Public Key Cryptography

Septimiu Crivei

Index

1. Coordinates
2. Some Number Theory
   - The Euclidean Algorithm
   - Congruences Modulo *n*
   - Euler’s Function
   - Repeated Squaring Modular Exponentiation
3. The Story of Alice and Bob
4. Public Key Cryptography
5. Algorithms Complexity
6. Primality and Factorization

Coordinates

- Semester: 5
- Status: optional course
- Hours: 2C+1L
- Course contents:
  1. Notions of algorithm complexity.
  2. Elements of number theory.
  3. Primality testing.
  4. Factorization methods.
  5. Public key cryptography.
  6. Applications.
- Assessment: assignments (50%) and labs (50%)

The Euclidean Algorithm

Division Algorithm

$\forall a, b \in \mathbb{N},\ b \neq 0,\ \exists! q, r \in \mathbb{N}$ such that $a = bq + r$, where $r < b$.

One of the most efficient ways to compute $gcd(a, b)$, also denoted (a, b), is the Euclidean Algorithm.

Example. We have $(1547, 560) = 7$, because:

$$\begin{aligned}
1547 &= 2 \cdot 560 + 427 \\
560 &= 1 \cdot 427 + 133 \\
427 &= 3 \cdot 133 + 28 \\
133 &= 4 \cdot 28 + 21 \\
28 &= 1 \cdot 21 + 7 \\
21 &= 3 \cdot 7
\end{aligned}$$

The Extended Euclidean Algorithm

Theorem

Let $a, b \in \mathbb{N}$ and $d = (a, b)$. Then $\exists u, v \in \mathbb{Z}$: $d = au + bv$.

Corollary

Let $a, b \in \mathbb{N}$. Then $(a, b) = 1 \Leftrightarrow \exists u, v \in \mathbb{Z} : 1 = au + bv$.

Example. We already know that $(1547, 560) = 7$. Then:

$$\begin{aligned}
7 &= 28 - 1 \cdot 21 \\
&= 28 - 1 \cdot (133 - 4 \cdot 28) \\
&= 5 \cdot 28 - 1 \cdot 133 \\
&= 5 \cdot (427 - 3 \cdot 133) - 1 \cdot 133 \\
&= 5 \cdot 427 - 16 \cdot 133 \\
&= 5 \cdot 427 - 16 \cdot (560 - 1 \cdot 427) \\
&= 21 \cdot 427 - 16 \cdot 560 \\
&= 21 \cdot (1547 - 2 \cdot 560) - 16 \cdot 560 \\
&= 21 \cdot 1547 - 58 \cdot 560.
\end{aligned}$$

Congruences Modulo n

Definition

Let $n \in \mathbb{N}$, $a, b \in \mathbb{Z}$. We have:

$$a \equiv b \pmod{n} \Leftrightarrow n | a - b.$$

If $n \neq 0$, then $a \equiv b \pmod{n} \Leftrightarrow$ a and b give the same remainder when divided by n.

Theorem

(i) $(\mathbb{Z}_n, +, \cdot)$ is a ring, where

$$\begin{aligned} \widehat{a} + \widehat{b} &= \widehat{a+b} \\ \widehat{a} \cdot \widehat{b} &= \widehat{a \cdot b}. \end{aligned}$$

(ii) $\widehat{0} \neq \widehat{a}$ is invertible in $\mathbb{Z}_n \Leftrightarrow (a, n) = 1 \Leftrightarrow \exists u, v \in \mathbb{Z}$ such that $au + nv = 1$. In this case $\widehat{a}^{-1} = \widehat{u}$.
(iii) $(\mathbb{Z}_n, +, \cdot)$ is a field $\Leftrightarrow$ n is prime.

Euler's Function

Definition

The function $\varphi : \mathbb{N}^* \to \mathbb{N}^*$,

$$\varphi(n) = |\{k \in \mathbb{N} \mid k < n \textit{ and } (k, n) = 1\}|$$

is called Euler's function.

Theorem

(i) If $(m, n) = 1$*, then* $\varphi(mn) = \varphi(m)\varphi(n)$.
(ii) If p *is prime, then* $\varphi(p) = p - 1$.
(iii) If $n = p^k$ *for some prime* p*, then* $\varphi(n) = n\left(1 - \frac{1}{p}\right)$.
(iv) If $n = p_1^{k_1} \ldots p_j^{k_j}$ *for some primes* $p_1, \ldots, p_j$*, then*

$$\varphi(n) = n\left(1 - \frac{1}{p_1}\right) \ldots \left(1 - \frac{1}{p_j}\right).$$

One can easily compute $\varphi(n)$ GIVEN the factorization of n.

Repeated Squaring Modular Exponentiation

Let us compute $b^k \bmod n$, where $b, k \in \mathbb{N}$ are large. Write k in binary, say $k = \sum_{i=0}^{t} k_i 2^i$. We have

$$b^k = \prod_{i=0}^{t} b^{k_i 2^i} = (b^{2^0})^{k_0} (b^{2^1})^{k_1} \dots (b^{2^t})^{k_t}.$$

Example. Let us compute $42^{51} \bmod 73$.
We have $51 = 2^0 + 2^1 + 2^4 + 2^5$. Compute modulo 73:

$$\begin{aligned}
42^{(2^0)} &= 42, \\
42^{(2^1)} &= 42^{(2^0)} \cdot 42^{(2^0)} = 42 \cdot 42 = 12, \\
42^{(2^2)} &= 42^{(2^1)} \cdot 42^{(2^1)} = 12 \cdot 12 = 71, \\
42^{(2^3)} &= 42^{(2^2)} \cdot 42^{(2^2)} = 71 \cdot 71 = 4, \\
42^{(2^4)} &= 42^{(2^3)} \cdot 42^{(2^3)} = 4 \cdot 4 = 16, \\
42^{(2^5)} &= 42^{(2^4)} \cdot 42^{(2^4)} = 16 \cdot 16 = 37.
\end{aligned}$$

Then $42^{51} = 42^{2^0+2^1+2^4+2^5} = 42 \cdot 12 \cdot 16 \cdot 37 = 17 \pmod{73}$.

The Story of Alice and Bob

From "The Alice and Bob After Dinner Speech" given at the Zurich Seminar, April 1984, by John Gordon:

> *So let's talk about coding theory [*note: actually, cryptography*]. There are perhaps some of you here tonight who are not experts in coding theory, but rather have been dragged here kicking and screaming. So I thought it would be a good idea if I gave you a sort of instant, five minute graduate course in coding theory.*
>
> *Coding theorists are concerned with two things. Firstly and most importantly they are concerned with the private lives of two people called Alice and Bob. In theory papers, whenever a coding theorist wants to describe a transaction between two parties he doesn't call them A and B. No. For some longstanding traditional reason he calls them Alice and Bob.*
>
> *Now there are hundreds of papers written about Alice and Bob. Over the years Alice and Bob have tried to defraud insurance companies, they've played poker for high stakes by mail, and they've exchanged secret messages over tapped telephones.*

If we put together all the little details from here and there, snippets from lots of papers, we get a fascinating picture of their lives. This may be the first time a definitive biography of Alice and Bob has been given.

In papers written by American authors Bob is frequently selling stock to speculators. From the number of stock market deals Bob is involved in we infer that he is probably a stockbroker. However from his concern about eavesdropping he is probably active in some subversive enterprise as well. And from the number of times Alice tries to buy stock from him we infer she is probably a speculator. Alice is also concerned that her financial dealings with Bob are not brought to the attention of her husband. So Bob is a subversive stockbroker and Alice is a two-timing speculator.

But Alice has a number of serious problems. She and Bob only get to talk by telephone or by electronic mail. In the country where they live the telephone service is very expensive. And Alice and Bob are cheapskates. So the first thing Alice must do is minimize the cost of the phone call.

The telephone is also very noisy. Often the interference is so bad that Alice and Bob can hardly hear each other. On top of that Alice and Bob have very powerful enemies. One of their enemies is the Tax Authority. Another is the Secret Police. This is a pity, since their favorite topics of discussion are tax frauds and overthrowing the government.

These enemies have almost unlimited resources. They always listen in to telephone conversations between Alice and Bob. And these enemies are very sneaky. One of their favorite tricks is to telephone Alice and pretend to be Bob.

Well, you think, so all Alice has to do is listen very carefully to be sure she recognizes Bob's voice. But no. You see Alice has never met Bob. She has no idea what his voice sounds like.

So you see Alice has a whole bunch of problems to face. Oh yes, and there is one more thing I forgot so say - Alice doesn't trust Bob. We don't know why she doesn't trust him, but at some time in the past there has been an incident.

Now most people in Alice's position would give up. Not Alice. She has courage which can only be described as awesome. Against all odds, over a noisy telephone line, tapped by the tax authorities and the secret police, Alice will happily attempt, with someone she doesn't trust, whom she cannot hear clearly, and who is probably someone else, to fiddle her tax returns and to organize a coup d'etat, while at the same time minimizing the cost of the phone call.

A coding theorist is someone who doesn't think Alice is crazy...

Public Key Cryptography

Private key cryptosystem

- characteristic for classical cryptography
- once the encryption key was known, the decryption key could be easily recovered, hence the message deciphered.

Public key cryptosystem

- 1976: Diffie and Helman
- Idea: given the encryption key, one cannot determine the decryption key in a "reasonable" time.
  Hence $f : \{plaintext\} \to \{ciphertext\}$ can be easily computed once the encryption key K_E is known, but f^{-1} is very difficult (impossible) to be computed in practice without knowing the decryption key K_D.

The story of Alice, Bob and their friends begins...

RSA - Rivest, Shamir, Adleman (1977)

1. Key generation. Alice creates a public key and a private key.

1.1. Generates 2 random large distinct primes p, q.

1.2. Computes $n = pq$ and $\varphi(n) = (p-1)(q-1)$.

1.3. Randomly selects $1 < e < \varphi(n)$ with $gcd(e, \varphi(n)) = 1$.

1.4. Computes $d = e^{-1} \ mod \ \varphi(n)$.

1.5. Alice's public key is $K_E = (n, e)$; her private key is $K_D = d$.

2. Encryption. Bob sends an encrypted message to Alice.

2.1. Gets Alice's public key $K_E = (n, e)$.

2.2. Represents the message as a number m between 0 and $n-1$.

2.3. Computes $c = m^e \ mod \ n$.

2.4. Sends the ciphertext c to Alice.

3. Decryption. Alice decrypts the message from Bob.

3.1 Alice uses the private key $K_D = d$ to get the message $m = c^d \ mod \ n$.

RSA example

Example. General setting:

- Use the RSA cryptosystem.
- Use a 27-letters alphabet for plaintext and ciphertext: _ (blank) with numerical equivalent 0 and letters $A - Z$ (the English alphabet) with numerical equivalents 1-26.

_	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*	*K*	*L*	*M*	*N*	*O*	*P*	*Q*	*R*	*S*	*T*	*U*	*V*	*W*	*X*	*Y*	*Z*
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26

- Plaintext message units are blocks of k letters, whereas ciphertext message units are blocks of l letters. The plaintext is completed with blanks, when necessary.
- We must have $27^k < n < 27^l$.

RSA example - encryption

Let $k = 2$, $l = 3$, $K_E = (n, e) = (1643, 67)$. We have $27^2 < 1643 < 27^3$.

- Plaintext: algebra
- Split the plaintext: al /ge /br /a_
- Write the numerical equivalents: 39 194 72 27
  (al $\mapsto \boxed{1} \cdot 27 + \boxed{12} = 39$, ge $\mapsto \boxed{7} \cdot 27 + \boxed{5} = 194$,
  br $\mapsto \boxed{2} \cdot 27 + \boxed{18} = 72$, a_ $\mapsto \boxed{1} \cdot 27 + \boxed{0} = 27$)
- Encrypt ($m^e \bmod n$): 1428 498 919 1503
  ($39^{67} \bmod 1643 = \cdots = 1428$, $194^{67} \bmod 1643 = \cdots = 498$,
  $72^{67} \bmod 1643 = \cdots = 919$, $27^{67} \bmod 1643 = \cdots = 1503$)
- Write the literal equivalents: AYX _RL AGA BAR
  ($1428 = \boxed{1} \cdot 27^2 + \boxed{25} \cdot 27 + \boxed{24} \mapsto$ AYX,
  $498 = \boxed{0} \cdot 27^2 + \boxed{18} \cdot 27 + \boxed{12} \mapsto$ _RL,
  $919 = \boxed{1} \cdot 27^2 + \boxed{7} \cdot 27 + \boxed{1} \mapsto$ AGA,
  $1503 = \boxed{2} \cdot 27^2 + \boxed{1} \cdot 27 + \boxed{18} \mapsto$ BAR)
- Ciphertext: AYX_RLAGABAR

RSA example - encryption (cont.)

Some details:

We compute 39^{67} mod 1643 by repetead squaring modular exponentiation.
We have $67 = 2^0 + 2^1 + 2^6$. Compute modulo 1643:

$$\begin{aligned}
39^{(2^0)} &= 39, \\
39^{(2^1)} &= 39^{(2^0)} \cdot 39^{(2^0)} = 39 \cdot 39 = 1521, \\
39^{(2^2)} &= 39^{(2^1)} \cdot 39^{(2^1)} = 1521 \cdot 1521 = 97, \\
39^{(2^3)} &= 39^{(2^2)} \cdot 39^{(2^2)} = 97 \cdot 97 = 1194, \\
39^{(2^4)} &= 39^{(2^3)} \cdot 39^{(2^3)} = 1194 \cdot 1194 = 1155, \\
39^{(2^5)} &= 39^{(2^4)} \cdot 39^{(2^4)} = 1155 \cdot 1155 = 1552, \\
39^{(2^6)} &= 39^{(2^5)} \cdot 39^{(2^5)} = 1552 \cdot 1552 = 66.
\end{aligned}$$

Then $39^{67} = 39^{2^0+2^1+2^6} = 39 \cdot 1521 \cdot 66 = 1428 \pmod{1643}$.

RSA example - decryption

We have $n = 1643 = 31 \cdot 53$, hence $\varphi(n) = 30 \cdot 52 = 1560$.
We have $K_D = d = e^{-1} \bmod \varphi(n) = 67^{-1} \bmod 1560 = \cdots = 163$.

- Ciphertext: AYX_RLAGABAR
- Split the ciphertext: AYX /_RL /AGA /BAR
- Write the numerical equivalents: 1428 498 919 1503
  (AYX $\mapsto 1428 = \boxed{1} \cdot 27^2 + \boxed{25} \cdot 27 + \boxed{24}$,
  _RL $\mapsto 498 = \boxed{0} \cdot 27^2 + \boxed{18} \cdot 27 + \boxed{12}$,
  AGA $\mapsto 919 = \boxed{1} \cdot 27^2 + \boxed{7} \cdot 27 + \boxed{1}$,
  BAR $\mapsto 1503 = \boxed{2} \cdot 27^2 + \boxed{1} \cdot 27 + \boxed{18}$)
- Decryption ($c^d \bmod n$): 39 194 72 27
  ($1428^{163} \bmod 1643 = \cdots = 39$, $498^{163} \bmod 1643 = \cdots = 194$,
  $919^{163} \bmod 1643 = \cdots = 72$, $1503^{163} \bmod 1643 = \cdots = 27$)
- Write the literal equivalents: al ge br a_
  ($39 = \boxed{1} \cdot 27 + \boxed{12} \mapsto$ al, $194 = \boxed{7} \cdot 27 + \boxed{5} \mapsto$ ge,
  $72 = \boxed{2} \cdot 27 + \boxed{18} \mapsto$ br, $27 = \boxed{1} \cdot 27 + \boxed{0} \mapsto$ a_)
- Plaintext: algebra

RSA example - decryption (cont.)

Some details:

We compute $67^{-1} \bmod 1560 = 163$ by the extended Euclidean algorithm.

$$\begin{aligned}
1560 &= 23 \cdot 67 + 19 \\
67 &= 3 \cdot 19 + 10 \\
19 &= 1 \cdot 10 + 9 \\
10 &= 1 \cdot 9 + 1 \\
9 &= 9 \cdot 1
\end{aligned}$$

Then $(1560, 67) = 1$, hence there exists $67^{-1} \bmod 1560$.
We have:

$$\begin{aligned}
1 &= 10 - 1 \cdot 9 = 10 - 1 \cdot (19 - 1 \cdot 10) = 2 \cdot 10 - 1 \cdot 19 \\
&= 2 \cdot (67 - 3 \cdot 19) - 1 \cdot 19 = 2 \cdot 67 - 7 \cdot 19 \\
&= 2 \cdot 67 - 7 \cdot (1560 - 23 \cdot 67) = 163 \cdot 67 - 7 \cdot 1560.
\end{aligned}$$

hence $67^{-1} \bmod 1560 = 163$.

RSA example - decryption (cont.)

Some details:

We compute $1428^{163} \bmod 1643$ by repetead squaring modular exponentiation.
We have $163 = 2^0 + 2^1 + 2^5 + 2^7$. Compute modulo 1643:

$$\begin{aligned}
1428^{(2^0)} &= 1428,\\
1428^{(2^1)} &= 1428^{(2^0)} \cdot 1428^{(2^0)} = 1428 \cdot 1428 = 221,\\
1428^{(2^2)} &= 1428^{(2^1)} \cdot 1428^{(2^1)} = 221 \cdot 221 = 1194,\\
1428^{(2^3)} &= 1428^{(2^2)} \cdot 1428^{(2^2)} = 1194 \cdot 1194 = 1155,\\
1428^{(2^4)} &= 1428^{(2^3)} \cdot 1428^{(2^3)} = 1155 \cdot 1155 = 1552,\\
1428^{(2^5)} &= 1428^{(2^4)} \cdot 1428^{(2^4)} = 1552 \cdot 1552 = 66,\\
1428^{(2^6)} &= 1428^{(2^5)} \cdot 1428^{(2^5)} = 66 \cdot 66 = 1070,\\
1428^{(2^7)} &= 1428^{(2^5)} \cdot 1428^{(2^5)} = 1070 \cdot 1070 = 1372.
\end{aligned}$$

Then $1428^{163} = 1428^{2^0+2^1+2^5+2^7} = 1428 \cdot 221 \cdot 66 \cdot 1372 = 39 \pmod{1643}$.

Algorithms Complexity

Definition

- *Polynomial-time algorithm*: an algorithm of complexity $O(N^k)$ bit operations, where N is the size of the input (that is, *log n* if n is the input) and k is a constant.
- *Exponential-time algorithm*: any non-polynomial-time algorithm.

Algorithm	Complexity	No. operations for $N = 10^6$	Time needed at 10^6 operations / sec.
constant	$O(1)$	1	1 μ sec.
linear	$O(N)$	10^6	1 sec.
quadratic	$O(N^2)$	10^{12}	1,6 days
cubic	$O(N^3)$	10^{18}	32000 years
exponential	$O(2^N)$	10^{301030}	10^{301006}. age of universe

- Polynomial-time algorithms ↦ efficient.
- Exponential-time algorithms ↦ inefficient.

Primality and Factorization

- **Primality**
  - The largest known prime: $2^{82589933} - 1$ (December 7, 2018). It has 24862048 digits.
  - Electronic Frontier Foundation: $ 100,000 award for the discovery of the first 10 million digit prime.

Problem

Is a given large number prime?

  - *probabilistic* polynomial-time algorithms
  - AKS test - *deterministic* polynomial-time (2003)

- **Factorization**

Problem

Find a prime factor of a large composite number n.

  - exponential-time algorithms

Selective Bibliography

- M. Cozzens, S.J. Miller, *The Mathematics of Encryption: An Elementary Introduction*, American Mathematical Society, 2013.
- A.J. Menezes, P.C. van Oorschot, S.A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1997. [http://www.cacr.math.uwaterloo.ca/hac]
- C. Paar, J. Pelzl, *Understanding Cryptography*, Springer, 2009.