

The Rivest-Shamir-Adleman Cryptosystem

MATH2068 Number Theory & Cryptography
Week 6 Lecture 2

University of Sydney
NSW 2006
Australia

28th August 2007



Number theory prerequisites

Proposition: Let $e, m \in \mathbb{Z}$ with $\gcd(e, m) = 1$. Then e has an inverse mod m . (i.e. there exists $d \in \mathbb{Z}$ with $de \equiv 1 \pmod{m}$.)

Indeed, the E.E.A. tells us how to find integers r, s with $re + sm = \gcd(e, m) = 1$. We can choose $d = r$.

Suppose that p and q are prime numbers, and let $n = pq$. We proved in Lecture 1 of Week 4 that $\phi(n) = (p - 1)(q - 1)$.

We also proved the following proposition.

Proposition: Let $n = pq$, where p, q are primes and $p \neq q$. Then $a^{k\phi(n)+1} \equiv a \pmod{n}$ for all $a \in \mathbb{Z}$ and all positive integers k .

So if e is coprime to $\phi(n)$ and d is the inverse of $e \pmod{\phi(n)}$ then $a^{de} \equiv a \pmod{n}$, for all a .

A public key cryptosystem



RSA is a public key cryptosystem. This means that the encryption key is public knowledge.

But the decryption key is kept secret.

The idea is that RSA user Bob posts his public key on the web. Then if Alice needs to send a secure message to Bob she can encrypt it with Bob's public key, confident that only Bob can decrypt it.

It is essential that knowing the encryption key (public key) does not enable one to work out the decryption key (private key).

RSA keys



For the RSA system a public key is a pair of integers (n, e) .

The integer n should be the product of two very large primes, $n = pq$ say, and e must be coprime to $(p - 1)(q - 1)$.

The primes p and q must be kept secret.

The private key is (n, d) , where d is the inverse of e modulo $(p - 1)(q - 1)$.

Plaintext and ciphertext



If the public key is (n, e) then the plaintext should be a sequence of residues mod n .

The ciphertext will also be a sequence of residues mod n .

This means that one must have a method of representing any message as a sequence of residues mod n . There are many ways one could choose to do this.

For example, the ascii code assigns a natural number less than 128 to each letter, numeral or punctuation mark occurring in normal English text. One could simply concatenate the ascii code numbers of the symbols in the message to obtain a string of numerals.

And then chop up this string to make a sequence of, say, 200 digit numbers.

Assuming n has more than 200 digits, this sequence of numbers is a sequence of residues mod n , as needed.

Encryption



Assume that the public key is (n, e) , and the plaintext has been put in the form of a sequence $[a_1, a_2, \dots, a_k]$, where each a_i is a natural number less than n .

The ciphertext is then the sequence $[b_1, b_2, \dots, b_k]$, where b_i is the residue of $a_i^e \pmod{n}$.

Note that since n is huge, the a_i 's are usually huge too. And e could very well be huge.

So it is probably not feasible to compute a_i^e . But one can compute the mod n residue of a_i^e without computing a_i^e . Moreover, these computations can be done rapidly, despite the size of the numbers.

Decryption



Assume that the private key is (n, d) . The ciphertext will have the form $[b_1, b_2, \dots, b_k]$, where each b_i is a natural number less than n .

For each i let c_i be the residue of $b_i^d \pmod{n}$. Then $[c_1, c_2, \dots, c_k] = [a_1, a_2, \dots, a_k]$, the original plaintext.

(Of course, $[a_1, a_2, \dots, a_k]$ is really the numerically encoded form of the plaintext. Presuming that this encoding is done by a sensible method, recovering the text will be trivial.)

Why does it work?



By the encryption algorithm, $b_i \equiv a_i^e \pmod{n}$.

By the decryption algorithm, $c_i \equiv b_i^d \pmod{n}$.

Thus $c_i \equiv (a_i^e)^d \pmod{n}$.

But $(a_i^e)^d = a_i^{ed} \equiv a_i \pmod{n}$, by the proposition we recalled earlier, since $ed \equiv 1 \pmod{\phi(n)}$.

So $c_i \equiv a_i \pmod{n}$, and since c_i and a_i are residues mod n they must be equal.

Why is it secure?



Everyone knows n and e . To be able to decrypt messages they would need to know d (and n).

Now d is the inverse of $e \bmod \phi(n) = (p-1)(q-1)$. So anyone who knows $\phi(n)$ can quickly compute d .

But knowing n doesn't tell you $\phi(n)$, unless you can factorize n and thus find p and q .

At present it is easy to find numbers n that are large enough to defeat the best known factorization algorithms, yet small enough for RSA encryption and decryption to be fast.

An example with small numbers



Let $n = 3 \times 11$, a product of two primes. Thus $\phi(n) = (3-1)(11-1) = 20$.

The encryption exponent e must be chosen coprime to $\phi(n)$; let us take $e = 3$.

The public key will then be the pair $(n, e) = (33, 3)$.

To find the private key we must compute the inverse of $e \bmod \phi(n)$. The answer is $d = 7$, found by the E.E.A.:

$$\begin{array}{r} 20 \ 3 \ 2 \ 1 \ 0 \\ \ 6 \ 1 \ 2 \\ \ 0 \ 1 \ 6 \ 7 \\ \ 1 \ 0 \ 1 \ 1. \end{array}$$

The theory tells us that 7×3 must be congruent to $(-1)^2 \pmod{20}$. This is obviously true!

The RSA private key in this example is therefore $(33, 7)$.

Encrypting a message



The plaintext must be a sequence of residues mod 33. As an example, let us choose $[2, 5, 6, 20, 11]$.

Since $e = 3$ the ciphertext is $[2^3, 5^3, 6^3, 20^3, 11^3]$, computed using residue arithmetic modulo 33.

$$2^3 = 8.$$

$$\text{Mod } 33 \text{ we have } 5^3 \equiv 5 \times 25 \equiv 5 \times (-8) \equiv -40 \equiv 26.$$

$$\text{Similarly } 6^3 \equiv 6 \times 36 \equiv 6 \times 3 \equiv 18.$$

$$\text{And } 20^3 \equiv (-13)^3 \equiv -13 \times 169 \equiv -13 \times 4 \equiv 14.$$

$$\text{And } 11^3 \equiv 11 \times 121 \equiv 11 \times -11 \equiv 11.$$

The ciphertext is $[8, 26, 18, 14, 11]$.

Decrypting it



Since $d = 7$ we decrypt $[8, 26, 18, 14, 11]$ by calculating $[8^7, 26^7, 18^7, 14^7, 11^7]$.

$$8^2 \equiv 64 \equiv -2; \text{ so } 8^6 \equiv -8 \text{ and } 8^7 \equiv (-8) \times 8 \equiv 2.$$

$$26 \equiv -7; \text{ so } 26^2 \equiv 49 \equiv 16; \text{ so } 26^3 \equiv -112 \equiv -13;$$

$$\text{so } 26^6 \equiv 169 \equiv 4; \text{ so } 26^7 \equiv -28 \equiv 5.$$

$$18^2 \equiv 324 \equiv -6; \text{ so } 18^3 \equiv -108 \equiv -9; \text{ so } 18^6 \equiv 81 \equiv -18;$$

$$\text{so } 18^7 \equiv 324 \equiv 6.$$

$$14^2 \equiv 196 \equiv -2; \text{ so } 14^6 \equiv -8; \text{ so } 14^7 \equiv -112 \equiv 20.$$

$$11^2 \equiv -11; \text{ so } 11^7 \equiv 11.$$

Thus deciphering gives $[2, 5, 6, 20, 11]$. And so it should, since the plaintext was $[2, 5, 6, 20, 11]$.

Authenticating announcements



Sometimes mischievous people distribute bogus announcements purporting to come from a reputable organization, but containing false or misleading information.

How can reputable organizations deal with this problem?

One possible method is to always encrypt public announcements using a system that has a secret encryption key but a public decryption key.

No one but the holder of the encryption key will be able to construct an encrypted announcement that decrypts into something meaningful.

So if a member of the public can decrypt the announcement and finds that the result is not nonsense, he can be sure that it originated in the right place.

RSA in reverse



To do this with RSA you encrypt the announcement using your private key (n, d) , and tell people to use the public key (n, e) to decrypt it.

So if the unencrypted announcement (numerically encoded as a sequence of residues mod n) is $[a_1, a_2, \dots, a_k]$ then the publically released encrypted form will be $[b_1, b_2, \dots, b_k]$, where b_i is the residue of $a_i^d \pmod{n}$.

Decryption using (n, e) works since $b_i^e \equiv a_i^{de} \equiv a_i \pmod{n}$.