Encryption: The History and Implantation

by

Jacob Levinsky

Submitted to the Department of Mathematics and Computer Science
School of SUNY Purchase
in partial fulfillment of the requirements
for the degree of Bachelor of Arts

Purchase College
State University of New York

May 2022

Sponsor: [[Enter Sponsor's name here]]

Second Reader: [[Enter Second Reader's name here]]

Table of Contents

Abstract

This thesis seeks out to look back at the history of encryption and see how it has evolved. The research will first look at what encryption started out as and as time went on, see how encryption has as well. The research will show how at each step of the development of our encryption methods, how secure were they as well as the steps taken to keep the information away from those who aren't supposed to have it. Along with the research, I will create my own encryption software and write about what steps were taken to ensure the security of encryption. From all of these points of research and development, I will conclude how cryptography and encryption are used to provide security, as well as how important it is for today's age.

Keywords: Encryption; Cryptography; Software; Security

1. Introduction

Throughout history of humans, there has been one form or another of humans keeping text. As time went on, people started to hide their texts so that others couldn't know what they were, and that process has evolved. Encryption in its most basic idea is cryptography, which is just a technique in which messages can be secured. Back as far back as 1900 BC with the Egyptians, they used hieroglyphics to write their texts and some evidence shows that there are some different hieroglyphics that are not same as others suggesting the use of cryptography. Throughout history, evidence of this continues, even before what we call the current era, and we can see that civilizations have their own versions of cryptography and encryption. What I will be researching and looking at is that history of encryption and cryptography from its early stages to how it is used today.

1.1.History of encryption and cryptography

Back as early as 1900 BC, Egyptians kept their texts in their own language that was universally used called hieroglyphics. Back then, we have seen how Egyptians used hieroglyphics to keep their texts, communicate, create plans, and just about any and everything else. There is some evidence on how some of the hieroglyphics used in their texts are different then some of the typical hieroglyphics presenting the theory that it was used as a cypher to create cryptography. Another civilization that has evidence of the use of cryptography are the ancient Spartans. Back in 600 BC, the Spartans had a device that those theorize that it was used to messages. What the Spartans had was a device that was "called a scytale to send secret messages during battle" (Thales). The device had a leather strap that had letters on it, and the "letters on the leather strip are meaningless when it's unwrapped, and only if the recipient has the correctly sized rod does the message make sense" (Thales). Later in the more modern era, around 1917, there is an American named Edward Hebern who invented an "an electro-mechanical contraption" that "uses a single rotor, in which the secret key is embedded in a rotating disc" (Sidhpurwala) that was called the Hebern rotor machine. The device was used to encrypt messages by having the key encode "substitution table and each key press from the keyboard resulted in the output of cipher text" (Sidhpurwala). This was an early encryption device that later the Enigma machine built itself upon. In Germany during WW1, around 1918, a German engineer named Arthur Scherbius started work and invented the Enigma machine. The Enigma machine used 3 or 4 more rotors and how it works is that the "rotors rotate at different rates as you type on the keyboard and output appropriate letters of cipher text" (Sidhpurwala). There are more examples of the progression of the history of cryptography and encryption devices that will be explored further, as well as the examples brought up.

1.2.The difference between cryptography Vs encryption

There is not much of a difference between cryptography and encryption except that one goes into the other. Cryptography is "the science of concealing messages with a secret code" (Thales) which is typically used by a cipher. A cipher is a "method of transforming a message to conceal its meaning" (Britannica). Encryption is "the way to encrypt and decrypt the data" (Thales). To encrypt a message or data usually used the use of cryptography and the use of a cipher. Some examples of different encryption methods and ciphers are the Caesar Cipher, Hill Cipher, and Reverse Cipher. All of these have different levels of "security" depending on how complex the

cipher is. By using the Reverse Cipher, all that is done is that the alphabet is reversed so the A becomes Z, B becomes X, and so on. By taking the plain text and putting it through the cipher, all is needed to be done is reverse the process. Similar is done with the Caesar Cipher, which is the alphabet, but the letters are moved 3 spaces down so that A becomes D, B becomes E, and so on. Another similar cipher is the Hill Cipher where the alphabet is associated with its corresponding number, so A becomes 1, B becomes 2, and so on. Cryptography has a lot to do with encryption, especially now when it comes to algorithms.

1.3. Algorithms

As cryptography evolves and advances, so does its methods. Today, most cryptography and encryption methods use algorithms. An algorithm is a process that is used in calculations typically by a computer. Today's encryption methods use these algorithms, typically with software, and they are based on the old cryptography methods. One kind of algorithm is called the Symmetric-key algorithm and it can also be "referred to as a secret-key algorithm" and it "transforms data to make it extremely difficult to view without possessing a secret key" (Turner). Another kind of algorithm is called the Asymmetric-key algorithm, also known as the "public-key algorithms" and it "uses paired keys (a public and a private key) in performing their function" (Turner). Just like in the encryption methods where they have different ciphers, algorithms also have their own version of a cipher, which is called the key. Without the ciphers, the encrypted message won't be able to be seen, just like without the key whatever the message is won't be able to either. Algorithms are a huge advancement and importance to cryptography because they are used for things "such as data encryption, authentication, and digital signatures" (Spies). Just about everything in today's life revolves around algorithms either controlling or being a huge part of how things work. Without them, anyone could access bank accounts, steal personal information, log into people's emails, just about do anything.

2. Early cryptography

Looking back to how cryptography was used, the Spartans and other early civilizations like the Egyptians were early adopters of them. The Egyptians had their hieroglyphics which was used for their texts. Hieroglyphics showed things such as how they lived their lives, their rules, and just about everything else. One of the ways that Egyptians used their cryptic hieroglyphics were by monks and they used "non-standard hieroglyphics to keep anyone outside of their inner circle from understanding what was being communicated" (Fuerst).

Spartans also had their own version of cryptography that was used during war time which was called the Scytale Cipher. Just like modern ciphers, it uses a type of key to decipher the messages. How the Scytale Cipher works is that it "used a cylinder base, such as a stick or baton, wrapped with a leather or parchment strip wound spirally around it" (Fuerst), and the stick or baton was considered to be the key. The message would be on the leather strip and without the correct size stick, the message would just be random letters and wouldn't make sense. The key in this case is crucial to decipher the messages, just like we need the cipher keys in encryption to know what the messages are.

- Talk further about the Spartans and other early civilizations that use cryptography
- List examples of how they were used
- How did these methods evolve and differ from one and other
- How does modern cryptography/encryption be based on these early developments

3. Modern encryption

In more modern encryption, things are still physical like how cryptography methods were in the first iteration but with a touch of modern technology. Machines such as the Engima machine was a “rotor-based cipher machine” (Fuerst). How the Enigma machine worked was it used the rotors to turn each time a letter was pressed, which in the process would assign that letter a new random letter. As the person using the machine typed out the message, after one letter was pressed a “electrical current went through a series of rotors and a light was illuminated on a lamp panel” (Fuerst) and the person would take note of what was lighting up. The new letter that was assigned becomes part of the encrypted message, but to decipher it the person must also know how the Enigma machine was set up before the message was typed. The Enigma machine itself is both the key as well as the encryption machine.

- Talk about more current encryption devices such as the Hebern rotor machine and Enigma machine
- Give more examples of current era devices up until devices that were created recently
- How have these devices evolved and take inspiration from previous devices/methods
- What do these lead to as we further develop them in the future

Bibliography

Fuerst, N. (2021, November 11). *Who should we thank for modern cryptography? the Egyptian monks.Neal Fuerst*. Entrust Blog. Retrieved December 14, 2021, from https://www.entrust.com/blog/2020/12/who-should-we-thank-for-modern-cryptography-the-egyptian-monks/.

Sidhpurwala, Huzaifa. "A Brief History of Cryptography." Red Hat Customer Portal, 19 Mar. 2019, https://access.redhat.com/blogs/766093/posts/1976023.

Spies, T. (2017, May 19). *Public key infrastructure*. Computer and Information Security Handbook (Third Edition). Retrieved December 14, 2021, from https://www.sciencedirect.com/science/article/pii/B978012803843700048X.

Thales. "A Brief History of Encryption." Thales Group, 18 Apr. 2016, https://www.thalesgroup.com/en/markets/digital-identity-and-security/magazine/brief-history-encryption.

Turner, D. M. (n.d.). *Summary of cryptographic algorithms - according to NIST*. Cryptomathic. Retrieved December 13, 2021, from https://www.cryptomathic.com/news-events/blog/summary-of-cryptographic-algorithms-according-to-nist.

Progress with code:

- Have created a class for the encryption
    - Generate an array for a key of random numbers
        - The “strength” of the key will be determined by the user
        - The “strength” is the length of the array
    - Can display the key
    - Encrypt the message
        - Takes in a message and a key
        - My encryption method uses the Caesar Cipher
            - Shifts the alphabet a certain amount
        - The key determines how much the alphabet is shifted
    - Can display the encrypted message
- Have created a class for decryption
    - Decrypt the message
        - Takes in a message and a key
        - Also uses the Caesar Cipher
        - The key determines how much the alphabet is shifted
    - Can display the decrypted message

```
import random

class encryption:

    def __init__(self):
        self.key = []
        self.secureness = 0
        self.encryptMessage = ""

    #create a key that will be used for the ecnryption
    def generateKey(self, secureness):
        self.secureness = secureness
        i = 0
        while i < self.secureness:
            x = random.randint(1,9)

            self.key.append(x)
            i += 1
        # print("Your key has been generated!")
        return self.key
```

```
def displayKey(self):
    return self.key


#encrypt the message
def encrypt(self, message, key):
    encrypted = ""
    i = 0
    shiftKey = key

    for char in message:
            if char.isupper(): #check if it's an uppercase character
                if i >= len(key):
                    i = 0
                char_index = ord(char) - ord('A')
                # shift the current character by key positions
                char_shifted = (char_index + shiftKey[i]) % 26 + ord('A')
                char_new = chr(char_shifted)
                encrypted += char_new
                i += 1
            elif char.islower(): #check if its a lowecase character
                if i >= len(key):
                    i = 0
                # subtract the unicode of 'a' to get index in [0-25) range
                char_index = ord(char) - ord('a')
                char_shifted = (char_index + shiftKey[i]) % 26 + ord('a')
                char_new = chr(char_shifted)
                encrypted += char_new
                i += 1
            elif char.isdigit():
                if i >= len(key):
                    i = 0
                # if it's a number,shift its actual value
                char_new = (int(char) + shiftKey[i]) % 10
                encrypted += str(char_new)
```

```
                i += 1
            else:
                # if its neither alphabetical nor a number, just leave it like that
                encrypted += char

        self.encryptMessage = encrypted
        return encrypted

    def displayMessage(self):
        return self.encryptMessage


class decryption:

    def __init__(self):
        self.key = []
        self.decryptMessage = ""


    #take in key from sender
    def keyImport(self):
        #checks if key is correct

        #if key is correct, run decrypt

        #if not, throw exception
        pass

    def decrypt(self, encMessage, key):
        decrypted = ""
        shiftKey = key
        i = 0

        for char in encMessage:
            if char.isupper():
                if i >= len(key):
                    i = 0
```

```
            char_index = ord(char) - ord('A')
            # shift the current character to left by key positions to get its original position
            char_og_pos = (char_index - shiftKey[i]) % 26 + ord('A')
            char_og = chr(char_og_pos)
            decrypted += char_og
            i += 1

        elif char.islower():
            if i >= len(key):
                i = 0
            char_index = ord(char) - ord('a')
            char_og_pos = (char_index - shiftKey[i]) % 26 + ord('a')
            char_og = chr(char_og_pos)
            decrypted += char_og
            i += 1

        elif char.isdigit():
            if i >= len(key):
                i = 0
            # if it's a number,shift its actual value
            char_og = (int(char) - shiftKey[i]) % 10
            decrypted += str(char_og)
            i += 1

        else:
            # if its neither alphabetical nor a number, just leave it like that
            decrypted += char

    self.decryptMessage = decrypted
    return decrypted

def displayMessage(self):
    return self.decryptMessage



from encryption import encryption
```

```
from encryption import decryption

encrypt = encryption()
decrypt = decryption()
random = encryption()

encrypt.generateKey(15)
print(encrypt.displayKey())
encrypt.encrypt("This is my test message. If you can see this, you have the correct key", encrypt.key)
print("They encrpyted message is: ")
print(encrypt.displayMessage())
print(" ")
decrypt.decrypt(encrypt.encryptMessage, encrypt.key)
print("The decrypted message is: ")
print(decrypt.displayMessage())
print(" ")

random.generateKey(15)
print("Someone else trying to get message: ")
print(random.displayKey())
decrypt.decrypt(encrypt.encryptMessage, random.key)
print(decrypt.displayMessage())
```

[4, 3, 8, 6, 7, 4, 9, 8, 5, 6, 2, 5, 3, 5, 8]
They encrypted message is:
Xkqy pw vg ykuy pjawdok. Pj hwz ics vjm xkqy, fsd pfbg ykj ksuzkjx tmd

The decrypted message is:
This is my test message. If you can see this, you have the correct key

Someone else trying to get message:
[5, 5, 9, 7, 5, 2, 2, 5, 1, 9, 9, 1, 8, 8, 3]
Sfhr ku tb xblx hbxryfd. Kh fry ztr nbj sfhr, aqb kesx xcb hnpqdev rhc