Cryptography & Encryption:6G7Z1011

Keith Yates

January 18, 2019

Cryptography & Encryption:6G7Z1011 : Introduction

Keith Yates

January 18, 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 $= 010000100110010101100100$

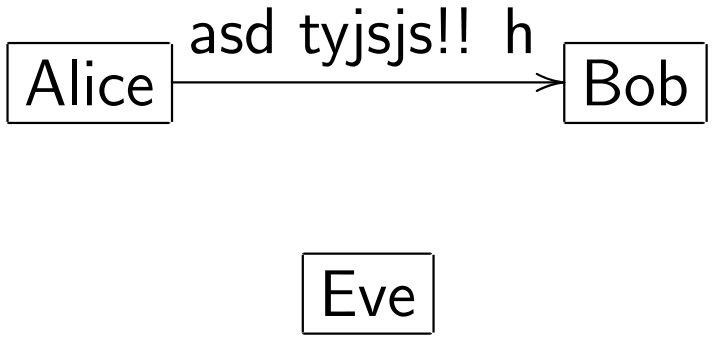
Privacy

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

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

Privacy

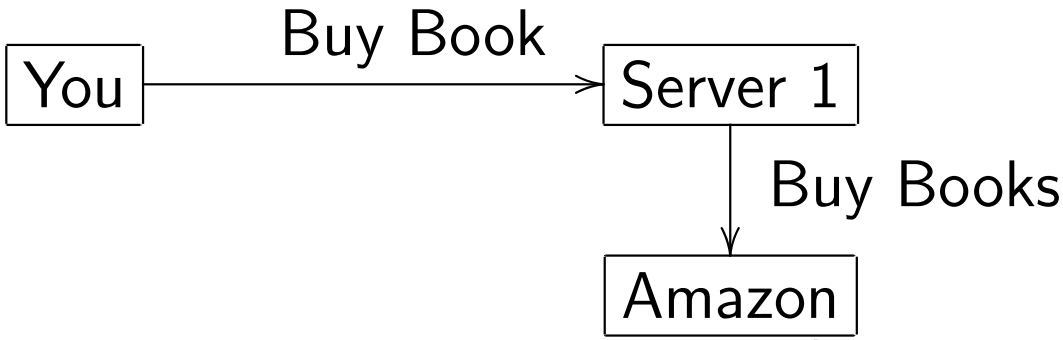
Encryption allows two users to communicate securely. Alice and Bob are in love; Alice states ‘asd tyjsjs!! h’



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");
           }

    }

}
```