RSA

Alejandra Alvarado

June 23, 2014

The Communication Scenario

Suppose Alice and Bob want to communicate with each other privately.

If Alice wants to send Bob a message, she will encrypt it and send Bob the encrypted message.

Assuming only he has the decryption key, he can then determine the original message.

Eve is a third party who wants to eavesdrop on their conversation.

Symmetric vs Asymmetric

Two basic types of cipher systems.

Symmetric: encryption and decryption keys are known only to Alice and Bob. Keys are possibly the same, or one can be deducted from the other.

Asymmetric (public key): encryption key is public, decryption key is not.

RSA is a public key cipher system.

Advantage of public key is that providing authentic public keys is easier than distributing secret keys securely.

RSA

The RSA Algorithm is based on the difficulty of factoring large primes.

The largest known number factored has 232 digits. (768 bits)

Recommended that *n* should be 617 digits (2048 bits) long.

Ten years ago, 155 digits (512 bits) was unbreakable.

RSA Algorithm

- Alice wants to send Bob a message.
- Bob secretly chooses two distinct primes p and q.
- He computes $n = pq$, which is public.
- He chooses e such that $\gcd(e, (p-1)(q-1)) = 1$. The encryption key e is public.
- He chooses his decryption key d such that $d \equiv e^{-1} \mod (p-1)(q-1)$.
- Alice encrypts her message m, $0 \leq m < n$, as $c \equiv m^e \mod n$, which is sent to Bob.
- Bob decrypts by computing $m \equiv c^d \mod n$.

Why Does This Work?

Definition (Euler's totient function)

$\phi(n) = |\{x : 1 \leq x < n, \gcd(x, n) = 1\}|$

Theorem (Euler's Theorem)

If $\gcd(a, n) = 1$*, then* $a^{\phi(n)} \equiv 1 \mod n$*.*

Theorem (Fermat's Little Theorem)

If p is prime and $\gcd(a, p) = 1$*, then* $a^{p-1} \equiv 1 \mod p$*.*

Why Does This Work? (cont.)

Recall:

$n = pq$ where n public, p and q private.

m = message (private)

c = encrypted message (public)

e = encryption key (public), $\gcd(e, (p-1)(q-1)) = 1$

d = decryption key (private), $de \equiv 1 \mod (p-1)(q-1)$

Why Does This Work? (cont.)

gcd(*m*,*n*)=1

By Euler's Theorem, $m^{\phi(n)} \equiv 1 \mod n$ where

$\phi(n) = \phi(pq) = (p-1)(q-1)$.

$c^d \equiv m^{ed} \mod n \equiv m^{1+k\phi(n)} \mod n \equiv m \mod n$

Why Does This Work? (cont.)

Theorem (Chinese Remainder Theorem)

If $\gcd(p, q) = 1$*, then the congruences*

$$x \equiv a \mod p, \quad x \equiv b \mod q$$

have a unique solution $x \mod pq$*.*

gcd(*m*,*n*)$\neq$ 1 Then $\gcd(m, n) = p$ or q or pq.

If $\gcd(m, n) = p$ (or $\gcd(m, n) = q$ but not both), then $m = pm_1$.
By Euler's Theorem, $m^{q-1} \equiv 1 \mod q$.
Since $m \equiv 0 \mod p$, by Chinese Remainder Theorem,
$c^d \equiv m^{ed} \mod n \equiv m^{1+k\phi(n)} \mod n \equiv m \mod n$.

If $\gcd(m, n) = pq$, then $c^d \equiv m^{ed} \equiv 0 \mod n \equiv m \mod n$.

Example

Bob chooses $p = 557$ and $q = 4349$. Then $n = 2422393$.

His encryption key is $e = 3$ since $\gcd(e, (p-1)(q-1)) = 1$.

Bob computes his decryption key to be
$d \equiv 3^{-1} \mod 2417488 = 1614929$.

Alice wants to send Bob the message "PASS ME".

Let 00 represent A, 01 to represent B, and so on. Then her message can be represented as an integer
$m = 150018181204$.

Example (cont.)

She then looks up e and n.

Since her message is larger than n, she breaks up her message as $m_1 = 150018$ and $m_2 = 181204$, then encrypts each one individually.

She sends Bob $c_1 \equiv 150018^3 \mod 2422393 = 1554793$ and $c_2 \equiv 181204^3 \mod 2422393 = 1618552$.

Bob obtains the original message by computing $c_1^d \mod n$ and $c_1^d \mod n$.

Factorization vs Primality Testing

Largest known prime has 12,978,189 digits.

Factorization and primality testing are not the same.

It is easier to prove a number is composite than it is to factor it.

In 2002, largest announced factorization of an integer with 155 digits. At the same time, it is possible to prove a 1000 digit number is prime.

Largest integer factored has 232 digits. Largest integer factored using elliptic curves has 73 digits.

Largest prime proved with elliptic curve primality proving is over 20,000 digits.

Primality Testing

Proposition (Pocklington-Lehmer)

Let $n > 1$ and $n - 1 = rs$ with $r \geq \sqrt{n}$ and $\gcd(r, s) = 1$. For each prime l that divides r, there exists an integer a_l with $a_l^{n-1} \equiv 1 \mod n$ and $\gcd(a_l^{(n-1)/l} - 1, n) = 1$ if and only if n is prime.

Example

Is $n = 5279$ prime?
Factor $n - 1 = 5278 = 2 \cdot 7 \cdot 13 \cdot 29$.
Let $r = 91$ and $s = 58$.
Then $a_7 = 3$ and $a_{13} = 432$ satisfies the properties in the proposition.
So 5247 must be prime.

References

- eccworkshops.org
- Elliptic Curves in Cryptography (Blake)
- Elliptic Curves, Number Theory and Cryptography (Washington)
- primes.utm.edu
- research.microsoft.com/en-us/um/people/klauter
- rsa.com/rsalabs/node.asp?id=2013
- en.wikipedia.org/wiki/RSA_Factoring_Challenge