

Cryptography

Helmer Aslaksen

Department of Mathematics

National University of Singapore

aslaksen@math.nus.edu.sg

www.math.nus.edu.sg/aslaksen/sfm/

Basic Concepts

There are many situations in life where you want to encode messages, so that even if they are seen, they will not be understood by other people than the intended recipient. The study of such encodings and decodings is called cryptography.

The message you want to hide is called the plaintext, and the act of encoding it is called encryption or enciphering. The encoded plaintext is called the ciphertext or the ciphertext, and the act of decoding it is called decryption or deciphering. The encryption system, also called a cipher, uses an encryption key, K_E , and a decryption key, K_D , for the encoding and decoding process.

Substitution Ciphers

One of the most common ciphers is substitution. This scheme is the one used in the Sherlock Holmes story “Mystery of the Dancing Men”. The idea is straightforward: choose a rearrangement of the letters of the alphabet, and replace each letter in the plaintext by its corresponding one.

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
T	H	E	Q	U	I	C	K	B	R	O	W	N	F	X	J	M	P	S	V	L	A	Z	Y	D	G

The sentence “please do not read this” then becomes

JWUTSU QX FXV PUTQ VKBS

More About Substitution Ciphers

Note that our alphabet contains a space, which corresponds to a space in our key. We didn't need to do this, but it left the message in words of the same length. This, of course, makes the message easier to guess. Also, notice that we ignored case, i.e, "A" and "a" are the same. We could have distinguished between upper and lower case, in which case we would need to use a 53 letter alphabet and a 53 letter key.

The usual way to break such a code is a combination of frequency analysis, that is, knowing that the most commonly occurring letter in the English language is "E", followed by "T", plus a bit of trial and error. Leaving the spaces between words helps the cracker greatly.

The Caesar Cipher

A variation on the arbitrary permutation discussed above is the Caesar cipher, named after Julius Caesar, who supposedly invented it himself. Here we convert our alphabet to numeric equivalents, say $A = 0$, $B = 1$, and so on, add an offset to each numeric equivalent (Caesar used 3), then re-encode the numbers as letters. For example, Caesar would replace A with D, B by E, and Z by C.

Public Key Cryptography

All of the ciphers so far discussed are symmetric encryption routines, also called secret key ciphers. If you know the key used to encode the message, you can decrypt the message without too much work.

This was true of all crypto systems up until the mid 1970's: knowledge of how to encode a message allowed one also to decipher it. However, in 1976, W. Diffie and M. Hellman invented public key cryptography. In a public key system, someone who knows how to encipher a message cannot determine how to decipher the message without a prohibitively large computation. It is important to realize that there may be a procedure for doing so, but to carry out this process would take a prohibitively long time, say, hundreds of years on the fastest known computers.

Why Public Key Cryptography?

Since knowledge of the encoding key K_E does not give information about the decoding key K_D , the encoding key K_E can be made public (hence the name “public key”). This allows anyone to encode messages that only the recipient can decode. For example, suppose you want to make a credit card purchase over the Internet. Data sent across the Internet unencrypted is not secure. However, if the merchant provides a public key, you can encrypt your credit card number and transmit it without worry. Only the merchant can decrypt the message, even though anyone may send one.

Authentication

Another feature of public key cryptography is authentication. In order to digitally sign a message, I could append my name encrypted with my deciphering key. The recipient can then decode it using my enciphering key. Then anyone can check that I was the actual sender, because only I could have encoded my name using my deciphering key. This feature is commonly used to digitally sign e-mail by software such as PGP.

Implementation

Public key cryptography is very computationally intensive, so typically its use is limited to allow for the secure transmission of a secret key; this secret key is then used to encrypt the rest of the message using a symmetric encryption method such as DES (Data Encryption Standard).

A common public key system in current use is RSA, named for its inventors: Rivest, Shamir and Adelman.

Some Number Theory

Most public key systems rely on number theoretic results. Before we can discuss the implementation of one, we need to quickly go over the necessary background.

Two numbers are said to be relatively prime if their greatest common divisor is 1.

How can we determine the gcd of two numbers? If the numbers are not too large, just looking at their factors does the trick.

$$\gcd(138, 126) = 6$$

since $138 = 2 \cdot 3 \cdot 23$ and $126 = 2 \cdot 3 \cdot 2 \cdot 7$.

Theorem 1 *Let a and b be two positive integers. Then there are integers x and y so that $ax + by = \gcd(a, b)$.*

Modular Arithmetic

Given three integers a , b and m , we say that a is congruent to b modulo m and write

$$a \equiv b \pmod{m} \quad \text{if } a - b \text{ is divisible by } m.$$

$$11 \equiv 5 \pmod{3} \text{ since } 11 - 5 = 6 = 2 \cdot 3.$$

We denote the set of integers modulo n as \mathbb{Z}_n . That is,

$$\mathbb{Z}_n = \{0, 1, 2, \dots, n - 1\}.$$

If r is such that $ar \equiv 1 \pmod{m}$, then r is called the multiplicative inverse of a .

Theorem 2 *Let a and n be integers, with $n \geq 2$. Then a has a multiplicative inverse modulo n if and only if $\gcd(a, n) = 1$.*

$2 \cdot 2 = 4 \equiv 1 \pmod{3}$, while $2x \not\equiv 1 \pmod{4}$ for any x .

Fermat's Little Theorem

Theorem 3 (Fermat's Little Theorem, 1640)

Let p be a prime. Any integer a satisfies

$$a^p \equiv a \pmod{p}$$

and any integer a not divisible by p satisfies

$$a^{p-1} \equiv 1 \pmod{p}.$$

The second part follows from the first, since if a is not divisible by p , then a and p are relatively prime, so a is invertible mod p , and we can divide by a .

Euler's ϕ function

When n is a positive integer, we define $\phi(n)$ to be the number of positive integers less than or equal to n that are relatively prime to n . If p is prime, then $\phi(p) = p - 1$, and if a and b are relatively prime, then $\phi(ab) = \phi(a)\phi(b)$. It can be shown that

$$\phi(n) = n \prod_{p|n} (1 - 1/p).$$

We now come to Euler's generalization of Fermat's result. This result is the central idea underlying the RSA public key cryptosystem.

Theorem 4 (Euler, 1750) *Let a and n be relatively prime integers, with $n \geq 2$. Then*

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

Notice that if $n = p$, we get Fermat's Theorem.

RSA

If you know that $n = pq$, then $\phi(n) = (p - 1)(q - 1)$, but in general, calculating $\phi(n)$ is as hard as factoring n , which, for very large n (say, 200 digits), is very hard indeed. This is the basic fact behind the RSA public key system.

To set up the system, we pick at random two large primes p and q , of about 100 digits each. We then set $n = pq$ so $\phi(n) = (p - 1)(q - 1)$. We also pick some other large number $e < \phi(n)$, which is relatively prime to $\phi(n)$. We make the numbers (n, e) public — these form the key needed for encoding. We also compute the multiplicative inverse d of e . Then $de - y\phi(n) = 1$. The number d is the part of the key we keep private.

RSA: How?

To encode a message: The sender divides the message up into blocks of equal length and to each block assigns an integer M with $0 < M < n$. For each block of plaintext, the sender transmits $m \equiv M^e \pmod n$.

To decode the message: For each unit m of the ciphertext received, the recipient computes $M \equiv m^d \pmod n$.

RSA: Why?

$$m^d \equiv M^{ed} \equiv M^{1+y\phi(n)} \equiv M(M^y)^{\phi(n)} \equiv M.$$

If $\gcd(M, n) = 1$, the last step follows from Euler's Theorem. It turns out that it is true in general.

RSA: Example Encoding

We choose $p = 29$ and $q = 53$. Then $n = pq = 1537$ and $\phi(n) = (p - 1)(q - 1) = 1456$. We then choose $e = 47$, which is prime, but doesn't divide 1456. Hence $\gcd(47, 1456) = 1$ and the encryption key is $(1537, 47)$.

Suppose we want to send "NO WAY". We set space: 00, A: 01, B: 02 etc. This gives

141500230125.

Since $n = 1537$, we break the plaintext into block of three digits, so we get

141 500 230 125

If you use a computer, you can check that

$$\begin{aligned} 141^{47} &\equiv 658, & 500^{47} &\equiv 1408, \\ 230^{47} &\equiv 1250, & 125^{47} &\equiv 1252 \pmod{1537}. \end{aligned}$$

The ciphertext is therefore

0658140812501252.

Notice that we use blocks of four digits, since $n = 1537$.

RSA: Example Decoding

To decode, we need to know that if we set $d = 31$, we get $de = 47 \cdot 31 \equiv 1 \pmod{1456}$. This is easy to compute if you know $\phi(n) = (p-1)(q-1) = 1456$, but if you only know n and not that $n = pq$, then you cannot compute $\phi(n)$ and d . If you use a computer, you can check that

$$\begin{aligned} 658^{31} &\equiv 141, & 1408^{31} &\equiv 500, \\ 1250^{31} &\equiv 230, & 1252^{31} &\equiv 125 \pmod{1537}. \end{aligned}$$

This gives us back

$$141500230125.$$