Scribe: Cryptography and Network Security (Class.10.1)

Venu Gopal Bandhakavi

4-Nov-2020

1 Introduction

In this scribe we will cover the following topics:

1. Public Key Cryptography
2. One way functions
3. RSA Algorithm

2 Public Key Cryptography

In public key cryptography the receiver and sender both have a set of two keys public key and private key. If Alice wants to send a message to Bob in a insecure channel , Alice encrypts the message using the Bob's public key and this ciphertext can only be decrypted by Bob's private key. The public key of a user is public where as the private key is only known to user. Here private key can give out information about public key but public key should not give out any information about private key. The public key cryptography provides not only confidentiality but also authenticity in the form of digital signatures.

2.1 One way Functions

These are the functions which are easy to compute in one given direction i.e when given x it is easy to find $f(x)$ but not the other way around. In these functions "trapdoors" are used to create keys. These functions should be average time hard to find x from $f(x)$. The one way functions with these trapdoors are called trapdoor one way functions. The functions used for public key cryptography are canditate one way functions i.e they are potentially a one way function.

For example if N was given then it is harder to find p, q such that $N = pq$ as it is hard to find the factors of a given number (which is very large). But the rise of quantum computing threatens the hardness of these functions.

2.2 Encryption

If we encrypt the plaintext with Bob's public key then only Bob's private key can decrypt the ciphertext. here Bob's public key is known to the public.

$$E_{Pr_k}(E_{Pu_k}(P)) = P$$

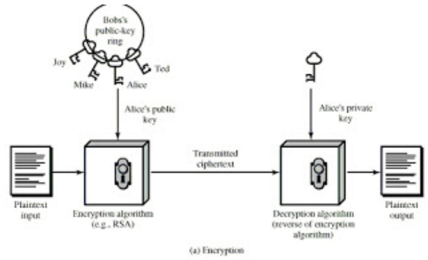

Figure 1: Encryption

2.3 Authentication

Public key cryptography provides authentication by the help of digital signatures. The digital signature is signed by encrypting with the user's private key and anyone can decrypt the signature with user's public key and validate the signature that it is from the user. Anyone can validate the signature by decrypting it but only user can sign it with encrypting it.

$$E_{Pu_k}(E_{Pr_k}(S)) = S$$

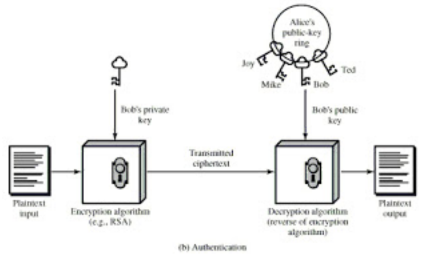

Figure 2: Authentication

3 RSA Cryptosystem

It is a public key cryptosystem. Let $n = pq$ where p, q are prime numbers(generally very large). Let $P = C = Z_n$ and we define key

$$K = \{(n, p, q, a, b) : ab \equiv 1(mod\phi(n)\}$$

for $k = (n, p, q, a, b)$

$$e_k(x) = x^b(modn)$$

and

$$d_k(y) = x^a(modn)$$

Here , $x, y \in Z_n$ are plaintext and ciphertext repectively. (n, b) is the public key and (p, q, a) is the private key. We can see that given the private key it is easy to find the public key. But it is hard to find the private key given the public key. We should find p, q, a to know the private key and it is very hard to factorize a number. It is also very hard to find a as it is very hard itself to calculate $\phi(n)$.

The size of n could vary from 1024 bits to 4096 bits. We can see that these numbers are very large. It is very difficult to prime factorize n into p, q whcih could be atleast be 512 bits.

3.1 Proof of Correctness

$ab \equiv 1(mod\phi(n) \Rightarrow ab = 1 + t\phi(n),\ t \geq 1$ and $t \in Z$

Suppose , $x \in Z_n^*$ then,

$x \equiv x^{ab} \equiv x^{1+t(n)}$

From Euler's Theorem,

$x(x^{\phi(n)})^t \equiv\ x\ (mod\ n)$

Now, Let us consider $x \in Z_n \setminus Z_n^*$,

so $gcd(x, n) \neq 1 \Rightarrow x$ is multiple of p or x is a multiple of q

$\Rightarrow gcd(x, p) = p$ or $gcd(x, q) = q$

if $gcd(x, p) = p$ then $gcd(x, q) = q$

Both p and q cannot be multiples of x because then x cannot be less than n.

from Euler's Theorem,

$x^{\phi(q)} \equiv 1\ (mod\ q)$

$\Rightarrow x^{t\phi(n)} \equiv 1\ (mod\ q)$

$x^{t\phi(q)\phi(p)} \equiv 1\ (mod\ q)$

$\Rightarrow x^{t\phi(n)} \equiv 1\ (mod\ q)$

Thus , $x^{t\phi(n)} = 1 + kq,\ k \in Z^{+}$

Multiply , Both sides with x,

$x^{t\phi(n)+1} = x + kqx,\ k \in Z^{+}$

$gcd(x, p) = p$

$\Rightarrow x = cp$ where c is a positive integer

$x^{t\phi(n)} \equiv x^{ab} \equiv x(mod n)$ Similarly , We can prove for the case $gcd(x, q) = q$

let us take an example where $p = 131$ and $q = 137$ and $n = pq = 17947$.

$\phi(n) = (130)(136) = 17680$
$\phi(n) = 5 \times 2^4 \times 13^1 \times 17^1$

So , b should be not a multiple of $5, 2, 13, 17$. let us take $b = 729$. Here, we factorized $\phi(n)$ for not to take its multiples for b. Instead of this we can check whether $gcd(b, \phi(n)) = 1$. We get a from $ab \equiv 1(mod\phi(n))$. $a = 10089$

To get y we do encryption using public key b ,
$x = 1547$ then $y = (1547)^{729}(mod\ 17947) = 3784$

To get x we do decryption by using private key a ,
$x = 3784^{10089}\ (mod\ 17984) = 1547$.

3.2 Efficient Exponentiation

This algorithm is used for efficient modular exponentiation.

Algorithm 1: Square-And-Multiply(x,c,n)

```
z ← 1
for $i \leftarrow l-1$ downto 0 do
    z ← $z^2 (mod\ n)$
    if $c_i == 1$ then
        $z \leftarrow (z \times x)(mod\ n)$
    end
end
return z
```

Here , we represent c in binary such as $c = \sum_{i=0}^{l-1} c_i 2^i$. This algorithm decreases the time complexity of algorithm to bit length of c.

3.3 RSA parameter generation

1. generate two large prime numbers p and q, such that $p \neq q$.
2. $n \leftarrow pq$ and $\phi(n) \leftarrow (p-1)(q-1)$.
3. Choose a number b $(1 < b < \phi(n))$ and $gcd(b, \phi(n)) = 1$.
4. $a \leftarrow b^{-1}$ $(mod\ \phi(n))$.
5. Public key is (n, b) and private key is (p, q, a).

- n is known , but factors are not known. It is very hard to factorize n.
- b is known and to compute a from b we need to know $\phi(n)$ which cannot be known without p , q.
- typically n is 1024 bits so p and q would be around 512 bits which are quite large.

4 conclusion

In this scribe we discussed about Public key cryptosystems. We discussed about One way functions and how trapdoors are used in them. Encryption , Decryption and authentication using Public key cryptosystems. We discussed about RSA algorithm. We discussed about efficient ways of calculating modular exponentiation and parameter generation of RSA.