RSA for beginners - version 0.2

Louis Botterill

2025-04-04

Updates

V0.2 — LJB — 2025-09-01 — Further updates
V0.1 — LJB — 2025-04-04 — Initial version

1 RSA public key cryptography

1.1 Introduction

This document serves as an introduction to RSA cryptography and the essential underlying mathematics that it is based on. It is intended to be relatively concise yet self-contained, assuming the reader has a basic understanding of certain fundamental mathematical concepts. The glossary and resource sections at the end of this document may provide useful further reading or help bridge any gaps in prerequisite knowledge.

All information presented in this document is publicly available and not confidential. The goal of this document is to bring together key concepts in a clear, easily accessible and digestible format, making it suitable for beginners or as a refresher for those already familiar with the topic.

RSA is a type of public key (asymmetric) cryptographic system. To understand what this means, we will first look at the general concepts of cryptography, then explore how RSA works, including in particular, the mathematical principles behind it.

Cryptographic systems are generally divided into two main categories: symmetric cryptography and asymmetric cryptography (also known as public key cryptography).

Symmetric - uses one private key. Encryption and decryption are inverses, using the same privately shared key for both.

- Advantages - speed and efficiency. Smaller keys for same cryptographic strength.
- Disadvantages - difficulty in distributing the private keys, without risk of compromising them.

Asymmetric - there are two keys, a public and a private key, mathematically related as a pair. Anyone may obtain the public key, but the private must be kept secret.

- Advantages - key distribution is vastly simplified. Private keys do not need to be distributed to both parties in secret.
- Disadvantage - generally slower, computationally more expensive. Larger keys for same cryptographic strength.

For the purposes of describing the process of sending secure messages from a to b, we will introduce the standard parlance, Alice and Bob. Alice (a) is the sender and Bob (b) is the recipient. Since we are discussing RSA here, which is a public asymmetric system, Bob will have a related key pair, a private and public key. Only Bob will know his private key, but his corresponding public key is shared. Alice will know of Bob's public key and use it to send secure messages to Bob (that only Bob can decrypt). Because the private and public keys are related they form a secure encrypt-decrypt process and are generated together at the same time from a mathematical process using common parameters. How exactly this is done will be worked through in the following sections of this document.

This document will help take you through the necessary definitions, background mathematics and how it is used in RSA asymmetric public key cryptography. We will see that the building blocks are not very complex. Further reading may be necessary but the aim is to have this relatively self contained in a single document intended for beginners, hopefully without being either too high or low-level.

1.2 RSA

RSA is a popular cryptographic system, it stands for the authors Rivest-Shamir-Adleman who publicly described the algorithm in 1977. It is a is a widely used public key (asymmetric) encryption algorithm.

RSA uses a pair of keys – a public key for encryption and a private key for decryption and is based on modular arithmetic. The scheme was originally invented by Clifford Cocks while working at GCHQ in Great Britain, back in 1973. This project remained secret until 1997 and hence RSA took hold in the public domain in the meantime.

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

The central formula

$m^{e.d} \equiv m \ (mod\ n)$ __________(1)

m - by convention we denote m as the message
e.d - the product of two integers, with a relation to Euler totient function which will be explained later

To set this up, we need to choose

e - a public exponent, typically a fixed known value - most commonly this is $2^{16} + 1 = 65537$
d - private value found such that $e.d$ satisfy particular criteria to be explained later
n - a modulus made from the product of two large unique primes, p and q (where p != q)

Both e and n are components of the public key, which are published and shared.
A sender Alice wishing to send a secret message m to a recipient Bob, 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 really works, mathematically (and how it is used) this document will run through the relevant parts and supporting mathematics from the ground up.

Given all this, we use it in RSA, as follows

To encrypt

To Encrypt a plain text message m for Bob, Alice requires Bob's unique public key (e, n) and uses it to compute the corresponding cipher-text as follows

$c = m^e \ mod\ n$

where c is the encrypted ciphertext, m is the plaintext message, n is the public modulus, and e the public exponent.

The ciphertext c can now be transmitted safely in private from Alice to the recipient Bob over a public network. Only the intended recipient Bob, with the corresponding private key d can decrypt c back to m. Anyone else obtaining the ciphertext should not be able to decrypt it (easily, in theory).

To decrypt

To decrypt the ciphertext c of message m for Bob from Alice, this requires Bob's (matching) private key (d, n) which of course only Bob should know. Then Bob performs the following:

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

Summary

We choose the public modulus n = p.q where p and q are very large non-equal primes and e is selected as a fixed publicly shared '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 non-equal 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

The exact details do mater for this to work, e and d are carefully selected such that

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

i.e.

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

See section 2.7 for further explanation of how this actually works in more precise detail.

2 Mathematical building blocks

The following subsections break down some of the relevant key mathematical concepts and building blocks.

2.1 Prime numbers and prime factorization

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

Prime numbers are the natural 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 natural numbers, the positive integers here, not real numbers and so on.

Prime factoriztion:

The fundamental theorem of arithmetic (also known as the unique factorization theorem), states that every integer greater than 1 can be uniquely represented as a product of prime numbers. For some $n > 1$, where can express n as follows

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

where the prime factorization is unique. That is to say every integer greater than 1 has a unique prime factorization. Thus all integers have a unique representation as a decomposition of primes.

2.2 Modular arithmetic and congruence

This is a system of arithmetic for integers, where numbers "wrap around" at a certain value, called the modulus.

Example, for a modulus n say 3, the numbers modulo n (called mod n) are [0, 1, 2] (and no other numbers mod n are allowed)

This can be thought of as the remainder $< n$ after dividing out the original number by n

When two numbers have the same remainder modulo a particular modulus, we say they are congruent.

Example, 5 and 8 are both congruent to 2 mod 3

This would be written in mathematical notation as

$5 \equiv 8 \equiv 2\ mod\ 3$

where $\equiv$ is the symbol for congruence in modular arithmetic. Meaning having the same remainder, after division by the modulus.

e.g.

$5 - 3 = 2$ and $8 - 2 \times 3 = 8 - 6 = 2$, thus both 5 and 8 are congruent to 2 mod 3

2.3 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 greatest common divisor function (a binary function of two integers)

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.

2.4 Fermat's little theorem (FLT)

This is an important theorem in general and it's generalization is vitally important for RSA public key cryptography.

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

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 mathematical result, in general and for public key cryptography, such as in RSA.

2.5 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, say

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

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 (unique) 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 which are p and q)

Note here this works because the totient function is a multiplicative function

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.

2.6 Euler's generalization of Fermat's Little Theorem

This is an important theorem in general and an important part of RSA public key cryptography.

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

note here we're extending Fermat's little theorem to $\phi(n)$ 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 (known but secret) primes, we can calculate it trivially, as explained earlier (see Euler totient from primes factors).

2.7 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 relatively insignificant here)

i.e.

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

This works because

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

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

Another important aspect of this is that $\phi(n)$ is generally hard to calculate for (large) n (unless you know the prime factors of n). This is why n is calculated as the product of 2 large unique prime numbers.

If we did know, or were able to easily calculate (or guess), $\phi(n)$ then we could trivially compute d the private key and the whole cryptographic system would be compromised.

$d = (\phi(n) + 1)/e$ where everything on the right hand side is publicly available, so d would be easily obtained and compromised purely from publicly available data.

To ensure this is not the case, n is calculated from the product of 2 large primes, p and q.

This means if you know the primes, it is easy to calculate $\phi(n) = (p-1)(q-1)$, however if you don't (and these should not be shared) then it is very hard to find them (factorizing for large n and/or find $\phi(n)$ by any other means, is difficult.

Thus this is a vitally important part of making RSA public key cryptography secure against compromises.

Another important consideration is the authenticity of Bob's public key. It must be trustworthy and genuine for the system to work. Typically this is achieved by obtaining the public keys from a trusted published source.To prove authenticity other cryptographic mechanisms are employed, such as signatures and certificates.

Typically this is called PKI (public key infrastructure). Namely, some trusted certification authority issues a (digital) certificate confirming that the public key belongs to a particular person. This authority is verified and trustworthy.

Alternative mechanisms exist, for example the key may be shared directly in person. Or by other secure channels, even by secure symmetric cryptographic channels.
If we consider the potential thread if the public key is not genuine, we can consider Eve, e a malicious eavesdropper who has spoofed a public key for Bob. Alice believes this to be Bob's public key and uses it to encrypt a secure message for Bob. Alas if Bob ever receives, he finds he cannot decrypt it, becasue it was made using a fake public key. The author of that fake key, Eve however, finds it trivial to decrypt the message intended for Bob and thus the system is totally compromised.

The reader may with to follow up on this aspect with further reading. We won't get into that here but there might be a follow up document on this topic in the future.

3 Summary

In this short document we have seen how the popular public key (asymmetric) cryptographic system known as RSA can be used and how it works, using relatively simple mathematical building blocks and concepts.

The key building blocks of modular arithmetic, prime numbers, Fermat's Little Theorem (FLT) and Euler's Generalization of FLT have been introduced along the way to provide an understanding of the mathematical mechanisms by which this system works.

Whilst there is more to know in this area generally, such as the use of PKI, hopefully this document has illustrated how RSA works, mathematically (and practically) from the ground up.

Notes on cryptographic strength.

Whilst RSA is no longer seen as the height of security, RSA-2048 is still considered secure enough for general use. However, its lifespan is limited to around 2030 according to the National Institute of Standards and Technology (NIST).
Increased security can of course be found using larger key sizes, such as RSA-3072 or RSA-4096.

Better cryptographic systems such as Lattice-based and cryptographic hash algorithms may currently be the best options as improvements over RSA, as they are both resistant to classical and quantum methods. The latter being an important vital aspect to consider for future proofing.

The primary concern for the future is not classical computing but the development of quantum computers, which could break RSA encryption more easily.

Any comments, issues, mistakes, anomalies or omissions in this document, feel free to let the author know for correction - since it is hosted in the public domain in GitHub, a GitHub Issue is suitable for providing the feedback.

Source location of the document: `https://github.com/LouisJB/mathsy/blob/main/crypto/rsa/LaTeX/rsa.pdf` - please check here the latest revisions.

Further reading

A Method for Obtaining Digital Signatures and Public-Key Cryptosystems - R.L. Rivest, A. Shamir, and L. Adleman

https://people.csail.mit.edu/rivest/Rsapaper.pdf

Wikipedia on RSA

https://en.wikipedia.org/wiki/RSA_cryptosystem

GCHQ

https://en.wikipedia.org/wiki/GCHQ

ITU

ITU Introduction to cybersecurity

Stanford

https://crypto.stanford.edu/~dabo/cryptobook/draft_0_3.pdf

NIST Special Publication 800-175B

https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-175Br1.pdf

Glossary

Coprime - two numbers sharing no common factors other than 1. Two numbers are considered coprime if their greatest common divisor (GCD) or highest common factor (HCF) is 1

Chinese hypothesis - The hypothesis that an integer n is prime iff it satisfies the condition that $2^n - 2$ is divisible by n

FTL - Fermat's Little theorem. This is a generalization of the Chinese hypothesis and a special case of Euler's totient theorem.

GCD - greatest common divisor. The highest divisor common to two numbers

GCHQ - Government Communications Headquarters (GCHQ)

HCF - see GCD

Iif - short-form of if and only if

Integer - whole numbers, e.g. -1, 0, 1, 2, 3 .. n

Integer factorization - decomposition of a positive integer into a product of integers

Modular arithmetic - wrap-around integers from 0 to $n - 1$

Mutually prime - see coprime

Natural numbers - positive integers - e.g. 1, 2, 3, 4, .. n

PKI - public key infrastructure. The system and services used to share trusted, authenticated public keys.

Prime factorization - reduction to the unique factors of any number

Prime number - numbers with no divisors (other than 1 and themselves)

Relatively prime - see coprime

Totient - for n, it is the number of coprimes from 1 to n

Recommended further resources

https://mathworld.wolfram.com/RSAEncryption.html

https://mathworld.wolfram.com/GreatestCommonDivisor.html

https://mathworld.wolfram.com/TrapdoorOne-WayFunction.html