Deciphering the Mysteries

SoS 2023(Cryptography)

Arya Vishe
21D070018
Mentor- Nilabha Saha

IIT-Bombay

Table of Contents

General information

Cryptography is the practice and study of techniques for secure communication in the presence of adversarial behavior.

General information

Cryptography is the practice and study of techniques for secure communication in the presence of adversarial behavior.

In general, *Alice* is trying to communicate with *Bob* over an unsecure channel where *Eve* is trying to eavesdrop.

General information

Cryptography is the practice and study of techniques for secure communication in the presence of adversarial behavior.

In general, *Alice* is trying to communicate with *Bob* over an unsecure channel where *Eve* is trying to eavesdrop.

In general, we always deal with binary data when it comes to advanced forms of encryption as we are usually talking about communication between two computers considering the computation required to encrypt/decrypt.

General information

Cryptography is the practice and study of techniques for secure communication in the presence of adversarial behavior.

In general, *Alice* is trying to communicate with *Bob* over an unsecure channel where *Eve* is trying to eavesdrop.

In general, we always deal with binary data when it comes to advanced forms of encryption as we are usually talking about communication between two computers considering the computation required to encrypt/decrypt.
We call the original message to be communicated as *plaintext* and the encrypted data that is sent as *ciphertext*. The *key* is the information known to only *Bob* and *Alice*, that can be used for encryption/decryption of data.

Today's topics

We will try to cover the Key distribution problem, Asymmetric cryptosystems and RSA in specific.

The Key Distribution Problem

Traditional cryptosystems required the key to be physically passed to every recepient.

The Key Distribution Problem

Traditional cryptosystems required the key to be physically passed to every recepient.

- Even advanced machines like the Enigma would require high ranking officers to carry around an Enigma machine and a sheet with all codes for a specific month.
- Banks in the primitive age of the internet would hand deliver keys to their most important clients to ensure that they can remotely access some services.

Obviously it is not practical to meet each and every person face to face if you want to start communication with them over the internet.

Symmetric and Asymmetric cryptosystems

We can divide cryptosystems into 2 classes-

- Symmetric Cryptosystems- Anyone who can encrypt a *plaintext* also has enough information to be able to decrypt any *ciphertext*.

Symmetric and Asymmetric cryptosystems

We can divide cryptosystems into 2 classes-

- Symmetric Cryptosystems- Anyone who can encrypt a *plaintext* also has enough information to be able to decrypt any *ciphertext*.
- Asymmetric Cryptosystems- Anyone can encrypt (*Alice*) but only the receiver has enough information to decrypt (*Bob*).

As you can already see, asymmetric cryptosystems do show some promise for having a solution to the key distribution problem.

Table of Contents

The abstract concept

It consists of an encryption function and a decryption function (publicly known) which use a public and private key respectively.

The abstract concept

It consists of an encryption function and a decryption function (publicly known) which use a public and private key respectively.
Let $\mathcal{C}$ and $\mathcal{M}$ be the domains of ciphertext and plaintext respectively.

$$e_{k_{public}}() : \mathcal{M} \to \mathcal{C} \text{ AND } d_{k_{private}}() : \mathcal{C} \to \mathcal{M}$$

The abstract concept

It consists of an encryption function and a decryption function (publicly known) which use a public and private key respectively.

Let $\mathcal{C}$ and $\mathcal{M}$ be the domains of ciphertext and plaintext respectively.

$$e_{k_{public}}() : \mathcal{M} \rightarrow \mathcal{C} \text{ AND } d_{k_{private}}() : \mathcal{C} \rightarrow \mathcal{M}$$

If m = plaintext and c = corresponding ciphertext then,

$$c = e_{k_{public}}(m) \Rightarrow m = d_{k_{private}}(c) = d_{k_{private}}(e_{k_{public}}(m))$$

The abstract concept

It consists of an encryption function and a decryption function (publicly known) which use a public and private key respectively.
Let $\mathcal{C}$ and $\mathcal{M}$ be the domains of ciphertext and plaintext respectively.

$$e_{k_{public}}() : \mathcal{M} \rightarrow \mathcal{C} \text{ AND } d_{k_{private}}() : \mathcal{C} \rightarrow \mathcal{M}$$

If $m =$ plaintext and $c =$ corresponding ciphertext then,

$$c = e_{k_{public}}(m) \Rightarrow m = d_{k_{private}}(c) = d_{k_{private}}(e_{k_{public}}(m))$$

In a way, you can say that $d_{k_{private}}()$ is the inverse of $e_{k_{public}}()$

Requirements of this function

The function $e_{k_{public}}()$ should one such that-

1. Finding $k_{private}$ from any information about k_{public} should be a difficult task.

Requirements of this function

The function $e_{k_{public}}()$ should one such that-

1. Finding $k_{private}$ from any information about k_{public} should be a difficult task.
2. Figuring out the input of $e_{k_{public}}()$ from the output should be difficult without knowing $k_{private}$.

One candidate for such a function is exponentiation in modulo.

Modulo function

$$a^b \equiv c \bmod (d)$$

You can get a lot of variation in c by varying b in the above equation and thus can be used as a "trapdoor function".

Depending on the trapdoor information, we can use it for 2 different cryptosystems

Discrete Logarithm Problem (DLP)

$$g^x \equiv h \bmod (p)$$

The problem of finding x can be used

1. Diffie-Hellman Cryptosystem
2. ElGamal Cryptosystem

Integer Factorisation Problem

$$x^e \equiv h \bmod (p \cdot q)$$

The problem of finding x here can be used in

1. RSA cryptosystem implementation.

Table of Contents

Some necessary math

The math involved would be too heavy for such a short video so instead I will just ask you to suspend disbelief for a few moments with the following few theorems.[1].

[1] You can refer to Chapter: RSA and Integer Factorisation in the endterm report

Some necessary math

If we are looking at the equation,

$$x^e \equiv h \bmod (p) \text{ where } p \in \text{prime numbers}, \ h \in (1, p)$$

Then solving for x is trivially easy and involves finding d such that $d \cdot e \equiv 1 \bmod (p-1)$

Some necessary math

If we have to solve any equation

$$x^e \equiv h \bmod (p \cdot q)$$

We can easily decompose it to two congruences

$$x^e \equiv h \bmod (p)$$

and

$$x^e \equiv h \bmod (q)$$

and then use Chinese Remainder Theorem to stitch the solutions of the 2 congruences to get the solution for the original congruence.

But wait!
Did we just say that

$$x^e \equiv h \bmod (p \cdot q)$$

is easy to solve?

But wait!
Did we just say that

$$x^e \equiv h \bmod (p \cdot q)$$

is easy to solve?
Yes.

But wait!
Did we just say that

$$x^e \equiv h \bmod (p \cdot q)$$

is easy to solve?
Yes.
Then what is the difficult part in integer factorisation?

Some necessary math

We can only solve this equation

$$x^e \equiv h \bmod (p \cdot q)$$

If we know both p and q individually. If you just know $N = p \cdot q$, you will HAVE to factorise N else you cannot proceed with solving it.

Some necessary math

We can only solve this equation

$$x^e \equiv h \bmod (p \cdot q)$$

If we know both p and q individually. If you just know $N = p \cdot q$, you will HAVE to factorise N else you cannot proceed with solving it.
And factorisation does not have any easy algorithm known as of yet.

Table of Contents

RSA Implementation

1. *Bob*$\Rightarrow$Chooses a $p, q \in$ primes to construct $N = p \cdot q$.[2]

[2] e is then selected such that $\gcd(e, (p-1) \cdot (q-1)) = 1$. There are some more nuances in selecting e

RSA Implementation

1. *Bob*$\Rightarrow$Chooses a $p, q \in$ primes to construct $N = p \cdot q$.[2]
2. *Alice*$\Rightarrow$Converts her plaintext to an integer m such that $1 \leq m < N$. She then sends $c \equiv m^e \bmod (N)$ to *Bob*.

[2] e is then selected such that $\gcd(e, (p-1)\cdot(q-1)) = 1$. There are some more nuances in selecting e

RSA Implementation

1. *Bob*⇒Chooses a $p, q \in$ primes to construct $N = p \cdot q$.[2]
2. *Alice*⇒Converts her plaintext to an integer m such that $1 \leq m < N$. She then sends $c \equiv m^e \bmod (N)$ to *Bob*.
3. *Bob*⇒Finds d using $d \cdot e \equiv 1 \bmod \big((p-1) \cdot (q-1)\big)$ and calculates $c^d \equiv m^{e \cdot d} \equiv m \bmod (N)$.

[2] e is then selected such that $\gcd(e, (p-1) \cdot (q-1)) = 1$. There are some more nuances in selecting e

RSA Implementation

1. *Bob*⇒Chooses a $p, q \in$ primes to construct $N = p \cdot q$.[2]
2. *Alice*⇒Converts her plaintext to an integer m such that $1 \leq m < N$. She then sends $c \equiv m^e \bmod (N)$ to *Bob*.
3. *Bob*⇒Finds d using $d \cdot e \equiv 1 \bmod \big((p-1)\cdot(q-1)\big)$ and calculates $c^d \equiv m^{e \cdot d} \equiv m \bmod (N)$.
4. *Eve*⇒This whole while had m^e but inversion required her to factor N which is a difficult task without an easy algorithm as is the case with DLP.

[2] e is then selected such that $\gcd(e, (p-1)\cdot(q-1)) = 1$. There are some more nuances in selecting e

Thank you. :)