Public Key Cryptography

G. Eric Moorhouse, UW Math

References

A.J. Menezes, P.C. van Oorschot and S.A. Vanstone, *Applied Cryptography,* CRC Press, 1997.

D.R. Stinson, *Cryptography: Theory and Practice,* CRC Press, 1995.

R.L. Rivest, A. Shamir and L.M. Adleman, 'A method for obtaining digital signatures and public-key cryptosystems', *Communications of the ACM,* **21** (1978), 120–126.

Many recent textbooks in abstract algebra, applied algebra and number theory, e.g.

J. Gallian, *Contemporary Abstract Algebra,* 4th ed., Houghton Mifflin, 1998.

Coding Theory
(Theory of Error-Correcting Codes)

The design and study of codes which protect information against bit errors during transmission or storage.

Codes add redundancy to a message so that errors can be corrected when the message is read.

Cryptography

The design and study of schemes (cryptosystems) for the exchange of information which provide for one or more features such as:

Confidentiality—preservation of the content of the information from all but the intended recipient(s).

Authentication—guarantee of the identity of the author (and possibly the date, time and place of origin) of a message.

Cryptanalysis

The study of methods of defeating cryptosystems, including

- the extraction of private information from an encrypted message by unauthorised means;
- the unauthorised alteration of encrypted data; or
- the impersonation of a participant in the information exchange.

Cryptology = Cryptography + Cryptanalysis

Public Key Encryption

By this scheme, everyone is able to encrypt messages to send to Alice, which no one but Alice can decrypt.

The **encryption** algorithm is well known, efficient and easily performed on any computer. Alice's **public key** is required in the encryption process.

The **decryption** algorithm is also efficient but requires Alice's **private key**, known only to her. It is impossible or computationally infeasible to deduce the private key from the public key.

RSA Public Key Cryptography

Alice privately chooses two large primes $p \neq q$ and two large integers d, e such that de mod $(p-1)(q-1)$ is 1.

She publishes the pair (n, e) as her public key, where $n = pq$.

Bob encrypts the message m $(1 < m < n)$ as $m' = m^e$ mod n, which he sends to Alice.

To decrypt the message m', Alice computes $(m')^d$ mod n, which equals the original message m.

Security of the System

Alice's private key d cannot be determined without a knowledge of the factorisation of n. Without this information, it is presumably infeasible to recover the original message m given the encrypted message m'.

Example

Alice chooses

$p = 99103$, $q = 80177$

$d = 5144067833$, $e = 2968833449$

so

$$(p-1)(q-1) = 7945601952$$

and

de mod 7945601952 is 1.

(e is determined from d by Euclid's Algorithm.)

She publishes

$n = pq = 7945781231$ and $e = 2968833449$.

Encryption

Using blank=00, A=01, B=02, . . . , Z=26
we translate

Bob's message:	S E N D	M O N E Y
Translation:	19 05 14 04 00	13 15 14 05 25

		Encrypted message
$1905140400^e \bmod n$	$=$	6774683355
$1315140525^e \bmod n$	$=$	4105272362

Decryption

$$6774683355^d \bmod n = 1905140400 = \mathsf{S\ E\ N\ D}$$

$$4105272362^d \bmod n = 1315140525 = \mathsf{M\ O\ N\ E\ Y}$$

Why RSA works

Let $n = pq$ where $p \neq q$ are primes.

Let S be the set of positive integers $x < n$ such that $\gcd(x, n) = 1$. Then $|S| = (p-1)(q-1)$. The product of all elements of S is

$$\prod_{x \in S} x = \prod_{x \in S} (mx)$$

$$= m^{(p-1)(q-1)} \prod_{x \in S} x$$

(mod n) so $m^{(p-1)(q-1)}$ mod n is 1.

If de mod $(p-1)(q-1)$ is 1, i.e.

$$de = k(p-1)(q-1) + 1$$

then

$$m^{de} = m^{k(p-1)(q-1)+1}$$

$$= (m^{(p-1)(q-1)})^k \cdot m = m$$

(mod n).

RSA Authentication Scheme

As before, Alice privately chooses two large primes $p \neq q$ and two large integers d, e such that
de mod $(p-1)(q-1)$ is 1.

She publishes the pair (n, e) as her public key, where $n = pq$.

Alice encrypts the message m $(1 < m < n)$ as $m' = m^d$ mod n, which she sends to Bob.

Bob (or anyone) can decrypt the message m' by computing $(m')^e$ mod n, which equals the original message m. This demonstrates that the original message must have originated from Alice.

It is also possible to achieve *both* confidentiality *and* authentication for a network of individuals communicating over an open channel.

Rabin Encryption Scheme

The advantage of this scheme is that decrypting messages by unauthorised individuals is *known* to be as hard as factorising n.

Alice secretly chooses two large primes $p \neq q$ and publishes the value of $n = pq$. (For simplicity we'll assume p and q are both 3 mod 4.)

Bob encrypts a message m ($1 < m < n$) as $m' = m^2 \bmod n$, which he sends to Alice.

Alice decrypts the message as follows: determine

integers a, b such that $ap + bq = 1$;

$r = (m')^{(p+1)/4} \bmod p$;

$s = (m')^{(q+1)/4} \bmod q$;

$x = (aps + bqr) \bmod n$; and

$y = (aps - bqr) \bmod n$.

The four possible values of m are $\pm x \bmod n$ and $\pm y \bmod n$.

Modular exponentiation, while implemented efficiently in polynomial time, may still be too slow for some applications. In such situations, a conventional (faster) encryption process may be used, having one-time encryption/decryption key, e.g.:

Vernam Cipher

Until very recently, secure communication between Washington and Moscow used the following cipher scheme (with key exchange using a trusted courier service).

The two communicating parties secretly agree on a binary string $d = (d_1, d_2, \ldots, d_k)$ ($d_i = 0$ or 1).

A long message is broken up int binary strings of length k and encrypted as

$$(x_1, x_2, \ldots, x_k) \mapsto (x_1 \oplus d_1,\ x_2 \oplus d_2,\ \ldots,\ x_k \oplus d_k)$$

where $\oplus$ is addition mod 2.

Repeating this operation returns the original message. Both encryption and decryption (which are the same process) are performed very efficiently.

This is secure if

- the key d can be agreed upon with confidentiality, and
- each key is only used once and then destroyed.

We will describe how it is possible for two individuals, communicating over an open channel, to agree on an encryption key which is inaccessible to any eavesdroppers.

The security of this protocol rests on the assumed intractability of the *discrete logarithm problem.*

Discrete Logarithm Problem

For every prime p, there exists a *generator* a such that the powers

$$1, a, a^2, a^3, \ldots, a^{p-2} \bmod p$$

give *all* the nonzero integers mod p.

E.g. $p = 13$ has $a = 2$ as a generator:

k	$2^k \bmod 13$	
0	1	
1	2	
2	4	
3	8	
4	3	
5	6	$\longleftarrow \log_2(6) = 5$
6	12	
7	11	
8	9	
9	5	
10	10	
11	7	

Problem: Given $0 < x < p$, find $0 \leq k \leq p - 2$ such that

$$a^k \text{ mod } p \text{ is } x.$$

We write $k = \log_a(x)$.

The best known algorithm on a conventional computer finds $\log_a(x)$ in time

$$e^{O(L^{1/3}(\log L)^{2/3})}$$

where $L = \log p$ (Gordon, 1993). Shor's quantum algorithm computes discrete logarithms in time polynomial in L.

Diffie-Hellman Key Exchange

A large prime p and a generator a for the integers mod p, are agreed upon beforehand. (This information is not confidential.)

Alice secretly chooses a random integer $1 < x < p - 2$ and sends Bob the value of a^x mod p, using an unsecured channel.

Bob secretly chooses a random integer $1 < y < p - 2$ and sends Alice the value of a^y mod p, using the unsecured channel.

The secret encryption key is $d = a^{xy}$ mod p, which Alice computes as $(a^y)^x$ mod p, using the value of a^y which she obtains from Bob.

Bob determines the same key as $(a^x)^y$ mod p, using the value of a^x which he obtains from Alice.

Security of the Key Exchange

An eavesdropper can deduce the value of the secret key $d = a^{xy}$ from the values of a, a^x and a^y if he can first find $x = \log_a(a^x)$ and $y = \log_a(a^y)$, but this is presumed to be intractable. No faster method is known for breaking the security of this key exchange.

ElGamal Encryption Scheme

Alice chooses a large prime p, a generator a for the integers mod p, and a power a^x mod p where $1 < x < p-2$ is chosen randomly.

She publishes $(p, a, a^x \text{ mod } p)$ as her public key; x is her private key.

Bob encrypts a message m $(1 < m < p-2)$ as follows: He chooses $1 < k < p-2$ at random, and computes

$$m_1 = a^k \text{ mod } p \quad \text{and} \quad m_2 = m_1 (a^x)^k \text{ mod } p.$$

He sends the encrypted message (m_1, m_2) to Alice.

Alice decrypts the message by computing $m_1^{p-1-x} m_2$ mod p, which equals the original message m.

Breaking this scheme is presumed to be as difficult as the discrete logarithm problem.

This scheme has the advantage that the same message will not always be encrypted in the same way.