MATH 433

Applied Algebra

Lecture 12:
Public key encryption.
The RSA system.

Euler's Theorem

$\mathbb{Z}_n$: the set of all congruence classes modulo n.
G_n: the set of all invertible congruence classes modulo n.

Fermat's Little Theorem Let p be a prime number. Then $a^{p-1} \equiv 1 \bmod p$ for every integer a not divisible by p.

Theorem (Euler) Let $n \geq 2$ and $\phi(n)$ be the number of elements in G_n. Then

$$a^{\phi(n)} \equiv 1 \bmod n$$

for every integer a coprime with n.

Theorem Let $n = p_1^{s_1} p_2^{s_2} \dots p_k^{s_k}$, where $p_1, p_2, \dots, p_k$ are distinct primes and $s_1, \dots, s_k$ are positive integers. Then

$$\phi(n) = p_1^{s_1 - 1}(p_1 - 1) p_2^{s_2 - 1}(p_2 - 1) \dots p_k^{s_k - 1}(p_k - 1).$$

Public key encryption

Suppose that Alice wants to obtain some confidential information from Bob, but they can only communicate via a public channel (meaning all that is sent may become available to third parties, in particular, to Eve). How to organize secure transfer of data in these circumstances?

The **public key encryption** is a solution to this problem.

Public key encryption

The first step is **coding**. Bob digitizes the message and breaks it into blocks $b_1, b_2, \dots, b_k$ so that each block can be encoded by an element of a set $X = \{1, \dots, K\}$, where K is large. This results in a **plaintext**. Coding and decoding are standard procedures known to public.

Next step is **encryption**. Alice sends a **public key**, which is an invertible function $f : X \to Y$, where Y is an equally large set. Bob uses this function to produce an encrypted message (**ciphertext**): $f(b_1), f(b_2), \dots, f(b_k)$. The ciphertext is then sent to Alice.

The remaining steps are **decryption** and **decoding**. To decrypt the encrypted message (and restore the plaintext), Alice applies the inverse function f^{-1} to each block. Finally, the plaintext is decoded to obtain the original message.

Trapdoor function

For a successful encryption, the function f has to be the so-called **trapdoor function**, which means that f is easy to compute while f^{-1} is hard to compute unless one knows special information ("trapdoor").

The usual approach is to have a family of fuctions $f_\alpha : X_\alpha \to X_\alpha$ (where $X \subset X_\alpha$) depending on a parameter α (or several parameters). For any function in the family, the inverse also belongs to the family. The parameter α is the trapdoor.

An additional step in exchange of information is **key generation**. Alice generates a pair of **keys**, i.e., parameter values, α and β such that the function f_β is the inverse of f_α.
α is the **public key**, it is communicated to Bob (and anyone else who wishes to send encrypted information to Alice).
β is the **private key**, only Alice knows it.

The encryption system is efficient if it is virtually impossible to find β when one only knows α.

The RSA system

The **RSA (Rivest-Shamir-Adleman)** system is a public key system based on the modular arithmetic.

$X = \{1, 2, \dots, K\}$, where K is a large number (say, 2^{128}).

The **key** is a pair of integers (n, α), **base** and **exponent**. The domain of the function $f_{n,\alpha}$ is G_n, the set of invertible congruence classes modulo n, regarded as a subset of $\{0, 1, 2, \dots, n-1\}$. We need to pick n so that the numbers $1, 2, \dots, K$ are all coprime with n.

The function is given by $f_{n,\alpha}(x) = x^\alpha \bmod n$.

Key generation: First we pick two distinct primes p and q greater than K and let $n = pq$. Secondly, we pick an integer α coprime with $\phi(n) = (p-1)(q-1)$. Thirdly, we compute β, the inverse of α modulo $\phi(n)$.

Now the public key is (n, α) while the private key is (n, β).

By construction, $\alpha\beta = 1 + \phi(n)k$, $k \in \mathbb{Z}$. Then

$$f_{n,\beta}(f_{n,\alpha}(x)) = [x]_n^{\alpha\beta} = [x]_n([x]_n^{\phi(n)})^k,$$

which equals $[x]_n$ by Euler's theorem. Thus $f_{n,\beta} = f_{n,\alpha}^{-1}$.

Efficiency of the RSA system is based on impossibility of efficient prime factorisation (at present time).

Example. Let us take $p = 5$, $q = 23$ so that the base is $n = pq = 115$. Then $\phi(n) = (p-1)(q-1) = 4 \cdot 22 = 88$.

Exponent for the public key: $\alpha = 29$. It is easy to observe that -3 is the inverse of 29 modulo 88:

$$(-3) \cdot 29 = -87 \equiv 1 \bmod 88.$$

However the exponent is to be positive, so we take $\beta = 85$ $(\equiv -3 \bmod 88)$.

Public key: $(115, 29)$, private key: $(115, 85)$.

Example of plaintext: 6/8 (two blocks).

Ciphertext: 26 $(\equiv 6^{29} \bmod 115)$, 58 $(\equiv 8^{29} \bmod 115)$.

Primality tests

For an efficient implementation of the RSA system, we should be able to quickly generate large prime numbers in a given range (say, between 2^{127} and 2^{128}).

There is no (efficient) deterministic algorithm to generate large primes. Instead, a probabilistic algorithm is used.

First we generate a random number p in the given range such that p is not divisible by small prime numbers from a selected (very short) list. Then we run a number of **primality tests**. If p passes all tests, it is accepted. Otherwise we generate another number.

Primality tests are designed so that all prime numbers pass them while most composite numbers do not.

Fermat primality test

To test if a large number p is prime, we choose a (very small) number a and verify the congruence $a^{p-1} \equiv 1 \bmod p$. If the congruence does not hold, the number p is surely composite (due to Fermat's Little Theorem).

Suppose a is a positive integer coprime with p, where p is composite. The number a is called a **Fermat witness** for (compositeness of) p if $a^{p-1} \not\equiv 1 \bmod p$. Otherwise a is called a **Fermat liar**.

Proposition The product of two Fermat liars (for the same p) is also a Fermat liar. The product of a Fermat liar and a Fermat witness is a Fermat witness.

Corollary If there is at least one Fermat witness for p, then the number of Fermat liars does not exceed the number of Fermat witnesses.

There exist composite numbers with no Fermat witnesses (the **Carmichael numbers**). Example: $3 \cdot 11 \cdot 17 = 561$.