
Applied Cryptography

Unit-2 Symmetric key Cryptography

Lecture Notes 2:

Symmetric Key Cryptography- Why Computational Secrecy?

Recommended Reading:

Katz-Lindell: Chapter 2

1. Introduction:

Private key cryptography formally described practically can't be used to provide perfect secrecy. We are aiming for something called as Computational secrecy. This section gives you an overview about why exactly we need to understand computational secrecy. To define security for the described cryptosystem, it involves two components:

- ***Threat model defines the real capabilities that the attacker can possess.***
- ***Objective is what should be our capability to make sure the attacker is useless.***

Let us understand this in detail.

1.1. Threat Models:

- The threat model denotes the capacities of an attacker to guess the key.
- Based on the type of accessing he get the type of threat model varies.

Cipher Text Only Attack:

The least effective threat model is cipher text only attack wherein the attacker gets exposed only to the cipher text. Using the cipher alone the adversary tries to identify the key. This is the least effective attack methodology as this can be done only through brute forcing. Stronger computational capable adversary might be in a position to attack.

Known Plain text attack:

A stronger threat model is the so-called known-plaintext attack. Here, the attacker will get access to both the cipher as well as plain text. Using the before and after information, attacker tries to identify the pattern using which he tries to guess the key. Assume if Alice and Bob both shares the

same type of cipher multiple times can make the attackers work easy. so this threat model is more vulnerable than the cipher text only attack.

Chosen Plain Text Attack:

Still stronger model involves the chosen plain text attack. The attacker will again get access to cipher and plain and now he possesses the capability of performing encryption and trying to observe the pattern through which key guessing happens. observe one or more ciphertexts, whose underlying plaintext is unknown. For example, imagine an attacker typing at a terminal where anything that's typed gets encrypted using a key unknown to the attacker. In that case, the attacker really does have the ability to mount a complete Chosen-plaintext attack.

Chosen Cipher Text Attack:

The strongest threat model typically considered is a Chosen-ciphertext attack. Now in addition to having the ability to carry out a Chosen-plaintext attack like before. We also assume the attacker can get the parties to decrypt certain cipher texts of that attacker's choice. This may sound totally unrealistic, but we'll see later in the course that the ability to carry out some limited form of chosen cipher text attack is actually very common and must be defended against.

1.2 Assumptions to be made for defining security:

i. "Key should be made strong"

A system is secure if the attacker is unable to break the key. This doesn't really hold sense y because assume a room where in you are sending Bob inside the room. The room do not have anything else apart from a deck of card, a box and different types of lock. The rule of the game is Bob can choose one card, but Eve should not be able to decipher which card Bob chose. He can't take any card outside. Bob think for a while and choose one card from the deck and place it inside the box. He finds lot of locks; he feels lock with a key should not be used so he chooses a functional lock and lock the card. Is he not giving any chance for Eve to guess the card, not really because he leaves the card deck there using which Eve can guess Bob's card? So, the goal of security is not to create unbreakable key.

ii. "Cipher should give no clue about plain even if cipher is given"

Can we think the system is secure, if it provides a possibility of not giving any information to the attacker to guess plain text even if cipher is provided? Is it possible? Consider the same old example we gave, wherein

if the deck of card is left there is always possible that the attacker can at least guess 90% of the plain text. Again, we do not want this as well.

- iii. Can we restrict the point 2 as attacker should not learn any character of plain text? This also might because the message need not be only characters. Say for example attacker do not have any knowledge what the salary of an employee is since they were not able to read any character of plain text, however he is able to tell this might be the range of salary. That again do not hold as a definition of security.

So, what should be the practical way of defining security... “Regardless of any prior information the attacker has about the plaintext the ciphertext observed by the attacker should leak no additional information about the plaintext”. In our card game, this can be made possible only by taking the card deck include the card which Bob selected and kept aside and do a shuffling and leave the card. So even if EVE enters and observe the card, she might not be able to guess the card. This is what you call it as perfect secrecy. This definition of perfect secrecy is informally we have given.

So how to define this formally is by using the concepts of probability.

2. Basic Concepts of Probability:

Probability Distribution:

A probability distribution is a table or an equation that links each outcome of a statistical experiment with its probability of occurrence.

Random Variable:

- When the value of a variable is the outcome of a statistical experiment, that variable is a **random variable**.

Generally, statisticians use a capital letter to represent a random variable and a lower-case letter, to represent one of its values. For example,

- **X** represents the random variable X.
- **P(X)** represents the probability of X.
- **P(X = x)** refers to the probability that the random variable X is equal to a particular value, denoted by x. As an example, P(X = 1) refers to the probability that the random variable X is equal to 1.

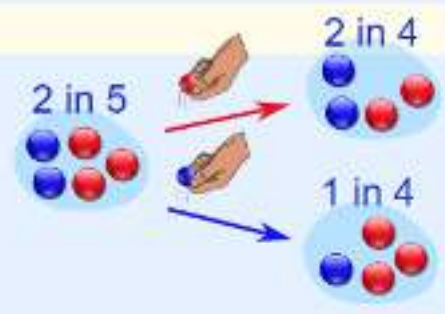
Conditional Probability:

The outcome of an event is dependent on the previous event.

Example: Marbles in a Bag

2 blue and 3 red marbles are in a bag.
What are the chances of getting a blue marble?
The chance is **2 in 5**

But after taking one out the chances change!
So the next time:



if we got a **red** marble before, then the chance of a blue marble next is **2 in 4**

if we got a **blue** marble before, then the chance of a blue marble next is **1 in 4**

Chances of getting blue marble depends on the previous ball picked and probability value varies depending on the previous value selected.

Formally this is denoted by the

$$P[A|B] = \frac{P(A \cap B)}{P(B)}$$

Independence of Random Variables:

If two random variables, X and Y, are **independent**, they satisfy the following conditions.

- $P(x|y) = P(x)$, for all values of X and Y.

-
- $P(x \cap y) = P(x) * P(y)$, for all values of X and Y.

The above conditions are equivalent. If either one is met, the other condition is also met; and X and Y are independent. If either condition is not met, X and Y are **dependent**.

Note: If X and Y are independent, then the correlation between X and Y is equal to zero.

3. Formal Definition of Security:

A Symmetric Key Encryption on scheme (Gen, Enc, Dec) with message space M and ciphertext space C is perfectly secret if for every distribution over M, every $m \in M$, and every $c \in C$ with $P[C=c] > 0$, it holds that

$$P[M = m \mid C = c] = P[M = m].$$

Since it demands complete independence among plain and Cipher text.

This formal definition of security can be verified using shift or one-time pad and it holds that one-time pad is perfectly secret.

So, the condition to define security for our symmetric key cryptosystem is illustrated as there should be absolute no relationship between plain and cipher. This independence makes sure the guessing most of the time ends up with brute forcing that can become successful only with very high computational power.

Now the question that arises is since one-time pad is perfectly secret, is it possible to define perfect secrecy for any system that we consider. Practically this becomes as an infeasible solution as because it demands the key size should be as same size of the actual message and we can never reuse the key.

These restrictions on perfect secrecy makes it impossible to use under practical conditions and always we end up with compromising on certain relaxations in security definition.

That clearly explains you the need for computational secrecy, in the next session lets discuss the most practical definition of security referred to as Computational Secrecy.