[CS304] Introduction to Cryptography and Network Security

Course Instructor: Dr. Dibyendu Roy
Scribed by: Dipean Dasgupta (202151188)

Winter 2023-2024
Lecture 15,16 (Week 9)

1 RSA (Rivest Shamir Adleman) Encryption

RSA (Rivest-Shamir-Adleman) is a widely used public-key cryptosystem for secure data transmission. It is named after its inventors, Ron Rivest, Adi Shamir, and Leonard Adleman, who introduced it in 1977.
It is the first public key encryption algorithm.RSA is based on the mathematical properties of large prime numbers. Before beginning any discussion on RSA encryption, let's first recall a few concepts.

- The number of integers less than n that are co-prime to n, or the number of x such that gcd(x, n) = 1 where $1 \leq x \leq n-1$, is indicated by the Euler's Totient Function $\phi(n)$. $\phi(8) = 4, \{1, 3, 5, 7\}]$, for instance, is co-prime to 8. Also, we know,

$$\phi(p) = p - 1, \text{ if p is prime}$$
$$\phi(p^k) = p^k(1 - \tfrac{1}{p})$$

- Let there be a set $S = \{x \bmod \text{m}\}$ such that $|S| = m$.

$$S = \{r_1, r_2, \ldots, r_m\}$$

Every element in the set S is distinct; typically, they range from 0 to m-1. Let a be an integer for which $gcd(a, m) = 1$. Suppose there exists an additional set S_1 such that,

$$S_1 = \{ar_1 \bmod m, ar_2 \bmod \text{m}, \ldots, ar_m \bmod m\}$$

$\{ar_1 \bmod m, ar_2 \bmod \text{m}, \ldots, ar_m \bmod m\}$ will also be m unique elements since $\{r_1, r_2, \ldots, r_m\}$ are distinct elements and gcd(a, m) = 1. Contradiction can be used to demonstrate this. Assume that for every $r_i \neq r_j$, $ar_i = ar_j$. Consequently,

$$ar_i \equiv ar_j \bmod m$$

$1 = ab + ms$ since gcd(a, m) = 1 (from Bezout's Identity). Therefore, $ab \equiv 1$ mod m exists for some integer b. We refer to the value of b as the multiplicative inverse of a, and the Extended Euclidean Algorithm can be used to find it. As a result, multiplying both sides of the preceding equation by b yields,

$$b \cdot a \cdot r_i \equiv b \cdot a \cdot r_j \bmod m$$

$$r_i \equiv r_j \bmod \text{m } (\because ab \equiv 1 \bmod \text{m})$$

Hence, it is a contradiction to our initial assumption that $r_i \neq r_j$. Hence, elements in the set S_1 will be unique iff gcd(a, m) = 1.

1.1 Euler's Theorem

$a^{\phi(m)} \equiv 1$ (mod m) if gcd(a, m) = 1. Assuming that we have a set S, let's say that

$$\text{S} = \{ \text{ x} \mid \text{gcd(x, m)} = 1\}$$
$$\text{S} = \{s_1, s_2, s_3, s_4, ..., s_{\phi(m)}\}$$

Let us consider gcd(a, m) = 1 and create another set S_1 such that

$$S_1 = \{ \ as_1, as_2, as_3..., as_{\phi(m)} \ \}$$

As was mentioned in the last section, if $as_i \equiv as_j$ (mod m) $\Rightarrow s_i \equiv s_j$(mod m)
Considering that $b.a \equiv 1$ (mod m) and gcd(a, m) = 1

$$|S| = \phi(m)$$
$$|S_1| = \phi(m)$$

Given that a and s_i are co-prime with m, there has to be some correspondence between the elements of S and S_1.

$$s_i \equiv as_j \text{ (mod m)}$$

Let us now take product on both sides,

$$\prod_{i=1}^{\phi(m)} s_i \equiv \prod_{j=1}^{\phi(m)} as_j (mod\ m)$$
$$\Rightarrow \prod_{i=1}^{\phi(m)} s_i \equiv a^{\phi(m)} \prod_{j=1}^{\phi(m)} s_j (mod\ m)$$

As gcd(s_i, m) = 1, each s_i will have multiplicative inverse under mod m. So, after simplifying,

$$a^{\phi(m)} \equiv 1 (mod\ m)$$

1.2 Fermat's Theorem

If p is a prime number and p does not divide a(means that p is co-prime to a), then

$$a^{p-1} \equiv 1 (mod\ p)$$

Using Fermat's theorem,

$$\Rightarrow a^p \equiv a (mod\ p)$$

Note: If $p|a$(P divides a), then,

$$\text{a} \equiv 0 \text{ mod p}$$
$$\Rightarrow a^p \equiv 0\ mod\ p$$
$$\Rightarrow a^p \equiv a\ mod\ p$$

But the Fermat's theorem will not hold when p does not divide a.

1.3 RSA Cryptosystem

Few facts:

- g(a, m) = 1, then $a^{\phi(m)} \equiv 1(mod\ m)$
- $a^{p-1} \equiv 1(mod\ p)$

Now, let us understand the components of RSA

1. n = pq, where p, q are primes
2. Plaintext space = $_n$
   Ciphertext space = $_n$
3. Key space = {K = (n,p,q,e,d) | $ed \equiv\ (mod\ \phi(n))$}
4. Encryption:

$$E(x, K) = c$$
$$c = E(x, K) = x^e (\text{mod } n)$$

5. Decryption:

$$Dec(c, K) = x$$
$$c = Dec(c, K) = c^d (\text{mod } n)$$

We know that e and d are related as:

$$ed \equiv 1 \text{ mod } \phi(n)$$
$$\Rightarrow ed - 1 = t.\phi(n)$$
$$\Rightarrow 1 = ed + t_1.\phi(n)$$
$$1 = gcd(e, \phi(n)) = ed + t_1.\phi(n)$$

Encryption:

$$c = x^e (\text{mod } n)$$

Decryption:

$$x = c^d (\text{mod } n)$$
$$c^d = (x^e)^d \text{ mod } n$$
$$c^d = x^{ed} \text{ mod } n$$
$$\text{Now using } ed = 1 + t.\phi(n) \text{ from above}$$
$$c^d = x^{1+t.\phi(n)} \text{ mod } n$$
$$c^d = x.x^{t.\phi(n)} \text{ mod } n$$
$$\text{Since p, q are primes and n = pq, then } \phi(n) = (p-1)(q-1)$$
$$c^d = x.x^{t[(p-1)(q-1)]} \text{ mod } n$$
$$\text{Finally, } c^d = x.x^{t[(p-1)(q-1)]} \text{ mod } (pq)$$

Now, simplifying the part $x^{t[(p-1)(q-1)]}$ mod (pq), where x $\in$
We check $x^{t[(p-1)(q-1)]}$ mod p,

$$\equiv x^{p-1^{t(q-1)}} \text{ mod p}$$
$$\equiv 1(\text{mod p}) \text{ [As } x^{p-1} \equiv 1 mod\ p]$$

Now we check $x^{t[(p-1)(q-1)]}$ mod q

$$\equiv x^{q-1^{t(p-1)}} \text{ mod q}$$
$$\equiv 1(\text{mod p}) \text{ [As } x^{q-1} \equiv 1 mod\ q]$$

We finally have,

$$x^{t[(p-1)(q-1)]} \equiv 1(\text{mod p})$$
$$x^{t[(p-1)(q-1)]} \equiv 1(\text{mod q})$$
$$\Rightarrow x^{t[(p-1)(q-1)]} \equiv 1(\text{mod pq})$$

Substituting the result,

$$c^d = \text{x}.x^{t[(p-1)(q-1)]} \text{ mod (pq)}$$
$$c^d = \text{x}.1 \text{ mod (pq)}$$
$$c^d = \text{x mod (pq)}$$

Hence, decryption is successful!

Now let us consider a scenario where Alice is trying to communicate with Bob. Here, two keys play the main role-one is public key and the other is secret key. Here, Bob is encrypting the message and sending it to Alice and Alice has both the keys. Public key is known to Bob but the secret key is not known to Bob.

Alice

n = pq and p, q:large prime numbers

ed $\equiv$ 1 (mod ϕ(n))

choosing e

Public key of Alice = (n, e)

generate d using Extended Euclidean algorithm

Secret key of Alice = (p, q, d)

Bob

Now Bob selects a message x from $_n$

x

n, e for Alice are known, so he can encrypt

y = $x^e(modn)$

Now the message y is sent to Alice, she can decrypt it with her secret key as :

$$x = y^d \text{ (mod n)}$$

How can we find p and q in polynomial time if we were given n?
Using a loop that goes from 2 to $\sqrt{n}$, we can determine if a factor has been found if each time we calculate n % i is zero. If so, we can proceed to determine if the factor is prime. If so, we calculate q by dividing n by p. When n is big, finding the prime factors of that number is a computationally challenging problem.
Note: We shall be able to calculate ϕ(n) if we can calculate p,q from n. And since we already know e, the security of RSA will be compromised since we may use the extended Euclidean technique to find d. Thus, the foundation of RSA is the difficulty of the factorization problem.

1.3.1 RSA Problem

We have public key (n, e) and c. If from this we can find x ($c = x^e$), we will be able to break security of RSA.
We have an algorithm to solve the RSA problem, i.e., it can find the decryption without the factorization. Is this true? **Note:** Finding the factors is not certain even if we manage to crack the RSA. But we can always break the RSA if we have the necessary elements. However, this isn't always the case. Thus, RSA is secure if two conditions are met:

- factorization is very difficult.
- decryption is very difficult.

Note : In general, public key encryption is complex due to the x^e and c^d operations. There are a lot of exponential operations. Thus, people typically steer clear of them.
RSA is considered secure due to the difficulty of factoring large composite numbers, which is the basis of breaking the system. The security of RSA relies on the difficulty of factoring the product of two large prime numbers, which forms the public key.

2 Diffie Hellman Key Exchange

Known as Diffie—Hellman key exchange, the first public-key algorithm was disclosed in the groundbreaking article by Diffie and Hellman that established public-key cryptography. This key exchange technology is used in many commercial goods.
The algorithm's goal is to make it possible for two users to safely trade keys so that messages can be symmetrically encrypted later on. The algorithm is restricted to exchanging confidential values. The Diffie-Hellman algorithm depends for its effectiveness on the difficulty of computing discrete logarithms.A primitive root of a prime number p is one whose powers modulo p generate all the integers from 1 to $p-1$. That is, if a is a primitive root of the prime number p, then the numbers

$$a \mod p, a^2 \mod p, \ldots, a^{p-1} \mod p$$

are distinct and consist of the integers from 1 through $p-1$ in some permutation. For any integer b and a primitive root a of prime number p, we can find a unique exponent i such that

$$b \equiv a^i \pmod{p} \quad \text{where } 0 \leq i < (p-1)$$

Algorithm
For this scheme, there are two publicly known numbers: a prime number q and an integer α that is a primitive root of q. Suppose the users A and B wish to create a shared key.
User A selects a random integer $X_A < q$ and computes $Y_A = \alpha^{X_A} \mod q$. Similarly, user B independently selects a random integer $X_B < q$ and computes $Y_B = \alpha^{X_B} \mod q$. Each side keeps the X value private and makes the Y value available publicly to the other side. Thus, X_A is A's private key and Y_A is A's corresponding public key, and similarly for B. User A computes the key as

$$K = (Y_B)^{X_A} mod q$$

and user B computes the key as

$$K = (Y_A)^{X_B} mod q.$$

These two calculations produce identical results:

$$
\begin{aligned}
K &= (Y^b)^a \mod q \\
&= (x^a \mod q)^a \mod q \\
&= (x^{ab} \mod q)^a \mod q \\
&= x^{ab} \cdot x^a \mod q && \text{by the rules of modular arithmetic} \\
&= (x^{ab} \cdot x^b) \mod q \\
&= (x^{ab} + ab) \mod q \\
&= ((x^a mod q)^b) mod q
\end{aligned}
$$

The outcome is the exchange of a hidden value between the two parties. This secret value is typically utilized as a shared symmetric secret key. Now imagine a malicious party who is able to see the key exchange and wants to find the secret key K. An enemy only has the following resources at their disposal because X_a and X_b are private: q, a, Y_a, and Y_b. In order to mine the key, the opponent is thus compelled to take a discrete logarithm. For example, to determine the private key of user B, an adversary must compute

$$X_b = dlog_a(Y_b).$$

The adversary can then calculate the key K in the same manner as user B calculates it. That is, The adversary can calculate K as

$$K = (Y_a)^{X_b} mod q.$$

Because it is reasonably cheap to calculate exponentials modulo a prime, but very difficult to calculate discrete logarithms, the Diffie-Hellman key exchange is secure. The latter task is deemed unfeasible for large primes.

Man-in-the-Middle Attack

The man-in-the-middle attack (MITM) is a significant threat in cryptographic systems. It occurs when an adversary, often called Darth, intercepts and modifies the communication between two parties, Alice and Bob, without their knowledge. Here's how it typically unfolds:

1. Alice sends a message to Bob, which is intercepted by Darth.
2. Darth modifies the message and sends it to Bob, pretending to be Alice.
3. Bob receives the message, believing it is from Alice, and responds.
4. Darth intercepts Bob's response, modifies it, and sends it to Alice, pretending to be Bob.

This attack can compromise the confidentiality and integrity of the communication, allowing Darth to gain unauthorized access to sensitive information.