

PUBLIC-KEY CRYPTOGRAPHY

1. INTRODUCTION TO CRYPTOGRAPHY

Prime numbers, the history of math, inverse functions, technology, and a contemporary application of mathematics all come together in public-key cryptography. Cryptography (sometimes cryptology) is the study of methods of writing a message that hides its meaning from everyone except the message's intended recipient. For instance, we could take the message MATH RULES and write it backwards as SELUR HTAM.

Encryption is a method of changing *plaintext*, the message to be hidden, to *ciphertext*, the message in its hidden form. For the message given above, we have

Plaintext	Encryption rule	Ciphertext
MATH RULES	write the plaintext backwards	SELUR HTAM

Decryption is the procedure that changes ciphertext back to plaintext. For the simple example above, the decryption rule is "write the ciphertext backwards." The pair of an encryption rule and decryption rule is called a *cipher*.

Encryption can be viewed as a function that maps a plaintext message to ciphertext. Decryption is the inverse of the encryption function. To illustrate how this might work, suppose we assign each letter of the alphabet and the space between words to a number. One possible scheme is shown below.

A	B	C	...	Y	Z	space
10	11	12	...	34	35	36

The plaintext message MATH RULES would be represented as 22102917362730211428. Usually this message is broken into blocks. For instance, we could choose blocks of 4 digits as shown below.

2210 2917 3627 3021 1428

Now choose a function that has an inverse, say $f(x) = 2x + 1$, where $f^{-1}(x) = \frac{1}{2}x - \frac{1}{2}$. To encrypt the message, evaluate f at each block. (Note that the function *write the message backwards*, used above, is its own inverse.)

$$\begin{aligned}f(x) &= 2x + 1 \\f(2210) &= 2(2210) + 1 = 4421 \\f(2917) &= 2(2917) + 1 = 5835 \\f(3627) &= 2(3627) + 1 = 7255 \\f(3021) &= 2(3021) + 1 = 6043 \\f(1428) &= 2(1428) + 1 = 2857\end{aligned}$$

The message would be sent as

4421 5835 7255 6043 2857

The person receiving the message evaluates f^{-1} at each received block.

$$f^{-1}(x) = \frac{1}{2}x - \frac{1}{2}$$

$$f^{-1}(4421) = \frac{1}{2}(4421) - \frac{1}{2} = 2210$$

$$f^{-1}(5835) = \frac{1}{2}(5835) - \frac{1}{2} = 2917$$

$$f^{-1}(7255) = \frac{1}{2}(7255) - \frac{1}{2} = 3627$$

$$f^{-1}(6043) = \frac{1}{2}(6043) - \frac{1}{2} = 3021$$

$$f^{-1}(2857) = \frac{1}{2}(2857) - \frac{1}{2} = 1428$$

Once blocks are evaluated, the recipient can use the correspondence between number and letter to read the message.

The problem with the above method is that the message recipient must know the encryption function so that the inverse function can be computed. But how is that encryption function made known to the recipient? It cannot be broadcast on an open communications channel because someone could retrieve the function, determine its inverse, and thus decrypt the message.

One way to solve this problem would be to have an encryption function whose inverse is so difficult to find that making the encryption function public would not compromise the security of a message encrypted by that function. Enter public-key cryptography, which uses a public encryption function whose inverse, the decryption function, is practically impossible to find without knowing very specifically how the encryption function was created.

The basic idea is as follows. Use a clever scheme to create f and f^{-1} . Keep f^{-1} your secret function. Publish f in a public database of encryption functions. If Olivia wants to send a message to Henry, Olivia would use the public function Henry entered into the database. She would encrypt the message using that function and send it to Henry on an open communication channel which could be intercepted by anyone. However, because Henry is the only person with the inverse function, Henry is the only one who can decrypt the message.

To see how this can be done, we must borrow from the ancients, use prime numbers, and properties of modular functions. We begin with an ancient Egyptian multiplication algorithm.

2. EGYPTIAN MULTIPLICATION

The Rhind Papyrus, discovered in 1858, gave a rich account of early Egyptian mathematics. In that document, the procedure to multiply two numbers by successively doubling one of the numbers was given. For instance, to find the product $23 \cdot 26$, a scribe would proceed as shown below.

TABLE 1. Egyptian Multiplication

1	26
2	52
4	104
8	208
16	416

The scribe would stop doubling at this point because $32 > 23$, the multiplier. Now the scribe would notice that $23 = 1 + 2 + 4 + 16$ and place a mark next to those numbers.

TABLE 2. Egyptian Multiplication

→ 1	26
→ 2	52
→ 4	104
	8 208
→ 16	416

The product $23 \cdot 26 = (1 + 2 + 4 + 16)26 = 26 + 52 + 104 + 416 = 598$.

Another way to view the product $23 \cdot 26$ is to use a binary representation of 23.

$$23_{10} = 10111_2$$

Using the last expression, $23 \cdot 26$ can be written as

$$(1 \cdot 2^0 + 1 \cdot 2^1 + 1 \cdot 2^2 + 0 \cdot 2^3 + 1 \cdot 2^4) \cdot 26 = 598$$

where we have written the binary expansion of 23 in ascending powers of 2.

This suggests that Egyptian multiplication is a subtle use of base 2 arithmetic. It is not clear how the Egyptian's came to this algorithm but it has persisted for centuries. A similar multiplication algorithm, which has base 2 roots, is called *Russian Peasant Multiplication* or *Ethiopian Peasant Multiplication*.

Egyptian multiplication can be performed by using a graphing calculator program. The idea is to represent one number, say 23 in the above calculation, as a list of the coefficients of the binary representation of that number. Thus $L_1 = \{1, 1, 1, 0, 1\}$ where, again, we have listed the coefficients from low order to high order. In a second list, we place the results of doubling the multiplicand. For our example, $L_2 = \{26, 52, 104, 208, 416\}$. Now $L_1 \cdot L_2 = \{26, 52, 0, 208, 416\}$. Adding the elements in the product of the two lists gives 598, $\text{sum}(L_1 \cdot L_2) = 598$.[†]

[†]Another way to look at Egyptian multiplication is that the algorithm is the inner product of two vectors: $\vec{v} = (1, 1, 1, 0, 1)$, $\vec{w} = (26, 52, 104, 208, 416)$. Then $\vec{v} \cdot \vec{w} = 1 \cdot 26 + 1 \cdot 52 + 0 \cdot 104 + 1 \cdot 208 + 1 \cdot 416 = 598$.

An extension of the Egyptian multiplication algorithm can be used as a method of computing powers. Variations of this method are incorporated into some computers to evaluate exponential expressions. We will illustrate the method for the product 26^{23} . If we were to proceed as in the following,

$$\begin{aligned} 26 \cdot 26 &= 676 \\ 676 \cdot 26 &= 17,576 \\ 17,576 \cdot 26 &= 456,976 \\ &\dots \end{aligned}$$

22 multiplications would be required. However, we can use a variation of Egyptian multiplication that squares successive numbers instead of doubling them. We will call it *Egyptian Exponentiation* and it is shown in the following table.

TABLE 3. Egyptian Exponentiation

→	1	26
→	2	26^2
→	4	26^4
	8	26^8
→	16	26^{16}

Using this method, 26^{23} requires only 7 multiplications: $26^2 = 26 \cdot 26$, $26^4 = 26^2 \cdot 26^2$, $26^8 = 26^4 \cdot 26^4$, $26^{16} = 26^8 \cdot 26^8$, and the three products $26 \cdot 26^2 \cdot 26^4 \cdot 26^{16}$. This is an important savings in public-key cryptography because it is necessary to compute large powers of a number. In terms of the lists we used for Egyptian multiplication, 26^{23} equals the product of the *nonzero* elements of L_1 , the binary representation of the exponent, and L_2 , the successive squares of the base.

$$\begin{aligned} L_1 \cdot L_2 &= \{1, 1, 1, 0, 1\} \cdot \{26, 26^2, 26^4, 26^8, 26^{16}\} \\ &= \{26, 26^2, 26^4, 0, 26^{16}\} \\ &= 26 \cdot 26^2 \cdot 26^4 \cdot 26^{16} \quad \text{The product of the } \textit{nonzero} \text{ elements} \end{aligned}$$

3. MODULAR ARITHMETIC

We now explore the role of modular functions in public-key cryptography. The modular function $a = b \bmod n$ returns the remainder a when b is divided by n . Here are a few examples.

$$\begin{aligned} 3 &= 8 \bmod 5 \\ 15 &= 15 \bmod 27 \\ 0 &= 14 \bmod 7 \end{aligned}$$

By the division algorithm, $\frac{a}{b}$ implies $a = b \cdot q + r$, where $0 \leq b < r$. Therefore, $r = a \bmod b$. We can determine r as follows.

$$\begin{aligned} a &= b \cdot q + r \\ a - b \cdot q &= r \\ a - b \cdot \left\lfloor \frac{a}{b} \right\rfloor &= r \end{aligned}$$

where $\left\lfloor \frac{a}{b} \right\rfloor$ is the integer part of $\frac{a}{b}$. From this,

$$a \bmod b = a - b \cdot \left\lfloor \frac{a}{b} \right\rfloor$$

Here are three properties of modular functions.

$$\begin{array}{ll} \text{Addition Property} & (a \bmod n) + (b \bmod n) = (a + b) \bmod n \\ \text{Multiplication Property} & (a \bmod n)(b \bmod n) = (ab) \bmod n \\ \text{Exponentiation Property} & (a \bmod n)^p = a^p \bmod n \end{array}$$

The exponentiation property, which is really a result of the multiplication property, is particularly important in cryptography because it is necessary to compute $m^r \bmod n$ for large values of m and r . To see how these properties are used, we combine Egyptian Exponentiation and modular arithmetic. Here is an example of computing $233^{25} \bmod 537$.[†]

TABLE 4. $233^{25} \bmod 537$

→	1	$233 \bmod 537$	$= 233 \bmod 537$
	2	$233^2 \bmod 537$	$= 52 \bmod 537$
	4	$233^4 \bmod 537 = (52 \bmod 537)^2$	$= 19 \bmod 537$
→	8	$233^8 \bmod 537 = (19 \bmod 537)^2$	$= 361 \bmod 537$
→	16	$233^{16} \bmod 537 = (361 \bmod 537)^2$	$= 367 \bmod 537$

Because the exponent $25 = 1 + 8 + 16$, we have

$$\begin{aligned} 233^{25} \bmod 537 &= (233 \bmod 537) \cdot (361 \bmod 537) \cdot (367 \bmod 537) \\ &= (341 \bmod 537) \cdot (367 \bmod 537) \\ &= 26 \bmod 537 \end{aligned}$$

Note from the table above that we never had to square or multiply two numbers larger than the modulus 537. This is very important for computer implementations of cryptography because m^r would quickly create an overflow for large values of m and r .

[†]Texas Instruments 83/84 programs are available at <http://idisk.mac.com/galoisgroup-Public> that can be used to verify these calculations.

4. RSA CRYPTOGRAPHY

One form of public-key cryptography is known as RSA, from the first letters of the last names of Ron Rivest, Adi Shamir, and Leonard Adleman, who published the method in 1977 while professors at the Massachusetts Institute of Technology. A basic tenet of this form of cryptography is that given a large (over 200 digits) integer, it is extremely difficult to find the prime factorization of that integer. This fact is exploited in RSA public-key cryptography by creating a modular function for which the inverse modular function is extremely difficult to determine without knowing the prime factorization of the modulus.

To create a cipher in RSA cryptography, we begin by choosing two large, distinct prime numbers p and q [†] and forming their product $n = p \cdot q$. The value of n is the modulus of a modular function. Compute the *Euler Totient Function*, $\phi(n)$, which is the cardinality of $\{0 < x < n \mid \gcd(x, n) = 1\}$. When p and q are prime numbers, $\phi(pq) = (p-1)(q-1)$. Now randomly choose a number e , where $1 < e < \phi(n)$ and $\gcd(e, \phi(n)) = 1$. The value of e along with the modulus n are made public. Solve the equation $ed = 1 \pmod{\phi(n)}$ for d . This can be done by using the *Extended Euclidean Algorithm*.

To receive encrypted messages, Olivia posts n and e to a public-key encryption service, called a *Certificate Authority*, which guarantees the integrity of her public key. If Henry wants to send Olivia a message, M , he turns M into a number m ($m < n$) and then computes $c = m^e \pmod{n}$ and sends c to Olivia. Olivia can reconstruct the message by using the fact that $c^d \pmod{n} = m$.

Here is an example of encrypting and decrypting a message using RSA public-key cryptography. Begin by choosing two prime numbers that are known only to you. We will use $p = 83$ and $q = 89$. In practice, these numbers would be quite large, around 100 digits. Then $n = pq = 83 \cdot 89 = 7387$.

Compute $\phi(n) = (p-1)(q-1) = 82 \cdot 88 = 7216$. Randomly choose e so that $1 < e < \phi(n)$ and $\gcd(e, \phi(n)) = 1$. We will use $e = 23$. This is the public exponent, published for all to see along with n , the modulus. Then solve $de = 1 \pmod{\phi(n)}$ for d . Applying the Extended Euclidean Algorithm, $d = 1255$. This is the private exponent and must be kept secret from everyone.

For Henry to send the message MATH RULES to Olivia, he would convert the message to a number (We will use the same scheme as we used earlier.), $m = 22102917362730211428$. Now break the message into blocks of equal length keeping in mind that the number in each block must be less than n . We will use the same five blocks, m_i , as before, 2210 2917

[†]Frequently the primes p and q are chosen so that p_1 and q_1 are primes, where $p_1 = 2p+1$ and $q_1 = 2q+1$. The primes p and q are called Sophie St. Germain primes. The numbers 83 and 89 are St. Germain primes.

3727 3021 1428. Henry now computes $c_i = m_i^e \bmod n$ for $i = 1, 2, 3, 4, 5$.

$$2210^{23} \bmod 7387 = 6117$$

$$2917^{23} \bmod 7387 = 1088$$

$$3627^{23} \bmod 7387 = 6030$$

$$3021^{23} \bmod 7387 = 1874$$

$$1428^{23} \bmod 7387 = 5878$$

Olivia receives the message 6117 1088 3994 1874 5878 and uses her private exponent to decrypt the message. The modular function she uses, $c^{1255} \bmod 7387$, is the inverse of $m^{23} \bmod 7387$ used by Henry.

$$6117^{1255} \bmod 7387 = 2210$$

$$1088^{1255} \bmod 7387 = 2917$$

$$6030^{1255} \bmod 7387 = 3627$$

$$1874^{1255} \bmod 7387 = 3021$$

$$5878^{1255} \bmod 7387 = 1428$$

The decrypted message is 2210 2917 3627 3021 1428, or MATH RULES.

5. OTHER APPLICATIONS

Public-key cryptography is also used for *digital signatures*. A digital signature is an electronic signature that can be used to verify the identity of the sender of a message. For instance, if Olivia wants to send a message to Henry (and Henry wants to be assured it came from Olivia), she would encrypt her message using her private key. When Henry receives the message, if he can decode it using Olivia's public key, he knows that it came from Olivia. Of course, this simple example suffers from the fact that anyone, because the public key is known, could decrypt the message. One way to foil this would be for Olivia to encrypt her message using Henry's public key before sending it to Henry. Now when Henry receives the message, he can verify that it came from Olivia by using her public key and he can decode it using his private key.

Web browsers also use public-key cryptography to allow users to make secure purchases with a credit card over the Internet, which is quite public. A Transport Sockets Layer (TSL), the successor of Secure Sockets Layer (SSL), is a protocol that lies between, for instance, HTTP and TCP/IP. Although more involved, the TSL basically works as follows. When a purchaser (the client) contacts the company (the server) to make a purchase, the server sends its public key to the client. The TSL intercepts the public key and encrypts the credit card number and sends it to the server which decrypts the message with its private key. All of this is built into the technology of most Web browsers.

6. SECURITY OF PUBLIC-KEY CRYPTOGRAPHY

We have suggested that the security of public-key cryptography is based on the fact that it is extremely difficult to factor a large integer. To give a simple instance, according to the Prime Number Theorem, there are approximately $\frac{n}{\ln(n)}$ prime numbers less than n . Suppose we tried to find the prime factors of a number n with 100 digits using prime number trial factors up to \sqrt{n} . If a computer could test one trillion prime factors every second, it would take trillions and trillions of years to find the factors. Consequently, mathematicians have invested a lot of research into more efficient methods of finding prime factors. There are now some efficient algorithms to test for prime factors but these algorithms still require hundreds of years of computation to determine the factors of a number.

7. THE DETAILS

The example we gave earlier showed that Olivia could recover a message that was encrypted using her public key. To prove that the RSA procedure will always work, we need the help of the following three theorems.

Theorem 1. Euler's Theorem *If $\gcd(a, k) = 1$, then $a^{\phi(k)} \equiv 1 \pmod{k}$.*

Proof. Suppose $c_1, c_2, \dots, c_{\phi(k)}$ are the elements of $\{0 < c_i < k \mid \gcd(c_i, k) = 1\}$. Multiply each c_i by a where $\gcd(a, k) = 1$.

$$ac_1, ac_2, \dots, ac_{\phi(k)}$$

We claim that these numbers are, mod k , just a permutation of the original c_i . First, no $ac_i \equiv 0 \pmod{k}$. For if that were the case, then $ac_i \equiv 0 \pmod{k}$ implies k divides ac_i . But $\gcd(a, k) = 1$ and $\gcd(a, c_i) = 1$ so k cannot divide ac_i . Thus $ac_i \not\equiv 0 \pmod{k}$ for all i .

Now suppose $ac_i \equiv ac_j \pmod{k}$ but $c_i \not\equiv c_j \pmod{k}$. Then

$$ac_i - ac_j \equiv 0 \pmod{k} \Rightarrow k \mid a(c_i - c_j).$$

Because $\gcd(a, k) = 1$, $k \mid (c_i - c_j)$. However, this means that $c_i = c_j + mk$ for some integer m , or $c_i \equiv c_j \pmod{k}$, which contradicts that the c_i are distinct mod k . Because $c_1, c_2, \dots, c_{\phi(k)}$ and $ac_1, ac_2, \dots, ac_{\phi(k)}$ are permutations of the same numbers mod k

$$a^{\phi(k)}(c_1 \cdot c_2 \cdot \dots \cdot c_{\phi(k)}) \equiv (c_1 \cdot c_2 \cdot \dots \cdot c_{\phi(k)}) \pmod{k}$$

Because $\gcd(k, c_1 \cdot c_2 \cdot \dots \cdot c_{\phi(k)}) = 1$, both sides of the congruence can be divided by $c_1 \cdot c_2 \cdot \dots \cdot c_{\phi(k)}$ giving $a^{\phi(k)} \equiv 1 \pmod{k}$. \square

Theorem 2. Fermat's Little Theorem *If p is prime number and $\gcd(a, p) = 1$, then $a^{p-1} \equiv 1 \pmod{p}$.*

Proof. When p is prime, $\phi(p) = p-1$. Apply *Euler's Theorem* to establish the theorem. \square

Theorem 3. Chinese Remainder Theorem Suppose $x \equiv y \pmod{m}$ and $x \equiv y \pmod{n}$, where $\gcd(m, n) = 1$. Then $x \equiv y \pmod{mn}$.

Proof.

$$(1) \quad x \equiv y \pmod{m} \Rightarrow x = y + k_1m$$

$$(2) \quad x \equiv y \pmod{n} \Rightarrow x = y + k_2n$$

Subtract Equation (2) from Equation (1) to give $k_1m = k_2n$ or $k_1 = \frac{k_2n}{m}$. Because $\gcd(m, n) = 1$, $m|k_2 \Rightarrow k_2 = k_3m$. Substituting for k_2 in Equation (1), we have

$$x = y + k_3mn \Rightarrow x \equiv y \pmod{mn}$$

□

Theorem 4. RSA Cryptography Suppose p and q are prime numbers with $n = pq$. Choose e such that $1 < e < \phi(n)$, where $\phi(n) = (p-1)(q-1)$ and $\gcd(e, \phi(n)) = 1$. Let $c \equiv m^e \pmod{n}$, where m is a message and $0 \leq m < n$. Then there exists d such that $de \equiv 1 \pmod{\phi(n)}$ and $m \equiv c^d \pmod{n}$.

Proof. We first establish that we can find d such that $de \equiv 1 \pmod{\phi(n)}$. Let $k = \phi(n)$. Because e was chosen so that $\gcd(e, k) = 1$, we have, by Theorem 1, $e^{\phi(k)} \equiv 1 \pmod{k}$ or $e^{\phi(\phi(n))} \equiv 1 \pmod{\phi(n)}$. Let $d = e^{\phi(\phi(n))-1}$. Then

$$\begin{aligned} de \pmod{\phi(n)} &\equiv e^{\phi(\phi(n))-1} \cdot e \pmod{\phi(n)} \\ &\equiv e^{\phi(\phi(n))} \pmod{\phi(n)} \\ &\equiv 1 \pmod{\phi(n)} \end{aligned}$$

Now consider $c \equiv m^e \pmod{n}$. Then $c^d \equiv (m^e)^d \pmod{n} \equiv m^{de} \pmod{n}$. To complete the proof, we must show that $m^{de} \pmod{n} \equiv m$.

Because $de \equiv 1 \pmod{\phi(n)}$, $de = 1 + t \cdot \phi(n) = 1 + t \cdot (p-1)(q-1)$ for some integer t . Suppose, first, that $\gcd(m, p) = 1$. Then

$$\begin{aligned} m^{de} &= m^{1+t(p-1)(q-1)} \\ &= m \cdot m^{t(p-1)(q-1)} \\ (3) \quad &= m \cdot (m^{p-1})^{t(q-1)} \\ &\equiv m \cdot (1)^{t(q-1)} \pmod{p} \quad \text{By Fermat's Little Theorem} \\ &\equiv m \pmod{p} \end{aligned}$$

By symmetry, $m \equiv m^{de} \pmod{p}$. If $\gcd(m, p) \neq 1$, then both m and m^{de} are divisible by p so that $m \equiv 0 \pmod{p}$ and $m^{de} \equiv 0 \pmod{p}$ in which case $m \equiv m^{de} \pmod{p}$.

We can repeat the above but rewrite Equation (3) as $m \cdot (m^{q-1})^{t(p-1)}$ from which we obtain $m^{de} \equiv m \pmod{q}$. Thus

$$m \equiv m^{de} \pmod{p}$$

$$m \equiv m^{de} \pmod{q}$$

Applying the Chinese Remainder Theorem, we have $m \equiv m^{de} \pmod{pq}$ or, with $n = pq$, $m \equiv m^{de} \pmod{n}$. \square