ENHANCED ENCRYPTION-DECRYPTION ALGORITHM BASED ON NUMERICAL METHODS

April 23, 2022

Written and Submitted By

Daniel Giftson - 20110051

Jinay Dagli - 20110084

Mumuksh Tayal - 20110116

Kush Patel - 20110131

Patel Vrajesh - 20110134

Under the guidance of

Prof. Satyajit Pramanik

Prof. Tanya Srivastava

Prof. Uddipta Ghosh

Contents

Abstract

Cryptography is not a new concept. Rather, it has been in use for thousands of years now. It is the study of mathematical methods pertaining to aspects of security of information like data integrity, authentication, confidentiality, and data origin authentication. It does not only mean hiding information. Rather it is a set of techniques which can be implemented to code a data, send it to the intended receiver, and should be easily unpackable by them.

Keywords: Encryption, Decryption, Diffie-Hellman Algorithm,Cryptography, Steganography, Bisection method, Newton-Raphson method, Fixed-Point iteration.

1 Introduction

Information security has become a significant issue in today's generation. With the increase in various threats and risks in the digital domains, there is an increasing need to come up with and develop newer ways of encrypting one's crucial data and information. Therefore, an extra layer of protection shield is usually added on top of the basic encryption-decryption setup with the help of steganography. Steganography is a widely preferred and implemented technique due to the extra security that it provides. This is what we would like to focus on during this project.

1.1 Important terms to know before going ahead

1. ASCII values: ASCII was the first encoding standard (character set) used across the computers on internet. It is a 7-bit character set containing 128 characters.

2. Encryption-Decryption: Encryption is the process of converting a plain (often important) text to a meaningless message, which is called Cipher-text. On the other hand, decryption is the reverse process of encryption, that is, converting the cipher-text back to the plain message.

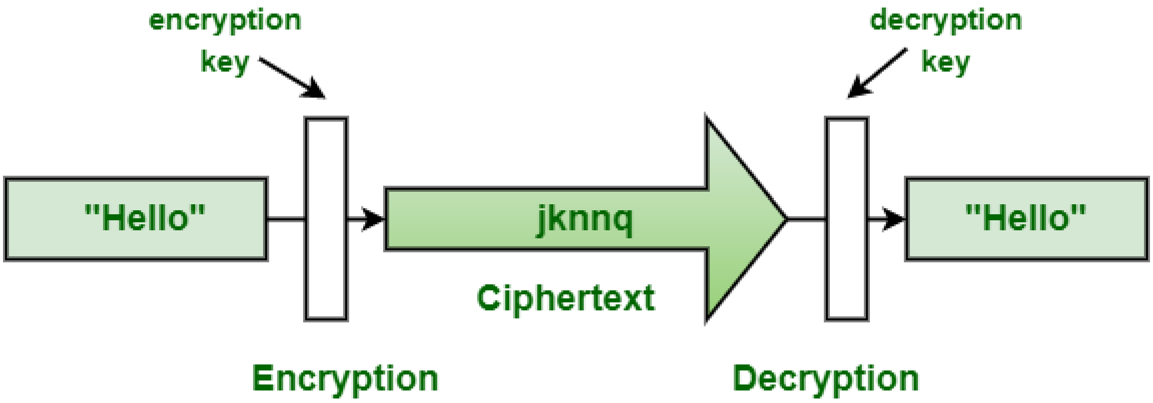


Figure 1: Representation of encryption-decryption algorithm

3. Cryptography: Cryptography, derived from the Greek word kryptos, are techniques that allow the secure transfer of data, and only the sender and the receiver know the actual contents of the message. There are two types of cryptography:

- Symmetric Key Cryptography-> The sender as well as the receiver use a single, common key.
- Asymmetric Key Cryptography-> Also called public-key steganography, it uses two related keys: a public key and a private key.

4. Steganography: Steganography is also a method of encrypting data, the only difference being that it encrypts data in a non-susceptible manner. For example,playing an audio track backwards to reveal the stored message.

1.2 Symmetric Key Exchange Algorithms

There are a number of symmetric key exchange algorithms. Most basic ones are AES, IDEA, RC4, RC5, and many more. Practically, these algorithms are not too secure. Key generated through AES can be detected by our computer in one day and the key generated through other algorithms can also be detected in a couple of days. Hence, we as a group decided to use one of the most secure algorithms - Diffie Hellman. Theoretically it is possible to detect the key but practically it is not feasible since it takes almost a year to crack it since there is loss of data during data transfer period. For encrypted communication between two people, they first need to exchange secret keys by any physical means. This key exchange method enables two people who have no prior knowledge of each other to use their own secret keys and subsequently establish a communication jointly using a symmetric key cipher.
It could be easily observed from the image how the Diffie Hellman Key Exchange works!

1.3 Assumptions taken

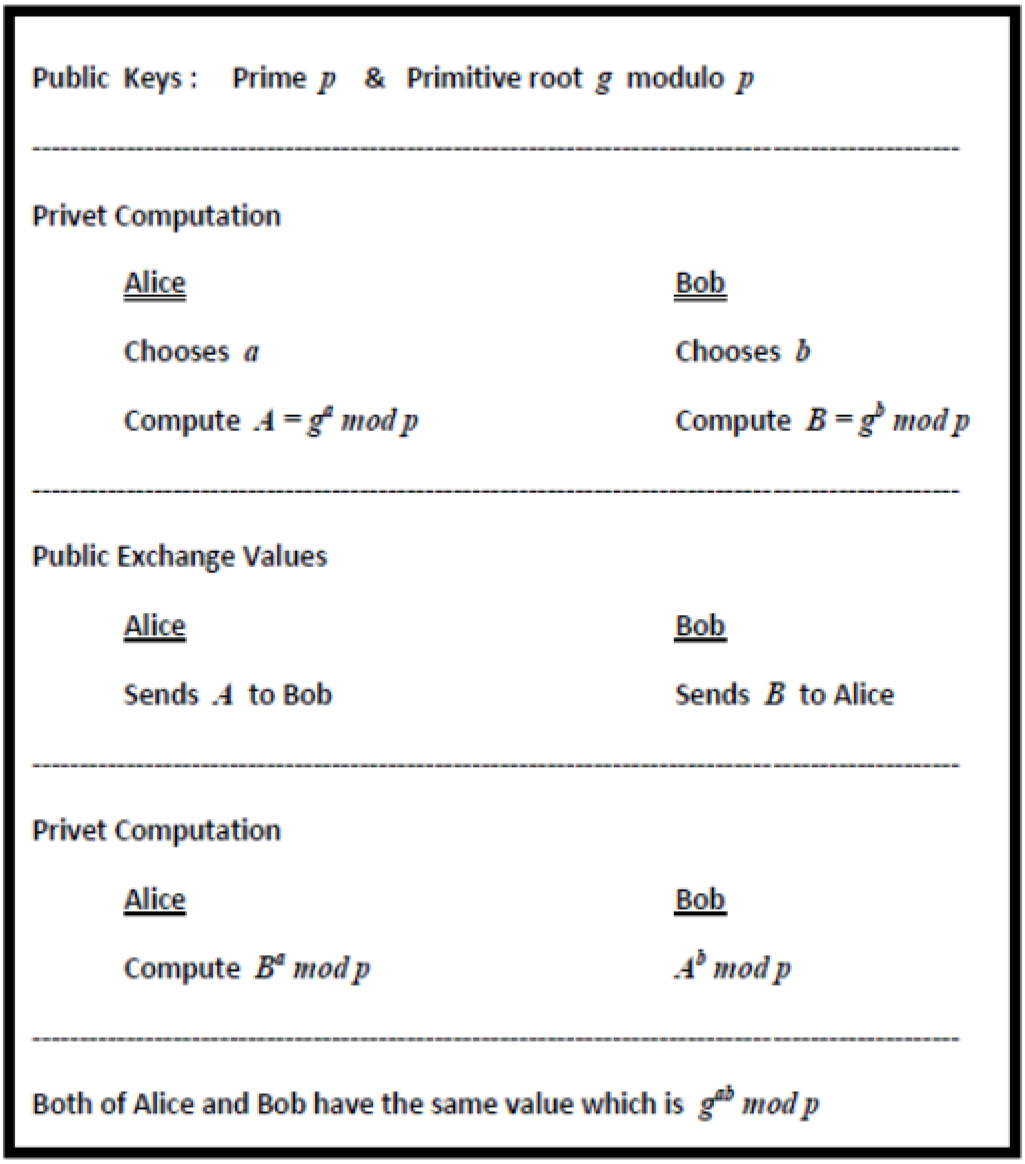
1. The function f(x) taken for generating cipher text is used only for the purpose of explaining the use of Numerical Methods. Practically much more complex one-way functions are used.
2. We are only assuming communication between two users. Practically, for multiple users, the algorithm can be run in a cyclic form.
3. The tolerance is assumed to be 0.001 in all the numerical methods.

2 Solution Methodology

2.1 Mathematical Approach

In the practical world, one way functions are the functions which can be computed easily. It is like finding f(x) at a given value of x, but when we talk about images then it is difficult to obtain an image from a random input. One way functions are used to give an unique address to the input but there is no way to go back to get the original input. So naturally, one way functions are used in cryptography to encode the messages.

We can also check or verify whether they are accurate or not by knowing the input and output form the original phase. As part of this project, we have used three different types of numerical methods to approximate the root of the one way function in cryptography. From the conclusion perspective, we have compared the iterations and results of all the methods.



Public Keys : Prime p & Primitive root g modulo p

Privet Computation

Alice

Chooses a

Compute $A = g^a \; mod \, p$

Bob

Chooses b

Compute $B = g^b \; mod \, p$

Public Exchange Values

Alice

Sends A to Bob

Bob

Sends B to Alice

Privet Computation

Alice

Compute $B^a \; mod \, p$

Bob

$A^b \; mod \, p$

Both of Alice and Bob have the same value which is $g^{ab} \; mod \, p$

Figure 2: Diffie Hellman Key Exchange

2.2 A Brief idea about our approach

There are many algorithms in order to enhance the encryption method but our project aims to implement a symmetric encryption-decryption algorithm. As we have mentioned above, encryption is done by the one-way function and Diffie-Hellman is used to encrypt it.

For the encryption purpose, we have used numerical methods to determine the roots of the one-way, which is the cipher-text for our message. Decryption follows the exact opposite approach. We would also compare and present the results about the performance of the various numerical methods.

3 Algorithm and Approach for encryption and decryption

3.1 Encryption algorithm

1. We convert the text message to a decimal number (the ASCII values of the letters).
2. The Diffie-Hellman algorithm is used to get the one-way function f(x) and the secret keys required by the user and the receiver.
3. Once we have the function f(x), we use one of the numerical methods to solve f(x) = ASCII value of the text message for x. The root of this equation gives us the cipher-text.
4. We now get the array of the solutions of the equation, which represents the encrypted data.

3.2 Decryption algorithm

The decryption algorithm is just the opposite of the encryption algorithm. It is as follows:

1. We have the array of solution for the equation representing the encrypted data. We can put these values into the equation to get the value of f(x).
2. This value of f(x) is equal to the ASCII value of the text message.
3. From this we can get back the original message.

3.3 Diffie-Hellman Algorithm

1. The sender and receiver agree on using the same prime number p and base g.
2. The sender chooses a secret key(integer) a and sends the receiver a value k = g^{a}mod(p).
3. Similarly, the receiver chooses a secret key(integer) b and sends the sender a value m = g^{b}mod(p).
4. The sender finds s = g^{m}mod(p), while the receiver finds s = g^{k}mod(p).
5. The sender and the receiver now have the same secret key s.

Example of Diffie-Hellman Algorithm: Consider that the public keys p and g are 13 and 6 respectively. The sender Alice is allocated a private key a = 4 and the receiver Bob is allocated a private key b = 2. The private keys are not known to anyone else.

Note that p is greater then g. The intermediate key generated by sender Alice is given by g^{a}mod(p) which is equal to 9. The intermediate key generated by Bob is given by g^{b}mod(p) which is equal to 10. Now the secret key is given by g^{ab}mod(p) which is equal to 3. The secret key will be the same either ways because the Diffie-Hellman Algorithm is a symmetric key exchange algorithm.

4 Numerical Methods

The solution of equations of the form f(x)=0 is obtained in many applications. If a polynomial f(x) is of degree two or three, exact formulae are obtainable. But, on the other hand, if f(x) is a polynomial of a higher order or is a transcendental function, the solution does not exist. In these cases, numerical methods are very important to find the approximate root. We would be describing the following numerical methods and use them to find the one-way function.

4.1 Bisection Method

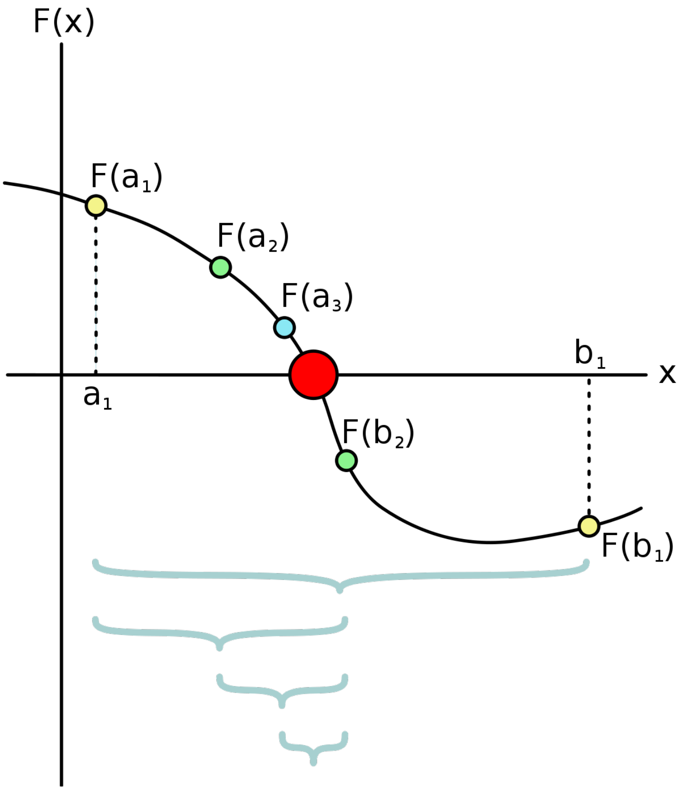


Figure 3: Bisection Method

The bisection method or interval halving is a type of incremental search method (locating an interval where function changes sign), where the search interval is always divided into half. If the function changes sign in this interval, we know that the root lies in this interval.

Stage 1: If for an interval f(x_1)f(x_2)<0, at least one root is there!

Let

$$x_{\mathrm{m}} = \frac{(x_1 + x_2)}{2}$$

.

Stage 2: We then compute f(x_m) and check for sign change between x_1 and x_m or x_2 and x_m. One of the intervals will be discarded.

- If f(x_1)f(x_2)=0, x_m is the root. End the iterations hereafter.
- If f(x_1)f(x_m)<0, the root lies in interval (x_1,x_m). Repeat Stage 1 with $x_2 = x_m$.
- If f(x_m)f(x_2)<0, the root lies in interval (x_m,x_2). Repeat Stage 1 with $x_1 = x_m$.

This is repeated at every stage until the iterations are stopped.
Stage 3: While coding, we stop the iterations when the error at that stage.The error at the (i+1)th stage is given by:

$$E_{\mathrm{a}}{}^{(\mathrm{i+1})} = \left| \frac{x_{\mathrm{m}}{}^{(\mathrm{i+1})} - x_{\mathrm{m}}{}^{(\mathrm{i})}}{x_{\mathrm{m}}{}^{(\mathrm{i+1})}} \right|$$

4.2 Newton-Raphson Method

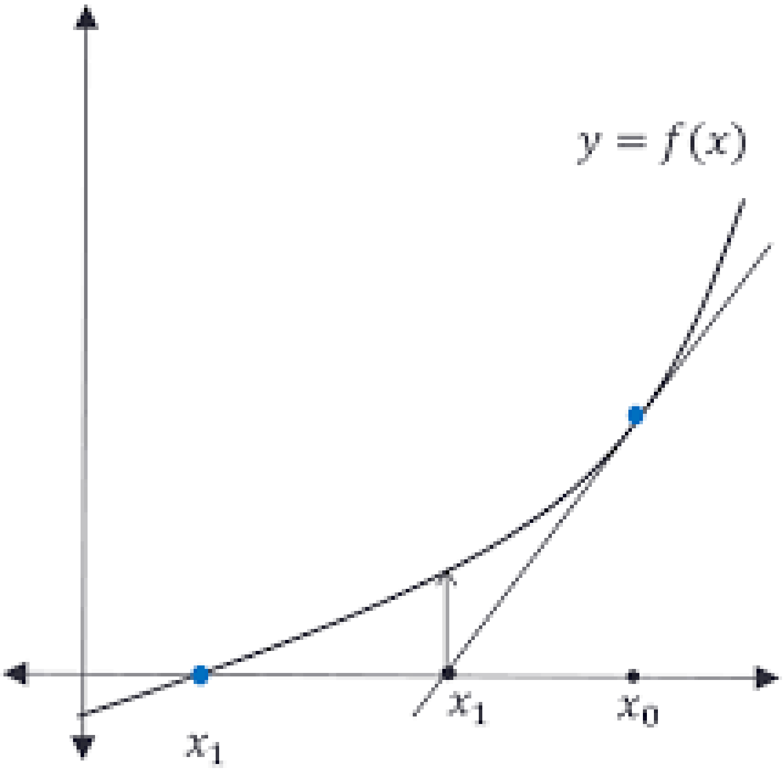


Figure 4: Newton-Raphson Method

Newton-Raphson Method, or simply called Newton's method, is a numerical method where we try to find the root of a function using an initial guess x_0 and approximating the function by its tangent line.
Stage 1: We consider an initial guess x = x_0 as the root of the function f(x).
Stage 2: The approximation of the root at the (i+1)th iteration is given by:

$$\boxed{x_{\mathrm{i+1}} = x_{\mathrm{i}} - \frac{f(x_{\mathrm{i}})}{f'(x_{\mathrm{i}})}} \quad (1)$$

Stage 3: While coding, we stop the iterations when the error at that stage.The error at the (i+1)th stage is given by:

$$E_{\mathrm{a}}{}^{(\mathrm{i+1})} = \left| \frac{x_{\mathrm{m}}{}^{(\mathrm{i+1})} - x_{\mathrm{m}}{}^{(\mathrm{i})}}{x_{\mathrm{m}}{}^{(\mathrm{i+1})}} \right|$$

4.3 Secant Method

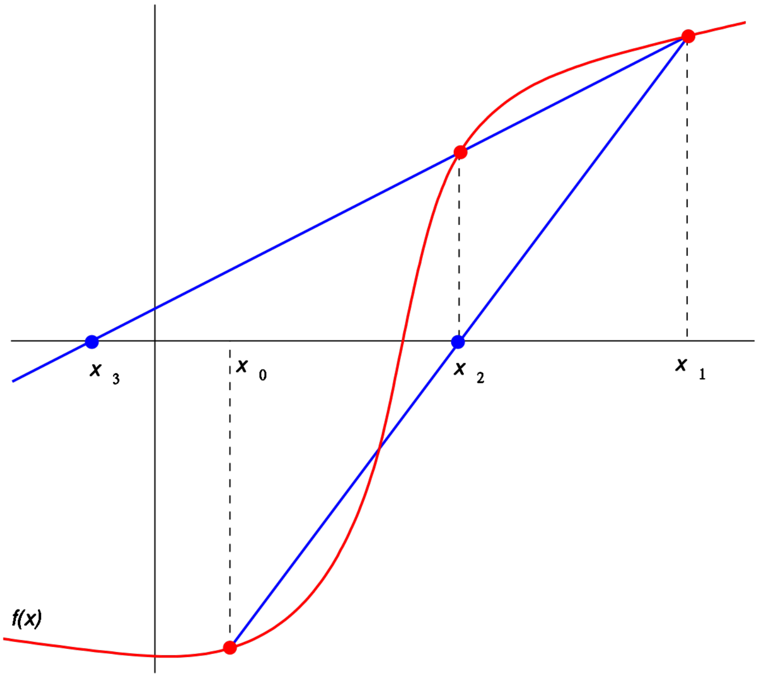


Figure 5: Secant method

The secant method is a recursive method for finding the root for polynomials by successive approximation.

In the secant method, we approximate the neighbourhoods of the roots by a secant line or chord to the function f(x).

Stage 1: We consider two initial guesses x = x_0 and x = x_1 as the neighbourhoods of the roots of the function f(x).

Stage 2: The approximation of the root at the (i+1)th iteration is given by:

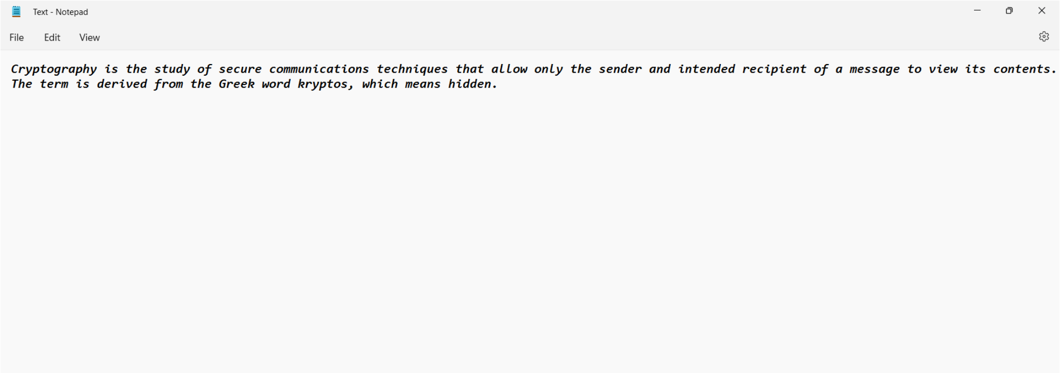
$$\boxed{x_{\mathrm{i+1}} = x_{\mathrm{i}} - \frac{(x_{\mathrm{i+1}} - x_{\mathrm{i}})f(x_{\mathrm{i}})}{f(x_{\mathrm{i+1}}) - f(x_{\mathrm{i}})}} \tag{2}$$

Stage 3: While coding, we stop the iterations when the error at that stage.The error at the (i+1)th stage is given by:

$$E_{\mathrm{a}}{}^{(\mathrm{i+1})} = \left| \frac{x_{\mathrm{m}}{}^{(\mathrm{i+1})} - x_{\mathrm{m}}{}^{(\mathrm{i})}}{x_{\mathrm{m}}{}^{(\mathrm{i+1})}} \right|$$

5 Python code and Results

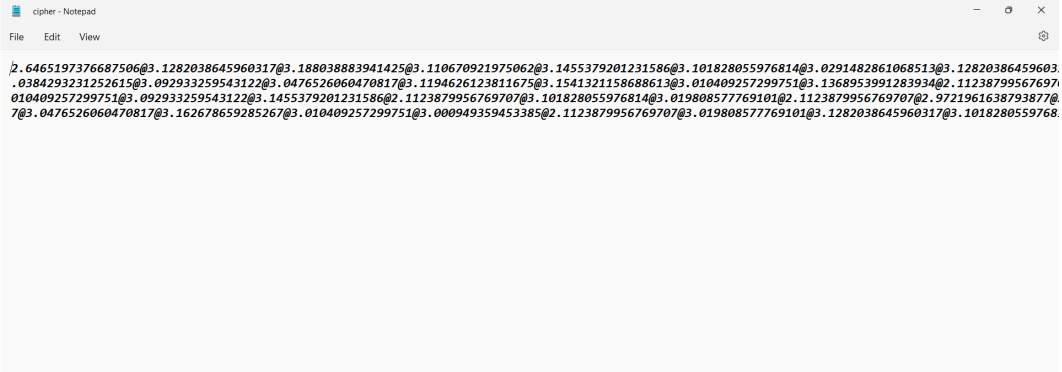
As an example we have used the following plain text:



```
Cryptography is the study of secure communications techniques that allow only the sender and intended recipient of a message to view its contents.
The term is derived from the Greek word kryptos, which means hidden.
```

Figure 6: Original Text

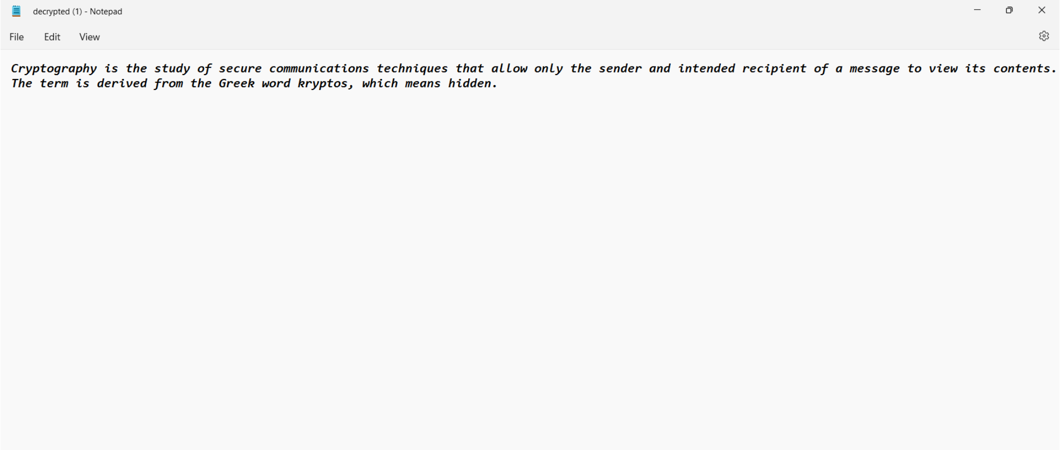
The encrypted text (that is, the cipher text) that we obtained after using function f(x) as (Secret Key)x^3 - 2.7x - (ASCII) = 0 is shown below:



```
2.6465197376687506@3.1282038645960317@3.188038883941425@3.110670921975062@3.1455379201231586@3.101828055976814@3.0291482861068513@3.12820386459603
.0384293231252615@3.092933259543122@3.0476526060470817@3.1194626123811675@3.1541321158688613@3.010409257299751@3.1368953991283934@2.11238799567697
010409257299751@3.092933259543122@3.1455379201231586@2.1123879956769707@3.101828055976814@3.019808577769101@2.1123879956769707@2.9721961638793877@
7@3.0476526060470817@3.162678659285267@3.010409257299751@3.000949359453385@2.1123879956769707@3.019808577769101@3.1282038645960317@3.10182805597 68
```

Figure 7: Encrypted Text (Cipher Text)

The decrypted text that we obtained using the decryption code is shown below:



```
Cryptography is the study of secure communications techniques that allow only the sender and intended recipient of a message to view its contents.
The term is derived from the Greek word kryptos, which means hidden.
```

Figure 8: Decrypted Text (Same as the original text)

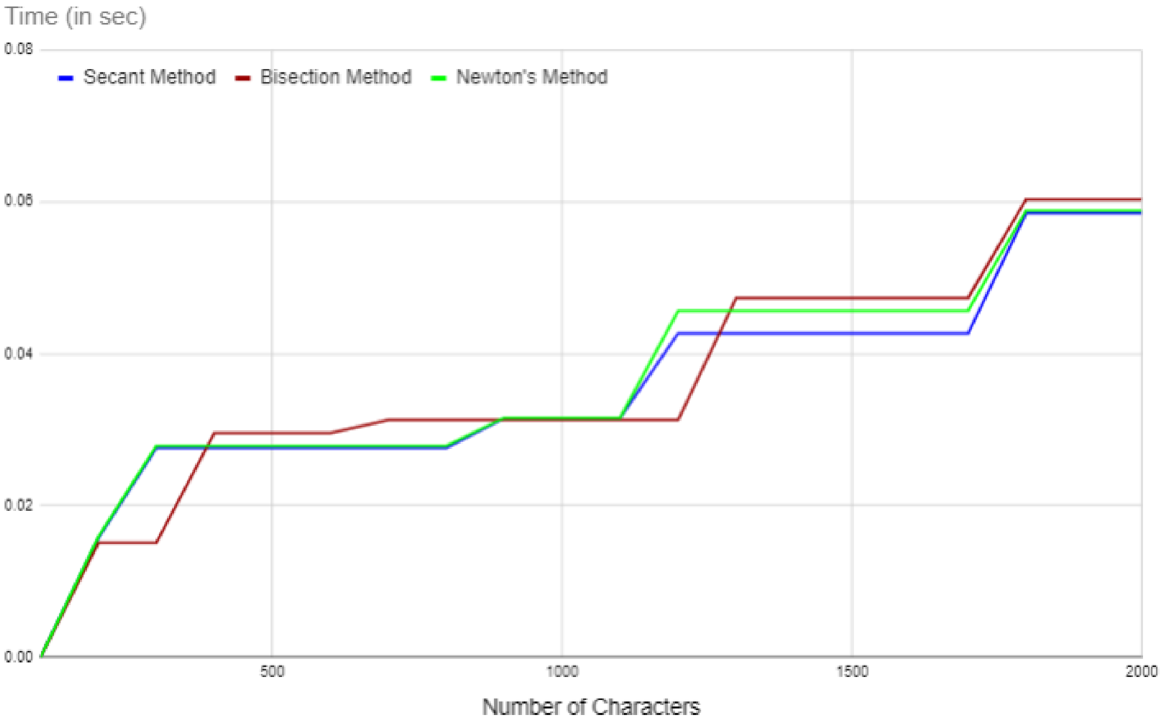
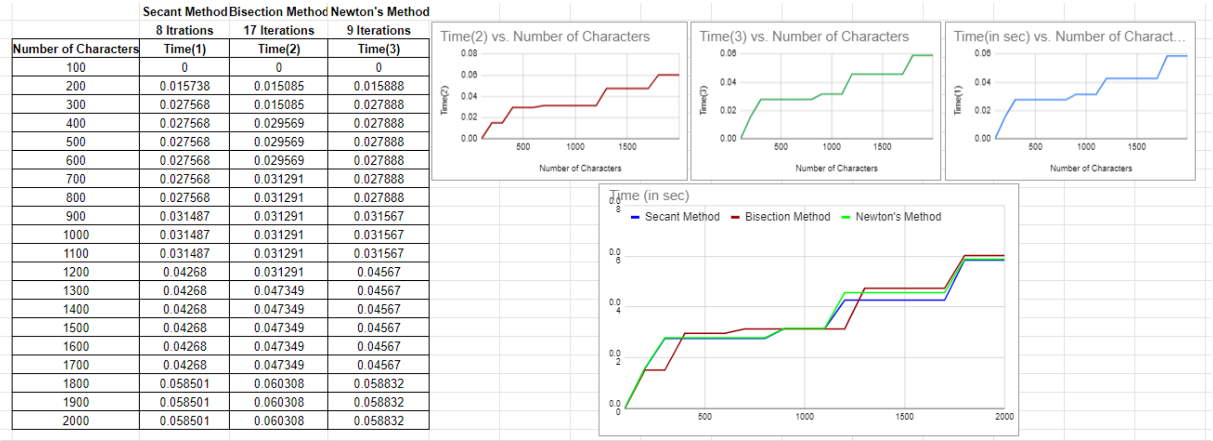


Figure 9: Plot of Computation Time vs Number of characters

6 Observations and Conclusions



	Secant Method	Bisection Method	Newton's Method
	8 Itrations	17 Iterations	9 Iterations
Number of Characters	Time(1)	Time(2)	Time(3)
100	0	0	0
200	0.015738	0.015085	0.015888
300	0.027568	0.015085	0.027888
400	0.027568	0.029569	0.027888
500	0.027568	0.029569	0.027888
600	0.027568	0.029569	0.027888
700	0.027568	0.031291	0.027888
800	0.027568	0.031291	0.027888
900	0.031487	0.031291	0.031567
1000	0.031487	0.031291	0.031567
1100	0.031487	0.031291	0.031567
1200	0.04268	0.031291	0.04567
1300	0.04268	0.047349	0.04567
1400	0.04268	0.047349	0.04567
1500	0.04268	0.047349	0.04567
1600	0.04268	0.047349	0.04567
1700	0.04268	0.047349	0.04567
1800	0.058501	0.060308	0.058832
1900	0.058501	0.060308	0.058832
2000	0.058501	0.060308	0.058832

Figure 10

1. The results of the comparison show that the Newton-Raphson method and Secant method are quite efficient, while methods like Bisection method do not work very efficiently for a larger length of text.

2. From Figure 6, it can be very well observed that the Newton-Raphson method and the Secant method take very less time and very less iterations for encrypting the same text message.

3. The results we obtained hold with our general expectation that the Newton-Raphson method and the Secant method takes lesser time to converge.

7 Future Scope

- We have used text-to-text encryption, that is, we are encrypting the required text in some other non-susceptible text.
  This encryption can be extended to more complex steganography techniques, such as encrypting text in an image, or even more complex such as, encrypting an image in a video file.

- Further we can extend the study to compare the results for other numerical methods as well, such as the Brent's method, Runge-Kutta method,etc.

- In the current world, there is a possibility of the insecurity of the data through encryption. Researchers are still looking for more secure encryption. There may exist some futuristic encryption method like Honey Encryption, in which information is created by wrong guess that looks like accurate, Quantum key encryption, where quantum atoms protects the data, Functional Encryption, where restricted secret keys enable a key holder to learn about only a specific function of encrypted data and nothing else.

8 Contributions

All of our team members contributed to the successful completion of the project, and worked hard and well above their potential. However the following work done by the members is worth mentioning:

- Vrajesh and Mumuksh helped formulate the entire problem statement and brief of the problem.

- Daniel and Mumuksh helped us in understanding the entire cryptography process. This includes topics from Diffie-Hellman process and the generation of secret keys to the entire encryption-decryption algorithm.

- Vrajesh, Daniel, and Mumuksh contributed towards the code for text-to-text encryption, while Jinay and Kush formulated the MATLAB code (which has not been included in this report).

- Jinay and Kush completed the entire documentation process, including the documentation in LaTex, writing the entire content. They also created the slides for the presentation.

9 References and Bibliography

1. Diffie-Hellman

2. Future of Encryption

3. Tech Target, Diffie-Hellman.

4. Wolfram

5. FedTechMagazine

6. Basics of Cryptography

7. Implementation of Diffie-Hellman

8. Implementation of Diffie-Hellman

9. Jonathan Blackledge, Cryptography Using Steganography: New Algorithms and Applications

10. AESHA N. ELGHANDOUR, AHMED M. SALAH1, YASSER A. ELMASRY,ABDELRAHMAN A. KARAWIA, An Image Encryption Algorithm Based on Bisection Method

11. Nagunwa T.;"Examining Usage of Web Browser Security Indicators in ebanking:A Case Study"; International Journal of Advanced Research in Computer Science and Software Engineering; Volume 4,Issue 9,September 2014.

12. Alfred J.Menezes, Paul C.Van Oorschot and Scott A.Vanstone; "Handbook of Applied Cryptography";1996.

13. Whitfield D. and Martin E.Hellman;"New Directions in cryptograhy" ;IEEE Transactions on Information Theory;Vol.22,No.6 November 1976.

14. Song Y.Yan; "Number Theory for Computing"; Second Edition.(2002).

15. J.Buchmann;" Introduction to cryptography"; 2nd ed.(2004)

16. Kandasamy , P. Thilagavathy, K. Gunavathy,k., Numerical Methods, S.Chand and Co.New Delhi 2008.

17. Saxena , H.C. ,Finite Differences and Numerical Analysis,S.Chand and Co.New Delhi 2008.

18. Amartya Ghosh and Anirban Saha, NUMERICAL METHOD BASED ENCRYPTION ALGORITHM