Cryptography & Encryption:6G7Z1011:All Slides

Keith Yates

March 19, 2019

Contents

Cryptography & Encryption:6G7Z1011 : Introduction

Keith Yates

March 19, 2019

6G7Z1011: Introduction

This is a MSc unit, Keith Yates is responsible for all of it (K.Yates@mmu.ac.uk, phone 1521, room E139). The unit is assessed as follows:

1. Course work
2. Exam 3 hours (in a lab)

Books

We cover the core ideas in encryption and there are lots of good books that cover the material:

1. Understanding Cryptography: A Textbook for Students and Practioners, Publisher:Springer, Authors: Paar and Pelzl.
2. Introduction to Cryptography, Publisher: Springer, Authors: Buchmann.

The Goals of the Unit

The main goals of the unit are (at a high level, not mentioning a specific algorithm) the following:

1. To understand from a theoretical and a computational view point the main algorithms used in encryption today.
2. To be able to code encryption algorithms.
3. To be able to assess the strengths and weaknesses of a particular algorithmn.

Alice, Bob and Eve

I introduce the main protoganists: Alice, Bob and Eve (they appeared in an early paper on cryptography and have been associated with the subject ever since). Alice and Bob wish to communicate with each other; however Eve (as in eavesdropper) is a malicious party who wishes to disrupt proceedings and cause trouble.

The Main Goal of the Unit

The unit concentrates on finding ways that Alice and Bob can communicate in a manner such that Eve cannot understand what Alice and Bob are communicating.

Cryptography and Encryption

What problems/tasks does Cryptography and Encryption solve? Essentially:

1. Privacy: It allows two people to communicate securely; that is, no one else can understand the message.
2. Integrity: It stops data tampering.
3. Non-repudiation: It stops one party denying a certain event/conversation took place.

Converting between Types

Most of the encryption algorithms we develop are based on ideas
from number theory. This means we need to convert our messages into a (typically large) number. There are various ways we can do this

Word	Characters	ASCII	Binary
Bed	'B' 'e' 'd'	$66, 101, 100$	$01000010, 01100101, 011000100$

(0.1)

So Bed $= 010000100110010101100100$

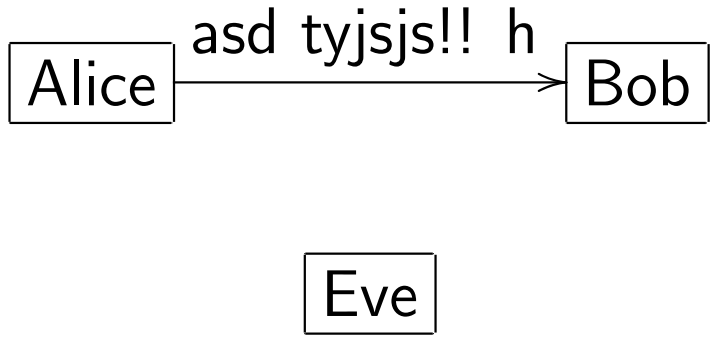
Privacy

Alice and Bob are in love; Alice states 'I love you'

Eve is none too happy (she is married to Bob).

Privacy

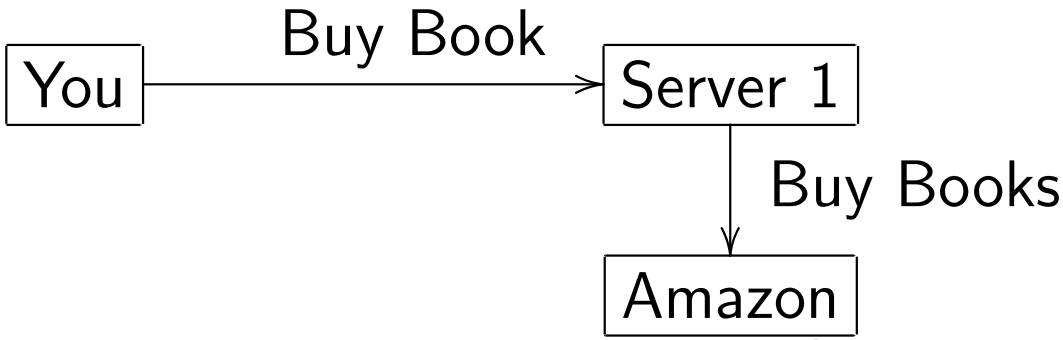
Encryption allows two users to communicate securely. Alice and Bob are in love; Alice states ‘asd tyjsjs!! h’



If Bob has a mechanism for turning ‘asd tyjsjs!! h’ into ‘I love you’ and Eve does not know the mechanism then Eve is none the wiser, and Bob has a marriage and a mistress.

Integrity

Encryption allows a user to check if their data has been tampered with. To be explicit, consider a scenario where you send data over the internet to (for example) Amazon.



If Server 1 modifies the data (data will be past through many servers before reaching Amazon) then you and Amazon need to know a problem has arose. Whilst a server may alter the data (there is nothing you can do about that) encryption techniques can tell you the data has been altered; this is very important it stops data tampering.

Non-repudiation

Encryption techniques allow non-repudiation. By this I mean if two parties agree on some business transaction and the deal falls through. For example, suppose party A claims the transaction fell through because a piece of paperwork from B did not arrive on time, then non-repudiation is a technique by which B could show they had sent the document.

First Step

Encryption requires a fair bit of mathematics. In fact at an advanced level it requires large parts of number theory. For now, I introduce:

1. the natural numbers $\mathbb{N} = \{1, 2, 3 \ldots\}$
2. the integers $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2 \ldots\}$
3. the prime numbers $\mathbb{P} = \{1, 2, 3, 4, 7, 11, \ldots\}$

Rules of Engagement

1. Note cryptography is not about 'hiding'information (that discipline is called steganography).
2. All the data is available; when Alice and Bob communicate all the data they send to each other will be available to Eve (recall she is an eavesdropper). The problem Eve faces is making sense of the data.

More ideas from Mathematics

Definition

A *function* is written

$$f : X \to Y, \quad x \mapsto f(x). \tag{0.2}$$

It is to be read: the function f assigns to every element of the set X an element of the set Y. The rule that does this is the function f.

The concept is very simple, consider a function g that simply adds one to every integer

$$g : \mathbb{Z} \to \mathbb{Z}, \quad x \mapsto x + 1. \tag{0.3}$$

so $g(3) = 4$

More Examples of Functions

Consider the function h that multiplys two integers together, add three and then squares everything.

$$h : \mathbb{Z} \times \mathbb{Z} \to \mathbb{Z}, \quad (a, b) \mapsto (ab + 3)^2 \tag{0.4}$$

The function i takes three integers, multiplies the first integer by 2, the second integer by 3, the third integer by 4; add the lot together and then cubes.

$$i : \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z} \to \mathbb{Z}, \quad (a, b, c) \mapsto (2a + 3b + 4c)^3 \tag{0.5}$$

The point: as our functions become more complicated, the equation form is easier to read.

One More Example

Another example:

$$f : K \times P \to E, \quad (k, p) \mapsto f(k, p) \tag{0.6}$$

f will be an encrypting function; here

1. K is the set of Keys.
2. P is a set of Plaintext messages; for example 'Hello Students'
3. $f(k, p)$ is the result of encrypting the plain text message $p \in P$ with the key $k \in K$ (draw a picture)
4. $f(k, p) = vi9re!'' * *ewqw;;$ and is the encryption

The Big Question

If f encrypts with f

$$f : K \times P \to E, \quad (k, p) \mapsto f(k, p) \tag{0.7}$$

Can we find f^{-1} the inverse of f

$$f^{-1} : E \to P \times K, \quad f(k, p) \mapsto (k, p). \tag{0.8}$$

In plainer terms, can we 'undo the action? This is the big question in Cryptography —we need to ensure finding the inverse function (in plain English: decrypt the message) is a difficult task for Eve.

Some Definitions

Let $f : X \to Y, x \mapsto f(x)$ denote a function. The function f is said to be:

1. *injective* if $x_1 \neq x_2$ implies $f(x_1) \neq f(x_2)$.
2. *surjective* if to every $y \in Y$ there is a $x \in X$ such that $f(x) = y$.
3. *bijective* if it is both injective and surjective.

Draw some graphs $f : \mathbb{R} \to \mathbb{R}$ illustrating the three definitions. What relevance do the definitions have in encryption?

Injective

The functions in Cryptography need to be injective; the reason is obvious: if the function was not injective this means for our function

$$f : \text{Plain Text} \rightarrow \text{Encrypted Text} \tag{0.9}$$

two messages, (say) "Hello" and "Goodbye" get encrypted to the same message "wert5he34"; that would be a disaster.

Toy Example

We continue with another example

$$f : \mathbb{Z} \to \mathbb{Z}, x \mapsto x + 3; \tag{0.10}$$

in plain English we add three to the number to encode it. So for a plaintext message of 10 the encrption key of 3 will produce 13; the message is decrypted by adding -3.

Symmetric and Asymmetric Encryption

An important definition:

1. A symmetric key encryption algorithm is when the same key is used to both encrypt and decrypt a message.
2. An asymmetric key encryption algorithm is when one key encrypt a message and a different key is used to decrypt the message.

Example :Asymmetric Encryption

The simple example

$$f : \mathbb{Z} \to \mathbb{Z}, x \mapsto x + 3; \tag{0.11}$$

is asymmetric

Example :Symmetric Encryption

To examine symmetric key encryption we need to recall binary addition with no carry

Plain Text	1	1	0	1	0	1	0	1	
Key	0	0	0	1	1	1	1	0	(0.12)
Encryption									

In this type of encryption the zeroes and ones are added (no carry) to produce an encrypted message. To decrypt the message you simply apply the key again. This technique is called the one time pad.

A little bit if Java

We need remarkably little Java to code the encryption algorithms. The concepts we will use repeatedly are:

1. if statement
2. for loop
3. functions.

There are examples on moodle using all these ideas.

types

In Java everything is an object: the only object that are of interest to us are

1. int - integers
2. char - characters
3. Strings - strings

Example For Loop

'Example for loop'

```
public class exampfor{                                          1
 public static void main(String a[])
  {
   for(int i=1;i<=10;i++)
       {
       System.out.println("i = "+i);
       }

  }

}                                                               11
```

Example if Statement

'Example if loop'

```
public class exampif{
 public static void main(String a[])
  {
        int x = 12,y=15;
        if(x>=y)
            {
                System.out.println("x >= y");
            }
        else{
            System.out.println("x < y");
           }

    }

}
```

Cryptography & Encryption:6G7Z1011 : Simple Ciphers , Statistical Analysis and $\mathbb{Z}(n)$

Keith Yates

March 19, 2019

Caesar

Recall from last week, we meet the Caesar cipher, it worked by using a key which 'pushed on' each letter by a fixed amount. So if key $K = 3$ then A mapped to A $+3$ = D, and so on.

Permutation Cipher

We now consider the *permutation cipher*, a permutation is (recalling notation from the last lecture) a bijection from the alphabet to itself, in plainer terms it jumbles the letters up.

plain	a	b	c	d	e	f	...
cipher	z	p	n	e	a	g	...

so 'cab' becomes 'nzp'.
The Caesar cipher had 26 possible keys, whilst the permutation cipher has $26! = 26 \times 25 \times 24$ keys; however it is no more secure than the Caesar cipher, why?

Statistical analysis

Both the Caesar cipher, the affine cipher and the permutation cipher suffer from the same major flaw, they are amenable to attack by statistical analysis.

Attacking the permutation cipher

The permutation cipher has 26! keys, so how long would a brute force attack take to do. Say you had a computer that could test 10^9 keys per second (clock speed of a chip is $\approx 10^9$) then to 'crack' the code would take

$$\frac{26!}{10^9} \approx 16! \text{ seconds} \tag{0.13}$$

Statistical attack

The flaw in the ciphers discussed to date is they always encode a letter in the same way, as such if you read an encrypted message and keep a count of how many times each character appears then the letter with the most counts will be one of the vowels, and the letters with the fewest counts will tend to be 'z'.
A lab question asks you to confirm this.

Affine Cipher

We now look at the affine cipher, it suffers the same problem as the Caesar and the permutation cipher but it uses techniques that are required in 'real world' ciphers.
We need some mathematics

Introducing $\mathbb{Z}(n)$

We define

$$\mathbb{Z}(n) = \{0, 1, 2, \ldots n-1\}\,. \tag{0.14}$$

And we define addition $+$ and multiplication $\times$ in the usual manner except we remove any multiples of n. For example, in $\mathbb{Z}(7)$ then

$$4 \times 3 =_{\mathbb{Z}(7)} 5. \tag{0.15}$$

because in $\mathbb{Z}$

$$4 \times 3 =_{\mathbb{Z}} 12 =_{\mathbb{Z}} (7)^{\text{remove}} + 5 \tag{0.16}$$

From now on I will omit the subscript in $=_{\mathbb{Z}(n)}$ and just write $=$, If I want to emphasise which system I am working in I may write

$$4 \times 3 = 5 \mod 7. \tag{0.17}$$

Simple Problem in $\mathbb{Z}(6)$

A simple problem, please evaluate the cells $(\mathbb{Z}(6), \times)$. I have done some, for example

$$2 \times 4 = 6 + 2, \quad \text{so } 2 \times 4 = 2 \mod 6. \tag{0.18}$$

	0	1	2	3	4	5
0						
1						
2				0	2	
3						
4						
5						

Table: $\mathbb{Z}(6)$

Simple Problem in $\mathbb{Z}(5)$

A simple problem, please evaluate the cells $(\mathbb{Z}(5), \times)$.

	0	1	2	3	4
0					
1					
2					
3					
4					

Table: $\mathbb{Z}(5)$

Coding

The above two examples are interesting from an encryption view point. Suppose we proposed an encryption algorithm that was just multiplication $\mod n$. Then we are in difficulties because we have lost injectivity in the $n = 6$ case, the $n = 5$ case does work.
Important point: our coding techniques require the n in $\mathbb{Z}(n)$ to have rather special properties.

Affine Cipher

The affine cipher is defined for a key $(a, b) \in \mathbb{Z}(26)$ by

$$\phi : \mathbb{Z}(26) \to \mathbb{Z}(26), \quad x \mapsto e(x) = ax + b \mod 26 \tag{0.19}$$

where x is the 'plain' integer and $ax + b$ is its encryption. For example, if $(a, b) = (3, 5)$ and $x = 6$ then it encrypts

$$6 \mapsto 3 \times 6 + 5 = 23. \tag{0.20}$$

Note if $a = 1$ then the affine Cipher reduces to the Caesar cipher.

Affine Cipher

The affine Cipher is interesting from a learning viewpoint, suppose I pick $(a, b) = (4, 7)$

$$\phi : \mathbb{Z}(26) \to \mathbb{Z}(26), \quad x \mapsto e(x) = 4x + 7 \mod 26 \tag{0.21}$$

Note

1. $x = 1$ then $e(x) = 4 \times 1 + 7 = 11 \mod 26$, and
2. $x = 14$ then

$$e(x) = (4 \times 14) + 7 = 56 + 7 = 63 = (26 \times 2) + 11 \mod 26. \tag{0.22}$$

Disaster, the function is no longer injective. In cryptographic terms $(4, 7)$ is NOT a valid key.

Determining the valid keys for the affine cipher

What are the valid keys for the affine cipher? Or, for what values of $a, b \in \mathbb{Z}(26)$ does

$$ax_1 + b = ax_2 + b \mod 26 \tag{0.23}$$

imply $x_1 = x_2$?

prime and relatively prime

Recall

1. The *greatest common denominator* of two numbers a and b is the larger integer that divides both a and b, and we write this $\gcd(a, b)$.
2. A number is termed *prime* if its only divisors are one and itself.
3. Two numbers are termed *relatively prime* if their largest common divisor is 1

For example 11 is prime; the numbers 15 and 22 are relatively prime (though neither is prime).

composite, pseudoprime , Carmichael

Let $n \in \mathbb{N}$ ($\mathbb{N} = \{1, 2, 3, \ldots\}$).

1. n is termed *composite* if $n > 1$ and n has a divisor.
2. n is termed *pseudoprime* to the base a if n is odd, composite and $a^{n-1} \equiv 1 \mod n$.
3. If n is pseudoprime to the base a for all integers a with $\gcd(a, n) = 1$ then n is termed a *Carmichael* number.

Carmichael numbers

Show that 561 is a Carmichael number.

Code

We have $561 = 3.11.17$, and the number is clearly odd and composite. We need to form the set

$$A = \{i \in \mathbb{N} \mid \gcd(i, 561) = 1\} \tag{0.24}$$

and for each $a \in A$ we need to check

$$a^{561-1} \equiv 1 \mod n \tag{0.25}$$

This is rote coding problem (that requires a for loop).

solutions to equations mod 26

Recall we wish the affine function to be injective. That is for any $y \in \mathbb{Z}(26)$ then

$$ax + b = y \mod 26 \tag{0.26}$$

has a unique solution for x. Writing this

$$ax = y - b \mod 26. \tag{0.27}$$

As y runs over all of $\mathbb{Z}(26)$ so does $y - b$ (recall b is fixed), so the question becomes: for what values of a does

$$ax = y \mod 26. \tag{0.28}$$

have a unique x solution for each $y \in \mathbb{Z}(26)$?

Key space sizes

Determining the key space

Caesar cipher	26
Permutation cipher	26!
Affine Cipher	?

Table: The key space is the number of keys available to an encryption algorithm, and for obvious reasons an encryption algorithm needs a large key space.

Solution

The equation

$$ax = y \mod 26. \tag{0.29}$$

has a unique solution in x for each $y \in \mathbb{Z}(26)$ if and only if $\gcd(a, 26) = 1$

Proof.

- $\Rightarrow$: Suppose $\gcd(a, b) = d > 1$ then picking (for example) $y = 0$ then we need to solve $ax = 0 \mod 26$, and we have at least two solutions: $x = 0$ and $x = \frac{26}{d}$.

Proof.

- $\Leftarrow$: Let $\gcd(a, 26) = 1$ and (arguing to the contrary) suppose there are two solutions

$$ax_1 = y \mod 26 \quad \text{and} \quad ax_2 = y \mod 26 \tag{0.30}$$

then

$$a(x_1 - x_2) = 0 \mod 26. \tag{0.31}$$

So $a(x_1 - x_2)$ is a multiple of 26, or turning this around $26 \mid a(x_1 - x_2)$. However $\gcd(a, 26) = 1$ so $26 \mid x_1 - x_2$ thus $x_1 = x_2 \mod 26$.

Euler ϕ Function

The *Euler* ϕ function is defined

$$\phi : \mathbb{N} \to \mathbb{N}, \quad n \mapsto |\{i \in \mathbb{N} \mid 1 \leq i \leq n, \gcd(n, i) = 1\}.| \tag{0.32}$$

In English, for each $n \in \mathbb{N}$ the $\phi(n)$ is the number of integers less than or equal to n that are relatively prime to n.

Euler's function

Evaluate the Euler ϕ function in the range $1 \leq i \leq 10$.

i	$\phi(i)$
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	

Table: Euler

Solution

i	$\phi(i)$
1	1
2	1
3	2
4	2
5	4
6	2
7	6
8	4
9	6
10	4

Table: Euler's totient

Some working out, $\phi(9)$

1	2	3	4	5	6	7	8	9
✓	✓	✗	✓	✓	✗	✓	✓	✗

Table: What is $\phi(9)$, a tick indicates $\gcd(i, 9) = 1$

$\phi(9) = 6$, count the ticks.

Some working out, $\phi(10)$

1	2	3	4	5	6	7	8	9	10
✓	✗	✓	✗	✗	✗	✓	✗	✓	✗

Table: What is $\phi(10)$, a tick indicates $\gcd(i, 10) = 1$

$\phi(10) = 4$, count the ticks.

key space for the affine cipher

Recall, we are still trying to evaluate the key space for the affine cipher. Now note that every integer has an unique (to within the order) multiplicative decomposition has a product of primes, for example

$$100 = 2^2 5^2. \tag{0.33}$$

Cryptography & Encryption:6G7Z1011 : Euclid, and the Fast Powering Algorithm

Keith Yates

March 19, 2019

Speed of Computation

We have already coded algorithms of relevance to cryptography. For example:

1. determining if a number is or is not prime;
2. determining the greatest common denominator of two integers.

Both are of importance in cryptography. Given the integers that are used in real world cryptography can be hundreds of digits long there is an interest in its speed.

Estimates

When we talk of the 'time' taken to of an algorithm, we will measure it in bit operations, not real time. If all the computers became twice as fast over night we would not say the algorithms are running twice as fast, because what the algorithm did in terms of bit operations has not changed. Let us return to a problem we have meet before, and see if we can find a faster algorithm.

Euclidean algorithm

Let a, b be positive integers with $b \leq a$, there is a decomposition

$$a = qb + r \quad \text{with} \quad 0 \leq r < b. \tag{0.34}$$

That is b fits 'into' a q times and there is a remainder r that is always less than b.

1. Suppose d divided into both a and b, then eqn. 0.36 implies d must divide into r, which we write $d \mid r$
2. Suppose e divided into both r and b, then eqn. 0.36 implies it must divide into a.

Thus

$$\gcd(a, b) = \gcd(b, r). \tag{0.35}$$

Euclid's algorithm

Euclid realised that as

$$a = qb + r \quad \text{with} \quad 0 \leq r < b. \tag{0.36}$$

and

$$\gcd(a, b) = \gcd(b, r) \tag{0.37}$$

both held, then (as $r < b$) it would be easier to evaluate the right-hand side of eqn. 0.37, and being clever he also realised he could apply the same trick to (b, r) and write $b = rq' + r'$ with $0 \leq r' < r$ and

$$\gcd(b, r) = \gcd(r', r) \tag{0.38}$$

An example will clarify all this.

Proof

We have

$$a = bq + r \quad \text{where} \quad 0 \leq r < b. \tag{0.39}$$

1. Let α divides a and b then

$$\frac{r}{\alpha} = \frac{a - bq}{\alpha} \in \mathbb{N} \tag{0.40}$$

   so α divides r.

2. Let α divides b and r then

$$\frac{a}{\alpha} = \frac{bq + r}{\alpha} \in \mathbb{N} \tag{0.41}$$

Euclid

We have shown that if

$$a = bq + r \quad \text{where} \quad 0 \leq r < b. \tag{0.42}$$

then

$$\gcd(a, b) = \gcd(b, r). \tag{0.43}$$

This is clever: we have moved the problem to the same problem but with smaller numbers, we have $r < b < a$. And we can apply the same procedure to $\gcd(b, r)$ —note as the remainder $0 \leq r$; this process must stop after a finite number of steps.

Example

That was rather abstract, have a go apply (pen and paper calculation) to work out

$$\gcd(2024, 748) \tag{0.44}$$

Working out

$$
\begin{aligned}
2024 &= 748 \times 2 + 528 \\
748 &= 528 \times 1 + 220 \\
528 &= 220 \times 2 + 88 \\
220 &= 88 \times 2 + 44 \\
88 &= 44 \times 2 + 0
\end{aligned}
\tag{0.45}
$$

We have

$$
\begin{aligned}
&\gcd(2024, 748) = \gcd(748, 528) = \gcd(528, 220) = \\
&\gcd(220, 88) = \gcd(88, 44)
\end{aligned}
\tag{0.46}
$$

The answer is 44; not we reached this in 5 iterations of the algorithm a large saving over 748 'checks' in the naive approach.

Coding Euclid's algorithm

The following is Euclid's algorithm in words; you should code it in Java in the lab session. Input a and b we seek $\gcd(a, b)$

1. Let $r_0 = a$, $r_1 = b$.
2. Set $i = 1$.
3. Divide r_{i-1} by r_i to get quotient q_1 and remainder r_{i+1}, that is

$$r_{i-1} = r_i q_i + r_{i+1} \quad \text{with} \quad 0 \leq r_{i+1} < r_i \tag{0.47}$$

4. If $r_{i+1} = 0$ then $r_i = \gcd(a, b)$ and quit algorithm
5. Set $i = i + 1$ and go to 3.

Note as r_i decreases each time it must reach 0 after finitely many steps - so the algorithm does stop!

How Fast is Euclid's algorithm?

Euclid's algorithm for gcd is our first 'proper' algorithm. To complete its study we need to determine how fast it runs. Recall this requires us to create a graph plotting n the size of the smallest number against 'time' how long the algorithm takes to complete.

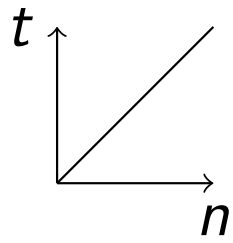


Figure: If we plot n (size of the smallest number argument to our gcd algorithm) against t time taken to run then for our naive gcd algorithm it will scale linearly; double the size of the number, then we double the checks to be performed. How does Euclid perform?

How fast is Euclid's algorithm?

Consider integers a, $b \in \mathbb{N}$ with $b \leq a$ then the algorithm loops at most

$$2\ln_2(b) + 1 \tag{0.48}$$

times $\ln_2$ is log to the base 2. This is some saving, consider our earlier example $\gcd(2024, 748)$ Then

$$2\ln_2(748) + 1 \approx 21. \tag{0.49}$$

Theoretical Analysis of Euclid's algorithm

Recall Euclid's algorithm generates a sequence of remainder

$$0 = r_n < r_{n-1} < .. < r_3 < r_2 < r_1. \tag{0.50}$$

At each stage the sequence decreases, and terminates at zero, the key issue is how fast does it decrease? We claim

$$r_{i+2} < \frac{r_i}{2}. \tag{0.51}$$

So after 5 iterations the remainder has got smaller by a factor of $\frac{1}{2^5}$.

Speed of Euclid's Algorithm

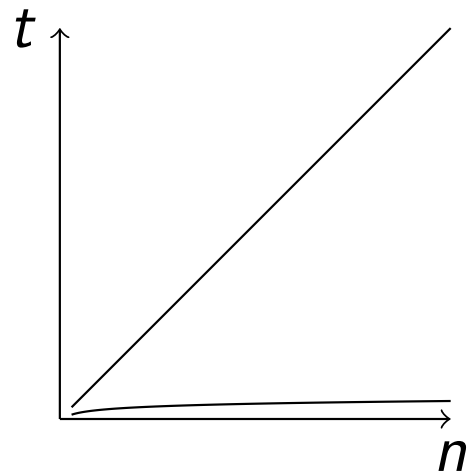


Figure: How does Euclid perform?The bottom line is $t = \log_2(n)$, the straight line $t = n$.

Proof

We need to prove $r_{i+2} < \frac{r_i}{2}$ holds for all i. There are two cases:

1. $r_{i+1} \leq \frac{r_i}{2}$: Then $r_{i+2} < r_{i+1} < \frac{r_i}{2}$.
2. $r_{i+1} > \frac{r_i}{2}$: r_{i+1} is so large that we have $r_i = 1.r_{i+1} + r_{i+2}$. Thus

$$r_{i+2} = r_i - r_{i+1} = r_i - \frac{r_i}{2} = \frac{r_i}{2} \tag{0.52}$$

We have shown $r_{i+2} < \frac{r_i}{2}$. As such

$$r_{2k+1} < \frac{r_{2k-1}}{2} < \frac{r_{2k-3}}{2^2} < \frac{r_{2k-5}}{2^3} < \ldots \frac{r_1}{2^k} = \frac{b}{2^k} \tag{0.53}$$

Thus if $b \leq 2^k$ then $r_{2k+1} < 1$ so it must be zero; and

$$\text{iterations required } \leq 2k = 2(k-1)+2 < 2\ln_2(b)+2. \tag{0.54}$$

End of the Story?

Given the importance of Euclid's algorithm it has been studied extensively and there are improvements (they are non-examinable but if you are interested they are discussed in [**?**]); the best to date is

$$t = 0.85 \ln_2(n) + 0.14 \tag{0.55}$$

The fast powering algorithm

In encryption algorithms we often need to compute x where

$$x = a^n \mod m \tag{0.56}$$

where a, n and m may be very large numbers. A naive approach would be

$$\begin{aligned} g_1 &= a \mod m \\ g_2 &= ag_1 \mod m \\ g_3 &= ag_2 \mod m \\ g_4 &= ag_3 \mod m \\ &. \end{aligned} \tag{0.57}$$

then $g_n = a^n \mod n$. Is it feasible?

How much time have you got?

Consider $x = a^n \mod m$, let a, m be large and let $n = 2^{1000}$. Consider a computer chip with a clock speed of 2^{20} (faster than any computer in the world) and let it process 2^{20} iterations in a second.

$$\text{time to complete all iterations} = \frac{2^{1000}}{2^{20}} > \text{ life time of universe.} \tag{0.58}$$

example

We — obviously — need a way of evaluating powers that is quicker than the above estimate. By way of an example let us compute

$$x = 3^{218} \mod 1000. \tag{0.59}$$

Any thoughts on how to do this?

Fast Powering Algorithm $x = 3^{218} \mod 1000$

First express 218 as the sum of powers of two

$$218 = 2 + 2^3 + 2^4 + 2^6 + 2^7 \tag{0.60}$$

Then

$$3^{218} = 3^{2+2^3+2^4+2^6+2^7} = 3^2.3^{2^3}.3^{2^4}.3^{2^6}.3^{2^7}. \tag{0.61}$$

We have

i	0	1	2	3	4	5	6	7
$3^{2^i} \mod 1000$	3	9	81	561	721	841	281	961

$$
\begin{aligned}
3^{2^6} &= 3^{2^5}.3^{2^5} \\
&= 3^{2^4}3^{2^4}3^{2^4}3^{2^4} \\
&= (43036.1000 + 721).(42046.1000 + 721) \\
&(43036.1000 + 721).(42046.1000 + 721) \mod 1000 \\
&= (721.721).(721.721) \\
&= 841.841 \mod 1000 \\
&= 281 \mod 1000
\end{aligned}
\tag{0.62}
$$

A bit more ...

We deduce

$$\begin{aligned} 3^{218} &= 3^2.3^{2^3}.3^{2^4}.3^{2^6}.3^{2^7} \\ &= 9.(6.1000+561).(1000.43046+721) \\ &(281).961 \mod 1000 \\ &= 480 \mod 1000. \end{aligned} \tag{0.63}$$

$$\begin{aligned} 3^{218} &= 3^2.3^{2^3}.3^{2^4}.3^{2^6}.3^{2^7} \\ &= 9.561.721.281.961 \mod 1000 \\ &= 480 \mod 1000. \end{aligned} \tag{0.64}$$

All done in two slides!

Extended Euclidean Algorithm

We return now to solution of equations. Let $a, b \in \mathbb{N}$ with $\gcd(a, b)$. Then there exists integers m and n such that

$$\gcd(a, b) = am + bn. \tag{0.65}$$

To see this consider the set

$$S = \{\alpha a + \beta b \mid \alpha, \beta \in \mathbb{Z}\}. \tag{0.66}$$

This set S has a smallest positive element, we claim it is $\gcd(a, b)$.

Proof of Claim

Let d denote the smallest number greater than or equal to zero in the set $S = \{\alpha a + \beta b \mid \alpha, \beta \in \mathbb{Z}\}$; say $d = \alpha_1 a + \beta_1 b$. We have two tasks:

1. Proving d divides a and b;
2. Any number dividing a and b is smaller than or equal to d (so $d = \gcd(a, b)$).

Proof

1. Using Euclid we have $a = dq + r$ where $0 \leq r < d$ then

$$r = a - dq = a - (\alpha_1 a + \beta_1 b)q = a(1 - \alpha_1 q) - (\beta_1 q)b \in S \tag{0.67}$$

but $0 \leq r < d$ and d is the smallest element in S; thus $r = 0$ and $d \mid a$. Similar reasoning gives $d \mid b$.

2. Finally let γ be any number dividing both a and b then as $d = \alpha_1 a + \beta_1 b$ we deduce $\gamma \mid d$; thus d is the greatest common denominator.

important property of the primes

Let p be prime and a, $b \in \mathbb{N}$ such that $p \mid ab$; then $p \mid a$ or $p \mid b$.
To prove this suppose $p \nmid a$ we show $p \mid b$. We have $\gcd(p, a) = 1$ so we have u and v such that

$$up + va = 1 \quad \text{thus} \quad upb + vab = b; \tag{0.68}$$

so as $p \mid b$ we deduce $p \mid ab$.
Note this result does not hold when p is not prime for example 15 divides $210 = 6 \times 35$ but 15 does not divide into 6 nor into 35.

Important Result

Recall, if $a, b \in \mathbb{N}$ with a greatest common denominator of $\gcd(a, b)$ then there exists integers m and n such that

$$\gcd(a, b) = am + bn. \tag{0.69}$$

This implies something of interest to us.

Existence of an inverse

Let a denote an integer then there is a solution to

$$ab = 1 \mod m \tag{0.70}$$

if and only if $\gcd(a, m) = 1$.

Proof.

- $\Rightarrow$: Suppose $\gcd(a, m) = 1$ then we have integers u and v such that $au + mv = 1$, and we deduce

$$au - 1 = mv, \quad \text{thus} \quad au = 1 \mod m. \tag{0.71}$$

- $\Leftarrow$: Suppose $ab = 1 \mod m$ then

$$ab - \alpha m = 1 \tag{0.72}$$

and any number d that divides both a and m must divide 1, which means $d = 1$.

Returning to our toy examples $\mathbb{Z}(5)$ and $\mathbb{Z}(6)$, then

1. $\mathbb{Z}(5)$: as $\gcd(a, 5) = 1$ for all $1 \leq a < 5$ (5 is prime) then every non-zero element of $\mathbb{Z}(5)$ has an inverse.
2. $\mathbb{Z}(6)$ $\gcd(3, 6) = 3 \neq 1$ so (for example) 3 has no inverse. Explicitly:
   a) $3.1 = 3 \neq 1$,
   b) $3.2 = 0 \neq 1$,
   c) $3.3 = 3 \neq 1$,
   d) $3.4 = 0 \neq 1$,
   e) $3.5 = 3 \neq 1$.

Introducing S_n

If I have time I will introduce S_n, the permutation group on n objects.

Cryptography & Encryption:6G7Z1011 : Mathematical Structures and a First Look at Diffie

Keith Yates

March 19, 2019

Overview

We continue with our study of real world (that is algorithms currently used in secure systems) cryptography. Goals of the lecture:

1. To define, discuss and work with the key mathematical structures used in Cryptography
2. Discuss the Diffie-Hellman key exchange protocol; this result started Public Key Cryptography [1] .

Groups, Rings and Fields

We have meet $\mathbb{Z}(n)$ under addition and multiplication, and we have used it in a few cases; it was just 'clock' arithmetic. A fact glossed over till know is the role of n. We have already seen that $\mathbb{Z}(6)$ and $\mathbb{Z}(7)$ are very different structures:

1. In $\mathbb{Z}(6)$ two non-zero elements could multiply to zero; for example, $2 \times 3 = 0$
2. In $\mathbb{Z}(7)$ two non-zero elements never multiplied to zero.

There are three fundamental algebraic constructs that we need to define.

Definition:Group

A *group* G is a set with an operation $\circ : G \times G \to G$; for brevity we write $g_1 g_2$ for $g_1 \circ g_2$. The operation satisfies:

1. The operation is associative that is $(g_1 g_2) g_3 = g_1 (g_2 g_3)$.
2. There is an identity (which we denote by 1) such that $1g = g1$ for all $g \in G$.
3. To every $g \in G$ there is an inverse g^{-1} such that $gg^{-1} = 1$.

If $g_1 g_2 = g_2 g_1$ holds for all g_1, $g_2 \in G$ then we have an *abelian group*.

Ring

A *ring* R is a set with two operations $+ : G \times G \to G$ (addition) and $\times : G \times G \to G$ multiplication.

1. $(G, +)$ is an abelian group
2. Multiplication is associative; $(a \times b) \times c = a \times (b \times c)$.
3. Multiplication is distributive; that is

$$a(b+c) = ab + ac \quad \text{and} \quad (b+c)a = ba + ca. \tag{0.73}$$

Field

A *field* F is a commutative ring in which every non-zero element has a multiplicative inverse.
The definitions of group, ring and field are abstract; we illustrate the definitions with some examples and classify them.

Examples

Groups are an abstract concept the 1 in a group is the identity element it means $1g = g$ for all g (it does not have to be the number 1) Decide if the following are groups and if they are determine if they are abelian, you need to decide what the identity is, and if each element has an inverse.

1. $\{1, 2, 3, \ldots\}$ under addition.
2. $\{\ldots, -2, -1, , 0, 1, 2, 3, \ldots\}$ under addition.
3. All two by two matrices under addition.
4. All two by two matrices under multiplication.
5. The set of bijections on a set X under function composition.

S_3

The collection of bijective functions on three elements (under function composition) is a non-abelian group. It is denoted by S_3 and its termed the *permutation group* on three objects (permutation groups are used a lot in cryptography)

1. What size is the group?
2. Can we evaluate its multiplication table?

This will get messy, in lectures we will experiment with small groups, in labs slightly larger groups.

S_3

Even with small groups, we need to use a simplified notation, this is going to get messy. First recall a bijection is an injective and surjective function.

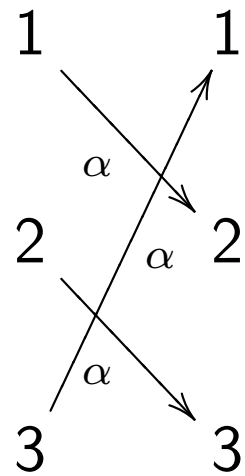


Table: A bijection between three objects. It is to be read: the bijection α takes 1 to 2, 2 to 3 and 3 to 1

Composition of bijections

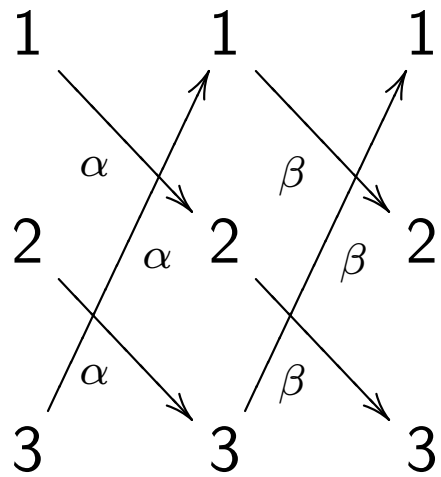


Table: The compostion of two bijections is a bijection; $\beta\alpha$, so for example $\beta\alpha(1) = 3$

Shorthand

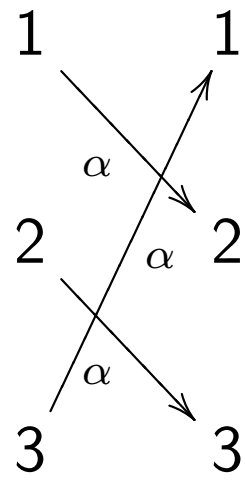


Table: $\alpha = (1, 2, 3)$

The bijection α may be written $(1, 2, 3)$ and it reads $1 \mapsto 2$, $2 \mapsto 3$ and $3 \mapsto 1$.

Evaluate S_3

It will be a bit messy, but construct the multiplication table for S_3 To help you let us agree to write

1. e for the bijection that fixes 1, 2 and 3, so does nothing!
2. $a = (1, 2, 3)$
3. $b = (2, 3)$

What does S_3 look like?

Blanks you need to fill in

.	e	a	a^2	b	ab	a^2b
e	e	a	a^2	b	ab	a^2b
a	a					
a^2	a^2					
b	b					
ab	ab					
a^2b	a^2b					

Table: S_3

Distinguishing between Groups

We have shown S_3 is a groups and its size is 6. The next question is

1. Are there any other groups of size 6 that are different from S_3?
2. If so, what do they look like?

Homomorphism, Isomorphism

We need to define what we mean by two groups being the same, and to turn it around what it means for two groups to be different. It is, hopefully, obvious that for two groups to be identical they need to have the same size. A *isomorphism* is a bijection between two groups

$$\phi : G_1 \to G_2, \quad g_1 \mapsto \phi(g_1) \tag{0.74}$$

such that

$$\phi(a \circ_{G_1} b) = \phi(a) \circ_{G_2} \phi(b) \tag{0.75}$$

holds for all a, $b \in G$.

Isomorphism

Let us find a simple example of two groups of the same size but they are not isomorphic.

1. S_3
2. $\mathbb{Z}(7)^*$

The important point:

1. For a given integer n there may exist numerous distinct groups of that size.

Symmetric and Asymmetric

Recall

1. A key is termed *symmetric* if the key both encrypts and decrypts a message. For example, the one time pad with any key $010001001101\ldots$ is symmetric, if you apply it twice you end up back where you started.
2. A key is termed *asymmetric* if the key that encrypts a mess-sage is different form the key that decrypts it; for example the Caesar cipher is asymmetric because the encryption and decryption keys differ.

Public /Private

We first need to clarify the difference between Public and Private cryptography. All examples prior to today's lecture are examples of private key cryptography. That is Alice and Bob have a private key by which they can securely communicate; for example Alice and Bob could use the one time pad with a particular key

$$K = 0100101010101010 \tag{0.76}$$

The key needs to be kept private, if Eve knows the value of the key then they can decrypt the message.

Some Notation

Our protagonists are still, Alice, Bob and Eve. The goal remains the same: Alice and Bob wish to communicate securely and Eve wishes to pry. We introduce

$$K_{A,Pu} = \text{Alice's Public Key known to All} \tag{0.77}$$

and

$$K_{A,Pr} = \text{Alice's Private Key known only to Alice} \tag{0.78}$$

$K_{B,Pu}$, $K_{B,Pr}$, $K_{E,Pu}$ and $K_{E,Pr}$ all have the obvious meanings.

Secure Connection across an Insecure Channel

The following is very important (and concepts such as e-commerce depend critically upon it)

Secure connection

The Diffie-Hellman algorithm allows Alice and Bob to establish a secure connection across an insecure connection.

If the above statement does not surprise you - you should think about it a little more

Alice, Bob and Eve in a Cafe

Forget about computers and the internet; imagine Alice, Bob and Eve meet in a cafe they have never meet and have no knowledge about each other. Alice and Bob wish to establish a 'shared secret' (typically this is a number) Given Eve is sat with them and Eve hears every word they say to each other how is it possible for Alice and Bob to exchange information and generate a 'shared secret'.
Note : Alice and Bob have never meet.

How it this Possible?

It was for many years thought to be impossible to establish a secure connection over an insecure network. The solution involves complicated ideas from discrete mathematics.

e-commerce variant

The e-commerce variant of the previous problems occurs frequently. Whilst in some e-commerce sites (Amazon) you have an account there are many sites you use for one off transactions in which your computer and the website of the company you are buying from establish a secure connection over an insecure channel
We commence the non-trivial task of showing how to do this.

Diffie Protocol

Fix a large prime p and a integer $g \mod p$; these numbers are known to Alice, Bob and Eve (in fact anyone in the world) can know their values. The protocol is as follows:

1. Alice picks a secret integer a.
2. Bob picks a secret integer b.
3. Alice works out $A = g^a \mod p$ and Bob works out $B = g^b \mod A$.
4. Alice sends A to Bob, Bob sends B to Alice; note Eve has knowledge of A and B. (Everybody knows the values of A and B.)
5. The clever bit follows;
   a) Alice computes $A' = B^a \mod p$
   b) Bob computes $B' = A^b \mod p$.

Congruent Mathematics

We find

$$A' = B^a = (g^b)^a = (g^a)^b = A^b = B' \mod p. \qquad (0.79)$$

Alice and Bob determine the same number and this is their 'shared secret'
We need an example with real numbers

Lab Question

We work on the Java Implementation of the Diffie-Hellman protocol. We will use small prime numbers — if a question asks you to verify something you are free to use a brute force attack.

1. Let $p = 941$ (prove 941 is prime), we let $g = 237$.
2. Suppose Alice chooses a secret key $a = 347$ what is A?
3. Suppose Bob chooses a secret key $b = 781$ what is B?
4. What is the value of A'?
5. What is the value of B'?

Knowledge is Power

At this point we have shown how Alice and Bob can generate a shared key — why is Eve not in a position to find A'.

	p	g	a	b	A	B
Alice	✓	✓	✓	✗	✓	✓
Bob	✓	✓	✗	✓	✓	✓
Eve	✓	✓	✗	✗	✓	✓

Table: Who knows what, a ✓indicates the person knows the value of the variable. Alice and Bob are one piece of data short, Eve — crucially is two pieces short — recall the goal is to evaluate A' (or B' as $A' = B'$).

Eve's hard problem

We left Eve trying to work out Alice and Bob's shared secret. Recall this amounts to solving

$$627^a = 390 \mod 941 \quad \text{or equivalently} \quad 627^b = 691 \mod 941. \tag{0.80}$$

A lab question asks you to 'crack' the problem by brute force; in our toy model $p = 941$ so brute force will work.

What are real world values?

Our algorithm is a correct implementation of Hiffie, however in real world encryption our primes are of the size 2^{1000}.

Diffie-Hellman Problem

If Eve discovers an algorithm that can solve the equations

$$x^a = y \mod p \quad \text{or equivalently} \quad x^b = c \mod p. \quad (0.81)$$

in a reasonable amount of time then the Diffie algorithm would be obsolete.
No one has yet found such an algorithm, and the study of eqn. 0.87 is referred to as the 'Diffie-Hellman' problem.

The Diffie-Hellman Problem

The Diffie-Hellman problem can be cast into a precise mathematical statememt; to do this we need to introduce some new mathematical concepts.
(There is little I can do about this as cryptography becomes more advanced computer scientists use ideas from more abstract areas of mathematics, for example elliptic curves.)

Cryptography & Encryption:6G7Z1011 : Coding Diffie

Keith Yates

March 19, 2019

Overview

We continue with our study of real world (that is algorithms currently used in secure systems) cryptography. Goals of the lecture:

1. Discuss the Diffie-Hellman key exchange protocol; this result started Public Key Cryptography [1] .
2. Implement Diffie-Hellman in JAVA
3. Assess the security of Diffie-Hellman

Symmetric and Asymmetric

Recall

1. A key is termed *symmetric* if the key both encrypts and decrypts a message. For example, the one time pad with any key $010001001101\ldots$ is symmetric, if you apply it twice you end up back where you started.
2. A key is termed *asymmetric* if the key that encrypts a mess-sage is different form the key that decrypts it; for example the Caesar cipher is asymmetric because the encryption and decryption keys differ.

Public /Private

We first need to clarify the difference between Public and Private cryptography. All examples prior to today's lecture are examples of private key cryptography. That is Alice and Bob have a private key by which they can securely communicate; for example Alice and Bob could use the one time pad with a particular key

$$K = 0100101010101010 \tag{0.82}$$

The key needs to be kept private, if Eve knows the value of the key then they can decrypt the message.

Some Notation

Our protagonists are still, Alice, Bob and Eve. The goal remains the same: Alice and Bob wish to communicate securely and Eve wishes to pry. We introduce

$$K_{A,Pu} = \text{Alice's Public Key known to All} \tag{0.83}$$

and

$$K_{A,Pr} = \text{Alice's Private Key known only to Alice} \tag{0.84}$$

$K_{B,Pu}$, $K_{B,Pr}$, $K_{E,Pu}$ and $K_{E,Pr}$ all have the obvious meanings.

Secure Connection across an Insecure Channel

The following is very important (and concepts such as e-commerce depend critically upon it)

Secure connection

The Diffie-Hellman algorithm allows Alice and Bob to establish a secure connection across an insecure connection.

If the above statement does not surprise you - you should think about it a little more

Alice, Bob and Eve in a Cafe

Forget about computers and the internet; imagine Alice, Bob and Eve meet in a cafe they have never meet and have no knowledge about each other. Alice and Bob wish to establish a 'shared secret' (typically this is a number) Given Eve is sat with them and Eve hears every word they say to each other how is it possible for Alice and Bob to exchange information and generate a 'shared secret'.
Note : Alice and Bob have never meet.

How it this Possible?

It was for many years thought to be impossible to establish a secure connection over an insecure network. The solution involves complicated ideas from discrete mathematics.

e-commerce variant

The e-commerce variant of the previous problems occurs frequently. Whilst in some e-commerce sites (Amazon) you have an account there are many sites you use for one off transactions in which your computer and the website of the company you are buying from establish a secure connection over an insecure channel
We commence the non-trivial task of showing how to do this.

Diffie Protocol

Fix a large prime p and a integer $g \mod p$; these numbers are known to Alice, Bob and Eve (in fact anyone in the world) can know their values. The protocol is as follows:

1. Alice picks a secret integer a.
2. Bob picks a secret integer b.
3. Alice works out $A = g^a \mod p$ and Bob works out $B = g^b \mod p$.
4. Alice sends A to Bob, Bob sends B to Alice; note Eve has knowledge of A and B. (Everybody knows the values of A and B.)
5. The clever bit follows;
   a) Alice computes $A' = B^a \mod p$
   b) Bob computes $B' = A^b \mod p$.

Congruent Mathematics

We find

$$A' = B^a = (g^b)^a = (g^a)^b = A^b = B' \mod p. \qquad (0.85)$$

Alice and Bob determine the same number and this is their 'shared secret'
We need an example with real numbers

Lab Question

We work on the Java Implementation of the Diffie-Hellman protocol. We will use small prime numbers — if a question asks you to verify something you are free to use a brute force attack.

1. Let $p = 941$ (prove 941 is prime), we let $g = 237$.
2. Suppose Alice chooses a secret key $a = 347$ what is A?
3. Suppose Bob chooses a secret key $b = 781$ what is B?
4. What is the value of A'?
5. What is the value of B'?

Knowledge is Power

At this point we have shown how Alice and Bob can generate a shared key — why is Eve not in a position to find A'.

	p	g	a	b	A	B
Alice	✓	✓	✓	✗	✓	✓
Bob	✓	✓	✗	✓	✓	✓
Eve	✓	✓	✗	✗	✓	✓

Table: Who knows what, a ✓indicates the person knows the value of the variable. Alice and Bob are one piece of data short, Eve — crucially is two pieces short — recall the goal is to evaluate A' (or B' as $A' = B'$).

Eve's hard problem

We left Eve trying to work out Alice and Bob's shared secret. Recall this amounts to solving

$$627^a = 390 \mod 941 \quad \text{or equivalently} \quad 627^b = 691 \mod 941. \tag{0.86}$$

A lab question asks you to 'crack' the problem by brute force; in our toy model $p = 941$ so brute force will work.

What are real world values?

Our algorithm is a correct implementation of Hiffie, however in real world encryption our primes are of the size 2^{1000}.

Diffie-Hellman Problem

If Eve discovers an algorithm that can solve the equations

$$x^a = y \mod p \quad \text{or equivalently} \quad x^b = c \mod p. \quad (0.87)$$

in a reasonable amount of time then the Diffie algorithm would be obsolete.
No one has yet found such an algorithm, and the study of eqn. 0.87 is referred to as the 'Diffie-Hellman' problem.

The Diffie-Hellman Problem

The Diffie-Hellman problem can be cast into a precise mathematical statememt; to do this we need to introduce some new mathematical concepts.
(There is little I can do about this as cryptography becomes more advanced computer scientists use ideas from more abstract areas of mathematics, for example elliptic curves.)

Groups, Rings and Fields

We have meet $\mathbb{Z}(n)$ under addition and multiplication, and we have used it in a few cases; it was just 'clock' arithmetic. A fact glossed over till know is the role of n. We have already seen that $\mathbb{Z}(6)$ and $\mathbb{Z}(7)$ are very different structures:

1. In $\mathbb{Z}(6)$ two non-zero elements could multiply to zero; for example, $2 \times 3 = 0$
2. In $\mathbb{Z}(7)$ two non-zero elements never multiplied to zero.

There are three fundamental algebraic constructs that we need to define.

Definition:Group

A *group* G is a set with an operation $\circ : G \times G \to G$; for brevity we write $g_1 g_2$ for $g_1 \circ g_2$. The operation satisfies:

1. The operation is associative that is $(g_1 g_2) g_3 = g_1 (g_2 g_3)$.
2. There is an identity (which we denote by 1) such that $1g = g1$ for all $g \in G$.
3. To every $g \in G$ there is an inverse g^{-1} such that $gg^{-1} = 1$.

If $g_1 g_2 = g_2 g_1$ holds for all g_1, $g_2 \in G$ then we have an *abelian group*.

Example

Let us write $\mathbb{Z}(7)^* = \mathbb{Z}(7) \setminus \{0\}$; that is we remove the zero element (this is common notation). Evaluate the multiplication tables for

$$(\mathbb{Z}(7)^*, \times) \quad \text{and} \quad (\mathbb{Z}(6)^*, \times). \tag{0.88}$$

In our new terminology one of the above sets is an abelian group — which one is it?

Examples

Groups are an abstract concept the 1 in a group is the identity element it means $1g = g$ for all g. Decide if the following are groups and if they are determine if they are abelian.

1. $\{1, 2, 3, \ldots\}$ under addition.
2. $\{\ldots, -2, -1, , 0, 1, 2, 3, \ldots\}$ under addition.
3. All two by two matrices under addition.
4. All two by two matrices under multiplication.
5. The set of bijections on a set X under function composition.

S_3 used in encryption

Recall S_3 is the group of permutations on three objects, we illustrate its usage in a 'toy encryption model', S_3 will be used to permute the blocks and the data in the blocks. It is a useful first approximation to the DES. Define a key

$$K = (\alpha, \beta, \alpha, \beta). \tag{0.89}$$

$\alpha = (1, 2)$, $\beta = (1, 2, 3)$. We define an encryption action on a string of length 9 by letting β act on the 3 blocks, and then applying α on the first block, β on the second block and α on the third block. Consider the string abcdefghi

$$\left| \text{a} \right| \text{b} \left| \text{c} \right| \text{d} \left| \text{e} \right| \text{f} \left| \text{g} \right| \text{h} \left| \text{i} \right|$$

$$\left| \text{a} \right| \text{b} \left| \text{c} \right\| \text{d} \left| \text{e} \right| \text{f} \left\| \text{g} \right| \text{h} \left| \text{i} \right|$$

We need to draw diagrams.

More Theory

To make headway we need to develop some more theory, some new concepts:

1. Equivalence relation;
2. Subgroup;
3. Generators;
4. Coset.

Relation, Equivalence Relation

Let X denote a set a *relation* R is simply a subset of $X \times X$.
As *equivalence* relation is a relation that is

1. reflexive: that is, $(x, x) \in R$ for all $x \in X$
2. symmetric: that is, $(x, y) \in R$ implies $(y, x) \in R$
3. transitive: that is, $(x, y) \in R$ and $(y, z) \in R$ imply $(x, z) \in R$

We write $x \sim y$ if $(x, y) \in R$

Examples

We need some examples:

1. In $\mathbb{Z}$ (set of all integers) define $x \sim y$ if and only if $x - y$ is a multiple of 10.
2. In $\mathbb{Z}$ (set of all integers) define $x \sim y$ if and only if $x \leq y$

Prove $\sim$ is an equivalence relation in 1, but $\sim$ is not an equivalence relation in 2.

Equivalence relations partition a set X

Let X denote a set, and $\sim$ an equivalence relation. For each $x \in X$ define

$$x^\bullet = \{y \in X \mid x \sim y\} \tag{0.90}$$

An equivalence relation partitions the set X. That is

$$X = \sqcup_{i=1}^{n} x_i^\bullet \tag{0.91}$$

where $\sqcup$ is disjoint union. A picture may help

Partitions

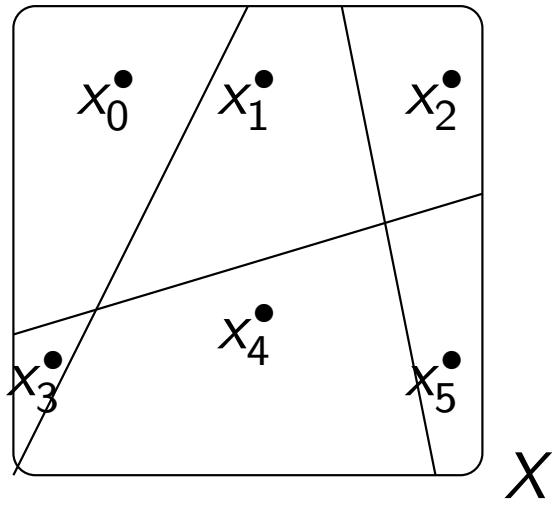


Figure: An equivalence relation partitions a set X into disjoint components.

Equivalence Relations

⌜An equivalence $\sim$ relation on a set X partitions a set; that is

$$X = \sqcup_{i=1}^{n} x_i^{\bullet} \tag{0.92}$$

⌟

Proof.

Let x, $y \in X$ then we claim either $x^{\bullet} = y^{\bullet}$ or $x^{\bullet} \cap y^{\bullet} = \emptyset$.
Have a go!

proof

Proof.

Recall

$$x^\bullet = \{a \in X \mid x \sim a\} \quad \text{and} \quad y^\bullet = \{a \in X \mid y \sim a\}. \tag{0.93}$$

and suppose $x^\bullet \cap y^\bullet \neq \emptyset$. There is a $z \in X$

$$z \in x^\bullet \cap y^\bullet. \tag{0.94}$$

we have $x \sim z$ and $y \sim z$.

1. Pick any $\alpha \in x^\bullet$, then $x \sim \alpha$ so $\alpha \sim x$ (symmetric) and as $x \sim z$ then $\alpha \sim z$ and as $z \sim y$ then $\alpha \sim y$ so $\alpha \in y^\bullet$. We deduce $x^\bullet \subseteq y^\bullet$.
2. Similarly $y^\bullet \subseteq x^\bullet$

Subgroup

Let G denote a group, a subset of G is termed a subgroup if it is itself a group. For example

$$GL(2) = \left\{ L \in M_{2,2}(\mathbb{R}) \mid L^{-1} \text{ exists} \right\} \tag{0.95}$$

(recall $M_{2,2}(\mathbb{R})$ is the set of 2×2 matrices) is a group and

$$H = \left\{ \begin{pmatrix} a & c \\ 0 & d \end{pmatrix} \mid ad \neq 0 \right\} \tag{0.96}$$

is a subgroup of $GL(2)$

Prove H is a subgroup

1. Is H closed under multiplication?
2. Does H has an identity?
3. Does each element have an inverse?
4. Is multiplication associative?

Prove H is a subgroup

1. Is H closed under multiplication? Yes
2. Does H has an identity? Yes
3. Does each element have an inverse? Yes

$$A = \begin{pmatrix} a & c \\ 0 & d \end{pmatrix} \quad \text{then} \quad A^{-1} = \frac{1}{ad} \begin{pmatrix} d & -c \\ 0 & a \end{pmatrix} \tag{0.97}$$

Generator

Let g be an element of a group G then we write

$$\langle g \rangle \tag{0.98}$$

for the smallest subgroup in G that contains g, and if A is a subset of G then we write

$$\langle A \rangle \tag{0.99}$$

for the smallest subgroup of G that contains A.

Examples

Let us return to our favourite group S_3

$$a = (1, 2, 3) \quad \text{and} \quad b = (2, 3) \tag{0.100}$$

	e	a	a^2	b	ab	a^2b
e	e	a	a^2	b	ab	a^2b
a	a	a^2	e	ab	a^2b	b
a^2	a^2	e	a	a^2b	b	ab
ab	ab	b	a^2b	a	e	a^2
a^2b	a^2b	ab	b	a^2	a	e

What are $\langle a \rangle$ and $\langle b \rangle$?

Subgroup

Easy

$$\langle a \rangle = \left\{ a, a^2, a^3 \right\} = \left\{ a, a^2, e \right\} \tag{0.101}$$

and

$$\langle b \rangle = \left\{ b, b^2 \right\} = \left\{ b, e \right\}. \tag{0.102}$$

Coset

Let H be a subgroup of G then the *right coset* of H by $g \in G$ is the set

$$Hg = \{hg \mid h \in H\} \tag{0.103}$$

The Cosets of S_3 by $\langle a \rangle$

Work out the cosets of S_3 by $\langle a \rangle$.

order

The number of elements in a group is termed its order and is denoted $|G|$, For example:

1. $|S_3| = 6$
2. $|\langle a \rangle| = 3$
3. $|\langle b \rangle| = 2$

Assess Security of Diffie-Helman

Read up on this, I will cover it later

CW1

Your assignment is out.

RSA Encryption

Today we discuss the RSA encryption algorithm (it is named after its creators Rivest, Shamir and Adleman), it is the most important and widely used public-key encryption algorithm. When you communicate securely over the internet, for example with bill payments and online shopping, then the algorithm used is RSA (or some small variant).

Theoretical Ideas

The algorithm rests on three theoretical concepts

1. The evaluation of inverses $\mod n$, recall for $x \in \mathbb{Z}$ then its multiplicative inverse $\mod n$ (if it exists) was the y such that $xy = 1 \mod n$.
2. The Euler function $\phi(n)$, recall

$$\phi : \mathbb{N} \to \mathbb{N}, n \mapsto \phi(n) = |\{i \mid 1 \leq i \leq n, \gcd(i, n) = 1\}| . \tag{0.104}$$

3. Euler's theorem: if a is invertible $\mod n$ then

$$a^{\phi(k)} = 1 \mod k. \tag{0.105}$$

Properties of ϕ

To get you thinking.

$$\phi : \mathbb{N} \to \mathbb{N}, n \mapsto \phi(n) = \left|\{i \mid 1 \leq i \leq n, \gcd(i, n) = 1\}\right|. \tag{0.106}$$

1. What is $\phi(p)$ for p prime?
2. Verify directly $\phi(15) = \phi(3)\phi(5)$
3. Prove $\phi(p^i) = p^{i-1}(p-1)$

Answers

1. $\phi(p) = p - 1$ because for every $1 \leq i < p$ we have $\gcd(i, n) = 1$
2. $\phi(15) = \phi(3)\phi(5)$:
   a) $\phi(3) = |\{1, 2\}| = 2$, $\phi(5) = |\{1, 2, 3, 4\}| = 4$,
   b) $\phi(15) = |1, 2, 4, 7, 8, 11, 13, 14|$ because $\gcd(3, 15) = 3$, $\gcd(5, 15) = 4$ $\gcd(6, 15) = 3$, $\gcd(9, 15) = 3$, $\gcd(10, 15) = 5$, $\gcd(12, 15) = 3$.
3. More of a challenge

Trapdoor Function

Let $f : X \to Y$ denote any function, with no loss of generality we assume the map is a bijection. f is termed a *trapdoor* function if the following conditions hold:

1. for every $x \in X$ it is easy (by easy, we mean the number of operations required to evaluate $f(x)$ is small) to determine $f(x)$.
2. for every $f(x) \in Y$ it is difficult to evaluate x.

Trap door Functions

Let $\mathbb{P}$ denote the primes, consider the function

$$f : \mathbb{P} \times \mathbb{P} \to \mathbb{N}, \quad (p_1, p_2) \mapsto_f p_1 p_2; \tag{0.107}$$

in plain terms: we multiply two primes together to form their product. For example if

1. $p_1 = 4575163$ and $p_2 = 4093567$ then $f(p_1, p_2) = 18728736276421$ and (not particularly sophisticated) algorithms can do this multiplication in n^2 steps (n the digit length of p_1 and p_2).
2. the reverse operation: that is, determining p_1 and p_2 from 18728736276421 takes 10^n steps.

so f is quadratic in operation count, but its inverse f^{-1} is exponential in operation count.

Fermat's little theorem

Fermat's little theorem is an interesting result in elementary number theory that is needed in our discussion of the RSA algorithm.

⌜ Let p be prime and a any integer then

$$a^p = a \mod p. \tag{0.108}$$

⌟

Proof of Fermats Theorem

I would like to present a proof of Fermats theorem, and to do this I need two ideas we have meet before :

1. Equivalence relations.
2. Partitioning.

Equivalence Relation

Fix a set X a relation R is simply a subset of $X \times X$ (the idea really is that simple). An *equivalence relation* is a relation that is

1. reflexive: that is $(x, x) \in R$ for all $x \in X$;
2. symmetric: that is if $(x, y) \in R$ then $(y, x) \in R$;
3. transitive: that is if $(x, y) \in R$ and $(y, z) \in R$ then $(x, z) \in R$;

If that is too abstract, observe that: congruent mod n on the integers $\mathbb{Z}$ is an equivalence relation.

Equivalence Relation Partition Sets

Let R denote an equivalence relation on X and define

$$x^\bullet = \{y \in X \mid (x, y) \in R\} \tag{0.109}$$

then the equivalence relation induces a partition in X. Why? Suppose

$$a \in x^\bullet \cap y^\bullet \tag{0.110}$$

so $(x, a) \in R$ and $(y, a) \in R$.

1. $x^\bullet \subseteq y^\bullet$: pick any $z \in x^\bullet$ then $(z, x) \in R$[symmetric] and as $(x, a) \in R$ and $(a, y) \in R$ [symmetric] then $(x, y) \in R$ [transitive] and $z \in y^\bullet$.
2. $y^\bullet \subseteq x^\bullet$: similar to above

So $x^\bullet = y^\bullet$, a picture would help a lot here.

Let G denote a group and H a subgroup then define $(x, y) \in R$ if and only $xy^{-1} \in H$ this is an equivalence relation. Proof:

1. reflexive: $xx^{-1} = 1 \in H$ [because $1 \in H$]
2. symmetric : if $xy^{-1} \in H$ then $xy^{-1} = h$ and we have $yx^{-1} = h^{-1} \in H$
3. transitive: similar

What is the big deal?

Lagrange's theorem

The size of a group G is denored $|G|$. Recall Lagrange's theorem: if H is a subgroup of a group G then $|H| \mid |G|$; that is the order of any subgroup divides the order of the group. For example if a group has order $12 = 2^2 3$ than any subgroup you find will have order 1, 2, 3, 2^2 or $2^2.3$. This allows us to split a group up via cosets.

Important Result

The previous result says for a group G and a subgroup H then

$$|G| = |H| \times \text{ number of cosets} \quad \text{so} \quad \frac{|G|}{|H|} = \text{ number of cosets.} \tag{0.111}$$

In particular if $|G|$ is a prime we deduce it has no subgroups!

proof of Fermat's little theorem

Proof.

Recall Lagrange's theorem: if H is a subgroup of a group G then $|H| \mid |G|$. The non-zero elements of $\mathbb{F}_p$ are under multiplication $\mod p$ a group, thus the order of any non-zero element divides $p-1$. Pick a $a \in \mathbb{F}_p^*$ then for some $n \in \mathbb{N}$ we have $a^n = 1 \mod p$ with $n \mid p-1$ so $p-1 = mn$

$$a^p = a^{mn+1} = aa^{mn} = a(a^n)^m = a1^m = a \mod p. \quad (0.112)$$

□

Variant

We have

$$a^p = a \mod p \quad \text{and thus} \quad a^{p-1} = 1 \mod p. \tag{0.113}$$

Can we have a go at proving this claim.

Fermat's little theorem

Fermat's little theorem has numerous applications. We mention two:

1. It establishes factorization properties for certain numbers, and some of these results are beyond modern day computing power.
2. It allows a method for constructing the multiplicative inverse in a field.

Fermat's little theorem - 1

Why is Fermat's little theorem so interesting. We look at an example from Hoffstein, consider the prime $p = 15485863$ then Fermat's little theorem (from the 16th century) states

$$2^{15485862} = 1 \mod 15485863 \mod p \qquad (0.114)$$

and thus with no computing we know $2^{15485862} - 1$ is divisible by 15485863; a fact that no computer (in the 21st century) could establish directly. These clever number theory ideas give cryptologists 'cold sweats', they live in fear of number theorists!

Inverses using Fermat - 2

Fermat's result

$$a^{p-1} = 1 \mod p. \tag{0.115}$$

implies

$$a^{p-2}a = 1 \mod p, \tag{0.116}$$

and we have a fast way of evaluating the inverse of a, its inverse is a^{p-2} (which we can evaluate quickly using a fast power algorithm).

Euler's formula

We need another technical result. Let p and q be distinct primes and set

$$g = \gcd((p-1),(q-1)) \tag{0.117}$$

then for all a satisfying $\gcd(a, pq) = 1$ we have

$$a^{(p-1)(q-1)/g} = 1 \mod pq \tag{0.118}$$

Proof of Euler

We are told $p \nmid a$ and $g \mid q-1$.

$$\begin{aligned} a^{(p-1)(q-1)/g} &= (a^{p-1})^{(q-1)/g} \mod p \\ &= 1^{(q-1)/g} \mod p \qquad (a^{p-1} = 1 \mod p) \\ &= 1 \mod p \end{aligned} \tag{0.119}$$

We are told $q \nmid a$ and $g \mid p-1$.

$$\begin{aligned} a^{(p-1)(q-1)/g} &= (a^{p-1})^{(q-1)/g} \mod q \\ &= 1^{(q-1)/g} \mod q \qquad (a^{p-1} = 1 \mod q) \\ &= 1 \mod q \end{aligned} \tag{0.120}$$

More Euler

We have

$$a^{(p-1)(q-1)/g} = 1 \mod p \quad \text{and} \quad a^{(p-1)(q-1)/q} = 1 \mod q. \tag{0.121}$$

We deduce

$$a^{(p-1)(q-1)/g} - 1 \tag{0.122}$$

is divisible by both p and q thus it is divisible by pq (this is true because p and q are prime)and

$$a^{(p-1)(q-1)/g} - 1 = 0 \mod pq \tag{0.123}$$

and

$$a^{(p-1)(q-1)/g} = 1 \mod pq. \tag{0.124}$$

The RSA algorithm

We have nearly covered all the theory to code the RSA algorithm, it has been very theory heavy so let us look at the actual algorithm and we will return to a few technical details later.

the usual problem

We have the usual problem: Alice and Bob wish to communicate securely across an insecure channel that Eve has access to.

Bob's Public Key

Anybody who wishes to communicate securely with Bob needs his Public Key. Bob picks two primes p and q (p, $q > 2^{1000}$) evaluates $N = pq$ and picks an encryption exponent e, where e satisfies

$$\gcd(e, (p-1)(q-1)) = 1. \qquad (0.125)$$

Bob's public key is the tuple (that is, it is a pair of numbers) $K_{B,Pu} = (N, e)$.

What Alice Does - Encryption

Alice has a plaintext message m (m an integer) and evaluates

$$c = m^e \mod N, \tag{0.126}$$

c is the ciphertext sent to Bob.

How Bob Decrypts

Bob solves two equations, first

$$ed = 1 \mod (p-1)(q-1). \tag{0.127}$$

The only term in eqn. 0.127 that Bob does not know is d, so this is simply finding the inverse of $e \mod (p-1)(q-1)$. He then evaluates

$$m' = c^d \mod N \tag{0.128}$$

and we find $m = m'$, and he has determined Alice's message.

ElGamal v. RSA

1. The security of ElGamal is linked to the difficulty in solving
$$a^x = b \mod p, \tag{0.129}$$
a, b and p (p prime) all known, and x is unknown.
2. The security of RSA is linked to the difficulty in solving
$$x^e = c \mod N, \tag{0.130}$$
e, c and N all known, and x is unknown.

Authenticity

Recall the four principles:

1. Confidentiality - The message the recipient gets can be proven not to have been read by anyone else.
2. Integrity - The message the recipient gets can be proven not to have been changed since it was encoded.
3. Authenticity - The message the recipient gets can be proven to have been encoded by a positively-identified sender.
4. Non-repudiation - The sender, given a message received by a recipient, cannot validly deny that the message was sent by him or that it was not the original content sent by him.

Authenticity

I mention (glossing over a few details) that in the real world RSA is not quite done as in this lecture or the lab questions.
The fundamental problem is that everyone has access to $K_{B,Pu}$ (Bob's public key) so when the message arrives from Alice then Bob cannot be sure the message was sent by Alice, it could have been Eve.
The resolution?

Resolution

Alice currently encrypts her plain text message with Bob's public key, what happens in the real world is

1. Alice encrypts her plain text message m with her own private key, denote this message $K_{A,Pr}(m)$
2. Alice then encrypts $K_{A,Pr}(m)$ with Bob's public key to get $K_{B,Pu}(K_{A,Pr}(m))$.
3. Bob can decrypt $K_{B,Pu}(K_{A,Pr}(m))$ to get $K_{A,Pr}(m)$.
4. Bob can use Alice's public key to decrypt $K_{A,Pr}(m)$ to get m.

He is now sure he is talking to Alice because the only person who could have sent the message has access to Alice's private key. This is the 'two way' handshake that allows you and (say) amazon to be sure you are talking to each other

Summary: RSA

1. We have developed most of the theory required to show that the RSA algorithm does encrypt and decrypt correctly.
2. Factorizing $N = pq$ when you N but do not know p or q is a very difficult task, and for that reason RSA is believed to be secure to attack from Eve.

Four Key Goals

Recall the four key goals :

1. Confidentiality : information is kept secret except from the authorised parties;
2. Integrity: data tampering is easily detectable;
3. Authentication: it is possible to determine the sender of the message.
4. Non-repudiation: the sender of a message can not deny the contents of a message

Note authentication and non-repudiation are not the same, the distinction is subtle.

Digital Signatures

Today we discuss Digital Signatures, they make use of ideas that are very similar to the RSA algorithm and the ElGamal algorithm (and we have already coded them).

Why bother with Signatures?

To date we have discussed algorithms that resolve the following problems:

1. the encryption and decryption of data;
2. showed it is possible to establish a shared secret between two people which they could then use to encrypt data.

All of this scenarios have involved Bob and Alice communicating securely and Eve has played the role of the trouble maker.

Why bother with signatures?

The above techniques allowed Alice and Bob to establish a shared key and thus communicate securely. The problem is that in many transactions whilst Alice and Bob need to communicate securely not only do they not trust Eve they do not trust each other. As such there is an interest in a method that would verify (to say a judge) that Alice did send a message to Bob and Bob cannot deny its contents. This can be achieved by a digital signature.

What is a Digital Signature?

As the term signature might suggest, a digital signature indicates that the person who signs a document agrees with the contacts of the document. We introduce two new characters:

1. Sam : signs a document.
2. Victor: verifies the document has been signed by Sam.

Obvious Criteria

We require the two obvious criteria:

1. Only Sam can create Sam's signature.
2. It is easy for anybody to prove that the document is signed by Sam.

Notation

1. K_{Pr} Sam's private signing key.
2. K_{Pu} Sam's public verification key.
3. Sign : an algorithm that takes a document D and a private key K_{Pr} and produce a signed document
4. Verify: an algorithm that takes the signed document and a verification key K_{Pu} and returns true if Sam signed the document or false if Sam did not sign the document.

The Algorithm

The algorithm described is from the original RSA paper, and is termed the RSA digital signature method. Sam does the following, he picks two large primes p and q he forms $N = pq$ and picks a v a verification exponent. Sam solves for s in

$$sv = 1 \mod (p-1)(q-1). \tag{0.131}$$

Note N and v are public knowledge.

Euler's Formula

Recall for distinct primes p and q then if

$$g = \gcd(p-1, q-1) \tag{0.132}$$

then

$$a^{(p-1)(q-1)/g} = 1 \mod pq \tag{0.133}$$

for all a satisfying $\gcd(a, pq) = 1$.

Signing and Verifying

1. Denote by D ($1 < D < N$) the document to be signed
2. The signed document is

$$S = D^s \mod N. \tag{0.134}$$

3. Victor has knowledge of S, N and v so he can evaluate

$$S^v \mod N \tag{0.135}$$

and this is D because

$$S^v = D^{sv} = D \mod N. \tag{0.136}$$

Covered the theory of this last week, page 170 of the text by Paar.

Solving Equations in Finite Fields

In Cryptography we are interested in solving equations of the form

$$a^x = b \mod c \tag{0.137}$$

for x, where all other terms known in eqn. 0.137 are known. We have already solved these equation by brute force, we now look at a new solution technique *collision algorithms* and *iterative algorithms*. To discuss collision algorithms we need to introduce some now ideas and terminology.

Probability

Formally a probability space is a function

$$\mathbb{P} : \Omega \to [0, 1], \quad A \mapsto \mathbb{P}(A). \tag{0.138}$$

1. Ω is the sample space;
2. For $A \in \Omega$ we read $\mathbb{P}(A)$ as the probability event A occurred.

Probability

The formal definition takes some getting used to, a simple example will help. Consider the following sample space

$$\omega = \{(m, n) \mid 1 \leq m, n \leq 6\}; \tag{0.139}$$

it has a natural representation as the events generated when you roll two dice, so $(2, 3)$ is the first row is 2 and the second throw is a 3 and the probability of this is

$$\mathbb{P}((2, 3)) = \frac{1}{36}. \tag{0.140}$$

Another example this time of a compound event

$$\mathbb{P}(\text{second throw } = \text{first throw}) = \frac{1}{6}. \tag{0.141}$$

problem

A fair coin is tossed 10 times. What are the probabilities of the following events:

1. $E_1 = \{\text{the first 5 tosses are heads}\}$
2. $E_2 = \{\text{the first 5 tosses are heads and the rest are tails}\}$
3. $E_3 = \{\text{exactly 5 tosses are heads}\}$

collision algorithms

There are forty people in a room.

1. What is the probability that someone has the same birthday as you?
2. What is the probability that at least two people in the room have the same birthday?

The answers are different, this question is termed ‘the birthday paradox’

Wrong Answer

- Q: There are 40 people in a room, what is the probability that someone has the same birthday as you?
- A: the answer is not $\frac{40}{365}$, if you think it is then what would your answer be if there were 500 people in the room?

Answer Part 1

$$\begin{aligned}
&\mathbf{P}(\text{some one has your birthday}) = \mathbf{1} - \mathbf{P}(\text{no one has your birthday}) \\
&= 1 - \prod_{i=1}^{40} \mathbf{P}(\text{person } \mathbf{i} \text{ does not have your birthday}) \\
&= 1 - \left(\frac{364}{365}\right)^{40} = 10.4\%
\end{aligned} \tag{0.142}$$

Answer Part II

The second question does not 'fix' a birthday

$$\begin{aligned}&\mathbf{P}(\text{two people share a birthday}) = \mathbf{1} - \mathbf{P}(\text{all 40 have different b'days})\\&= 1 - \prod_{i=1}^{40} \mathbf{P}(\text{person } \mathbf{i} \text{ birthday different from } \mathbf{1}, \ldots \mathbf{i-1})\\&= 1 - \left(\frac{365}{365}\,\frac{364}{365}\,\frac{363}{365}..\frac{326}{365}\right) = 89\%\end{aligned} \tag{0.143}$$

Summary

1. It requires 23 people to have a better than 50 % chance of a matched birthday
2. It requires 253 people to have a better than 50 % chance of someone having your birthday.

In a collision algorithm we (typically) generate two lists, a match (for example, a shared birthday) across two lists corresponds to a solution to our problem, and while we can not say for definite that we will have reached a solution by list length j we can say

1. If the list length is greater than m there is a $x\%$ probability of their being a solution in the list

This technique is popular in cryptography

Relevance to Cryptography

We can turn the argument around, suppose we are given a number m and ask if it is prime then we could do the following:

1. pick a random number $x < m$ if $x \mid m$ then the number is not prime and the process terminates
2. go back to step 1.

Suppose the algorithm had not terminated after x-steps then could we say something about its probability of being prime?

Searching Lists

We proceed in a abstract manner. How the lists are generated is obviously dependent upon the algorithm under consideration, but we can say something about matches in general lists. We wish a statement of the type:

- After completing n iterations of this process there is a $x\%$ probability you will have achieved y.

or to keep you focused

- After completing n iterations of this process there is a $x\%$ probability at least one of the iterates is a prime number.

collision theorem

⌜A box contains N balls, n are red and $N - n$ are blue. Bob

1. Removes a ball, notes the colour;
2. Replaces ball in box.

The above process is repeated m times then

a) The probability that Bob selected at least one red ball is

$$\mathbb{P}(\text{at least one red}) = 1 - \left(1 - \frac{n}{N}\right)^m \qquad (0.144)$$

b) A lower bound for

$$\mathbb{P}(\text{at least one red}) = 1 - \exp\left(\frac{-mn}{N}\right) \qquad (0.145)$$

Note m does not have to be be particularly large to ensure $\mathbb{P}(\text{at least one red})$ is near 1, eqn. 0.145 is important it tells us how long we need to keep picking balls to ensure there is a probability greater that (say) 0.99999 that we have found an answer

⌟

proof of collision theorem

The proof is straight forward

$$\begin{aligned}
\mathbb{P}(\text{at least one red in } m \text{ attempts}) &= 1 - \mathbb{P}(\text{all } m \text{ choices are blue}) \\
&= 1 - \prod_{i=1}^{m} \mathbb{P}(\text{choice } i \text{ is blue}) \\
&= 1 - \prod_{i=1}^{m} \left(\frac{N-n}{N}\right) \\
&= 1 - \left(1 - \frac{n}{N}\right)^{m}
\end{aligned} \tag{0.146}$$

proof of second part of collision theorem

The following is true. For all $x \in \mathbb{R}$

$$\exp(-x) \geq 1 - x \tag{0.147}$$

Thus setting $x = \frac{n}{N}$

$$\exp(\tfrac{-n}{N}) \geq 1 - \tfrac{n}{N} \quad \text{and} \quad \exp(\tfrac{-nm}{N}) \geq \left(1 - \tfrac{n}{N}\right)^m. \tag{0.148}$$

$$\begin{aligned} \mathbb{P}(\text{at least one red in } m \text{ attempts}) &= 1 - \left(1 - \tfrac{n}{N}\right)^m \\ &\geq 1 - \exp(\tfrac{-nm}{N}). \end{aligned} \tag{0.149}$$

Note this bound is independent of any algorithm, that is the balls are drawn are random.

Red v Blue

I kept the discussion to red and blue balls to keep it abstract. The important point here is that you can split the set you are searching in into two disjoint sets: Examples:

1. Red balls = prime numbers, Blue ball = non prime numbers;
2. Red balls = divisible by 11, 17 and 19 , blue balls = not divisible by at least one of 11, 17 or 19.

And note again, the bound in eqn. 0.149 is just randomly drawn balls.

nomenclature

1. Collision algorithms;
2. Meet-in-the-middle algorithm/attack;
3. Square root algorithm/attack.

The above terms are all used to describe collision algorithms.

Iteration

An iterative process is a function such that the state of the system at stage $n+1$ is dependent on the system at stage n. Given a start point x_1 we generate $\{x_i \mid i \in \mathbb{N}\}$. For example, let $x_1 = 1$

$$f : \mathbb{N} \to \mathbb{R}, \quad x \mapsto x^2 + 1. \tag{0.150}$$

then

$$\begin{aligned} x_1 &= 1 \\ x_2 &= (x_1)^2 + 1 = 1^2 + 1 = 2 \\ x_3 &= (x_2)^2 + 1 = 2^2 + 1 = 5 \\ x_4 &= (x_3)^2 + 1 = 5^2 + 1 = 26 \end{aligned} \tag{0.151}$$

Pollard's Rho Method

Let $f : X \to X$, $x \mapsto f(x)$ with X a finite set. For any $x_0 \in X$ define

$$x_1 = f(x_0), \quad x_2 = f(x_1), \quad x_3 = f(x_2), x_4 = f(x_3), \ldots \tag{0.152}$$

that is $x_n = f \circ f \circ f \ldots f(x_0)$, where $\circ$ denotes function composition. We seek

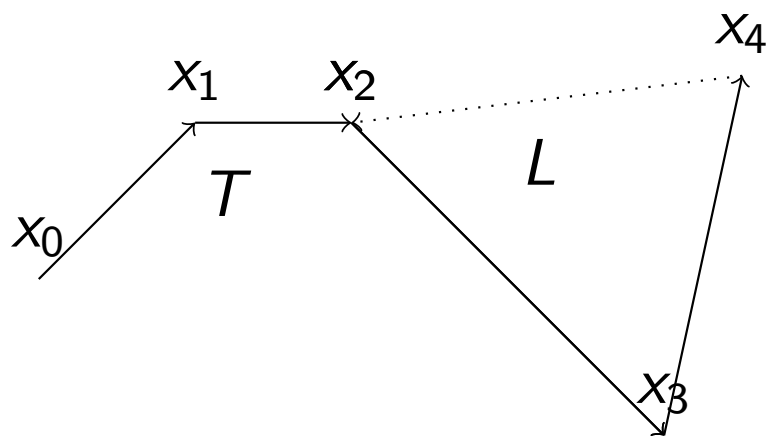


Figure: Iterative maps in finite sets need to 'return' to an earlier point, the loop L is the closed circle, and the tail T consists of those points not in the loop M

Pollard's ρ method

Let X be a finite set $f : X \to X$, $x \mapsto f(x)$ a function (any function!) and x_0 a start point (as we often work mod n then X is finite).

1. If the orbit $\{x_0, x_2, \ldots\}$ of x_0 has a tail of length T and a loop of length L then $x_{2i} = x_i$ for some $1 \leq i < T + L$.
2. If the map f is sufficiently random (I will expand on this) then the expected length E of $L + M$ before we find a solution is

$$E(T + L) \approx 1.2\sqrt{|X|}. \qquad (0.153)$$

Iterative algorithms

1. Start algorithm with a 'guess' x_1.
2. Update to $x_{i+1} = f(x_i)$ (the choice of f dependent on the problem).
3. Check some criteria : if criteria is satisfied we have an answer and quit.
4. Increment i.
5. Return to 2.

Solving Equations

We illustrate iterative equation solving with its simplest usage: *the fixed point method*. Suppose we wished to solve

$$x^3 - x - 1 = 0 \quad \text{then we are interested in} \quad f(x) = (1+x)^{1/3}. \tag{0.154}$$

Why? Because a fixed point of f corresponds to a solution to $x^3 - x - 1$. There are certain mathematical criteria that have to be in place (and they are) and the sequence

$$x_1, f(x_1), f^2(x_1), f^3(x_1) \tag{0.155}$$

will converge to a solution with a 'reasonable' choice of x_0.

Reading Exercise

It is a reading exercise/Homework to read up on Pollard's Rho method. I will illustrate the iterative mechanism with a simple example that you will code in labs.

Summary and Last Direction

1. We have covered the main encryption public key algorithms.
2. Today, we look at some of the newer encryption techniques, specifically
   a) Elliptic Curve Cryptography
   b) Quantum Crptography.
3. We explicitly construct $\mathbb{F}_{3^2}$.

Elliptic Curve Cryptography

We have a look at Elliptic Curve Cryptography, a newer approach to encryption that was developed after RSA, and will come to the fore in the next few years.
A misnomer: elliptic curves are not ellipses.
`http://www.maa.org/sites/default/files/pdf/upload_libra`

Why bother?

RSA is a widely-supported encryption technique with no feasible attack strategies. Elliptic curve cryptography also has no known attack strategies, but it has one big advantage Elliptic keys are smaller (reducing storage and transmission requirements) an elliptic curve group can use a 256-bit elliptic curve public key and it is believed to be as secure as 3072-bit RSA public key.
(same security, with a key a tenth of the length).

Elliptic Curves

An *elliptic curve* is a curve of the form

$$y^2 = x^3 + Ax + B \tag{0.156}$$

where A and B are parameters.

In a Nutshell

The RSA algorithm and the ElGamal algorithm involved us solving equations of the form

$$x^a = b \mod n. \tag{0.157}$$

The theory required us to develop a little of the structure theory of finite fields. Recall to each prime p there is one and only one field of size p^n, n a positive integer, and we wrote this $\mathbb{F}_{p^n}$.
Elliptic curve cryptogtaphy uses ideas from two dimensional geometry to create a new group structure that is used in encryption.

Example

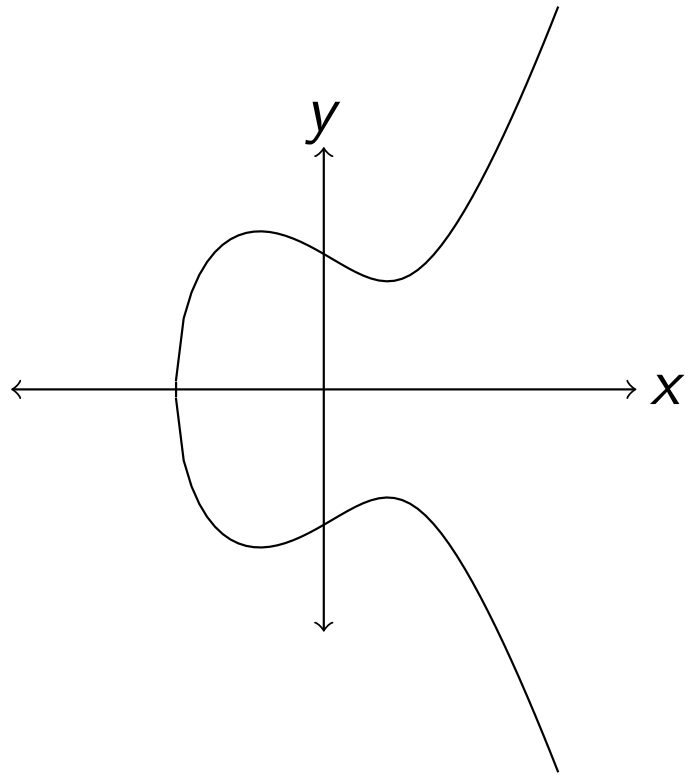


Figure: The elliptic curve $y^2 = x^3 + Ax + B$, $A = -3$, $B = 3$ $y^2 = x^3 - 3x + 3$.

Points on the Curve

The shape of the curve $y^2 = x^3 + Ax + B$ is dependent on A, B. For example consider the equation

$$y^2 = x^3 - 15x + 18. \tag{0.158}$$

Which of the following points lies on the curve in eqn. 0.158?

1. $P = (7, 16)$
2. $Q = (1, 2)$
3. $R = (2, 1)$

Just plug in the numbers

$P = (7, 16)$

We have

$$y^2 = x^3 - 15x + 18, \tag{0.159}$$

and setting $x = 7$ we find

$$x^3 - 15x + 18 = 7^3 - 15.7 + 18 = 256 \tag{0.160}$$

and $y = 16$ gives

$$y^2 = 256. \tag{0.161}$$

$Q = (1, 2)$

We have

$$y^2 = x^3 - 15x + 18, \tag{0.162}$$

and setting $x = 1$

$$x^3 - 15x + 18 = 1^3 - 15 + 18 = 4 \tag{0.163}$$

and $y = 2$

$$y^2 = 4. \tag{0.164}$$

Placing a Group Structure on the Curve

The new insight (Koblitz and Miller 1985) is that we can place a group structure on an elliptic curve. Recall a group consists of a set G and a binary operation $\circ : G \times G \to G$, $(a, b) \mapsto a \circ b$. The binary operation is required to satisfy the following properties:

1. $\circ$ is an associative operation, that is $a \circ (b \circ c) = (a \circ b) \circ c$.
2. There exists a $1 \in G$ such that $1 \circ g = g \circ 1 = g$ for all $g \in G$; the element 1 is termed the group identity.
3. To each $g \in G$ there exists a $g^{-1} \in G$ such that $g \circ g^{-1} = 1$; the element g^{-1} is termed the inverse of g.

Group addition $\oplus$

The group we construct will be abelian and in this case the group operation is usually denoted by $+$; however if you are given two points (a, b) and (c, d) in the plane $\mathbb{R}^2$ then $+$ is usually taken to mean

$$(a, b) + (c, d) = (a + c, b + d). \tag{0.165}$$

And that is not how we will define the group operation for an elliptic curve, we will use $\oplus$.

Group addition $\oplus$

Given two points on an elliptive curve P and Q we need to define their sum $R = P \oplus Q$, and when this is done we need to check all the axioms of a group are satisfied.

Group addition $\oplus$

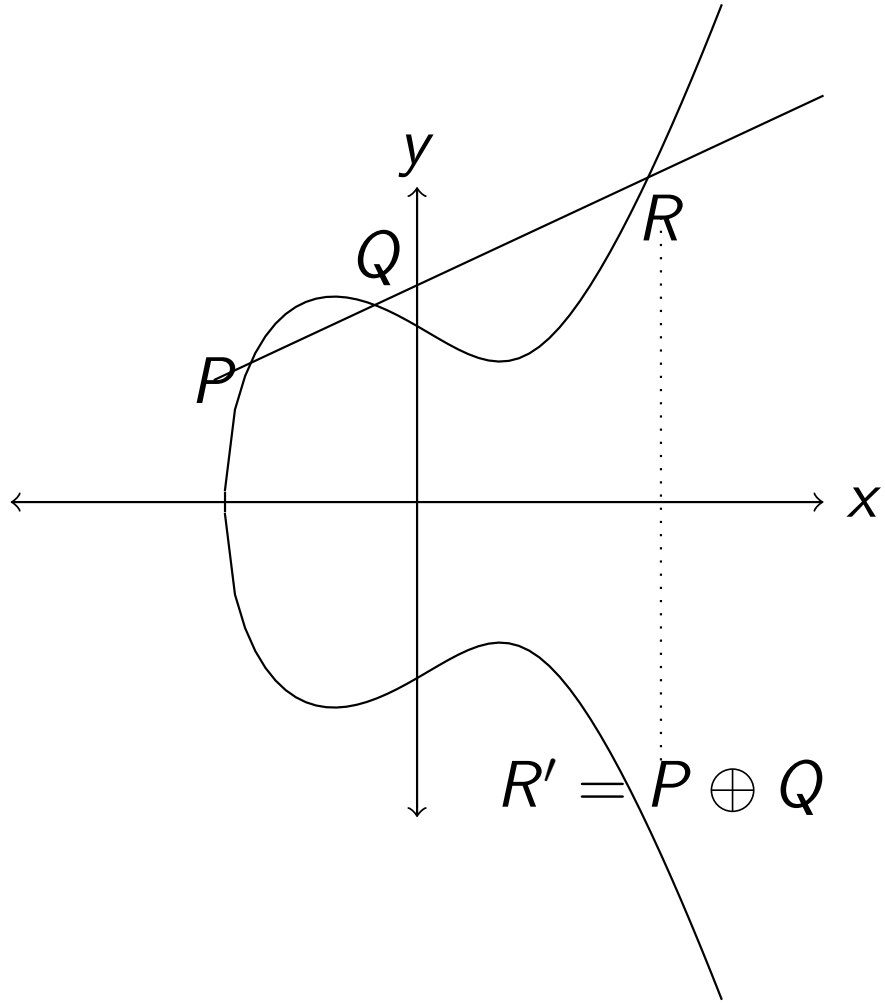


Figure: The elliptic curve $y^2 = x^3 + Ax + B$, $A = -3$, $B = 3$ $y^2 = x^3 - 3x + 3$.

Group addition in Words

Let points P and Q lie on the curve

1. Draw the straight line through P and Q it will will meet the curve at a third point R
2. Reflect R in the y-axis, this point is defined to be $P \oplus Q$.

Example

Let E be the elliptic curve

$$y^2 = x^3 - 15x + 18. \tag{0.166}$$

Let $P = (7, 16)$ and $Q = (1, 2)$. What is $P \oplus Q$?.
Give it a go —you need a little geometry

A bit more geometry

In case your mathematics is a little rusty, we need to describe the equation of the line that joins two points

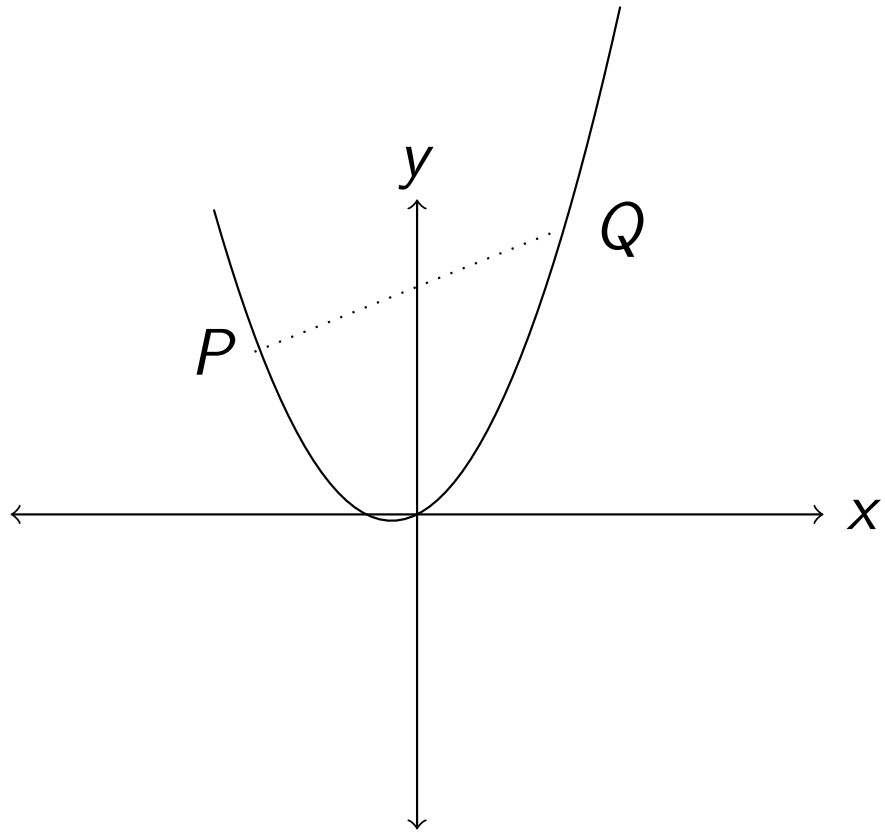


Figure: We require the equation of the straight line between points P and Q.

A little geometry

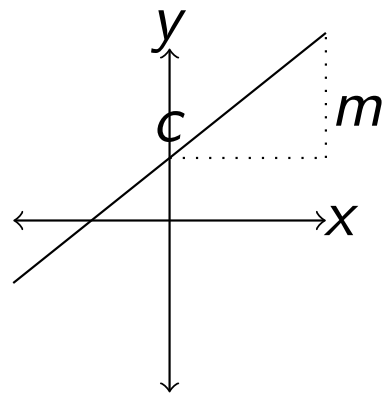


Figure: $y = mx + c$

Recall the equation of a line can be written

$$y = mx + c; \tag{0.167}$$

where m is the gradient of the line, and c is its intercept on the y axis. Consider points $P = (x_1, y_1)$ and $Q = (x_2, y_2)$.

Equation of Line

Using the gradient and intercept idea, if we know two points on the straight line (say) (x_1, y_1) and (x_2, y_2) then the line is described by the equation

$$y = y_1 + \frac{(y_2 - y_1)}{x_2 - x_1}(x - x_1) \tag{0.168}$$

So for $(7, 16)$ and $(1, 2)$ we deduce

$$y = 16 + \frac{(2-16)}{(1-7)}(x - 7) \quad \text{and rearranging} \quad y = \frac{7x}{3} - \frac{1}{3}. \tag{0.169}$$

Where are we now?

Recall we seek the point R, and we know P, Q and

$$y = \frac{7x}{3} - \frac{1}{3} \quad \text{and} \quad y^2 = x^3 - 15x + 18. \tag{0.170}$$

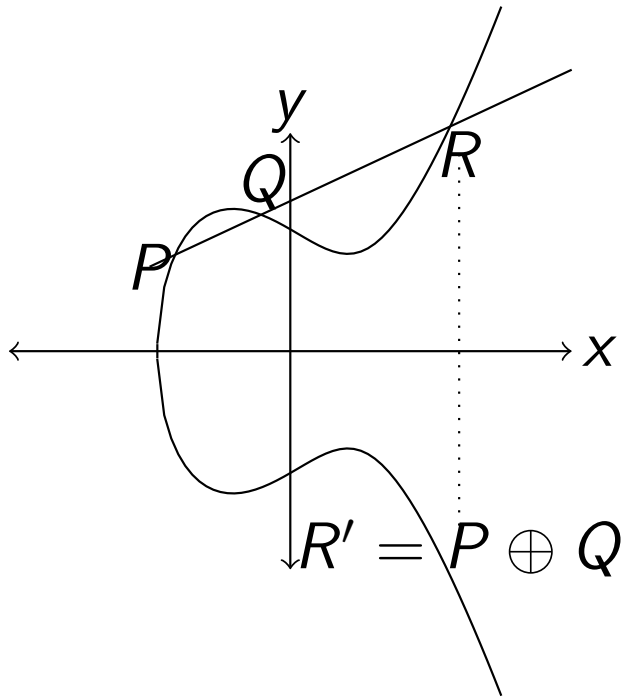


Figure: An elliptic curve

cubic equations

We seek the simultaneous solution of the two equations

$$y = \frac{7x}{3} - \frac{1}{3} \quad \text{and} \quad y^2 = x^3 - 15x + 18. \qquad (0.171)$$

The general solution of a cubic is readily available online
`https://en.wikipedia.org/wiki/Cubic_functionGeneral_fo`

cubic equations

We seek the solution to

$$\left(\frac{7x}{3} - \frac{1}{3}\right)^2 = x^3 - 15x + 18 \qquad (0.172)$$

Can we multiple it out and deduce the cubic please?

Some working out

$$\begin{aligned} \left(\frac{7x}{3} - \frac{1}{3}\right)^2 &= x^3 - 15x + 18 \\ \frac{49x^2}{9} - \frac{14x}{9} + \frac{1}{9} &= x^3 - 15x + 18 \\ 0 = \quad x^3 - \frac{49}{9}x^2 - \frac{121}{9}x + \frac{161}{9} & \end{aligned} \tag{0.173}$$

There is a general solution for a cubic, we already know two solutions $P = (7, 16)$ and $Q = (1, 2)$ thus

More Working Out

Two solutions are $P = (7, 16)$ and $Q = (1, 2)$ thus

$$x^3 - \frac{49}{9}x^2 - \frac{121}{9}x + \frac{161}{9} = (x-7)(x-1)(x+\frac{23}{9}) \quad (0.174)$$

and $x = \frac{-23}{9}$ so

$$y = \frac{7x}{3} - \frac{1}{3} \quad \text{implies} \quad y = \frac{7.(-23)}{27} - \frac{3}{27} = \frac{170}{27}. \quad (0.175)$$

And finally reflecting the y point

$$P \oplus Q = \left(\frac{-23}{9}, \frac{-170}{27}\right) \quad (0.176)$$

Caveats

I have glossed over a few details:

1. We need to add an identity element $O_\oplus$ to the solutions (recall a group needs an identiry element, it is a group axiom) this is no problem $O_\oplus + P = P = P + 0_\oplus$.
2. If $P = Q$ we are more trouble, because there is no line between P and P, so we can not solve the equation as we did earlier.

Elliptic Curve Addition Algorithm

Consider

$$y^2 = x^3 + Ax + B \mod \mathbb{F}_p \tag{0.177}$$

and let P_1 and P_2 be solutions to eqn. 0.177.

1. If $P_1 = O_\oplus$ then $P_1 + P_2 = P_2$.
2. If $P_2 = O_\oplus$ then $P_1 + P_2 = P_1$.
3. Otherwise let $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$
   a) If $x_1 = x_2$ and $y_1 = -y_2$ then $P_1 + P_2 = O_\oplus$
   b) Otherwise, define λ to be

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} \quad \text{if } x_1 \neq x_2, \quad \frac{3x_1^2 + A}{2y_1} \quad \text{if } x_1 = x_2 \tag{0.178}$$

set $x_3 = \lambda^3 - x_1 - x_2$ and $y_3 = \lambda(x_1 - x_3) - y_1$ and define

$$P_1 \oplus P_2 = (x_3, y_3). \tag{0.179}$$

Online

`http://www.christelbach.com/eccalculator.aspx`

Constructing $\mathbb{F}_{3^2}$

CW1 requires you construct $\mathbb{F}_{3^2}$ recall this is the unique field of size 3^2, and recalling the definition of a field

1. $\mathbb{F}_{3^2}$ is under addition an abelian group
2. the non-zero elements of $\mathbb{F}_{3^2}$ are under multiplication an abelian group.

So $\mathbb{Z}_9$ is not a field! Because

$$3 \times 3 = 0 \mod 9 \tag{0.180}$$

skew symmetric

A matrix M is termed skew-symmetric if its off-diagonal elements satisfy $M_{i,j} = -M_{j,i}$. A property of 2×2 skew-symmetric matrices with a single number on their diagonal is they commute. Please prove this

commutativity of skew-symmetric 2×2 matrices

In general multiplication of matrices need not be abelian however for skew-symmetric matrices we find

$$\begin{pmatrix} x_1 & y_1 \\ -y_1 & x_1 \end{pmatrix} \begin{pmatrix} x_2 & y_2 \\ -y_2 & x_2 \end{pmatrix} = \begin{pmatrix} x_1x_2 - y_1y_2 & x_1y_2 + x_2y_1 \\ -(x_1y_2 + x_2y_1) & (x_1x_2 - y_1y_2) \end{pmatrix}. \tag{0.181}$$

Constructing $\mathbb{F}_{3^2}$

The reason for proving the commutativity is that it saves us a lot of work, in particular when we work out the field table in the question sheet we know $AB = BA$ so we need only work out AB.

The following matrices are a field of cardinality nine over $\mathbb{Z}(3)$

$$a = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \quad b = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \quad c = \begin{pmatrix} 0 & 2 \\ -2 & 0 \end{pmatrix},$$
$$d = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad e = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}, \quad f = \begin{pmatrix} 1 & 2 \\ -2 & 1 \end{pmatrix},$$
$$g = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}, \quad h = \begin{pmatrix} 2 & 1 \\ -1 & 2 \end{pmatrix}, \quad i = \begin{pmatrix} 2 & 2 \\ -2 & 2 \end{pmatrix}. \tag{0.182}$$

Quantum Cryptography

Quantum Cryptography depends crucially on Heisenberg's uncertainity principle
`https://www.youtube.com/watch?v=a8FTr2qMutA`

Quantum Cryptography

Quantum Cryptography - does it lead to a one time pad?
https://www.youtube.com/watch?v=UiJiXNEm-Go

Quantum Cryptography

I think the most important idea here is that Eve

Zero Knowledge Proof Protocol

Suppose Peter has a piece of information and he wishes to prove to Vic that he does indeed have this information, but (and this is the important bit) he does not wish to tell Vic what the information is.
This seems impossible, let us look at a toy example

Peter and his Cave

Consider Peter in his circular cave, there is a magic word which he needs to shout 'Abr A C dAd Br' (whatever) that allows Peter to open the door at point B. He wishes to prove to Vic that he knows this word, but he does not want Vic to know what the word is — and because sound travels far in a cave, he cannot even let Vic be in the cave for fear of Vic over hearing the magic word.

Pete's Cave

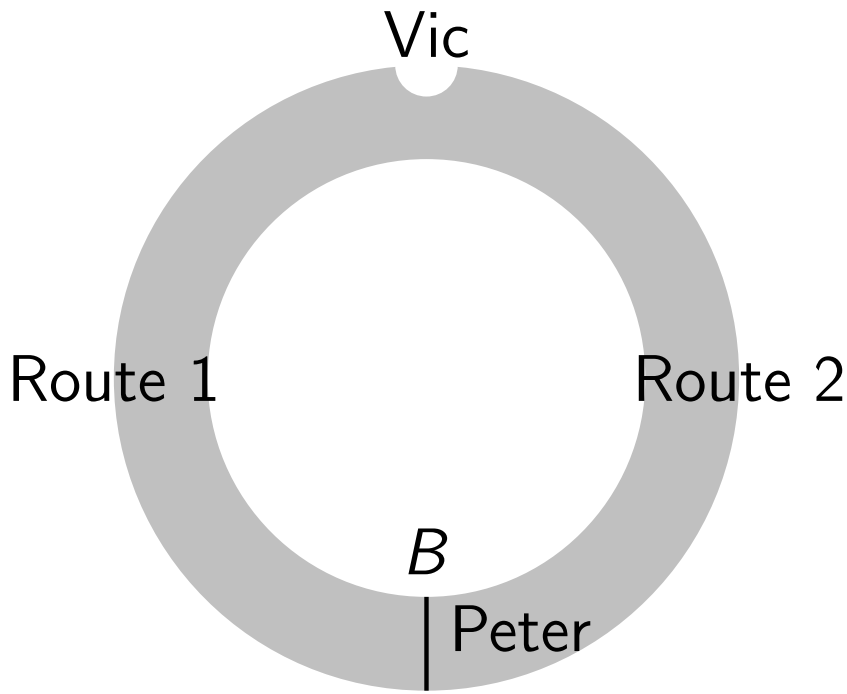


Figure: Pete's cave is the gray circle, Vic stands by the exit, and the magic door is at B

Start of a Protocol

Vic (Vic the verifier), is stood by the exit and he can see someone if they walk up Route 1 or Route 2. He phones Vic on his mobile (which works in a cave :-)) and does the following:

1. Vic flips a coin. If it is heads he tells Peter to come up via Route 1, and if it is tails he tells Peter to come up via Route 2.

Vic hangs up. Now if Peter has the magic word he can always do what is requested; however if Peter does not have the magic word he can achieve what he is asked 50 % of the time, it depends entirely on which side of the door B he is stood.

Probabilistic Protocol

We now have a way of establishing if Peter knows the magic word. The scenario stated earlier is played out n times: that is

1. Peter enters the cave, and moves to the door (Victor does not see the route Peter took to reach the door at B.)
2. Victor approaches the entrance of the cave, flips a coin, phones Peter, and specifies the route to be taken to the exit.

If Peter has the magic word then he will be 100% successful in adhering to all n of Vic's instructions; if he does not have the magic word then the chances of him adhering to all n instructions are 2^{-n}.

Zero Knowledge Proof Protocol

The set of rules just discussed do the following:

1. They allow Vic to establish if Peter does or does not know the stated fact (here the password for the cave entrance).
2. No matter how long you run the protocol for $n = 10$, $n = 10^{100000}$ Vic learns absolutely nothing about what the password is.

Such a procedure is termed a *Zero Knowledge Proof Protocol*.

Important Point

It is crucial that the coin flips are random and Peter does not know what they are in advance, if Peter knew that Vic was going to call: Heads, Heads, Tails, Heads then on each of those iterations he would position himself on the side of the door required to achieve the correct exit.

The Key Ideas

Having established the basic idea we look at some real examples of the zero-knowledge protocol. The common theme is that in all cases:

1. Vic (the verifier) poses a problem/question to Peter (the prover).
2. If Peter does have the knowledge he claims then answering the question is a simple task.
3. If Peter does not have the knowledge he claims then he is posed with a problem that he has no more that 50 % chance of getting correct.

The procedure is repeated by Vic until some probabilistic bound is reached; that is Vic is convinced that Peter is telling the truth to (say) 99.99999%.

Moving towards a Real World Scenario

We need to transplant the magic cave idea into a setting more amenable to computer science. As such let G be a very large group and $g \in G$ a generator ($|G|$ a large prime works here).

The Problem

Peter claims to have a solution x to the equation

$$g^x = A \tag{0.183}$$

(g and A are known to everybody) and he wishes to prove to Vic that he knows x, but is not prepared to tell Vic what x is. Can it be done?

Zero Knowledge Proof Protocol

The following are common knowledge G (the group) and a fixed element $g \in G$. Peter keeps x secret but passes $A = g^x$ to Vic.

1. Peter picks a random integer r and sends $C = g^r$ to Vic.
2. Vic flips a coin:
   a) head: Vic asks Peter for the value r
   b) tail: Vic asks Peter for the value $x + r$
3. Peter does as requested.
4. If it was heads then Vic evaluates g^r and checks it equals C, and if it was tails then Vic evaluates g^{x+r} and checks it equals AC.
5. We return to 1 (as often as Vic requests) and start again.

Peter's problem

Let us recap, we have equation

$$A = g^x \quad \text{and} \quad C = g^r \tag{0.184}$$

Peter has passed A and C to Vic, and g is known to everyone. Suppose Peter does not know the value of x, and tries to trick Victor into believing that he does.
The algorithm allows Peter to get away with cheating exactly half the time. Let us think about why this is so.

Sneaky Peter

Here is the loop Peter is obliged to be in:

1. Peter picks a random integer r and sends $C = g^r$ to Vic.
2. Vic flips a coin:
   a) head: Vic asks Peter for the value r
   b) tail: Vic asks Peter for the value $x + r$
3. Peter does as requested.
4. If it was heads then Vic evaluates g^r and checks it equals C, and if it was tails then Vic evaluates g^{x+r} and checks it equals AC.
5. We return to 1 (as often as Vic requests) and start again.

If Peter is trying to cheat, he of course does not have to follow the algorithm — he merely sends data that will best suit his agenda. So what should 'sneaky'Peter do?

Sneaky Peter Thinks Heads

I put in red what Peter actually sends to indicate he is trying to subvert the protocol. Suppose Peter thinks Vic will call heads:

1. Peter picks a random integer r and sends $C = g^r$ to Vic.
2. Vic flips a coin:
   a) head: Vic asks Peter for the value r
   b) tail: Vic asks Peter for the value $x + r$
3. Peter does as requested. Peter sends r.
4. If it was heads then Vic evaluates g^r and checks it equals C, and if it was tails then Vic evaluates g^{x+r} and checks it equals AC.

In this case, Peter adheres to the protocol, and if he guessed correctly with heads then he passes the test. If Vic however calls tails, then Peter is in trouble because he does not know x so does not know $x + r$.

Sneaky Peter Thinks Tails

Suppose Peter thinks Vic will call tails.

1. Peter picks a random integer r and sends $C = g^r$ to Vic. [Peter does no such thing he sends $C = \frac{g^r}{g^x}$]
2. Vic flips a coin:
   a) head: Vic asks Peter for the value r
   b) tail: Vic asks Peter for the value $x + r$
3. Peter does as requested. [Peter sends r not $x + r$ he does not know x!]
4. If it was heads then Vic evaluates g^r and checks it equals C, and if it was tails then Vic evaluates g^{x+r} (in reality g^r) and checks it equals AC.

If Peter guessed correctly with tails then he passes the test.

$$AC = g^x . \frac{g^r}{g^x} = g^r \tag{0.185}$$

If Vic however calls heads, then Peter is in trouble because $g^r \neq C$

Read Through

Read through the slides a few times, the idea is subtle. In summary:

1. If Peter does know the solution to $g^x = A$ then he can demonstrate this fact without telling Victor the value of x
2. If Peter trys to 'blag' it, then his chance of succeeding is exactly 50 % at each attempt
3. If Victor asks for 10^6 verifications then he can be sure Peter knows the answer, and Victor still has not got any idea what x is!

Another Example : From Graph Theory

We look at some more examples from Graph Theory. We need some new terminology, a *graph* is a tuple (V, E) consisting of a a set of vertices V and a set of edges E.

$$V = \{1, 2, 3, 4, 5\} \quad \text{and} \quad E = \{(1, 3), (3, 5), (1, 2), (2, 5), (3, 4)\} \tag{0.186}$$

is a graph, fig. 11 helps to clarify this, and makes it clear whilst the structure is called a graph.

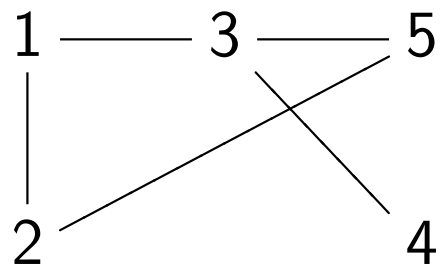


Figure: Example of a graph.

Slight Digression

1. `https://www.youtube.com/watch?v=JYyuPITzMgw`
   $R(3,3) = 6$
2. $R(5,5) =?$, the exact value of $R(5,5)$ is unknown, although it is known to lie between 43 and 48, computers are not fast enough to enumerate all possibilites: there are

$$\approx 2^{1000} \tag{0.187}$$

   scenarios. (There are 47 vertices so there are $47 * (46)/2 \approx 1000$ edges, each edge is either red or blue so there are 2^{1000} scenarios)

Slightly Simple Problem

To make the problem easier, let us color the vertices not the edges, and suppose we require that adjacent vertices (that is vertices joined by an edge) are given different colours. The minimal number of colours to colour a graph G is termed its *chromatic number* γ. Let a graph have n vertices, if each vertex is joined to every other vertex then clearly $\gamma = n$. Draw a picture

$\gamma(G) = 4$

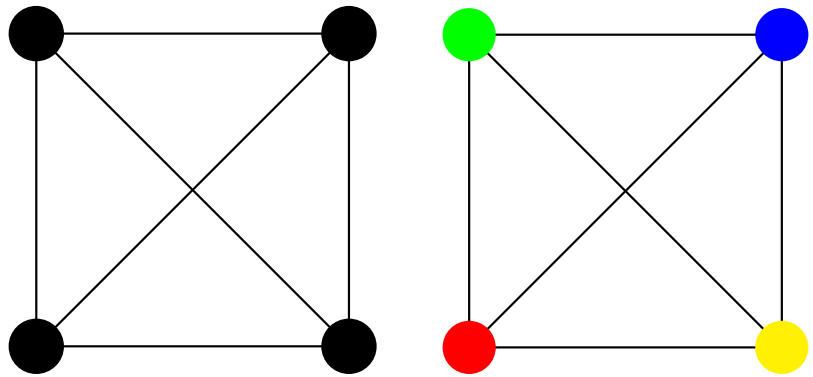


Figure: G clearly requires 4 colours. You can not colour G with 3 colours and have all adjacent vertices with differing colours.

A 3 colouring

Find a 3 colouring of the following graph:

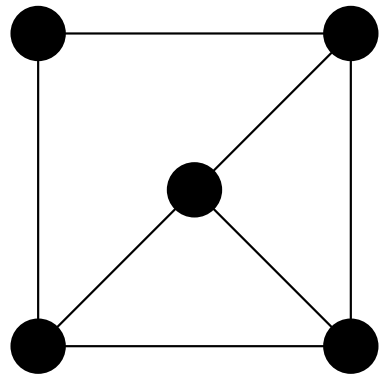


Figure: Find a 3 colour of the 5 vertex graph

Solution

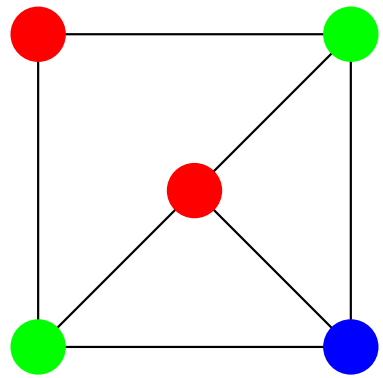


Figure: A 3 colour of the 5 vertex graph

A Zero Knowledge Proof

Suppose you have a graph with many vertices n and you claim there is a coloring with m colours, such graphs are important in many areas of combinatorics and applied mathematics. What is near amazing is the following:

1. Peter can prove that such a graph exists to Vic
2. Vic will be able to verify Peter's claim, but Vic will have no idea what the graph looks like!

A Zero Knowledge Proof: Algorithm

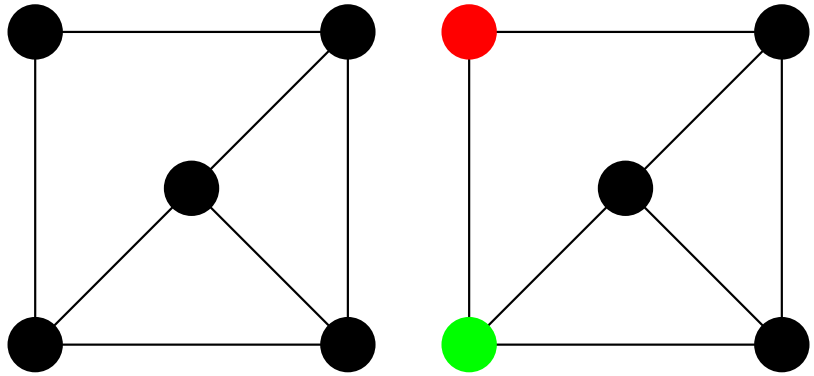


Figure: Zero knowledge proof.

1. Vic picks any edge.
2. Peter displays the colours on that edge.
3. Peter picks two distinct integers randomly between 1 and m say he picks i and j, Peter now changes the colour of all vertices that are colour i to colour j, and all vertices that are colour j to colour i.
4. Return to 1.

What if Peter is cheating?

If Vic plays this game many times and never sees more than m-colours, then he must acknowldege the proof. Why? Because if Peter uses $m+1$ (or more) colours then his chance of being caught is at least

$$\frac{1}{\text{number of edges}} \tag{0.188}$$

This may be a small number, but it is greater than zero, and after n-verifications we have

$$\begin{aligned} \mathbb{P}(\text{caught cheating } \leq n \text{ tests}) &= 1 - \mathbb{P}(\text{not caught cheating } \leq n \text{ tests}) \\ &\geq 1 - \left(\frac{\text{number of edges}-1}{\text{number of edges}}\right)^n \end{aligned} \tag{0.189}$$


W. Diffie and M.E.Hellman, *New directions in cryptography*, IEEE Trans. Information Theory, (1976).