Some slides for Mini-project in Algebra

Diego Ruano

Department of Mathematical Sciences
Aalborg University
Denmark

25-10-2010

Welcome to the mini project in Algebra. We will use the project description by Christian Thommesen in 2009

Purpose of the project

Among others:

- Using the mathematical theorems and methods studied in the subject Algebra (chapters 1 and 2 in [Lau]) in practice
- Deeper understanding of these mathematical methods. It should help a lot when studying for the exam in January.
- Last but not least: having fun!

Public-key crytography (from Wikipedia)

- The key used to encrypt a message is not the same as the key used to decrypt it.
- Each user has a pair of cryptographic keys-a public key and a private key. The private key is kept secret, while the public key may be widely distributed.
- Messages are encrypted with the recipient's public key and can only be decrypted with the corresponding private key.
- The keys are related mathematically, but the private key cannot feasibly (ie, in actual or projected practice) be derived from the public key.
- The discovery of algorithms that could produce public/private key pairs revolutionized the practice of cryptography beginning in the middle 1970s.

We will consider three cryptosystems:

1. Knapsack (Merkle-Hellman), see Wikipedia or slides from Christian Thommesen (2009) [Click here] and [Click here]
2. ElGamal, see page 160 in [Lau]
3. RSA, see page 24 in [Lau]

Knapsack is very fast and elegant, but was broken in 1982. However, there have been several improvements that have also been broken (in the 80's and 90's). It would be very nice to get a secure improvement.

ElGamal is nowadays used in practice. It has been improved using elliptic curves: it has smaller key sizes and faster operations. New standards are coming.

Project

The first aim of the project is to describe the three cryptosystems above. In particular, for each crytopsystem, the project should contain an example and a brief answer to the following questions:

- How are the public and private keys generated?
- How does the sender encrypt a message?
- How does the receiver decrypt a message?
- How can the receiver be sure that he/she will recover the original message?
- Why cannot an encrypted message be decrypted without the private key?

To answer the previous questions we will use the mathematical theorems and methods studied in the subject Algebra (chapters 1 and 2 in [Lau]). The second aim of the project is to put all this into practice in a series of exercises from [Tho], the exercises in the Problemkatalog (12 problems + Pollard's algorithms).

It is important that you point out which results and methods you have used when doing the computations.

Project groups?

- Each group has to deliver a project (2 copies) on Friday 29/10 at 12:00. One to Lisbeth and one to me.
- The project can be written in Danish or in English.
- We need to find a date for the exam, which will be oral, where the project contents will be discussed.
- The grading is: Bestået/Ikke bestået.

Computations:

You are welcome to use Maple for computations. You should however remember that it is a good idea to understand what the computer is doing.

To get some help type in Maple:

- >?mod
- >?isprime (or nextprime)
- >?ifactor
- >?igcdex

Do not forget that we use &^ to apply the repeated squaring algorithm in Maple.

In the Maple worksheet we did during lecture 5, you can find how to use some of these commands. Remember that we know how to compute μ in a smarter way now. The worksheet is at the course webpage.

Knapsack crytosystem (Merkle-Hellman)

A knapsack problem:

- Consider a knapsack (or rucksack) with volume N
- Consider n objects with volume $e_1, \ldots, e_n$
- Maybe we cannot put everything in the knapsack, but we want to fill it. That is, we want to find $I \subset \{1, \ldots, n\}$ such that
$$\sum_{i \in I} e_i = N$$

Our knapsack problem

Given $e_1, \ldots, e_n \in \mathbb{N}$ and $N \in \mathbb{N}$, find a binary number k with n bits $k = (\lambda_1, \ldots, \lambda_n)$ ($\lambda_i = 0$ means object e_i is not in the knapsack) such that:
$$\sum_{i=1}^{n} \lambda_i e_i = N$$

Our knapsack problem is NP-Complete, but there is an easy case:

$$e_i > \sum_{j=1}^{i-1} e_j, \quad \forall\, i$$

Example: $(e_1, \ldots, e_5) = (2, 3, 7, 15, 31)$ and $N = 24$.
$24 - 15 = 9 \quad \rightarrow e_4$
$9 - 7 = 2 \quad \rightarrow e_3$
$2 - 2 = 0 \quad \rightarrow e_1$
Hence $N = 2 + 7 + 15$ and $k = (1, 0, 1, 1, 0) = 24$.

Message: is binary (0's and 1's). We cut it in blocks of length n. Consider that we send M, a block of length n.

1. Bob chooses an easy knapsack $(e_1, \ldots, e_n)$ and $N \in \mathbb{N}$ such that $N > \sum_{i=1}^{n} e_i$ (why?$\longrightarrow$ unique encryption). He also chooses $w \in \mathbb{N}$ such that $0 < w < N$ and $\gcd(w, N) = 1$ (why?)
2. Bob computes $[w^{-1}]_N$ ($[w][w^{-1}] = [1]$) and $(a_1, \ldots a_n)$, with $0 < a_i < N$ where

   $$a_i \equiv w e_i (\text{mod } N)$$

   Secret Key: $(e_1, \ldots, e_n), N, w, w^{-1}$
   Public Key: $(a_1, \ldots, a_n)$
3. Alice wants to send $M = (M_1 \ldots, M_n) \in (\mathbb{Z}/2\mathbb{Z})^n$. She computes

   $$C = \sum_{i=1}^{n} M_i a_i$$

   and sends it to Bob

4. Bob gets C. He computes $[Cw^{-1}]_N$ because

$$w^{-1}C \equiv \sum_{i=1}^{n} w^{-1}a_iM_i \equiv \sum_{i=1}^{n} e_iM_i \pmod N)$$

   We have $[Cw^{-1}]_N = [\sum M_ie_i]_N$. Note that $\sum M_ie_i \leq \sum e_i < N$, then $0 < \sum M_ie_i < N$ and encryption is unique.
5. Bob uses the easy knapsack to find $(M_1, \ldots, M_n)$ from $\sum M_ie_i$.
6. Eve?, she gets $C = \sum_{i=1}^{n} M_ia_i$, but it is not an easy knapsack.

Example knapsack

- $M = (1, 1, 0, 0, 1)$
- $N = 61$, $w = 17$, $\gcd(17, 61) = 1$
- $w^{-1} \equiv 18 \pmod{61}$
- $a_1 = 17 \cdot 2 \equiv 34 \pmod{61}$
  $a_2 = 17 \cdot 3 \equiv 51 \pmod{61}$
  $a_3 = 17 \cdot 7 \equiv 58 \pmod{61}$
  $a_4 = 17 \cdot 15 \equiv 11 \pmod{61}$
  $a_5 = 17 \cdot 31 \equiv 39 \pmod{61}$
- Public Key=$(34, 51, 58, 11, 39)$, so to encrypt (1,1,0,0,1) we have $34 + 51 + 39 = 124$. Alice sends 124.
- Bob receives 124 and computes $124 \cdot 18 \equiv 36 \pmod{61}$. Then he has an easy knapsack for 36:
  $36 - 31 = 5 \quad \rightarrow e_5$
  $5 - 3 = 2 \quad \rightarrow e_2$
  $2 - 2 = 0 \quad \rightarrow e_1$, and recovers $M = (1, 1, 0, 0, 1)$
- Eve could do: $124 = a_1 + a_2 + a_5 = 34 + 51 + 39$ but this is a difficult knapsack (for large numbers!)

ElGamal

Based on discrete logarithm problem:

Given a prime p and $y, g \in \mathbb{N}$, find x such that

$$y \equiv g^x \pmod{p}$$

1. Alice and Bob choose p, a big prime, and $g \in \mathbb{N}$ s.t. $0 < g < p$ and g has order $p - 1$ in $(\mathbb{Z}/p\mathbb{Z})^*$ (a generator of $(\mathbb{Z}/p\mathbb{Z})^*$)
2. Alice chooses a, with $0 < a < p$ and computes $[g^a]_p$.
   Secret Key=a
   Public Key=$[g^a]_p$
3. Bob chooses b with $0 < b < p$ and computes $[g^b]_p$.
   Secret Key=b
   Public Key=$[g^b]_p$

4. Alice wants to send a message m, $0 < m < p$ to Bob. She sends:
$$\left([g^a]_p, [m(g^b)^a]_p\right)$$
5. Bob gets $([x_1]_p, [x_2]_p)$ and computes
$$[x_2]_p([x_1^b]_p)^{-1} = [mg^{ab}]_p([g^{ab}]_p)^{-1} = [m]_p$$
   and since $m < p$ he can recover m.

To encrypt the message one uses the public key of the receiver and the secret key of the sender.

6. Eve?: she had to compute b from $[g^b]_p$