

The Math and the key-concept of the "RSA Public-Key Cryptography"

By Urban Avierfjård

April 13, 2013

Abstract

The Math and the key-concept of The "RSA Public-Key Cryptography". Much of the text and examples Is copied from websites on the internet and compiled to a tutorial. The document was written in LaTeX with the texmaker TeXstudio for MS Windows.

1 The RSA Algorithm

1.1 The Key-Concept

The Key-Concept of "RSA Public-Key Cryptography" is "Fermat's little theorem" and using modular exponents of 1024-bit to 2048-bit prime numbers and by using techniques like "Exponentiation by squaring" for fastening up the calculations. First we need the

"Fundamental theorem of Arithmetic"

for the explanation of the RSA Algorithm.

Every integer n is decomposable into prime factors:

$$n = p_1^{\alpha_1} p_2^{\alpha_2} \dots p_k^{\alpha_k} \tag{1}$$

This decomposition is unique, except for a reordering of the factors.

then we have

"Euler's Theorem"

$$x^{\phi(N)} - 1 \equiv (\text{mod } N) \quad (2)$$

for any x prime to N .

Here $\phi(N)$ is

"Euler's function", "Euler's totient function" or "Euler's phi function"

It is an arithmetic function that counts the number of positive integers less than or equal to n that are relatively prime to n . That is, if n is a positive integer, then $\varphi(n)$ is the number of integers k in the range $1 \leq k \leq n$ for which $\text{gcd}(n, k) = 1$. The totient function is a multiplicative function, meaning that if two numbers m and n are relatively prime (to each other), then $\varphi(mn) = \varphi(m)\varphi(n)$. For example let $n = 9$. Then $\text{gcd}(9, 3) = \text{gcd}(9, 6) = 3$ and $\text{gcd}(9, 9) = 9$. The other six numbers in the range $1 \leq k \leq 9$, that is, 1, 2, 4, 5, 7 and 8, are relatively prime to 9. Therefore, $\varphi(9) = 6$. As another example, $\varphi(1) = 1$ since $\text{gcd}(1, 1) = 1$.

$$\phi(q) = \prod_{k=1}^{\infty} (1 - q^k) \quad (3)$$

then

"Fermat's little Theorem"

If p is a prime number and $p \nmid a$ the following statement is true:

$$a^{p-1} - 1 \equiv 0(\text{mod } p) \quad (4)$$

And the special case the

"RSA's Fundamental theorem"

$$x^{(p-1)(q-1)} - 1 \equiv 0(\text{mod } pq) \quad (5)$$

for any prime numbers p and q and any integer x having no common divisors with pq .

1.2 Example No.1

Choose two distinct prime numbers p and q . For security purposes, the integers p and q should be chosen at random, and should be of similar bit-length. Compute $n = pq$.

n is used as the modulus for both the public and private keys. Its length, usually expressed in bits, is the key length.

Compute $\varphi(n) = (p - 1)(q - 1)$, where $\varphi(n)$ is Euler's totient function.

Choose an integer e such that $1 < e < \varphi(n)$ and $\gcd(e, \varphi(n)) = 1$; i.e., e and $\varphi(n)$ are coprime. e is released as the public key exponent. e having a short bit-length and small Hamming weight results in more efficient encryption ? most commonly $216 + 1 = 65,537$. However, much smaller values of e (such as 3) have been shown to be less secure in some settings.

Determine d as $d \equiv e^{-1}(\text{mod } \varphi(n))$, i.e., d is the multiplicative inverse of $e(\text{mod } \varphi(n))$. This is more clearly stated as solve for d given $de \equiv 1(\text{mod } \varphi(n))$. This is often computed using the extended Euclidean algorithm. d is kept as the private key exponent. By construction, $de \equiv 1(\text{mod } \varphi(n))$. The public key consists of the modulus n and the public (or encryption) exponent e . The private key consists of the modulus n and the private (or decryption) exponent d , which must be kept secret. p, q , and $\varphi(n)$ must also be kept secret because they can be used to calculate d . An alternative, used by PKCS 1, is to choose d matching $de \equiv 1(\text{mod } \lambda)$ with $\lambda = \text{lcm}(p - 1, q - 1)$, where lcm is the least common multiple. Using λ instead of $\varphi(n)$ allows more choices for d . λ can also be defined using the Carmichael function, $\lambda(n)$. The ANSI X9.31 standard prescribes, IEEE 1363 describes, and PKCS 1 allows, that p and q match additional requirements: being strong primes, and being different enough that Fermat factorization fails.

1.2.1 Encryption

A transmits his public key (n, e) to B and keeps the private key secret. B then wishes to send message M to A. He first turns M into an integer m , such that $0 \leq m < n$ by using an agreed-upon reversible protocol known as a padding scheme. He then computes the ciphertext c corresponding to

$$c \equiv m^e(\text{mod } n) \tag{6}$$

This can be done quickly using the method of exponentiation by squaring. B then transmits c to A.

1.2.2 Decryption

A can recover m from c by using her private key exponent d via computing

$$m \equiv c^d(\text{mod } n) \tag{7}$$

Given m , she can recover the original message M by reversing the padding scheme. (In practice, there are more efficient methods of calculating c^d using the precomputed values below.)

1.3 Example No.2

Generate two large (e.g., 1024? 2048 bit) primes p and q .

Compute $n = pq$ and $m = (p - 1)(q - 1)$.

Choose an exponent e such that $1 < e < m$ and $e \perp m$.

Compute d such that $1 < d < m$ and $ed \equiv 1 \pmod{m}$.

The primes p and q should be roughly the same size but not so close that they can be found by trying integers near \sqrt{n} . Usually $q < p < 2q$.

There is no definite rule regarding how the encoding number e must be chosen (except that it must be relatively prime to m), but there are certain standard values that are commonly used because they speed up the calculations.

The decoding number d is computed using the extended Euclidean algorithm.

1.3.1 Public key encryption.

The public key is the pair (n, e) ; these numbers may be published and shared with anyone.

A plaintext message consists of a number $P, 0 < P < n$.

Encode P by

$$C = P^e \pmod{n} \tag{8}$$

This can be done efficiently using the fast modular exponentiation algorithm (the square-and-multiply algorithm).

1.3.2 Private key decryption.

Your private key is the number d , and it must be kept secret. The primes p and q must also be kept secret since anyone who knows p, q , and e can compute d . Although it is less obvious, it is possible to determine p and q from m and n , so m must also be kept secret. Decode ciphertext C by

$$P = C^d \pmod{n} \tag{9}$$

Take $p = 17, q = 19$, and $e = 17$. Then $n = 323, m = 288$ and $d = 17$.

To encrypt $P = 47$, using the the above formula (8)

$$C = 47^{17} \pmod{323} = 302 \tag{10}$$

Decode it by using the above formula (9)

$$D = 302^{17} \pmod{323} = 47 \tag{11}$$

1.3.3 Explanation

We must understand the Formula

$$P = P^{e^d} \pmod{n} \tag{12}$$

Recall that d was chosen so that $ed \equiv 1 \pmod{m}$.

Thus there exists an integer k such that

$de = 1 + km = 1 + k(p-1)(q-1)$. Therefore

$$P^{e^d} = P^{de} = P^{1+k(p-1)(q-1)} = PP^{k(p-1)(q-1)} \tag{13}$$

By Fermat's Little Theorem,

$$PP^{p-1} \equiv P \pmod{p} \tag{14}$$

Applying this result $k(q-1)$ times gives

$$PP^{k(p-1)(q-1)} \equiv P \pmod{p} \tag{15}$$

In a similar way we see that

$$PP^{k(p-1)(q-1)} \equiv P \pmod{q} \tag{16}$$

Thus $x = P$ and $x = P^{de}$ are both solutions to the system of congruences

$$x \equiv P \pmod{p} \tag{17}$$

$$x \equiv P \pmod{q} \tag{18}$$

2 Math

2.1 Modular Arithmetics

Let's use a clock as an example, except let's replace the 12 at the top of the clock with a 0.

Starting at noon, the hour hand points in order to the following:

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 0, 1, 2

This is the way in which we count in modulo 12. In modulo 5, we count

0, 1, 2, 3, 4, 0, 1, 2, 3, 4, 0, 1

We can also count backwards in modulo 5

.Any time we subtract 1 from 0, we get 4. So, the integers from -12 to 0, when written in modulo 5, are

4, 3, 2, 1, 0, 4, 3, 2, 1, 0, 1, 2

where -12 is the same as 3 in modulo 5.

Because all integers can be expressed as 0, 1, 2, 3, 4 or in modulo 5,

we give these integers their own name:

the residue classes modulo 5. In general, for a natural number n that is greater than 1, the modulo n residues are the integers that are whole numbers less than n :

0, 1, 2, , , , , $n - 1$

2.1.1 Congruence

There is a mathematical way of saying that all of the integers are the same as one of the modulo 5 residues.

For instance, we say that 7 and 2 are congruent modulo 5.

We write this using the symbol \equiv :

In other words, this means in base 5, these integers have the same last digit:

$2(\text{base}5) \equiv 12(\text{base}5) \equiv 22(\text{base}5) \equiv 32(\text{base}5) \equiv 42(\text{base}5) \equiv 7 \equiv 2(\text{mod } 5)$

The (mod 5) part just tells us that we are working with the integers modulo 5.

In modulo 5, two integers are congruent when their difference is a multiple of 5.

Thus each of the following integers is congruent modulo 5:

$-12 \equiv -7 \equiv -2 \equiv 3 \equiv 8 \equiv 13 \equiv 18 \equiv 23(\text{mod } 5)$

2.1.2 Rules

Consider four integers a, b, c, d and a positive integer m such that $a \equiv b(\text{mod } m)$ and $b \equiv d(\text{mod } m)$.

2.1.3 Addition

$a + c \equiv b + d(\text{mod } m)$

2.1.4 Subtraction

$a - c \equiv b - d(\text{mod } m)$

2.1.5 Multiplication

$ac \equiv bd(\text{mod } m)$

2.1.6 Division

$\frac{a}{e} \equiv \frac{b}{e} \pmod{\frac{m}{\gcd(m,e)}}$ where e is a positive integer that divides a and b .

2.1.7 Exponentiation

$a^e \equiv b^e \pmod{m}$ where e is a positive integer.

2.2 Linear Congruence

2.2.1 Definition

A Linear Congruence is a congruence \pmod{p} of the form

$$ax + b \equiv c \pmod{p}$$

where a, b, c, p and are constants and x is the variable to be solved for

2.2.2 Solving

Note that not every linear congruence has a solution.

For instance, the congruence equation $2x \equiv 3 \pmod{8}$ has no solutions.

A solution is guaranteed iff a is relatively prime to p . If a and p are not relatively prime, let their greatest common divisor be d ; then:

if d divides b , there will be a solution $\pmod{\frac{p}{d}}$

if d does not divide b , there will be no solution

2.3 Modular Exponentiation

If we have the base b , exponent e and the modulus m , The modular exponentiation c is

$$c \equiv b^e \pmod{m} \tag{19}$$

For Example given, $b = 5$, $e = 3$ and $m = 13$ the solution c is the remainder of dividing 5^3 by 13, which is the remainder of $125 \mid 13$ or 8

2.4 Euclidean Algorithm

The Euclidean algorithm proceeds in a series of steps such that the output of each step is used as an input for the next one. Let k be an integer that counts the steps of the algorithm, starting with zero. Thus, the initial step corresponds to $k = 0$, the next step corresponds to $k = 1$, and so on.

Each step begins with two nonnegative remainders r_{k-1} and r_{k-2} . Since the algorithm ensures that the remainders decrease steadily with every step, r_{k-1} is less than its predecessor r_{k-2} . The goal of the k th step is to find a quotient q_k and remainder r_k such that the equation is satisfied

$$r_{k-2} = q_k r_{k-1} + r_k \tag{20}$$

where $r_k < r_{k-1}$. In other words, multiples of the smaller number r_{k-1} are subtracted from the larger number r_{k-2} until the remainder is smaller than the r_{k-1} .

In the initial step ($k = 0$), the remainders r_{-2} and r_{-1} equal a and b , the numbers for which the GCD is sought. In the next step ($k = 1$), the remainders equal b and the remainder r_0 of the initial step, and so on. Thus, the algorithm can be written as a sequence of equations

$$a = q_0b + r_0 \tag{21}$$

$$b = q_1r_0 + r_1 \tag{22}$$

$$r_0 = q_2r_1 + r_2 \tag{23}$$

$$r_1 = q_3r_2 + r_3 \tag{24}$$

If a is smaller than b , the first step of the algorithm swaps the numbers. For example, if $a < b$, the initial quotient q_0 equals zero, and the remainder r_0 is a . Thus, r_k is smaller than its predecessor r_{k-1} for all $k \geq 0$.

Since the remainders decrease with every step but can never be negative, a remainder r_N must eventually equal zero, at which point the algorithm stops. The final nonzero remainder r_{N-1} is the greatest common divisor of a and b . The number N cannot be infinite because there are only a finite number of nonnegative integers between the initial remainder r_0 and zero.

2.4.1 An Example

For illustration, the Euclidean algorithm can be used to find the greatest common divisor of $a = 1071$ and $b = 462$. To begin, multiples of 462 are subtracted from 1071 until the remainder is less than 462. Two such multiples can be subtracted ($q_0 = 2$), leaving a remainder of 147 $1071 = 2 \times 462 + 147$. Then multiples of 147 are subtracted from 462 until the remainder is less than 147. Three multiples can be subtracted ($q_1 = 3$), leaving a remainder of 21 $462 = 3 \times 147 + 21$. Then multiples of 21 are subtracted from 147 until the remainder is less than 21. Seven multiples can be subtracted ($q_2 = 7$), leaving no remainder $147 = 7 \times 21 + 0$. Since the last remainder is zero, the algorithm ends with 21 as the greatest common divisor of 1071 and 462. This agrees with the GCD (1071, 462) found by prime factorization above.

Step k	Equation	Quotient and remainder
0	$1071 = q_0 462 + r_0$	$q_0 = 2$ and $r_0 = 147$
1	$462 = q_1 147 + r_1$	$q_1 = 3$ and $r_1 = 21$
2	$147 = q_2 21 + r_2$	$q_2 = 7$ and $r_2 = 0$; the algorithm ends

2.5 The Chinese Remainder Theorem

Suppose n_1, n_2, \dots, n_k are positive integers which are pairwise coprime. For any given sequence of integers a_1, a_2, \dots, a_k there exists an integer x solving the following system of simultaneous congruences.

$$x \equiv a_1 \pmod{n_1} \quad (25)$$

$$x \equiv a_2 \pmod{n_2} \quad (26)$$

$$x \equiv a_k \pmod{n_k} \quad (27)$$

Furthermore, all solutions x of this system are congruent modulo, the product, $N = n_1 n_2 \dots n_k$.

Hence

$$x \equiv y \pmod{n_i} \quad (28)$$

for all $1 \leq i \leq k$, if and only if

$$x \equiv y \pmod{N} \quad (29)$$

. Sometimes, the simultaneous congruences can be solved even if the n_i 's are not pairwise coprime. A solution x exists if and only if:

$$a_i \equiv a_j \pmod{\gcd(n_i, n_j)} \text{ for all } i \text{ and } j \quad (30)$$

All solutions x are then congruent modulo the least common multiple of the n_i .

2.6 Exponentiation by squaring

The method is based on the observation that, for a positive integer n , we have

$$x^n = x(x^2)^{\frac{n-1}{2}} \text{ if } n \text{ is Odd} \quad (31)$$

$$x^n = (x^2)^{\frac{n}{2}} \text{ if } n \text{ is Even} \quad (32)$$

A brief analysis shows that such an algorithm uses $O_{\log 2n}$ squarings and $O_{\log 2n}$ multiplications. For $n >$ about 4 this is computationally more efficient than naively multiplying the base with itself repeatedly.

2.7 Coprime Numbers

Two integers a and b are said to be coprime (also spelled co-prime), relatively prime or mutually prime if the only positive integer that evenly divides both of them is 1.

This is equivalent to their greatest common divisor being 1. In addition to $\gcd(a, b) = 1$ and $(a, b) = 1$ the notation $a \perp b$ is sometimes used to indicate that a and b are relatively prime.