RSA Proof

An, Trevor, Trinity

December 11, 2023

Introduction

Rivest Shamir Adleman (RSA) is a popular cryptographic encryption algorithm. It is asymmetric, which means that each user is assigned a public and private key. Anyone can access and use a another person's public key to encode a message to them. Only the recipient can decrypt the message easily using their private key. RSA is based around the product of two large prime numbers. While the product is public, it is extremely hard to figure out what its factors are. These factors are used to encrypt and decrypt such that it is each to compute on every input, but hard to invert given just an output.

Implementation

Algorithm
The basic RSA algorithm is as follows:

1) Choose two large distinct prime numbers p and q randomly
2) Calculate an $n = p * q$
3) Calculate Euler's Totient Function $\phi(n)$, which is defined as count of numbers in {1,2,..., n-1} that are co-prime to n.
4) Choose an $e \in [2, \phi(n) - 1]$ that is co-prime to $\phi(n)$. (e, n) becomes the public key, and is public information. This is our "public key" or "public exponent".
5) Calculate $d \in [2, \phi(n) - 1]$ that satisfies the equation $e * d = 1(\mod \phi(n))$. In other words, $\phi(n)$ must be co-prime to e * d. There is a unique d, and furthermore, d must be co-prime to ϕ. d is the modular multiplicative inverse of e.

Once we have calculated d and e, $\{e, n\}$ becomes a users public key, and anyone can find these numbers. $\{d, pq\}$ becomes a users private key. Notice that $pq = n$, so the only real private information is d (and $\phi(n)$ by association).

Encryption and Decryption
Anyone can find and use a person's public key to encrypt a message $m \leq n$ to send to them. If the

message is greater than n, the sender can segment the message into smaller messages. Knowing someone's public key (e, n), calculate a cipher text C as:

$$C = m^e(\mod n)$$

This is hard to hard to decrypt without d because while n is public, the calculation of e is dependent on $\phi(n)$, which is calculated using the prime factors p and q. Even with n being public, because p and q are not public, finding $\phi(n)$ becomes difficult, thereby making decryption difficult.
To decrypt a cipher text C, the recipient uses d and n:

$$m = C^d(\mod n)$$

To get back the original message.

Proof

To prove correctness of the RSA algorithm, we need to show that decryption algorithm results in the original message for any message m:

$$m = c^d \mod (n)$$

To do this, we will prove that

$$m = c^d \mod (p)$$

and

$$m = c^d \mod (q)$$

By the Chinese Remainder Theorem, if both of these are true, than the same is true for the product $p * q = n$.

$$\begin{aligned} C^d &= (m^e)^d \mod p \qquad \text{by the original encryption algorithm and modular arithmetic} \\ &= m^{ed} \mod p \end{aligned}$$

$ed = 1 \mod \phi(n)$. This is because this is the definition of how we calculated d. Eueler's Totient function $\phi(n)$ is defined as the number of numbers from 1 to n that are co-prime to n. In this case, $\phi(n) = (p-1)(q-1)$ via the Inclusion-Exclusion Principle. See proof of equality in appendix below. Subbing in for ϕ, we get that $ed = 1 \mod (p-1)(q-1)$. By definition of remainder, another way to say this is that we know $ed = t\phi(n) + 1 = t(p-1)(q-1) + 1$ for some integer t. Plugging that back in we get:

$$\begin{aligned} &= m^{1+t\phi(n)} \mod p \\ &= m^{1+t(p-1)(q-1)} \mod p \\ &= m * m^{t(p-1)(q-1)} \mod p \\ &= m * m^{(p-1)*t(q-1)} \mod p \end{aligned}$$

At this point, we use Fermat's little theorem, which states that if p is a prime number, for any non-zero $a \in Z_p$, it holds that $a^{p-1} = 1 \mod p$. We use it to get rid of the m^{p-1}:

$$\begin{aligned} &= m * 1^{t(q-1)} \mod p \quad \text{By Fermat's Little Theorem} \\ &= m \mod p \end{aligned}$$

We have proven that $m = c^d \mod (p)$. The same can be said for $m = c^d \mod (q)$ by symmetry. By the Chinese Remainder Theorem, $m = c^d \mod (p * q) = cd \mod (n)$. Thus, decryption of an encrypted message results in the original message for any message m.

Glossary

Co-prime
The definition of co-prime (sometimes called relatively prime) is that given a two or more numbers, none of these numbers have a common factor with each other, except one. In other words, the greatest common denominator between the numbers is only one, or $gcd(p, q) = 1$
Greatest Common Divisor
The greatest common divisor is a number which between two or more numbers, is the largest factor between the two. It is often denoted as $gcd(p, q)$
Least Common Multiple
The least common multiple is the smallest positive integer that is evenly divisible by two or more numbers. It is denoted as $lcm(p, q)$.
Modular Multiplicative Inverse
The modular multiplicative inverse of an integer, a, is defined as an integer value x so that the statement below is true.

$$ax = 1(\mod m)$$

Both of these numbers, a and x, must be co-prime. In the context of the RSA algorithm, our decryption key (d) must be a modular multiplicative inverse of e, our encryption key. In other words, we can express it as

$$d = e^{-1} \mod \phi(n)$$

Extended Euclidean Algorithm
The Extended Euclidean Algorithm finds the modular multiplicative inverse using Bezout's identity. In the context of RSA, we can use what knowledge we have of the values to solve for e or d if we use Bezout's identity to create an equation:

$$ed + \phi(n)y = gcd(e, \phi(n))$$

We know that $gcd(e, \phi(n)) = 1$, hence

$$\begin{aligned} ed + \phi(n)y &= 1 \\ ed - 1 &= \phi(n) \end{aligned}$$

According to the Modular Arithmetic Congruence Relation, where

$$a - b = kn \text{ can be expressed as: } a \equiv b \mod n$$

$$\begin{aligned} ed - 1 &= \phi(n) \\ ed &\equiv 1 \mod \phi(n) \end{aligned}$$

Because we would have already chosen e and known the value for $\phi(n)$, it would be moderately easy to calculate d.

Bezout's Identity

Bezout's Identity states that if there are two integers, a and b, who have a greatest common denominator, then there exists two integers, x and y so that

$$ax + by = gcd(a, b)$$

Euler's totient function

Euler's Totient function $\phi(n)$ is defined as given a number, n, $\phi(n)$ is the total sum of positive integers that are lesser than n and are co-prime numbers to n.

Given that in our algorithm n is made up of two co-prime numbers, p and q, we know that the factors for n are p and q. To find the sum of co-prime numbers to n that are lesser than n we can use the Inclusion-Exclusion Principle.

We know that the total set of all possible values for this function is [1, n-1]. We know that the sum of this set is $n - 1$, or $pq - 1$

Out of this set, because we are only interested in co-prime numbers, we need to eliminate all numbers that have a common denominator with n. According to the definition of n, it only has 2 possible factors. Hence, we need to find the sum of numbers that have a factor of p or a factor of q and exclude it from the total set.

$$\begin{aligned} A &= \text{Set of all numbers that have a factor of p} \\ B &= \text{Set of all numbers that have a factor of q} \end{aligned}$$

Hence the definition for $\phi(n)$ is

$$\phi(n) = (pq - 1) - A - B \tag{1}$$

To calculate the total size of the set of all possible values, it is just $n - 1$. To calculate the size of sets A & B, it is a little more complicated.

The sum of numbers from 1 to n that have p as a factor can be calculated logically. From the range of 1 to n, the pattern will be p, 2p, 3p, 4p.....qp. However, we can't count qp because we are only looking at values that are less than n, so we are looking at a value that is one less than q as the max value that has a factor of p in the set. Hence, our set contains elements of p, 2p, 3p, 4p.....(q-1)p. Hence, the total size of this set is $q - 1$.

$$\begin{aligned} A &= |\{p, 2p, 3p, ..., (q-1)p\}| \\ &= q - 1 \end{aligned}$$

The sum of numbers from 1 to n that have a factor of q can likewise be calculated, giving us a set with the elements q, 2q, 3q, 4q, ... pq, but since we can't count qp, the largest value that has a factor of p will be $(p-1)q$. hence

$$\begin{aligned} B &= |\{q, 2q, 3q, ..., (p-1)q\}| \\ &= p-1 \end{aligned}$$

Furthermore, because both q and p are co-prime we know that A and B do not intersect since no elements can be in both A and B. Hence, our final equation for Euler's totient function is:

$$\begin{aligned} \phi(n) &= (pq-1) - (q-1) - (p-1) \\ &= pq - 1 - q + 1 - p + 1 \\ &= pq - q - p + 1, \text{ this resembles the quadratic equation} \\ &= (p-1)(q-1), \text{ factoring} \end{aligned}$$

Hence, if we have $\phi(n)$ where $n = p \times q$ and p and q are co-prime,

$$\phi(n) = (p-1)(q-1)$$

Carmichael's Function
An alternative totient function to Euler's totient function is the Carmichael function $\lambda(n)$. When p and q are prime and $n = pq$, $\rho(n) = lcm(p-1)(q-1)$. Any possible key pairs obtained using Eulers Phi Function is also possible using the Carmichael function and can speeds up the key generation in some cases.

n	1	2	3	4	**8**	**12**	**15**	**16**	**20**	**21**
$\phi(n)$	1	1	2	2	2	2	4	4	4	6
$\lambda(n)$	1	1	2	2	4	4	8	8	8	12

Chinese Remainder Theorem
Let m and n be relatively prime positive integers. For all integers a and b, the pair of congruences $x \equiv a \mod m, x \equiv b \mod n$ has a solution, and this solution is uniquely determined modulo mn.

The theorem holds generally for any system of the following format: Let $m_1, m_2, ...m_n$ be pairwise relatively prime positive integers. The system

$$\begin{aligned} x &\equiv a_1(\mod m_1), \\ x &\equiv a_2(\mod m_2), \\ &... \\ x &\equiv a_n(\mod m_n), \end{aligned}$$

has a unique solution mod $m = m_1 m_2 ... m_n$.
Proof
Let $M_k = m/m_k$ for $k = 1, 2..., n$, meaning that M_k is the product of the moduli except for m_k. Since m_i and m_k have no common factors greater than 1 when $i \neq k$, that means that

$\gcd(m_k, Mk) = 1$, and that there is an integer y_k, an inverse of $M_k \mod (m_k)$ such that $M_k y_k \equiv 1(\mod m_k)$. To construct a simultaneous solution, we can form the sum

$$x = a_1 M_1 y1 + a_2 M_2 y_2 + ... + a_n M_n y_n$$

Since $M_j \equiv 0(\mod m_k)$ whenever $j \neq k$, all terms except the kth term in the sum are congruent to 0 mod m_k. Sine $M_k yk \equiv 1(\mod m_k)$:

$$x \equiv a_k M_k y_k \equiv a_k(\mod m_k$$

for $k = 1, 2..., n$. We have shown that x is a simultaneous solution to the n concurrences.

Modular Congruence
Modular congruence means that two numbers have the same remainder after the same modular and is written as $a \equiv b \mod c$. For example, $11 \equiv 16 \mod 5$
Fermat's Little Theorem
For any prime number p and any integer a such that p does not divide a, then $a^{p-1} \equiv 1 \pmod p$.

Proof. Generally, we are trying to prove that $a^p \equiv a(\mod p)$. If the greatest common divisor of a and p is 1, it follows that $a^{p-1} \equiv 1(\mod p)$ by dividing out a factor of a.
Fix p as a prime number. The base case, $1^p \equiv (\mod p)$, is clearly true. Inductive hypothesis: Suppose the statement $ap \equiv a(\mod p)$ is true for some integer $k >= 1$. This means that

$$kp \equiv k(\mod p)$$

Now we'll examine the $k + 1$ case:

$$\begin{aligned}(k+1)^p &\equiv k^p + \binom{p}{1} * k^{p-1} + \binom{p}{2} k^{p-2} + ... + \binom{p}{p-1} k + 1 \quad \text{By the Binomial Theorem} \\ &\equiv k + k(k)^{p-1} + p\frac{(p-1)}{2} k^{p-2} + ... + pk + 1 \\ &\equiv (k+1) + pk^{p-1} + p\frac{(p-1)}{2} k^{p-2} + ... + pk\end{aligned}$$

Because everything to the right of (k+1) is a multiple of p, that means that each of those terms is equal to zero when working in $\mod p$. Therefore we can conclude that

$$(k+1)^p \equiv (a+1)(\mod p)$$

We have proven that $p(1)$ is true and that if $p(k)$ is true for some $k \geq 2$, then $p(k+1)$ is true. Thus, by the principle of weak induction, $p(n)$ is true for all $n >= 1$, $a^p \equiv a(\mod p)$, and

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

■

Citations

View our annotated bibliography here