The limitations of RSA

By: Erdembileg.A

What are the limitations of RSA algorithm in terms of the key size, and how can these limitations be overcome?

Computer science

… words

Contents

Background Information

What is RSA?

RSA (Rivest-Shamir-Adleman) is an asymmetric encryption algorithm that is widely used in for secure data transmission. It is named after its creators, Ron Rivest, Adi Shamir, and Leonard Adleman, who first proposed it in 1977. RSA is a public-key cryptosystem, meaning it uses two different keys: a public key for encryption and a private key for decryption.

The security of RSA is based on the mathematical difficulty of factoring large composite numbers. The algorithm involves generating a pair of mathematically related keys: a public key and a private key. The public key is made available to anyone who wants to send encrypted data to the owner of the corresponding private key. The private key is kept secret and is used to decrypt the received messages.

Key Generation

1. Choose two distinct prime numbers, p and q. These should be large random prime numbers. The security of the RSA key relies on the difficulty of factoring the product of these two primes.

2. Compute n, the modulus, by calculating the product of p and q: $n = p * q$.

3. Calculate Euler's totient function of n, denoted as $\varphi(n)$. Euler's totient function counts the positive integers up to n that are relatively prime to n.

$$\varphi(n) = (p-1)*(q - 1).$$

4. Choose an integer e such that $1 < e < \varphi(n)$ and $\gcd(e, \varphi(n)) = 1$. The value of e is typically chosen as a small prime, such as 65537 (2^16 + 1), which is commonly used due to its efficient computation.

5. Compute the modular multiplicative inverse of e modulo $\varphi(n)$. In other words, find d such that $(d * e) \bmod \varphi(n) = 1$. This can be done using the Extended Euclidean Algorithm or other modular inverse algorithms.

6. The public key is the pair (e, n), and the private key is the pair (d, n). It's important to keep the private key secure, as it's used for decryption and signing operations.

An example of the RSA key generation step:

1) Let's say p = 61 and q = 53.
2) Calculate the modulus, n. The modulus is the product of p and q. n = p * q = 61 * 53 = 3233.
3) Calculate the totient, φ (phi).
   For RSA, φ(n) = (p - 1) * (q - 1). φ(n) = (61 - 1) * (53 -1) = 60 * 52 = 3120.
4) Common choices for e include 3, 17, or 65537. For this example, we'll use e = 17.
5) Compute the decryption exponent, d. The decryption exponent is the modular multiplicative inverse of e modulo φ(n). In other words, (d * e) mod φ(n) = 1. In this case, d = 2753.
6) The public key is composed of the modulus, n, and the encryption exponent, e. It is shared openly. Public key (e, n) = (17, 3233).
7) The private key is composed of the modulus, n, and the decryption exponent, d. It must be kept secret and should not be shared. Private key (d, n) = (2753, 3233).

GCD (Greatest Common Divisor)

GCD is a mathematical concept used to find the largest positive integer that divides two or more numbers without leaving a remainder. In other words, the GCD is the largest number that divides the given numbers evenly.

For example, let's consider the numbers 12 and 18. The divisors of 12 are 1, 2, 3, 4, 6, and 12, while the divisors of 18 are 1, 2, 3, 6, 9, and 18. The largest number that appears in both lists is 6, so the GCD of 12 and 18 is 6.

The GCD is often calculated using the Euclidean algorithm, which involves repeatedly dividing the larger number by the smaller number and replacing the larger number with the remainder until the remainder becomes zero. The final non-zero remainder obtained using this process is the GCD.

Modular Inverse

Modular inverse algorithms are used to compute the multiplicative inverse of a number modulo a given modulus. In modular arithmetic, the modular inverse of a number "a" modulo "m" is another number "b" such that (a * b) mod m = 1. In other words, it is the number that, when multiplied by "a" and then reduced modulo "m," gives a result of 1.

Step 1: Find the modular multiplicative inverse of 7 modulo 15 using Euler's algorithm. To find the modular multiplicative inverse, we need to find a number, let's call it a, such that (7 * a) mod 15 = 1. Using Euler's algorithm, we have: 15 = 2 * 7 + 1 7 = 7 * 1 + 0 Working backward: 1 = 15 - 2 * 7 Therefore, the modular multiplicative inverse of 7 modulo 15 is -2.

Step 2: Multiply both sides of the equation by the modular multiplicative inverse. (-2) * (7 * x) mod 15 = (-2) * 1 mod 15 Simplifying: (-14 * x) mod 15 = (-2) mod 15

Step 3: Compute the result on the left side. (-14 * x) mod 15 is equivalent to (1 * x) mod 15 since (-14 mod 15) is congruent to 1. So, the equation simplifies to: x mod 15 = (-2) mod 15

Step 4: Find the value of x. To find the value of x, we need to find an integer that is congruent to -2 modulo 15. Since -2 is congruent to 13 modulo 15, we can conclude that: x = 13 Therefore, the value of x that satisfies the equation (7 * x) mod 15 = 1 using Euler's algorithm is x = 13.

Euler's totient function

Euler's totient function, also known as Euler's phi function, is a mathematical function denoted by the symbol φ (phi). It is named after the Swiss mathematician Leonhard Euler, who introduced it in the 18th century. The totient function is defined for a positive integer n and calculates the count of positive integers that are relatively prime to n, i.e., the number of positive integers less than n that do not share any common divisors with n except for 1.

According to the https://brilliant.org, Euler's totient function (also called the Phi function) counts the number of positive integers less than n that are coprime to n.

Formally, for a positive integer n, the totient function φ(n) is defined as the number of positive integers k (1 ≤ k ≤ n) such that gcd(n, k) = 1, where gcd stands for the greatest common divisor. In other words:

φ(n) = Count of {k ∈ [1, n] : gcd(n, k) = 1}

Some important properties of Euler's totient function are:

1. If n is a prime number, then φ(n) = n - 1, as all positive integers less than n are relatively prime to n.
2. If p is a prime number and k is a positive integer, then φ(p^k) = p^k - p^(k-1).
3. If m and n are co-prime (i.e., gcd(m, n) = 1), then φ(m * n) = φ(m) * φ(n).
4. In general, for any positive integer n, φ(n) = n * (1 - 1/p1) * (1 - 1/p2) * ... * (1 - 1/pk), where p1, p2, ..., pk are the distinct prime factors of n.
5. Euler's totient function finds applications in various areas of number theory, cryptography, and algorithms, especially in RSA encryption, where it is used to calculate the totient of large

numbers as part of the key generation process. The totient function also plays a crucial role in modular arithmetic and the study of groups and cyclic groups.

Co-prime

a and b are Co-prime numbers if (GCD (a,b) = 1)

Certain RSA Limitations

Works Cited

“What is RSA algorithm?”. Assessed 1 July. 2023

Euler's totient for RSA https://nitaj.users.lmno.cnrs.fr/RSAnitaj1.pdf and https://www.youtube.com/watch?v=qa_hksAzpSg

https://brilliant.org/wiki/eulers-totient-function/#:~:text=Euler's%20totient%20function%20(also%20called,that%20are%20coprime%20to%20n.