Lecture 4

Review: RSA Public-Key Encryption

RSA Algorithm

Preparation by Bob

i.e. 1024 bits

1. Choose at random two large primes p and q w/ $p \neq q$
2. Let $n = pq$
3. Choose a small integer e relatively prime to $\varphi(n) = (p-1)(q-1)$
4. Compute $d := e^{-1} \bmod \varphi(n)$ using extended gcd algorithm
5. Publish the pair $P := (e, n)$ as Bob's RSA public key
6. Keep secret pair $S := (d, n)$ as Bob's RSA private key

Encryption by Alice:

Alice wants to send a secret message to Bob in the form of a number $M \bmod n$. She encrypts her message using Bob's RSA public key: $C := M^e \bmod n$, and sends the encrypted message C to Bob.

Decryption by Bob:

When Bob recieves the encrypted message C, he decrypts it using his RSA private key by calculating $C^d \bmod n$.

Proof: $C^d = M^{ed} \mod n$

Since $ed \equiv 1 \mod \phi(n)$, $\phi(n) = (p-1)(q-1)$, we have
$ed = 1 + k(p-1)(q-1)$ for some $k \in \mathbb{Z}$

If $M \not\equiv 0 \mod p$, we have
$$M^{ed} = M(M^{p-1})^{k(q-1)} \equiv M \cdot q^{k(q-1)} \equiv M \mod p$$
Fermat's Little Theorem

If $M \equiv 0 \mod p$, we also have $M^{ed} \equiv M \mod p$.

Similarly, we have $M^{ed} \equiv M \mod q$.
By the Chinese remainder theorem, we obtain $M^{ed} \equiv M \mod n$

What happens to the eavesdropper Eve?

He has got the encrypted message C and Bob's RSA public key $P = (e, n)$. It is very hard for him to find M such that $M^e \equiv C \mod n$ without knowing the prime factorization of n.
Calculating the prime factorization for large n is also very hard.

Example:

Bob picks: $p = 61$ and $q = 53$,

$n = pq = 3233$

$\phi(n) = 60 \cdot 52 = 3120$

picks $e = 17$, computes $d := e^{-1} = 2753 \mod 3120$

Alice wants to send secret message $m = 65$.

She encrypts it by $C := 65^{17} = 2790 \mod 3233$, and

sends $C = 2790$ to Bob.

Bob recieves $C = 2790$, and decrpts it by

$$C^d = \underset{\substack{\uparrow \\ \text{computed} \\ \text{logarithmically}}}{2790}^{2753} = 65 \mod 3233$$

Question: How to find large primes for RSA algorithm.

8. Primality testing

Goal: Find large random primes.

Idea: Large primes are not too rare; it is feasible to test large random integers until you find one that's prime.

Prime number theorem:

Prime distribution function $\pi(n) := \#$ of primes $\leq n$.

$$\lim_{n \to \infty} \frac{\pi(n)}{n/\ln(n)} = 1.$$

Corollary: For a large random integer n, the probability that n is prime is approximately $1/\ln n$.

So the expected number of trials before success is aproximately $\ln n$.

Next: How do we know whether a random large integer is prime?