Public Key Cryptography: Secrecy in Public Transcript

Date: Tuesday, 22 October 2013 - 1:00PM

Location: Museum of London
Welcome to the second of my lectures this academic year and thank you for coming along. This year I am taking as my theme some examples of using mathematics in various areas.

The lecture today is about cryptography, the science of transforming information so that it cannot be read or understood. I will look at how it has developed and, in particular, about how a subject which has secrecy at its heart is becoming more and more open and public over time and hiding less and less of its details and subjecting its algorithms, methods and systems to public scrutiny and analysis.

What is certain is that for centuries massive intellectual effort has gone into methods of ensuring secure communication, because of its importance in diplomatic, commercial, military and economic affairs. The consequences of secret messages being read by those for whom they were not intended can have serious economic and sometimes life-threatening consequences.

Let me give you an overview of the lecture.

Key terms and guidelines
As with any subject cryptography has its own language, notation and concepts and core among these concepts is the notion of a cipher system and its encryption and decryption keys. I will also talk about standard assumptions that are made about cipher systems and the different ways cipher systems can be attacked. The other distinction I will draw out is between conventional cipher systems and the newer public key approach to cryptography.

Caesar ciphers
Our look at particular cipher systems will start by considering one of the simplest, the Caesar cipher. This is not because the Caesar cipher is of any practical use but it does allow us to understand some of the key concepts and particularly concepts about keys and see potential weaknesses in cipher systems.

Substitution cipher
One of the weaknesses of the Caesar cipher is the small number of possible keys it has. By contrast the substitution cipher has a large number of keys we can use for encryption, about 4 times \(10^{27}\), a gigantic number, but it is vulnerable, this time due to the structure and statistics of natural language and in particular English which is the language I will be using for illustration. It can be broken because the frequency with which letters of the alphabet occur can be quantified.

Polyalphabetic cipher
Polyalphabetic ciphers try to hide the structure and statistics of the English language and do this essentially by using many different substitution ciphers. Which substitution cipher to use depends on where you are in the message you are trying to encrypt.

Enigma
The Enigma Machine mechanised a polyalphabetic cipher and cracking it was attempted and achieved by Polish cryptographers before World War II and by English cryptographers at Bletchley Park during the war. I will discuss how the Poles were able to use aspects of how the Enigma machine was used to crack it.

Modern ciphers
Stream ciphers
Block ciphers
Modern cipher systems are implemented electronically using computers and fall into the two broad categories of stream ciphers and block ciphers. Stream ciphers are very useful for encryption of digitised speech in, for example, a mobile phone network where there can be interference on the communications channel. Block ciphers have different strengths, for example, when it comes to the integrity of stored data and the reassurance that it has not been altered.

Diffie-Hellman key exchange
Now we will come to the seminal paper of Whitfield Diffie and Martin Hellman which was published in 1976 and set cryptography of in new directions. This paper revolutionised cryptography and in particular showed how two people, with no prior acquaintance, could communicate over an insecure communication channel to establish a key to use with their cipher system. They do not have to meet up or use a trusted courier to exchange keys!
RSA Public key cryptography

In this paper Diffie and Hellman also outlined how the cipher system called public key cryptography might work and I will outline a particular way of doing it called RSA Public key cryptography. This is named using the initials of its discoverers and its mathematical basis rests on the difficulty of finding the factors of large numbers.

To finish I will show how public key cryptography can be used to provide digital signatures which are used like ordinary written signatures to guarantee authenticity of the message.

Here is a diagram representing a cipher system. Two people usually called Alice and Bob wish to communicate securely. Suppose Alice, on the left of the diagram, wants to send a message, called the plaintext, to Bob. She makes use of an encryption algorithm and her encryption key to encrypt the message and obtains the cryptogram or ciphertext which is then sent to Bob. Bob uses a decryption algorithm and his secret decryption key to decrypt the ciphertext recovering the plaintext.

Another player, Eve, is often involved. Eve eavesdrops on the communication channel!

Cryptography is the science of designing cipher systems while cryptanalysis is the science of finding information about the plaintext from the ciphertext without being given the decryption key. Cryptology is the term for both cryptography and cryptanalysis.

The keys are vitally important for the security of a cipher system. For example, if you are encrypting data on your laptop computer and you write the decryption key on a Post-it and stick it on the laptop then if the computer is stolen the thief can also decrypt the data. Similarly writing your pin number on your bank card is not recommended! So, good key management is vital for security.

We can infer from this diagram that knowledge of the encryption key is not necessary to get information from the ciphertext and it was on this simple observation that Whitfield Diffie and Martin Hellman based their 1976 paper.

As a result cipher systems are now divided into two types:
A symmetric or conventional cipher system is one where it is easy to deduce the decryption key from the encryption key. For many symmetric cipher systems these two keys are, in fact, the same and the systems are known as secret key or one-key systems.

An asymmetric or public key cipher system is one in which it is practically impossible to deduce the decryption key from the encryption key. The difficulty comes in designing such systems.

I will discuss some symmetric ciphers and then asymmetric ciphers, but first a few comments about the security of cipher systems.

Key distribution
Our diagram of a cipher system says that Alice and Bob must have a matching pair of keys and in the symmetric case it might be difficult to reach that situation. Secure distribution of keys while keeping their value secret is a major issue as is the complete cycle of key management which includes, key generation, key distribution, key usage, key change and key destruction.

In the asymmetric case there is a different problem. Here the encryption key is public but how are you to be certain who the key really belongs to? This question of authenticity is very important and one approach is to set up Certification Authorities to counter against the possibility of impersonation.

Cover time
Assessing the security of a cipher system is not an exact science and it is helpful not to ask for absolute security but that the system is secure enough for its intended purpose. A helpful measure in answering the question of whether the system is secure enough for its intended purpose is to estimate the length of time for which the information needs protection. This desired time is called the cover time of the system and gives a rough indication of how secure the system is.

In order to make this estimate of how long it would take to break the system one could calculate how long it would take to try every possible candidate for the decryption, a so-called brute-force attack. To do this we would need to know the number of keys.

In assessing the security of a cipher system the following worst case conditions or assumptions are frequently made.

First: The cryptanalyst has a complete knowledge of the cipher system.
This means we should not rely on keeping the details of the system secret. Of course, this does not mean that we need to make them public and we could try to hide them in hardware or software but it is best to assume the worst case that they will be eventually uncovered and security must reside in the decryption key. This worst case scenario is important for cipher system manufacturers or designers as it removes from them a good deal of the responsibility for keeping the system secret.

Next: The cryptanalyst has obtained a considerable amount of the ciphertext.

This must be a reasonable assumption because if Eve cannot get hold of the ciphertext then there is no need for encryption! And the worst case assumption is that if she can get hold of some it then she can get hold of all of it.

Worst case scenario three: The cryptanalyst knows the plaintext equivalent of a certain amount of ciphertext.

This might be achieved by intelligent guessing perhaps due to the standard form of a communication say a letter always starting with the date and ending with yours faithfully. An example I’ve seen is where in the Second World War the RAF attacked German light buoys in order have knowledge of what was in the plaintext German report of the attack encoded using the Enigma machine.

The term known plaintext attack is a cryptanalytic attack in which the cryptanalyst possesses a substantial quantity of corresponding plaintext and ciphertext.

While a chosen plaintext attack is a cryptanalytic attack in which the cryptanalyst can have an unlimited number of plaintexts of attacker’s own choosing and examine the resulting cryptogram.

Having introduced some of the definitions and underlying concepts let me illustrate them by looking at some cipher systems.

This is a very simple cipher system in which each letter of the alphabet is “shifted” by the same amount, say 13 letters.

For example:
Write the 26 letters of the alphabet in a circle – the outer ring in the picture. Each letter in the alphabet is shifted 13 clockwise – the inner ring in the picture. So GRESHAM COLLEGE becomes TERFUNZ PBYYRTR
As G goes to T, R to A and so on.
Note that E is always encoded as R and L as Y.
Let us look at the cipher system diagram for the Caesar Cipher where the encryption key is rotating by clockwise by 3.

Here is the encryption algorithm is to rotate clockwise by the encryption key 3.
The word “MESSAGE goes to the ciphertext “PHVVDJH”
The decryption key is the same algorithm but with the decryption key 23 so we decrypt by rotating clockwise by 23. This decrypts because to encrypt we rotated by 3 and if we now additionally rotate by 23 we are rotating by 3 + 23 = 26 and shifting by 26 brings everything back to where it was.

Here we think of this cipher as having the same algorithm for enciphering and deciphering but with different enciphering and deciphering keys.

We could also describe it using the same key for enciphering and deciphering namely 3 but changing the deciphering algorithm by rotating anticlockwise by 3.

Suppose we have intercepted the Cryptogram: AFCCP which we know has been enciphered using a Caesar cipher. There are then only 26 possible keys to try. In the table:
The second column tells you what the original message would have been to give AFCCP under the enciphering key in the first column: similarly for columns 4 and 3 and finally for columns 6 and 5.

Assuming now that the word was an English word there are two possibilities for the message: either CHEER or JOLLY with enciphering keys of 24 and 17. We have ruled out all but 2 of the 26 possible keys. We cannot decide between the two possibilities unless we intercept some more ciphertext or have some idea of the context of the message. It will not take much more ciphertext to allow us to decide on the key.

There is another weakness of the Caesar cipher. It is vulnerable to a known plaintext attack. If we know the ciphertext for just one letter e.g. we know that A goes to E then we know that the shift is 4.

Write A as 0, B as 1, C as 2, ..., up to Z as 25.

Suppose the encryption key is y.

Encryption is achieved by replacing the letter with number x by the letter which is the remainder of dividing x + y by 26.

This is written \((x + y) \mod 26\)

Example: Suppose the encryption key is 18.

Then to encrypt J = 9 we obtain

\((9 + 18) = 1 \mod 26\). So J is encrypted as B.

There are other common uses of the modular idea, for example the 24 hour clock. This modular arithmetic is going to be vital when we come to public key cryptography.

Let us turn to more general substitution ciphers than the Caesar cipher. I want to use them to show that although a large number of keys are necessary for security it is by no means sufficient.

Write the alphabet in a randomly chosen order underneath the alphabet in alphabetical order.

\[
\begin{align*}
A & B C D E F G H I J K L M N O P Q R S T U V W X Y Z \\
P & Q O I U M E A Y L N F D X J K R C V S T Z W B \\
GRESHAM & is encrypted as MKIREPO
\end{align*}
\]

The encryption and decryption keys are equal and are just the order in which the blue letters above are written.

The encryption algorithm is: replace each letter by the one below it.

The decryption algorithm is: replace each letter by the one above it.

There are an enormous number of keys - they are all possible arrangements of writing the letters of the alphabet in a string.

Number of keys = \(26 \times 25 \times 24 \times 23 \times \cdots \times 3 \times 2 \times 1\)

\[
\text{(written as 26! and called 26 factorial)}
\]

\[
= 403,291,461,126,605,635,584,000,000
\]

This is approximately \(4 \times 10^{27}\) and is a very large number - if you were able to try a billion keys each second it would take nearly 13 billion years to try them all!

However the key is long and difficult to memorise, remember we are not allowed to write it down or if we do so we have to keep it securely! One approach to generating a key which is easier to remember is to use a key phrase.

Suppose we choose the key phrase:

Gresham College free public lectures.

Remove repetitions from the key phrase and complete by adding in alphabetical order the missing letters.

greshamcolfpubitdknjqvwxyz
The number of keys deducible from key phrases is many fewer than the 26! possible simple substitution keys but still enough to preclude a brute force attack.

However this cipher is very easy to break and the approach uses the statistics of the English language. A similar approach will apply to other languages.

If you take a passage of English text and count the number of times each letter appears it will be dominated by quite a small number of letters.

Here is a table of the letters occurring in the words listed in the main entries of the Concise Oxford Dictionary. The third and sixth column represents proportions, taking the least common letter Q as equal to 1. The letter E is over 56 times more common than Q in forming individual English words, A is 43 times, R is 38 as is I while O is 36 and T is 35. It might be easier to absorb if we display the information graphically.

The graph on the left shows the letters sorted by frequency starting with E and the right hand one with the letters in alphabetical order. This histogram is like a signature or fingerprint of the language. The next thing we note is that a substitution cipher does just that – it substitutes. If it substitutes E by W then it always substitutes E by W so if we have some ciphertext and count the number of times each letter appears in the ciphertext and find that W occurs most frequently we can guess that E is substituted by W.

Here is some ciphertext resulting from a substitution cipher. If we perform a frequency analysis of the letters in it we get the table you see, which is also represented graphically at the bottom right. The most frequent letter is H which we will guess is the substitution for E. The next most frequent in the ciphertext are I, B, G, and N which we try associating to the next four most frequent letters in English A, R, I, O. It doesn’t take long to finish it because we are assuming that the message is in English and has the structure, form and spelling of English text.

The key is in fact the one we had earlier derived from Gresham College and the message is just the publicity for this lecture. Let me make some other comments on the simple substitution cipher.

Remove English language spacing to hid the words!

How long is long enough?

200 characters is enough normally for the statistics to be reliable.

Structure of language and frequency analysis since the same plaintext letter always goes to the same ciphertext letter.

Simple substitution on bigrams that is, consecutive pairs of letters subject to frequency analysis as long messages are likely to be dominated by relatively few bigrams. The Playfair cipher is an example of this bigram approach.

What we need to try do is to eliminate the fingerprint given by the histogram of the frequencies of the letters. Ideally we would like all letters to appear equally often. One approach to achieving this attempt to flatten out the frequency histogram is to use polyalphabetic ciphers.

In a polyalphabetic cipher the ciphertext character used to represent a particular plaintext letter can vary throughout the cryptogram. The encrypting letter used might depend on for example, the position of the plaintext letter or on the content of the plaintext that come before it. Also the same ciphertext character can represent different plaintext letters.

It is easier if I show an example – the Vigenère cipher.

Plaintext
AGEDTWENTYSIXVIGENERE

Key is CHARLESV and if not long enough for the message it is repeated.

CHARESVCAREL

CipherText
CNEUEAWIVFSZIZABGUEIP

The way the square is used is that the key letter underneath gives the substitution cipher to use on the letter above it.

So C in the key means use row 3 where the A is encoded as C.

Then the next letter, G, in the plaintext is encoded using the row given by H and encodes as N.

The next row to use is given by A and E is encoded as E.
Because the key is of length 8 we are using 8 different substitution ciphers – in fact they are Caesar ciphers and we can see the effect they have on the letter frequencies of the ciphertext.

Consider the following text:
Aged twenty six, Vigenère was sent to Rome on a diplomatic mission. It was here that he became acquainted with the writings of Alberti, Trithemius and Porta, and his interest in cryptography was ignited. For many years, cryptography was nothing more than a tool that helped him in his diplomatic work, but at the age of thirty nine, Vigenère decided that he had amassed enough money to be able to abandon his career and concentrate on a life of study. It was only then that he began research into a new cipher.

The top histogram gives the usual frequency distribution for the plain text. The one lower down on the left is what we get if we apply a monoalphabetic cipher. It gives letter frequencies that are just a rearrangement of the letter frequencies of the plaintext. The lower right gives the letter frequencies for the Vigenère cipher and it is much flatter giving much less information to the analyst. It is flatter because it is using 8 substitution ciphers.

However the Vigenère cipher is very vulnerable if you know the length of the key. Suppose as here that it is 8. Then in the ciphertext, letters 1, 9, 17 ... come from the same Caesar cipher and can be attacked easily. Similarly for the letters 2, 10, 18, 26 ... and for 3, 11, 19....

But how to find the length of the key? You could just guess, but better is the following method!

It was used by Charles Babbage and searches for repeated long strings of characters in the ciphertext. When they occur they are likely to represent identical passages in the plaintext so spaced that they are encoded using the same letters of the keyword. Then it is likely that the gap between these repeated blocks is a multiple of the key length. Then you try those possibilities.

Let me turn now to one of the most important polyalphabetic cipher systems.

The Enigma was a German electromechanical rotor-coding machine. In the picture, when a key is pressed, a current flows through the three rotors, is reflected, and then flows back through the rotors to light up a lamp to give the code for the pressed key. Then, crucially, as shown on the right one of the rotors rotates, creating a new pathway for the signals. When the first rotor has made a complete turn, as a result of key presses, the middle one starts to move, and when it makes a complete turn the last one starts to move.

An additional complication is that there is a plug board that can be altered to connect letters in pairs so that they were swapped on entering and leaving the rotors.

The Enigma was a polyalphabetic cipher system with period 26 × 26 × 26 = 17,576.
In each state of the Enigma the substitution alphabet would be a swapping of pairs of letters and in particular no letter could be enciphered into itself
There were 17,576  Rotor settings and there were 6 ways for the Rotor orders.
The Plugboard connecting seven pairs of letters and could be achieved in 1,305,093,289,500 ways
This gives the total number of keys for the Enigma as
17,576 × 6 × 1,305,093,289,500

Marian Rejewski, 1905 - 1980, (picture probably 1932, the year he first solved the Enigma machine). Due to espionage, mainly by the French, the internal wiring of the rotors was known. Let me indicate how Rejewski broke the Enigma system. It was highly dependent on how the machine was used to communicate keys i.e rotor setting between users.
Code books were distributed to give the day-key, which was the plug board settings and the rotor settings and orders.
The Day-key was used to transmit a new key chosen by the sender.
For example suppose a particular day-key is RGF. The sender uses it to transmit chosen new key KJE and does so twice. Doing it twice was the weakness but was done in case of transmission errors.

We would then have KJEKJE transmitted using RGF and gives, say, ACKJDG. The receiver decodes this to get KJEKJE and both sender and receiver now change their rotor settings to KJE which they use for further transmissions.

So much was hidden from Rejewski. But he did know that the first six letters for all messages on a given day were sent using the same setting and further that in each message the 1st and 4th were encodings of the same letter and the 2nd and 5th were also as were the 3rd and 6th.
That meant there was a relationship between the first and third letter of each message. The table shows the first six letters of 4 messages.
This relationship can be written as in this table. Examining the first six letters of enough messages would give him the lower table where we know the association or relationship of all the letters of the alphabet.
We look at the structure of this relationship as follows: Start at A and keep going until you get back to A. Then choose a letter not found so far, start at it and keep going until you get back to it. Repeat until you have found all the letters. The lengths of these cycles are characteristic of the rotor settings and independent of the plug board. The plug board just changes the letters in the cycles not their lengths. We get another set of cycles for the 2nd and 5th letters and a final set for the 3rd and 6th. Rejewski and his colleagues spent a year obtaining these cycle lengths for all rotor positions. Then they could recognise the day setting! It was then fairly straightforward to determine the plugboard setting. Their success was highly dependent on how the system was used and failed when this changed.

Let me now turn to more modern approaches.

There are various ways of writing a message as a string of bits e.g. ASCII – American Standard Code for Information Interchange

Exclusive OR, often written XOR or ⊕ is a way of combining two bits as follows:

0 ⊕ 0 = 0, 0 ⊕ 1 = 1, 1 ⊕ 0 = 1, and 1 ⊕ 1 = 0

It is identical to addition modulo 2

We can combine two bit-streams of the same length by XORing the pair of bits in identical positions. Another way of thinking of it is that if the upper key is the message and the lower the keystream then:

if there is a 1 in the keystream it changes the corresponding number in the message and if there is a 0 in the keystream it leaves unaltered the corresponding number in the message.

A Stream Cipher takes a short key to generate a long keystream. There are various ways to do this. A good keystream generator should produce a keystream that, as far as is possible, is indistinguishable from a random sequence.

To encrypt: the plaintext is combined with the keystream, for example by using XOR, to get the ciphertext.

To decrypt: the ciphertext is combined with the keystream using XOR if that was used to encrypt.

A stream cipher can be easily implemented and is fast in operation.

A stream cipher is good for a noisy channel where errors during transmission can occur. This is because if a ciphertext bit is received incorrectly then there is only one incorrectly deciphered bit. This lack of error propagation means stream ciphers are used for encrypting digitised speech.

A stream cipher is vulnerable to a known plaintext attack as that will allow deduction of parts of the keystream from the known plaintext and ciphertext.

In a block cipher the bit-string is divided into blocks of a given length and the encryption algorithm acts on each block to produce a cryptogram block.

If the blocks are encrypted individually and independently we call this ECB (Electronic Code Book) mode. This is vulnerable to building up a dictionary of known corresponding plaintext and ciphertext.

To avoid this we can arrange for the encryption of each individual block to depend on all the message blocks that go before it using, for example, Cipher Block Chaining (CBC).

The major advantage of Cipher Block Chaining (CBC) mode over ECB mode lies in its ability to hide statistical properties of the plaintext blocks.

Whitfield Diffie and Martin E. Hellman published a very influential paper in cryptography in 1976. Its title New Directions in Cryptography captured its importance and in the abstract they said:

Abstract: Two kinds of contemporary developments in cryptography are examined. Widening applications of teleprocessing have given rise to a need for new types of cryptographic systems, which minimize the need for secure key distribution channels and supply the equivalent of a written signature. This paper suggests ways to solve these currently open problems.

Their idea for key distribution uses logarithms. Shortly after the invention of logarithms, Henry Briggs, first Gresham Professor of Geometry in London, heard about them and he produced an extensive collection of 30,000 logarithms to base 10 all calculated by hand to fourteen decimal places. Impressive though this is it is not very difficult to calculate logarithms to base 10. What is much more difficult is calculating discrete logarithms.

Let me illustrate:

Pick a prime, say 17. I choose a prime so that I have a mathematical object called a group which is the subject of my next lecture.

If \( y = 10x \mod 17 \) then \( x \) is the discrete logarithm of \( y \)

\( 5 = 107 \mod 17 \) and \( 14 = 103 \mod 17 \)

Here 7 is the discrete logarithm of 5.

Here 3 is the discrete logarithm of 14.

Knowing \( x \) it is easy to calculate \( y \).

But hard to find \( x \) if we know \( y \), for example,
8 = 10X mod 17

Now let me show you how Diffie and Hellman used the idea of discrete logarithms to enable Alice and Bob to communicate and agree on a secret key for further communication.

Here I am just repeating what we have already seen. Find a one-way function – popular choice is of discrete logarithms, say, Y = 10x mod 17

It is called a one-way function because knowing x it is easy to calculate y, for example,

5 = 107 mod 17 and 14 = 103 mod 17

But knowing y it is hard to find x, for example,

8 = 10X mod 17 (answer is 14)

Alice takes her private key is 7 and public key is 5
Bob’s private key is 3 and public key is 14

Message key for Alice is 147 mod 17. It is the public key of Bob to the power of the private key of Alice.
Message key for Bob is 53 mod 17. It is the public key of Alice to the power of the private key of Bob.

They are the same! Each is

103 x 7 mod 17 = 107 x 3 mod 17

Both equal to 6 which is their common secret key.

Let me now turn to the other important idea in the Diffie-Hellman paper, that of public key cryptography.

Using the London telephone directory easy to find the number given a person’s name and address but difficult to find their name and address given the telephone number.
The diagram below show how key generation works in a public key cryptographic system.

**Setup:** Bob chooses two secret prime numbers. We will call them p and q. To be secure, the numbers must be at least 100 decimal digits long.

Bob calculates n = p * q.

Bob finds a number e where the greatest common divisor of e and (p - 1) * (q - 1) is 1.
Bob finds a number d where d * e = 1 mod ((p - 1) * (q - 1)).
Bob publishes n and e as the public key. He keeps d secret and destroys p and q.

**Encryption:** Ciphertext = Me mod n

**Decryption:** Message = Cd mod n

Source: http://gdp.globus.org/gt4-tutorial/multiplehtml/index.html

Although I have used the Diffie-Hellman 1976 paper in my discussion it turns out that the principles and main techniques of public key cryptography had been developed in the early 1970s by James Ellis, Clifford Cocks and Malcolm Wilkinson at the UK Government’s Communications Electronics Security group. This work was classified and not made public for over two decades. It is interesting to speculate what else may now be secret!

A lot of work, research and money are being used on cryptography because secure communications are becoming more and more important in this electronic age of e-commerce. It does not matter just to governments, banks, retailers and other large organisations, it also matters to ordinary individuals. Last Saturday in the Guardian there was the following headline:

It appears that extortionists are using ‘ransomware’ to encrypt your files so that you can only get them decrypted by sending them money. CryptoLocker uses asymmetric cryptology to encrypt your files, photos and other data and it is impossible for you to decrypt them. So at least now after this lecture you know how encryption like this can be done and why it is so hard to undo. Perhaps when you get home you should back up the data on your computer!

Thank you.

Look forward to seeing you on Tuesday 19th November 2013 for Symmetries and Groups

Slide: References
David Kahn, The Codebreakers (Scribner, 1995)
Fred Piper and Sean Murphy, Cryptography, A Very Short Introduction (OUP, 2002).

© Professor Raymond Flood 2013