: Consider an encryption scheme $m :=\mathsf{Enc}(m)$. This encryption scheme does not reveal any information regarding the secret key since the key is not employed in the encryption of messages. However, this scheme is not secure since it reveals the entire plaintext to the adversary.
   - **Attempt 2:** A symmetric encryption scheme is secure if the ciphertext does not leak the entire plaintext i.e. it does not leak every bit of the message.
     Counterexample: Consider an encryption scheme which reveals the first $k < n$ bits of a n bit ciphertext. If the first k bits of the ciphertext correspond to the sensitive information such as a password, the scheme is not secure, although the entire plaintext is not leaked.
   - **Attempt 3:** A symmetric encryption scheme is secure if the ciphertext does not leak any bit of the message.
     Counterexample: Consider an encryption scheme which reveals the number of characters in the plaintext. If the data being encrypted is the confidential salary of an employee, the ciphertext reveals how many digit salary the employee is getting paid. Although this scheme does not leak even a single bit of the message, it is insecure.
   - **Attempt 4:** A symmetric encryption scheme is secure if the ciphertext leaks no additional information about the plaintext irrespective of the prior information possessed by the adversary.

Attempt 4 of the definitions of Break Model captures the right notion of security for a symmetric encryption scheme defined by algorithms Gen, Enc, Dec and the message space $\mathcal{M}$. However, it requires a concrete mathematical formulation. As mentioned in Section 2.1, the probability distributions M associated with the message space $\mathcal{M}$ is publicly known and thus it constitutes the prior information possessed by the adversary. That is, the adversary has prior information regarding the probability distribution $\{Pr[M = m]\}$. Moreover, on obtaining a ciphertext, the adversary should learn no additional information regarding the plaintext irrespective of the prior information held by it. This can be mathematically formulated as,

$$Pr[M = m|C = c] = Pr[M = m]$$

That is, the posteriori probability that m is encrypted in c should be equal to the a priori probability that m might be communicated. Such an encryption scheme is perfectly secret. Formally, perfect secrecy as defined by Claude Shannon [4] is as follows.

Definition 1 An encryption scheme (Gen, Enc, Dec) over a message space $\mathcal{M}$ is perfectly secret if for every probability distribution over $\mathcal{M}$, every message $m \in \mathcal{M}$, and every ciphertext $c \in \mathcal{C}$ for which $Pr[C = c] > 0$,

$$Pr[M = m|C = c] = Pr[M = m]$$

♢

4.2 Identifying Assumptions

As mentioned in Section 1, it is important to identify and clearly state the assumptions on which the security of the scheme relies. The assumptions being considered should be well studied and widely accepted, although not proven to be true. The underlying assumptions on which the security of schemes relies, allows for a comparison among the schemes. The cryptographic schemes which rely on widely popular and strongly believed assumptions are preferred over less studied assumptions.

Note that we have assumed a threat model in which the computational power of the adversary in unbounded. The adversary is assumed to be capable of solving any hard problem in a reasonable amount of time. Hence, we do not need to rely on any underlying assumptions for proving the security of the schemes under the threat model defined in the previous section.

4.3 Proof of Security

Having established a precise mathematical definition of security, we are now equipped to provide concrete proof of security of a symmetric encryption scheme under the stated definition. Note that the proof of security does not rely on any assumption for the threat model considered.

5 Vernam Cipher (One Time Pad)

Vernam designed and patented the Vernam cipher in 1917. However, the concrete proof that the Vernam cipher is perfectly secret was given much later by Claude Shannon [4]. The Vernam cipher is defined over all the binary strings of a fixed length l. That is, $\mathcal{M} = \mathcal{K} = \mathcal{C} = \{0,1\}^l$. The Gen, Enc, Dec algorithms of the Vernam cipher are defined as follows:

1. $k \leftarrow \mathsf{Gen}()$: This algorithm chooses $k \in_R \mathcal{K}$ (where $x \in_R X$ represents x is chosen uniformly at random from X) and outputs the key k. That is each key $k \in \mathcal{K}$ has equal probability $\frac{1}{2^l}$ of being chosen.
2. $c \coloneqq \mathsf{Enc}_k(m)$: This algorithm takes as input a message $m \in \mathcal{M}$ and a key $k \in \mathcal{K}$ and gives as output $c = m \oplus k$.
3. $m \coloneqq \mathsf{Dec}_k(c)$: This algorithm takes as input a ciphertext $c \in \mathcal{C}$ and a key $k \in \mathcal{K}$ and gives as output $m \coloneqq c \oplus k$.

Theorem 1 *The Vernam cipher (one-time pad) is a perfectly secret encryption scheme [4, 2].*

Proof Consider the Vernam cipher defined over $\mathcal{K} = \mathcal{M} = \mathcal{C} = \{0,1\}^l$. For any probability distribution $\mathcal{M}$, for every $m \in \mathcal{M}$ and $c \in \mathcal{C}$,

$$Pr[C = c | M = m] = Pr[K = m \oplus c] = \frac{1}{2^l}$$

Also from the law of total probability,

$$\begin{aligned} Pr[C = c] &= \sum_{m \in \mathcal{M}} Pr[C = c | M = m] \cdot Pr[M = m] \\ &= \frac{1}{2^l} \cdot \sum_{m \in \mathcal{M}} Pr[M = m] \\ &= \frac{1}{2^l} \end{aligned}$$

Hence, using Bayes' theorem,

$$\begin{aligned} Pr[M = m | C = c] &= \frac{Pr[C = c | M = m] \cdot Pr[M = m]}{Pr[C = c]} \\ &= Pr[M = m] \end{aligned}$$

Hence the Vernam cipher is perfectly secret. ■

5.1 Drawbacks of the Vernam Cipher

Although the Vernam cipher is proven perfectly secret, it has some practical issues associated with it.

1. A key used for encryption of a message m in Vernam cipher cannot be reused for encrypting a different message m'. Reusing a key compromises the security of the scheme. Consider the scenario wherein the same key k is used to encrypt the messages m and m'. That is,

$$c = m \oplus k$$
$$c' = m' \oplus k$$

Given the ciphertexts c and c', the adversary can obtain $m \oplus m'$ as follows:

$$\begin{aligned} c \oplus c' &= (m \oplus k) \oplus (m' \oplus k) \\ &= (m \oplus m') \oplus (k \oplus k) \\ &= m \oplus m' \end{aligned}$$

Hence, the adversary learns some additional information regarding the plaintexts by merely observing the ciphertexts encrypted under a common key, thus compromising the definition of perfectly secret encryption.

2. The length of the key is required to be equal to the length of the plaintext. For large plaintexts, this is a highly impractical requirement since
   - It is difficult for communicating parties to agree on a key when the key length is large.
   - The length of the message to be communicated is required to be predetermined in order to agree on a key of the same length.
3. The key is required to be a random sequence of binary bits. Typically the sources of obtaining randomness include radioactive decay, network delay, keyboard strokes, etc. However, randomness is very difficult to capture when the length of the random string required is very large.

However, the aforementioned drawbacks are inherent in all perfectly secret encryption schemes.

References

[1] A. Kerckhoffs, "La cryptographic militaire," *Journal des sciences militaires*, vol. IX, pp. 5–38, January 1883.

[2] J. Katz and Y. Lindell, *Introduction to modern cryptography.* Chapman and Hall/CRC, 2014.

[3] A. Patra. Csa e0_235 : Cryptography (august - december 2019). [Online]. Available: https://www.csa.iisc.ac.in/ cris/e0_235.html

[4] C. E. Shannon, "Communication theory of secrecy systems," *Bell system technical journal*, vol. 28, no. 4, pp. 656–715, 1949.