OLGA RADKO MATH CIRCLE, SPRING 2024: ADVANCED 3

FERNANDO FIGUEROA AND JOAQUÍN MORAGA

Worksheet 3: Public key crpytography

"Cryptography is the study of methods of sending messages in disguised form so that only the intended recipients can remove the disguise and read the message. The message we want to send is called the plaintext and the disguised message is called the ciphertext."

An enciphering transformation is a function that takes any plaintext message and gives a ciphertext message. In other word it is a function $f : \mathcal{P} \to \mathcal{C}$, where $\mathcal{P}$ and $\mathcal{C}$ are the sets of all possible plaintexts and ciphertexts, respectively. We will usually require that f is one-to-one and onto, i.e. for any ciphertext there is exactly one plaintext that gets encrypted into it. The deciphering transformation is f^{-1}, the inverse of the enciphering transformation.

Today we will mostly working with encryptions that go from $\mathbb{Z}/n\mathbb{Z}$ to itself.

Such a function can be for example of the form $f(x) = ax + b$, for fixed numbers a, b, with a invertible in the ring $\mathbb{Z}/n\mathbb{Z}$. If we know the value of n and that the encryption is by an affine function $f(x) = ax + b$, then we only need to determine the values of a, b to determine how to decipher the ciphertexts.

Problem 3.1:

Compute the inverse functions for the following functions. You must give the answer in the form $g(x) = cx + d$, where c and d are integers.

(1) $f(x) = 3x + 5$ in $\mathbb{Z}/7\mathbb{Z}$
(2) $f(x) = 7x + -4$ in $\mathbb{Z}/10\mathbb{Z}$
(3) $f(x) = 4x + 5$ in $\mathbb{Z}/15\mathbb{Z}$

Solution 3.1:

A public key cryptographic system is one that uses a pair of related keys, a *public key* and a *private key*. Anyone with a public key can encrypt a message, but only those that posses the private key can decrypt the ciphertext. The security of such a system relies on keeping the private key secret and the mathematical difficulty of finding the inverse to the enciphering transformation. As such, what can be thought of a public key cryptography relies on what is the state of art of algorithms to solve certain problems and the computational power that one has available.

In traditional cryptography (e.g. Caesar cipher) both the encryption and decryption have to be kept private, in order for no one else to be able to intercept messages.

sometimes we will write the key of the cryptosystem as $K = (f, g)$, where f and g are the enciphering and deciphering functions. Othertimes we will write the key as $f = (a, b)$, where (a, b) are parameters that define the encrypting function (which is of public knowledge).

Problem 3.2:

Suppose that m people want to communicate with each other. Each user wants to communicate with each other in such a way that the remaining $m - 2$ people cannot eavesdrop.

(1) How many different keys (cryptosystems) are needed to ensure this, if they are using classical cryptosystems?
(2) How many different keys are needed to ensure this, if they are using public key cryptosystems?

Solution 3.2:

The objective of this problem is to create a long distance coin flip. This could be used, for example if two countries are organizing a soccer match and they want to decide who will hosted the competition, without having to meet in person to flip a coin.

Let our encryption key be a two to one function $f : \mathcal{P} \to \mathcal{C}$, meaning that any element c in the image of f has exactly two distinct preimages p_1 and p_2, with $f(p_1) = f(p_2) = c$. And the decryption key g gives both of the preimages for a ciphertext.

Notice that if one has an element p_1, one can find the companion element p_2 if we know both f and g. But we will assume that only knowing the encryption function f, it is impossible to compute the companion element p_2 (because finding g is computationally too hard).

Problem 3.3:

Suppose that two people (Alice and Bob), want to use this setup (f and g) to flip a coin. Alice generates the functions f and g and sends f (but not g) to Bob.

Explain a process in which each player has a 50% chance of winning, and they can prevent the other person from cheating.

Solution 3.3:

Often one of the most important parts of a message is the signature, this ensures that the message was sent by a specific person and no one else.

Suppose there are two people A (Alice) and B (Bob), each having their public key cryptosystem $K_A = (f_A, g_A)$, $K_B = (f_B, g_B)$. Anyone can send a message to Bob, by simply encrypting using f_B.

Problem 3.4:

Find a way in which Alice can encrypt a message to Bob, ensuring that only Bob can decipher it, and no one other than Alice could have encrypted it.

Hint: If one switches plaintext and ciphertext, only Alice can encipher using g_A, while anyone can decipher using f_A.

Solution 3.4:

The objective of the following few problems is to describe the RSA cryptosystem.

First the user chooses two large prime numbers p q, and sets $n = pq$. These primes will need to be large, both for the implementation of the cryptosystme in problem 3.8 and for the deciphering functions to be hard to compute by knowing the enciphering function.

Let $\phi(n)$ denote the number of positive integer numbers coprime with n

Problem 3.5:

Show that $\phi(n) = n + 1 - p - q = (p - 1)(q - 1)$.

Solution 3.5:

Problem 3.6:
Show that $X^{(p-1)(q-1)} \equiv 1 \pmod{pq}$
Solution 3.6:

Next, the user chooses randomly an integer number e less than $\phi(n)$ that is coprime with $\phi(n)$. They find the inverse of e modulo $\phi(n)$, i.e. a positive integer d, such that $ed \cong 1 \pmod{\phi(n)}$.

From now on we will call the user Alice, we have fixed $p_A, q_A, n_A = p_A q_A, \phi(n_A) = (p_A - 1)(q_A - 1), e_A, d_A$.

The public enciphering key will be given by the entries (n_A, e_A), where the function will be $f(X) \equiv X^{e_A} \pmod{n_A}$.

The difficulty of finding the deciphering key of RSA relies on the difficulty to find the factorization of $n = pq$.

Problem 3.7:

Show that the deciphering key will be given by the entries (n_A, d_A), where the deciphering function will be $g(Y) \equiv Y^{d_A} \pmod{n_A}$

Solution 3.7:

In the previous algorithm the possible plaintext and ciphertexts depend on the choice of prime numbers, namely $\mathcal{P} = \mathcal{C}$ consists of $n_A = p_A q_A$ many elements.

The way we usually implement this ciphertext is to impose $N^k \leq n_A \leq N^l$, where N is the number of letters in an alphabet. Thus any string of text consisting of k-letters can be sent to a number smaller than n_A, enciphered by RSA and gives a unique block of text of at most l letters.

Here strings of letter as seen as numbers, via using base N.

For the following problem you may use a calculator.

Problem 3.8:

Let $N = 26, k = 3, l = 4$, meaning that we will send strings of two letters to strings of at most 3 letters, using an alphabet of 26 letters.

(1) Using cipher the key $(n, e) = (46927, 3)$ encipher the message YES.
(2) Using the fact that $46927 = 281 * 167$, find a deciphering key, and corroborate that the answer from the previous item gives you back the plaintext YES.

Solution 3.8:

Different choices of p q work better than others, in terms of how hard it is to break the code, i.e. find an enciphering key to RSA.

Let p, q be prime numbers, with $m := \text{lcm}(p-1, q-1)$

Problem 3.9:

(1) Show that $x^m \equiv 1 \pmod{pq}$, for any x coprime with pq.
(2) Use the previous item to exaplain why it is important in RSA to pick primes such that the least common multiples of $p-1$ and $q-1$ is large.

Solution 3.9:

The logarithm problem in $\mathbb{F}_q$ corresponds to, for given given x, b find an element y in $\mathbb{F}_q$, such that $y^b = x$. This is generally computationally hard, and the following cryptosystem relies on that. We won't worry much about the precise meaning of something being computationally hard, but in this setting it can be seen as any (known) algorithm taking a very large ammount of time compared to the entries.

The Massey-Omura cryptosystem:

There is a publicly agreed q (power of a prime number), and we work in the field $\mathbb{F}_q$.

Each user selects an integer between 0 and $q-1$, and these are kept secret. Let us call this numbers e_A for Alice and e_B for Bob. Alice and Bob can each find their own d_A, d_B such that $e_A d_A \equiv e_B d_B \equiv 1 \pmod{q-1}$, and still keep them secret.

In order for Alice to send a message P to Bob. She first sends P^{e_A} to Bob, then Bob replies back with the message $P^{e_A e_B}$.

Finally Alice gives back the message $P^{e_A e_B d_A}$ to Bob.

For this problem you may assume without proof that in $\mathbb{F}_q \setminus 0$ there is an element α, such that any number in $\mathbb{F}_q \setminus 0$ can be written in the form α^n, for some integer n.

Problem 3.10:

Explain why all the steps can be performed by Alic and Bob, with neither knowing the keys $(e_A, d_A), (e_B, d_B)$ of the other person, and Bob not knowing the starting number P.

Explain how Bob can determine the original P that Alice meant to send, but no one else can decipher this, even knowing all three messages that were sent between Alice and Bob.

Solution 3.10:

Let p be an odd prime number

Let $(\mathbb{Z}/p^n\mathbb{Z})^\times$ be the set of numbers less than p^n that are coprime with p^n

For this problem you may assume without proof that in $\mathbb{F}_p \setminus 0$ there is an element α, such that any number x in $\mathbb{F}_p \setminus 0$ can be written in the form $x = \alpha^m$, for some integer m.

Problem 3.11:

(1) Compute how many elements are in $(\mathbb{Z}/p^n\mathbb{Z})^\times$

(2) Show that there exists α in $(\mathbb{Z}/p^n\mathbb{Z})^\times$, such that any element x in $(\mathbb{Z}/p^n\mathbb{Z})^\times$ can be written as $x = \alpha^m$.

Solution 3.11:

Problem 3.12:
Show that in $(\mathbb{Z}/p^n\mathbb{Z})^\times$, any element can be written in the form 2^m, where m is an integer.
Solution 3.12:

UCLA Mathematics Department, Los Angeles, CA 90095-1555, USA.
Email address: fzamora@math.princeton.edu

UCLA Mathematics Department, Box 951555, Los Angeles, CA 90095-1555, USA.
Email address: jmoraga@math.ucla.edu