RSA for beginners - version 0.1

Louis Botterill

2025-04-04

1 RSA public key cryptography

1.1 Introduction

RSA is a public key cryptographic system. Let's see what that means and how it works. Firstly, there are two main types of cryptographic systems, Symmetric and asymmetric.

Symmetric - encryption and decryption are inverses, using the same privately shared key for both.

- Advantages - speed and efficiency. Smaller keys for same cryptographic strength.
- Disadvantages - distribution of these private keys, without risk of compromising them.

Asymmetric - there are two keys, a public and a private, mathematically related in a pair. Anyone may know the public key, but the private is kept secret.

- Advantages - key distribution is vastly simplified.
- Disadvantage - generally slower, computationally more expensive. Larger keys for same cryptographic strength.

1.2 RSA

The key premise of RSA public key cryptography is based on the following key mathematical result

Central formula

$m^{ed} \equiv m \ (mod \ n)$ __________(1)

m - by convention we denote m as the message

To set this up, we need to

e - chose a public exponent, called e - typically a fixed know value
d - private value found such that ed satisfy criteria explained below
n - chose a modulus made from the product of two large primes, p and q, called n

Both n and e are components of the public key, which are published and shared.
A sender wishing to send a secret message m to a recipient encrypts it for that recipient using their (e, n) public key.
The recipient has the corresponding private key (d, n) to decrypt and recover m.

This may or may not seem simple enough, to help understand how this works and is used this document will run through the relevant parts and supporting mathematics.

Given all this, then we use it like this

To encrypt

$c = m^e \ mod \ n$

where c is the encrypted cypher text and n is the public modulus, e the public exponent.
The cipher text c can now be transmitted safely in private to the recipient over a public net-

work. Only the intended recipient with the corresponding private key d can decrypt c back to m.

To decrypt

$m = c^d = (m^e)^d = m^{ed}\ (mod\ n) = m$ (the original message) by (1)

Summary

we chose n = pq where p and q are very large primes
and e is selected as a fixed public 'exponent'

Public Key - (e, n)

Used for encrypting data (and verifying digital signatures).
Shared publicly, used to encrypt data only intended for the recipient.
comprised of the modulus (n) and the public exponent (e) as (e, n).

n is the product of chosen large primes p and q
and e is a preselected exponent such as 65537 (a common value)

Private key - (d, n)

Used for decrypting data intended for the recipient.
Kept secret, not shared publicly, used to decrypt data only intended for the recipient.
comprised of the modulus (n) and the private value (d) as (d, n).

Details of e and d

Details mater, for this to work, e and d are selected such that

$ed = k\phi(n) + 1$ - where k is insignificant here

i.e.

$ed \equiv 1\ mod\ \phi(n)$

See section 2.6 for more detail of how this actually works in more detail

2 Mathematical building blocks

2.1 prime numbers

Most likely you know all about prime numbers, but as this is a beginners guide, let's briefly recap.

Prime numbers are the numbers that have only 2 factors, 1 and themselves. In other words they are **only** divisible by 1 and themselves (note - as are all numbers, but for primes these are the only factors) - therefore they contain no other factors, thus they are further indivisible. To really qualify this we should note we're speaking of positive integers here, not real numbers and so on.

2.2 co-primes and gcd

The Greatest Common Divisor (GCD), of two or more non-zero integers, is the largest positive integer that divides both of the integers. GCD is also known as also known as Greatest Common Factor (GCF) or Highest Common Factor (HCF)

We write g = gcd(a, b) where gcd is the greater common divisor number

Two integers are coprime (also called relatively prime or mutually prime) if their greatest common divisor (GCD) is 1.

In other words, they share no common factors other than 1.

2.3 Fermat's little theorem

$\boldsymbol{a^p \equiv p \; mod \; n}$

where p is prime and a is not divisible by p
therefore a and p are coprime i.e. gcd(a, p) = 1

a useful alternative formulation of this is

$a^{p-1} \equiv 1 \; mod \; n$

In either form, this is an important result, in general and for public key cryptography, such as in RSA.

2.4 Euler's Totient function

Euler totient function for n counts the number of coprimes from 1 to n

given some n has a prime factorization

$n = p_0^{e_0} \; p_1^{e_1} \; ... \; p_k^{e_k}$

The totient can be found using

$\phi(n) = (p_0 - 1)p^{e_0 - 1} \; (p_1 - 1)p^{e_1 - 1} \; (p_k - 1)p^{e_k - 1}$

when n is prime, p say

$\phi(p) = (p-1)p^0 = p-1$

Relevant to RSA, when n is a product of 2 primes, p and q, we have

$\phi(n) = \phi(p)\phi(q) = (p-1)(q-1)$ (which we note, is trivial to compute given we know the only prime factors of n, p and q)

This fact is taken advantage of to make a large number n with an easily known $\phi(n)$ incidentally, the cototient of n is defined as $n - \phi(n)$ i.e. the number of factors of n

2.5 Euler's generalization of Fermat's little theorem

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

note here we're extending Fermat's little theorem to any $\phi(n)$ of the modulus n

Furthermore whilst it would be hard to calculate $\phi(n)$ for large arbitrary n, because we created it from two large (secret) primes, we can calculate it trivially.

2.6 RSA mathematics using these building blocks

Recall we chose ed with special conditions

$ed = \phi(n) + 1$

or more generally

$ed = k\phi(n) + 1$ - where k is insignificant here

i.e.

$ed \equiv 1 \ mod \ \phi(n)$

This works because

$m^{k\phi(n)} = m^{\phi(n)}.m^{\phi(n)}...m^{\phi(n)} = 1 \ (mod \ n)$

$m^{ed} = m^{k\phi(n)+1} = m.m^{k\phi(n)} = 1.m \ (mod \ n) = m \ (mod \ n)$