Math 445 Number Theory

September 15 and 17, 2008

Public Key Cryptosystems: The idea behind a cryptosystem is to provide a method of encoding a message so that only the person intended to receive it can recover the original message. Public key systems take the added step of publishing the encoding method for all to see.

The RSA cryptosystem is a public key cryptosystem which uses exponentiation mod N as its encoding and decoding method. The idea is for someone to send a message as a number a mod N, by splitting the computation of a into two pieces, one part that the sender (and everybody else) knows, the other part that only the receiver knows. This is done using Euler's generalization if FLT: if $(a,N)=1$ and $k\equiv 1 \pmod{\phi(N)}$, then $k=1+\phi(N)m$ and $a^k=(a^{\phi(N)})^m\cdot a\equiv 1^m\cdot a=a$. The idea is to choose such a $k=de$, and publish n and e, but keep d secret. The sender computes $b=a^e \pmod N$ and sends b, and then the receiver can recover the original message by computing $a=b^d \pmod N$. This can all be accomplished if the receiver chooses an N for which he/she knows how to compute $\phi(N)$ (but nobody else can (easily) compute $\phi(N)$ from N). This in essence means choosing an N for which the receiver knows the factorization, but that nobody else can easily factorize.

Typically, to build the system, we need a pair of large (distinct) primes p,q and a number e with $\gcd(e,(p-1)(q-1))=1$. We set $n=pq$, and publish n and e . Privately, we also find (via the Euclidean algorithm) a number d (and x) satisfying $de-x(p-1)(q-1)=1$. We then keep d secret (and throw away our calculations, including the values of p and q).

To send us a message, the text is first converted into a string of numbers, using some standard procedure (e.g., use the ascii character codes for the symbols in the message), which are then cut into pieces each having fewer digits than n. Let A be one such string. You then compute, using my public key, the value

$$B=A^e \pmod n$$

and send me the number B. Then I compute

$$B^d=(A^e)^d=A^{ed}=A^{x(p-1)(q-1)+1}=A(A^{(p-1)(q-1)})^x=A1^x=A \pmod n$$

since $A^{(p-1)(q-1)}$ is $\equiv 1$ mod p and q, so is $\equiv 1$ mod n (since $\gcd(p,q)=1$) .

The security of this system lies in the fact that, to the best of our knowledge, the message A cannot be recovered from the cypher B, without knowing

d, which requires you to know $(p-1)(q-1)$ (to find it the way <u>we</u> did), which requires you to know p and q, which requires you to factor n. So its strength lies in the fact that (to the best of our knowledge) finding the prime factors of a large number is <u>hard</u>, especially when the primes are large!

To make things more interesting, if you also have a public key system, (n_1, e_1, d_1), then you can (after we have agreed to do this...) apply a two-step process to the message; take the message A and compute

$$B = A^{d_1} \pmod{n_1}\ \text{, and then compute } C = B^e \pmod{n},$$

and send me C. I then compute

$$B = C^d \pmod{n}\ \text{, and } A = B^{e_1} \pmod{n_1}\ ,$$

to recover the original message. This message, just because I can read it, tells me that only you could have sent it (because only you know d_1). The message can only be read by me, and could only have been sent by you.

The work involved in encrypting and decrypting lies in carrying out the exponentiations mod $n = pq$. The person holding the secret can speed this up by <u>keeping</u> p and q, and doing a pair of exponentiations mod p and q (which are much faster). The idea is that if we know $a^d \equiv a_1 \pmod{p}$ and $a^d \equiv a_2 \pmod{q}$, then we can recover $a^d \pmod{pq}$ by solving $x \equiv a_1 \pmod{p}$ and $x \equiv a_2 \pmod{q}$; by the Chinese Remainder Theorem, there is a unique solution mod pq, which can be effectively computing using the Euclidean algorithm. This speed-up is in fact used by some smart cards that implement RSA for security.

Public key cryptosystem vulnerabilities:

(1) The public key (N, e) is public! Anyone can spend any amount of time breaking it (by factoring N), without waiting for cyphertext to be intercepted. Which is why we want N to be so hard to factor....

(2) If someone can guess what message (or which 1,000,000 messages) you might be sending, they can compute what cyphertext $B = A^e \pmod{N}$ would correspond to that message, effectively reading the message A without knowing the secret key d.