BRIEF INTRODUCTION TO CRYPTOGRAPHY

By PAGVAC

February 8, 2004

What will I learn from this file?

- What cryptography is
- How encryption and decryption works
- Cryptography terms
- Symmetric cryptography
- Asymmetric cryptography
- PGP

Brief Introduction to Cryptography

Cryptography is the art and science of keeping messages secure. When we say "messages" we could be referring to plain text files as well as any other types of files such as executable files. Basically, the point of cryptography is to allow any user to keep his data secure and not readable from not desired individuals.

Before we examine how it works we need to be familiar with certain terms. In order to fully understand some cryptography-related terms, we are going to use the following example. Imagine that you were to send an attached text file by email to your boss. Suppose that the information you are sending him is quite sensitive and it is extremely important to you that only your boss gets to read the message. So, what you do is that you get the text file where that information is. This text file at this point is readable and therefore unsecured. In cryptography we call this the **plaintext** file. So now you need to make the plaintext file unreadable. This is called **encrypting** the data and the result is the **ciphertext** file (encrypted file). The cyphertext file cannot be read and it looks as a sequence of non-sense characters. Only if you decrypt the file you will be able to read it. After you email the cyphertext file to you boss then he would **decrypt** it, which means that he would convert it back to its plaintext form so he could read it. The following diagram (Figure 1) illustrates the encryption/decryption process.

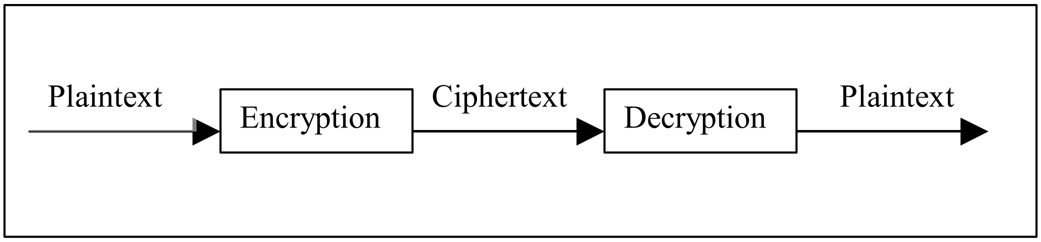

Figure 1

Now you must be wondering why would only your boss decrypt your ciphertext file and not someone else. How do I make sure that only my boss can decrypt the ciphertext file? Smart question. In order for only a specific person to be able to decrypt a file a **key** is used. This key is just like a real world door key. Only the people who own the key can open the door, and at the same time you keep you key hidden from other people so that no intruders get into your house. In cryptography, this is also called a key. This key is just a sequence of characters such as “firebird” or “this is my key”. This key should only be known by the sender and the receiver, so that if an authorized user gets access to the ciphertext file he won’t be able to read it since he wouldn’t know the key(see Figure 2).

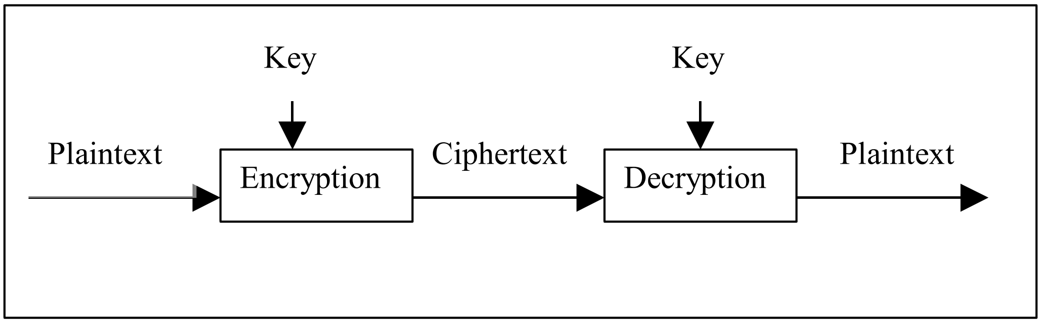

Figure 2

The key you choose is really up to you, although you should know that the longer your key is, the harder it will be for a cryptanalyst (a ciphertext breaker) to decrypt the file. Also, the encryption algorithm used makes a difference in how hard it is to break the encryption. There are all sorts of encryption algorithms out there, although this subject is really outside the scope of this paper.

Next, we will discuss the two general types of key-based algorithms: **symmetric** and **asymmetric**. In a symmetric algorithm the key used to encrypt and decrypt a certain message is the same in both cases. Since both sides have the same key we call this a symmetric algorithm. The example discussed at the beginning of this paper uses a symmetric algorithm to encrypt and decrypt the text file sent by email. With this type of algorithms all the security relies on only one key, so it must be kept in total secret. If anyone had access to the key, he could decrypt and encrypt the message with no problem. The second type is the asymmetric algorithm, also called **public-key** algorithm. As you can guess from the term "asymmetric" now each of the two parties have a different key. So now we have two different types of keys called **public** and **private** key. The first one is used to encrypt the message, whereas the private key is used to decrypt it. The public key can be given to anyone, since all you can do is encrypt a message but never decrypt it. However, the private key should only be known by whoever is supposed to decrypt the message.

Let's use an example to illustrate asymmetric-algorithm-based cryptography. We will use your boss again. Suppose that your boss now tells you that he wants to use a different type of cryptography in which only he knows the key to decrypt the messages that he receives. So you talk to him about asymmetric cryptography and he likes the concept. So now, your boss has a private key that is only known by him. He uses this key to decrypt messages that his employees send to him. On the other hand, his employees will use the public key in order to encrypt the messages (see Figure 3). Any of his employees will be able to encrypt a message in order to send it to their boss, but only their boss will be able to decrypt it and therefore be able to read the message, since only he knows the private key.

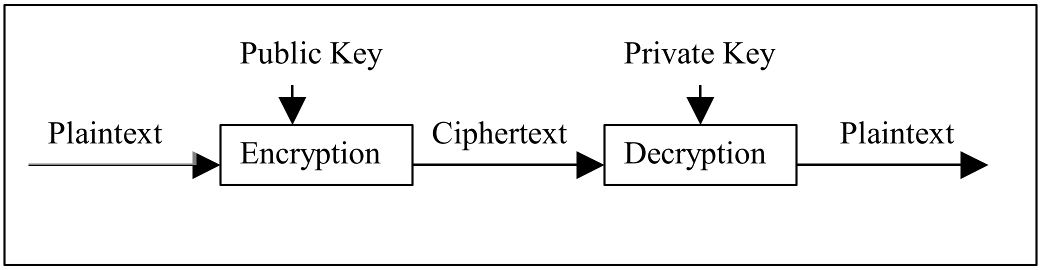

Figure 3

With this small but important background in mind you should be able to start encrypting and decrypting your own data. Which tool to choose is really up to you. There are tons of encryption/decryption tools out there, but with no doubt the most popular one is PGP (Pretty Good Privacy). You can find many different versions of PGP at http://www.pgpi.org/. If you are looking for a user-friendly version of PGP you might want to try PGP Desktop. PGP Desktop has mainly three parts: PGPkeys, PGPmail, and PGPdisk. PGPkeys allows you to create your personal key pairs (public and private keys). With PGPmail you can encrypt email messages. Finally PGPdisk allows you to encrypt portions of your hard drive so that sensitive data can only be accessed by you. There are also other secondary tasks you can perform with PGP Desktop such as securing your ICQ communications, completely deleting files (Windows by itself does not truly delete files), and create self-decrypting archives.