


Chapter 3

Lecture 3 - Preliminaries

Overheads and notes

You can find all sorts of stuff looking in

http://www.comp.nus.edu.sg/~cs3235/2003-semesterI/

Question box

If you have any questions, feel free to place them in the question box...

Or stick your hand up...

Or...

Last session

- Finish context
- Math preliminaries
  - XOR
  - Logarithms
  - Fields and groups

Recap - exclusive-or

Law XOR-1:
The cryptographer's favorite function is *Exclusive-Or*.

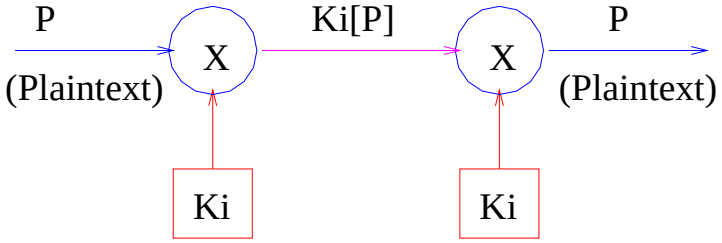
Message	A	B	C
m	0 1 0 0 0 0 0 1	0 1 0 0 0 0 1 0	0 1 0 0 0 0 1 1 . . .
Key= k	0 0 0 1 0 0 1 1	0 1 1 0 0 1 0 1	0 0 1 1 1 0 0 1 . . .
$K(m) = m \oplus k$	0 1 0 1 0 0 1 0	0 0 1 0 0 1 1 1	0 1 1 1 1 0 1 0 . . .
$K(m)$	R	'	z

Exclusive-Or

$K(m)$	R	'	z
	0 1 0 1 0 0 1 0	0 0 1 0 0 1 1 1	0 1 1 1 1 0 1 0 . . .
Key= k	0 0 0 1 0 0 1 1	0 1 1 0 0 1 0 1	0 0 1 1 1 0 0 1 . . .
$m = K(m) \oplus k$	0 1 0 0 0 0 0 1	0 1 0 0 0 0 1 0	0 1 0 0 0 0 1 1 . . .
Message	A	B	C

If the bit-stream for the key k is random, and not known to an eavesdropper, then this is the most secure system. It is known as a one-time-pad.

Another diagram



(Compare with previous representations).

Logarithms

Law LOG-1:
The cryptographer's favorite logarithm is *log base 2*.

- ✔ $y = \log_b x$ is the same as $b^y = x$
- ✔ $b^{(\log_b x)} = x$
- ✔ Logarithm is inverse of exponential.

Groups

- A *group* is
  - a set of *group elements* with
  - a *binary operation*

Law GROUP-1:
The cryptographer's favorite group is the *integers mod n*, Z_n.

Fields

- A field has two operations
  - $+$, with elements forming a commutative group.
  - $*$, with elements$\backslash 0$ forming another group,

Law FIELD-1:
The cryptographer's favorite field is the *integers mod p*, denoted Z_p, where *p* is a prime number.

Law FIELD-2:
The cryptographer's *other* favorite field is $GF(2^n)$.

This session

- Math preliminaries
  - Fermat's little theorem
  - Euler

This session

- Math preliminaries
  - Fermat's little theorem
  - Euler

Fermat's theorem

Theorem (Fermat): If p is a prime and a is any non-zero number less than p, then

$$a^{p-1} \bmod p = 1$$

Fermat's theorem, $p = 13$

p	a	a^1	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}
13	2	2	4	8	3	6	12	11	9	5	10	7	1
13	3	3	9	1	3	9	1	3	9	1	3	9	1
13	4	4	3	12	9	10	1	4	3	12	9	10	1
13	5	5	12	8	1	5	12	8	1	5	12	8	1
13	6	6	10	8	9	2	12	7	3	5	4	11	1
13	7	7	10	5	9	11	12	6	3	8	4	2	1
13	8	8	12	5	1	8	12	5	1	8	12	5	1
13	9	9	3	1	9	3	1	9	3	1	9	3	1
13	10	10	9	12	3	4	1	10	9	12	3	4	1
13	11	11	4	5	3	7	12	2	9	8	10	6	1
13	12	12	1	12	1	12	1	12	1	12	1	12	1

Fermat's theorem, $p = 13$

- ✔ Lengths of runs are always numbers that divide evenly into 12
- ✔ A value of a for which the whole row is needed is called a *generator*. 2, 6, 7, and 11 are generators.

An interesting observation..

Because a to a power mod p always starts repeating after the power reaches $p-1$, you can do this:

$$a^x \bmod p = a^{x \bmod (p-1)} \bmod p.$$

Thus modulo p in the expression requires modulo $p-1$ in the exponent. For $p = 13$ as above, then

$$a^{29} \bmod 13 = a^{29 \bmod 12} \bmod 13 = a^5 \bmod 13.$$


Another example

result $= 7^{1215} \bmod 13$


Another example

result=
6224702750673227370465564559079792689062398648329219130902078771092 4
8699107274058706519890781017383899497826793481300967770892782660131 3
5577736536148404478380085122281739226134142137076240050702683456450 1
6147888185801623358181550772919006073386381098582099841775377667037 2
8681473967012031571239691400018482234035235590645515566753410247396 4
5354137741258367626070635933104840329377905370464877106976413186542 2
6229950528055758428057418580269421329980228017932549456062894894073 9
3444822846491511971411686989595879473202428574269018023244940256710 1
0508311496735633429580921945571119113124697462717311124279255445332 1
1650491453007724199618935729850860520678012078988083552522234194051 4
5855673208684204238889320915704079986487190106499123086028865754587 8
5483803190210993511026450389154414587258074783062229406697804705969 8
0888822497677940491279201763309541131855593877680081677862469580790 9\
4970578719259627712779630348778181410614737537090462719599558908727 6
8469943 mod 13 = 5


How did I work that out?

I used bc

An arbitrary precision calculator language


Another example

result $= 7^{1215} \bmod 13$


Another example

$$\begin{aligned} \text{result} &= 7^{1215} \bmod 13 \\ &= 7^{1215} \bmod {}^{12} \bmod 13 \end{aligned}$$


Another example

$$\begin{aligned} \text{result} &= 7^{1215} \bmod 13 \\ &= 7^{1215} \bmod {}^{12} \bmod 13 \\ &= 7^{3} \bmod 13 \end{aligned}$$


Another example

$$\begin{aligned} \text{result} &= 7^{1215} \bmod 13 \\ &= 7^{1215} \bmod {}^{12} \bmod 13 \\ &= 7^{3} \bmod 13 \\ &= 343 \bmod 13 \end{aligned}$$


Another example

$$\begin{aligned} \text{result} &= 7^{1215} \bmod 13 \\ &= 7^{1215} \bmod {}^{12} \bmod 13 \\ &= 7^{3} \bmod 13 \\ &= 343 \bmod 13 \\ &= 5 \end{aligned}$$

Summary

We can do BIG NUMBER maths without calculating big numbers.

This session

- Math preliminaries
  - Fermat's little theorem
  - Euler

Euler

The Swiss mathematician Leonhard Euler (1707-1783) discovered a generalization of Fermat's Theorem which will later be useful in the discussion of the RSA cryptosystem.

Euler's theorem

Theorem (Euler): If n is any positive integer and a is any positive integer less than n with no divisors in common with n, then

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

where $\phi(n)$ is the *Euler phi function*:

$$\phi(n) = n(1 - 1/p_1) \ldots (1 - 1/p_m),$$

and $p_1, \ldots, p_m$ are all the prime numbers that divide evenly into n, including n itself in case it is a prime.

Special case 1

✔ If n is a prime, then using the formula,

$$\phi(n) = n(1 - 1/n) = n(\frac{n-1}{n}) = n - 1$$

Fermat's result is a special case of Euler's.

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

Special case 2

✔ Another special case needed for RSA comes when the modulus is a product of two primes: $n = pq$. Then

$$\phi(n) = n(1 - 1/p)(1 - 1/q) = (p-1)(q-1)$$

Special case 2

$$a^{(p-1)(q-1)} \bmod pq = 1$$

- assuming a has no divisors in common with pq
- and p and q are primes

Euler: $n = 15$ and $\phi(n) = 8$

a^1	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}
2	4	8	1	2	4	8	1	2	4	8	1	2	4
3	9	12	6	3	9	12	6	3	9	12	6	3	9
4	1	4	1	4	1	4	1	4	1	4	1	4	1
5	10	5	10	5	10	5	10	5	10	5	10	5	10
6	6	6	6	6	6	6	6	6	6	6	6	6	6
7	4	13	1	7	4	13	1	7	4	13	1	7	4
8	4	2	1	8	4	2	1	8	4	2	1	8	4
9	6	9	6	9	6	9	6	9	6	9	6	9	6
10	10	10	10	10	10	10	10	10	10	10	10	10	10
11	1	11	1	11	1	11	1	11	1	11	1	11	1
12	9	3	6	12	9	3	6	12	9	3	6	12	9
13	4	7	1	13	4	7	1	13	4	7	1	13	4
14	1	14	1	14	1	14	1	14	1	14	1	14	1

Table

Table illustrates Euler's theorem for $n = 15 = 3 \cdot 5$, with

$$\phi(15) = 15 \cdot (1 - 1/3) \cdot (1 - 1/5) = (3 - 1) \cdot (5 - 1) = 8$$

Notice here that a 1 is reached when the power is 8, but only for numbers with no divisors in common with 15.

For other base numbers, the value never gets to 1.

Euler

Arithmetic in the exponent is taken mod $\phi(n)$, so that, if a has no divisors in common with n,

$$a^x \bmod n = a^{x \bmod \phi(n)} \bmod n.$$

If $n = 15$ as above, then $\phi(n) = 8$, and if neither 3 nor 5 divides evenly into a, then $\phi(n) = 8$. Thus for example,

$$a^{28} \bmod 15 = a^{28 \bmod 8} \bmod 15 = a^4 \bmod 15.$$

Before we leave Euler...

We are interested in...

- ✔ Large prime numbers (p, q)
- ✔ Their product $n = pq$
- ✔ The Euler phi function $\phi(n) = (p - 1)(q - 1)$

Before we leave Euler...

- ✔ In a similar fashion to before we can do BIG number arithmetic easily
- ✔ Consider also the ease of multiplying, and difficulty of factoring...

Before we leave Euler...

29*37=?

The Euclidean algorithm

✔ Multiplicative inverse is not intuitive and requires some theory to compute.

✔ a^{-1} can be computed efficiently using *the extended Euclidean algorithm*

Law GCD-1:
The cryptographer's first and oldest favorite algorithm is the *extended Euclidean algorithm*, which computes the greatest common divisor of two positive integers *a* and *b* and also supplies integers *x* and *y* such that *x*a + y*b = gcd(a, b)*.

Finding GCD

- For the gcd of 819 and 462,
  - factor the numbers as:
    - $819 = 3 \cdot 3 \cdot 7 \cdot 13$
    - $462 = 2 \cdot 3 \cdot 7 \cdot 11$
  - gcd is $21 = 3 \cdot 7$

But there is no efficient algorithm to factor integers.

The Euclidean algorithm

1. Repeatedly divide the **`larger`** one by the **`smaller`**, and
2. Write **`larger = smaller * quotient + remainder`**
3. Repeat using the two numbers "**`smaller`**" and "**`remainder`**".
4. When you get a 0 **`remainder`**, then you have the **`gcd`** of the original two numbers.

Example

$$
\begin{array}{rcrcrcrl}
819 & = & 462 & \cdot & 1 & + & 357 & \text{(Step 0)} \\
462 & = & 357 & \cdot & 1 & + & 105 & \text{(Step 1)} \\
357 & = & 105 & \cdot & 3 & + & 42 & \text{(Step 2)} \\
105 & = & 42 & \cdot & 2 & + & 21 & \text{(Step 3, so GCD = 21)} \\
42 & = & 21 & \cdot & 2 & + & 0 & \text{(Step 4)}
\end{array}
$$

The extended Euclidean algorithm

Given the two positive integers 819 and 462, the extended Euclidean algorithm finds unique integers a and b so that

$$a \cdot 819 + b \cdot 462 = \gcd(819, 462) = 21$$

In this case,

$$(-9) \cdot 819 + 16 \cdot 462 = 21$$

(See notes...)

✘ How does this give us a mechanism to calculate the multiplicative inverse of an element?

The extended Euclidean algorithm

$$x * a + y * p = \gcd(x, y)$$

Now - if p is a prime, then $\gcd(x, y) = 1$, and so

$$x * a + y * p = 1$$

In the field Z_p, this indicates that $x * a = 1$, and so $x = a^{-1}$.

The extended Euclidean algorithm has given us a mechanism to calculate the multiplicative inverse of an element.

Fast integer exponentiation

Law EXP-1:
Many cryptosystems in modern cryptography depend on a fast algorithm to perform integer exponentiation.

Examples in notes... not so important, just nice to know it can be done.

Back to primes

For 2500 years mathematicians studied prime numbers just because they were interesting, without any idea they would have practical applications. Possible real-world uses:

1. Sometimes... a prime number of ball bearings arranged in a bearing, to cut down on periodic wear (also gear teeth).

2. Possibly... the 13 and 17-year periodic emergence of cicadas may be due to coevolution with predators (that lost and became extinct).

Since 1976

Now finally, in cryptography, prime numbers have come into their own.

Law PRIME-1:
A source of large random prime integers is an essential part of many current cryptosystems.

Checking for primes

- ✔ It is hard to check that an integer is "certainly" prime, but...
- ✔ It is easy to check that an integer is "probably" prime.
- ✔ Tests to check if a number is probably prime are called *pseudo-prime* tests.

Prime check

- ✔ Start with a property of a prime number, such as Fermat's Theorem, mentioned in the previous chapter
- ✔ if p is a prime and a is any non-zero number less than p, then $a^{p-1} \bmod p = 1$.
- ✔ If one can find a number a for which Fermat's Theorem does not hold, then the number p in the theorem is *definitely not a prime*.
- ✔ If the theorem holds, then p is called *a pseudo-prime with respect to* a, and it might actually be a prime.

Prime check

So the simplest possible pseudo-prime test would just take a small value of a, say 2 or 3, and check if Fermat's Theorem is true.

Simple Pseudo-prime Test: If a very large random integer p (100 decimal digits or more) is not divisible by a small prime, and if $3^{p-1} \bmod p = 1$, then the number is prime except for a vanishingly small probability, which one can ignore.

Prime check - 1105,1729

- ✔ One could just repeat the test for other integers besides 3 as the base, but unfortunately there are non-primes (called *Carmichael numbers*) that satisfy Fermat's theorem for all values of a even though they are not prime.
- ✔ Chances of a mistake less than 10^{-41}, in practice use better tests

Law PRIME-2:
Just one simple pseudo-prime test is enough to test that a very large random integer is probably prime.

Summary of topics

- ✔ We can do BIG arithmetic in these fields
- ✔ We can do fast exponentiation and modulo arithmetic
- ✔ We can check for primes