**Cryptography and Encryption**

You’ve learned about how messages can be sent across networks. The Internet is comprised of billions upon billions of such communications – every time you open a webpage in your browser, you make a request for a copy of a webpage from a server using a specific protocol (HTTP or HTTPS). When you send an email, you are sending a message wrapped in another protocol (POP3 or IMAP). Whenever using Snapchat, Instagram or Facebook Messenger, you communicate via HTTPS with a central server which relays your messages to the recipient.

Sometimes it’s important that these communications are kept secure. What do we mean exactly by secure? We can define it as the ability for the message to be read only by the intended recipient. This is especially important when sending certain messages – like using online banking, or sending a confidential email.

In practice, it turns out this is actually quite a difficult thing to do.

**Traditional Ciphers**

The problem of sending secure messages dates back hundreds of years. People came up with many ingenious schemes for secure communication using ciphers or codes. The classical ciphers are quite easy to understand.

*The Roman Cipher*

Using the following key, translate the message below. Look up each letter from the message in the bottom row (the *encrypted* *text*). The top row will provide the actual letter (the *plain text*).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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** |
| Z | C | D | Y | X | M | W | B | L | K | V | E | N | O | J | A | I | T | F | U | H | P | Q | R | G | S |

BXEEJ QJTEY

There are other versions of this idea, such as the *transposition cipher* (move each letter N places to the left or right in the alphabet). Historically, spies used one-time pads to communicate messages. These were more complex versions of the Roman cipher. Without the corresponding *key*, an encrypted message was difficult to decipher. In World War II, the British used the Bombe machine to crack encoded messages from the Germans, which were encoded using complex ciphers.

**Today’s Ciphers**

Today, we use two main forms of encoding for messages, symmetric-key encryption and asymmetric, or public-key, encryption.

For each form of encoding, there are many different standards available. Some of the most common are AES and DES (for symmetric-key encryption); PGP, GNUPG and RSA (for asymmetric encryption). These standards are simply descriptions of the steps to take to encrypt data and decrypt data. In the Roman cipher, the step (or algorithm) is to look up the corresponding letter in the supplied table. For modern ciphers, it’s a bit more complicated.

The DES algorithm, for example, takes a key of length 56 bits. It performs something called the Feistel function (or F function) which is simply a series of steps on the message, using the key, to produce the *ciphertext*. Similarly, a decryption key can reverse those steps using the same key to obtain the plaintext.

**Exercise –**

Using the following instructions, encode this message: TOP SECRET: USA INVADES CUBA. Use the key “72”.

1. For each letter in the message:
   1. Look up the letter’s decimal ASCII code (you can find an ASCII table here ->   
      <https://upload.wikimedia.org/wikipedia/commons/thumb/1/1b/ASCII-Table-wide.svg/875px-ASCII-Table-wide.svg.png>) and write it down.
   2. If the number you get is **lower** than the key number, subtract 1.
   3. If the number you get is **equal to or higher** than the key number, add 1.
   4. Use the ASCII table to look up the letter for your new number and write this down.  
      This is your ciphertext.
2. Using the reverse of the instructions in 1), can you decipher the following message? The key is “68”.

Remember to reverse your arithmetic – so if the number you get is lower than the key, **add** 1, and so on.

QF@BF JT EFBM@SFE JO JSFM@OE

ASCII: 81 70 64 66 70 / 74 84 / 69 70 66 77 64 83 70 69 / 74 79 / 74 83 70 77 64 79 69

+/-: 80 69 65 67 69 / 73 83 / 68 69 67 76 65 82 69 68 / 73 78 / 73 82 69 76 65 78 68

Plaintext: PEACE IS DECLARED IN IRELAND

But how do we communicate secret symmetric keys without the keys themselves being intercepted?

We can use something called the Diffie-Hellman key exchange algorithm. Watch the following video to find out more.

<https://www.youtube.com/watch?v=cM4mNVUBtHk>

There is an alternative to symmetric keys – we can use asymmetric keys, also called public key encryption.

**Public-Key Encryption**

Public key encryption, or asymmetric encryption, uses a pair of keys instead of a single key. The central idea is this – the public key is available for anyone to use and it is used only to encrypt a message. It is **not** kept secret. However the public key is used to decrypt the message, and this **is** kept secret.

Why is this secure?

Well, this means that two entities can swap public keys and use these keys to communicate securely. Only the receiving parties can decrypt the messages. There is no need to swap a private key. Note that the public key **cannot** be used to decrypt a message.

So how can you use a private key to decrypt a message that’s been encrypted by a public key? Surely the two keys are completely different?

Not necessarily. The keys are **related**, or more formally, **derived** from one another.

The strength of this derivation – or in plain English, how easy it is to work out the private key given the public key – is critical to the strength of the keys.

We can use very large prime numbers to guarantee the strength of a key pair. The essential idea is that it is very easy to multiply two large prime numbers to obtain another number. It is a lot harder to take a large number and work out which two prime factors were multiplied to obtain it!

If we imagine our product (our large number) to be our public key, then our prime factors are our private key.   
We can tell anyone the large number and let them use it to encode a message. But, we need our two prime factors to decode the message, so we keep these secret – these are the basis of our private key.

Watch the following short video:

<https://www.youtube.com/watch?v=56fa8Jz-FQQ>

Most Internet communications are carried out using public-key encryption.

The following video demonstrates some of the capabilities that hackers have, and why securing our communications is so important:

<https://www.youtube.com/watch?v=hqKafI7Amd8&t=110s>