MathCrypt Secure Algorithm

Sobitha Ahila S
Department of Computer Science
Vellore Institute of Technology Chennai
Chennai, India
sobithaahila.s@vit.ac.in

Amish Kedia
Department of Computer Science
Vellore Institute of Technology Chennai
Chennai, India
amish.kedia2021@vitstudent.ac.in

Manmeet Singh
Department of Computer Science
Vellore Institute of Technology Chennai
Chennai, India
manmeet.singh2021@vitstudent.ac.in

***Abstract*—In an era where data security is paramount, our research focuses on the implementation of a novel symmetric encryption-decryption algorithm. This project uses a unique method to identify the roots of one-way functions that serve as ciphertext for messages. For the initial exchange of keys, Diffie-Hellman is used to ensure secure communication. The encryption process uses these numerical methods to increase security. Decryption is done the opposite way. This paper provides a detailed description of the encryption and decryption process, as well as comparing the performance of the various numerical methods used.**

***Index Terms*—data security, cryptographic performance, one-way function, root-finding algorithms, Diffie-Hellman algorithms, steganography, bisection method, Newton-Raphson method, fixed-point iteration**

I. Introduction

The need for robust data protection mechanisms is more important than ever in the digital age. In an age where information is constantly exchanged via various communication channels, it has become increasingly difficult to ensure the integrity and confidentiality of data. Cryptography is a key component in addressing security concerns. It provides methods for encoding information so that only the intended recipients can access and decode it.

Among the myriad of cryptographic techniques available, symmetric encryption-decryption algorithms stand out due to their efficiency and simplicity. The symmetric encryption process relies on the same shared key to encrypt and decrypt data, making it an ideal choice for fast and secure data transmission. The robustness of encryption methods is crucial to the security of these algorithms.

This research introduces a novel symmetric encryption-decryption algorithm that leverages numerical methods to enhance security. This approach, unlike traditional algorithms, uses one-way functions to generate ciphertext from plaintext. These one-way functions are determined by various numerical methods and serve as encrypted data. The Diffie-Hellman algorithm is used for the initial key exchange, a method that has been proven to be efficient and secure in sharing secrets over unsecured channels.

This paper aims to present a comprehensive overview of the encryption-decryption algorithm, detailing the processes involved in both encryption and decryption. It also provides a comparison of the different numerical methods used to create algorithms, evaluating both their performance and effectiveness for cryptographic applications. The goal is to explore these innovative techniques and contribute to the development of more efficient and secure cryptographic systems.

There are many algorithms in order to enhance the encryption method but this project aims to implement a symmetric encryption-decryption algorithm. Encryption is done by the one-way function and Diffie-Hellman is used to encrypt it.

To encrypt our message, we use numerical methods to find the root of the one-way function. The decryption process is the exact opposite. The results of the numerical methods are compared and presented.

A. Symmetric key algorithms

There are many symmetric key exchange algorithms. AES, IDEA, and RC4 are the most basic. These algorithms are not very secure in practice. Our computer can detect the key generated by AES in just one day, and keys generated using other algorithms in only a few days. Therefore, we use Diffie-Hellman, one of the safest algorithms. It is theoretically possible to detect the secret key, but it is practically not feasible because it takes about a year for it to be cracked due to data loss during the data transfer period. To exchange secret keys between two individuals, the first step is to physically exchange them. The key exchange method allows two people to communicate using a symmetric-key cipher, even if they have never met before.

B. Assumptions taken

1) The function $f(x)$ taken for generating ciphertext is used only for the purpose of explaining the use of numerical methods. Practically, much more complex one-way functions are used.
2) It only assumes 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.

C. Diffie-Hellman Algorithm

1) Agree on a prime number p and base g.
2) Sender chooses secret key a, receiver chooses b.
3) Sender computes $k = g^a \mod p$, receiver computes $m = g^b \mod p$.

4) Sender finds $s = g^{bm} \mod p$, receiver finds $s = g^{ak} \mod p$.
5) Both 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 $b = 2$. These private keys are known only to Alice and Bob.

Note that p is greater than g. The intermediate key generated by 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 way because the Diffie-Hellman algorithm is a symmetric key exchange algorithm.

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. However, if $f(x)$ is a polynomial of higher order or is a transcendental function, an exact solution may not exist. In these cases, numerical methods become crucial for finding an approximate root. We describe the following numerical methods and use them to find the one-way function.

D. Numerical Methods

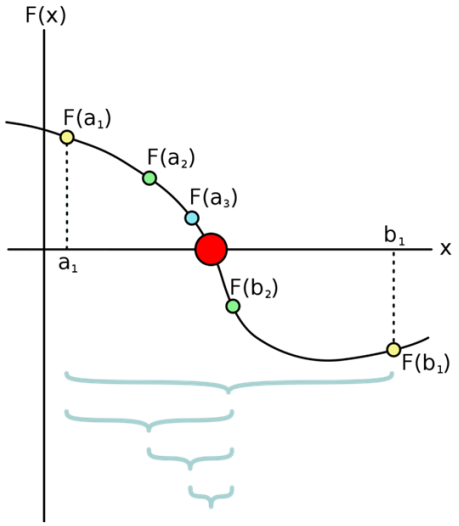

Fig. 1. Bisection method

1) Bisection Method: The bisection or interval halving method is a type of incremental search method that always divides the interval into halves (finding an interval where a function changes signs). Knowing that the root is in the interval where the function changes signs.

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

$$x_m = \frac{x_1 + x_2}{2} \tag{1}$$

Stage 2: 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 $f(x_m) = 0$, x_m is the root. End the iterations hereafter.
- If $f(x_1)f(x_m) < 0$, the root lies in the 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 the 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, stop the iterations when the error at that stage is less than the desired tolerance. The error at the $(i+1)^{\text{th}}$ stage is given by:

$$E^{(i+1)} = \left| x_m^{(i+1)} - x_m^{(i)} \right| / \left| x_m^{(i+1)} \right| \tag{2}$$

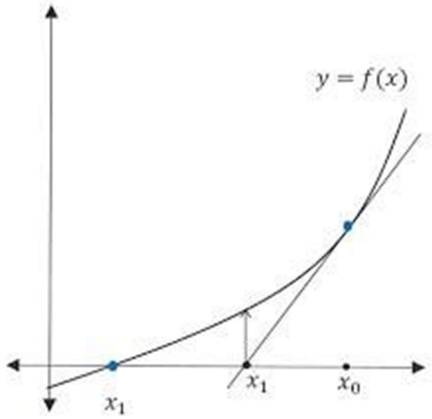

Fig. 2. Newton-Raphson method

2) Newton-Raphson Method: The Newton-Raphson method, or simply called Newton's method, is a numerical method to find the root of a function using an initial guess x_0 and approximating the function by its tangent line.

Stage 1: Consider an initial guess $x = x_0$ as the root of the function $f(x)$.

Stage 2: The approximation of the root at the $(i+1)^{\text{th}}$ iteration is given by:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \tag{3}$$

Stage 3: While coding, stop the iterations when the error at that stage is less than the desired tolerance. The error at the $(i+1)^{\text{th}}$ stage is given by:

$$E^{(i+1)} = |x_{i+1} - x_i| \,/\, |x_{i+1}| \tag{4}$$

3) 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: Consider two initial guesses $x = x_0$ and $x = x_1$ as the neighbourhoods of the roots of the function $f(x)$.

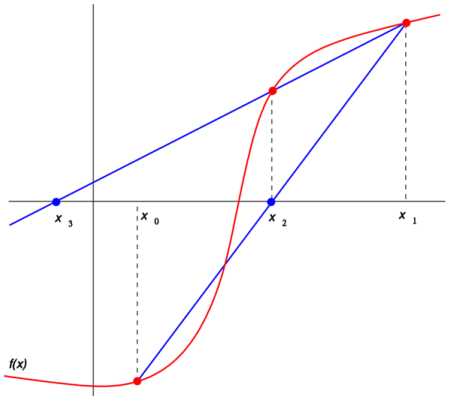

Fig. 3. Secant method

Stage 2: The approximation of the root at the $(i+1)^{\text{th}}$ iteration is given by:

$$x_{i+1} = x_i - f(x_i)\left(\frac{x_i - x_{i-1}}{f(x_i) - f(x_{i-1})}\right) \tag{5}$$

Stage 3: While coding, stop the iterations when the error at that stage is less than the desired tolerance. The error at the $(i+1)^{\text{th}}$ stage is given by:

$$E^{(i+1)} = |x_{i+1} - x_i| \,/\, |x_{i+1}| \tag{6}$$

E. Mathematical Approach

One-way functions are those functions that can be easily computed in the real world. However, given an output, it's difficult to find the input that produces that output. It's like finding x for a certain value of $f(x)$. The one-way functions give a unique address to an input, but it is impossible to retrieve the original input. In cryptography, one-way functions are used to encode messages.

It is possible to check the accuracy of the results by checking the inputs and outputs from the initial phase. In this project, we will use three different numerical methods to approximate the root of a one-way function in cryptography. Comparing the results and iterations of all methods from the perspective of the conclusion.

II. Literature Survey

Jonathan Blackledge [1] explores new methods to embed encrypted data in other media, enhancing security by obscurity. This dual-layered data protection approach is especially relevant to our work where we combine encryption with robust mathematical methods. The study shows that steganography is effective in hiding data and provides an additional layer to our encryption algorithm.

A. Elghandour, A. M. Salah, Y. A. Elmasry, and A. A. Karawia [2] present an image encryption algorithm based on the bisection method. The authors show how numerical methods are effective in cryptographic applications, reinforcing our approach to using numerical techniques to determine one-way functions.

Examining usage of web browser security indices in e-banking: A case study, Nagunwa T. [3] highlights the importance for users to be aware of security issues and adopt secure practices when conducting digital transactions. This study emphasizes the importance of robust encryption mechanisms in safeguarding sensitive information.

Handbook of Applied Cryptography, by Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone [4] offers a comprehensive guide to cryptographic principles. This handbook has been instrumental in shaping our understanding of various cryptographic protocols, including the implementation of our symmetric encryption-decryption algorithm.

Whitfield Diffie and Martin E. Hellman in "New Directions in Cryptography" [5] introduce the concept of public key cryptography and the Diffie-Hellman algorithm for key exchange. Their groundbreaking work is the foundation of our key-exchange process, ensuring secure communication between parties.

Song Y. Yan [6] delves deep into the mathematical foundations that are essential for cryptographic algorithms. This text provides valuable insight into the numerical methods used for encryption, focusing on the computational aspects of number theory.

J. Buchmann [7] provides a comprehensive overview of cryptographic algorithms and principles, with a focus on practical applications. This work has been crucial in guiding the development and implementation of our encryption-decryption algorithm, ensuring its theoretical soundness and practical viability.

In *Numerical Methods*, Kandasamy, Thilagavathy, and Gunavathy [8] provide detailed methodologies for solving mathematical problems, including the bisection method. Their exploration of different numerical techniques informed our approach to finding the root in encryption.

H.C. Saxena [9] in *Finite Differences and Numerical Analysis* offers a detailed examination of numerical analytical techniques with an emphasis on accuracy and efficiency. This was essential to our comparative evaluation of the performance and effectiveness of different numerical methods used in cryptographic applications.

"Numerical Method Based Encryption Algorithm," by Amartya Ghosh and Anirban Saha [10], gives practical insight into the use of numerical methods for cryptography. Their project report served as a reference to implement and test our algorithm, providing real-world examples.

Through this literature survey, we have identified and integrated key concepts and methodologies from existing research, enabling us to develop a robust and innovative approach to symmetric encryption-decryption. Our algorithm uses numerical methods to enhance security, but it also relies on cryptographic principles for secure communication.

III. System Design

The proposed system is designed to ensure secure communication using a novel encryption-decryption algorithm based on numerical methods and Diffie-Hellman key exchange. The

overall workflow comprises plaintext processing, key generation, encryption using numerical methods, and decryption to recover the original message. The following subsections detail each component of the system.

A. System Overview

The system can be divided into the following stages:

1) Input Processing: The plaintext message is converted into numerical values for processing.
2) Key Exchange: Secure keys are generated and shared between the sender and receiver using the Diffie-Hellman algorithm.
3) Encryption: The plaintext values are encoded into ciphertext using a one-way function and numerical methods.
4) Decryption: The ciphertext is decoded back into plaintext by reversing the encryption process.
5) Output Reconstruction: The decrypted numerical values are converted back into the original message.

The flow of these stages is depicted in Figure 4.

B. Encryption Algorithm

The encryption algorithm involves converting plaintext into secure ciphertext using numerical methods. Each step is detailed below:

1) Text Conversion: The plaintext message is converted into its numerical representation using ASCII encoding. Each character in the text corresponds to a unique integer value, ensuring a consistent numerical format for further processing.
2) Key Generation: The Diffie-Hellman algorithm is used to securely exchange keys between the sender and receiver. This ensures that both parties have access to the same secret keys without exposing them during transmission.
3) One-Way Function: A one-way mathematical function $f(x)$ is constructed using the generated keys. This function is computationally easy to evaluate in one direction but infeasible to reverse without the secret keys.
4) Numerical Methods for Encryption:
   - The one-way function ($f(x)$ = ASCII value of the text message) is solved for x using numerical methods (e.g., Bisection, Newton-Raphson, or Secant method).
   - The root of this equation becomes the ciphertext for the corresponding character in the plaintext message.
5) Ciphertext Generation: The solutions (roots of $f(x)$) for all characters are stored as an array, representing the encrypted data.

C. Decryption Algorithm

Decryption is the reverse process of encryption, designed to recover the original plaintext from the ciphertext. The steps are outlined below:

1) Ciphertext Retrieval: The array of solutions (ciphertext) is received by the intended recipient.
2) Reconstruction of ASCII Values: Each value in the ciphertext is substituted back into the one-way function $f(x)$. This step computes the corresponding ASCII values by solving the function in reverse.
3) Text Conversion: The ASCII values are converted back into their corresponding characters, reconstructing the original plaintext message.
4) Validation: The recovered plaintext is compared with the original message to ensure the integrity and accuracy of the decryption process.

D. Flow Diagram

Figure 4 illustrates the flow of the system design, showing the key stages: input processing, key exchange, encryption, decryption, and output reconstruction.

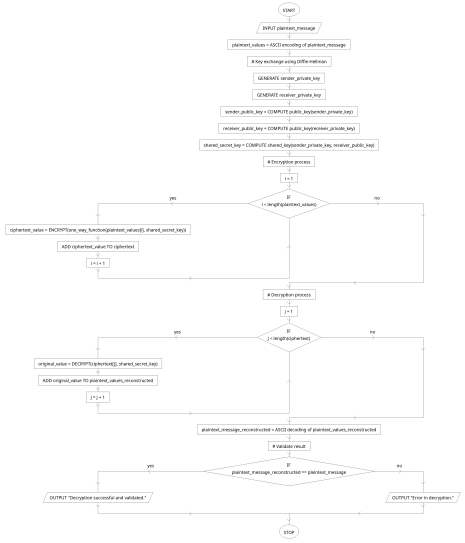

Fig. 4. Flow diagram of the system design showcasing key stages: input, key exchange, encryption, decryption, and output.

IV. Results

The results obtained from the implementation of the proposed encryption-decryption algorithm demonstrate its effectiveness and efficiency. This section provides an overview of the key outputs, including the original plaintext, encrypted ciphertext, and decrypted plaintext, along with a performance analysis of the numerical methods used.

A. Original Plaintext

As an example, we used a simple text message as input for the encryption process. The original plaintext message is depicted in Fig. 5. This serves as the baseline for evaluating the accuracy and reliability of the encryption and decryption processes.

Fig. 5. Original plaintext message.

B. Encrypted Ciphertext

The encryption algorithm was applied to the plaintext message using the one-way function:

$$f(x) = (\text{Secret Key})x^3 - 2.7x - (\text{ASCII}).$$

Numerical methods were used to find the roots of $f(x)$, which were stored as the encrypted ciphertext. The resulting ciphertext is shown in Fig. 6. The ciphertext is secure and computationally infeasible to reverse without access to the secret keys and the numerical decryption process.

Fig. 6. Encrypted text (ciphertext) obtained using the encryption algorithm.

C. Decrypted Plaintext

The decryption algorithm successfully reversed the encryption process, recovering the original plaintext message. By substituting the ciphertext values into the one-way function and solving for the ASCII values, the decryption process reconstructed the original text accurately. The decrypted plaintext is identical to the original message, as shown in Fig. 7.

Fig. 7. Decrypted text (same as the original plaintext).

D. Performance Analysis

The computational performance of the encryption algorithm was evaluated by analyzing the computation time for different numbers of characters in the plaintext message. The results are depicted in Fig. 8.

Key observations from the performance analysis:

- The Secant and Newton-Raphson methods significantly reduced the computation time compared to the Bisection method, particularly for longer texts.
- The Bisection method, while robust, demonstrated slower convergence, making it less suitable for large-scale encryption tasks.
- Both the Secant and Newton-Raphson methods scaled well with the increase in message size, maintaining efficient performance.

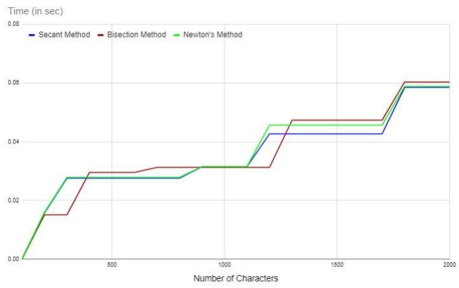

Fig. 8. Computation time vs. number of characters in the plaintext message.

E. Summary of Results

The results confirm the viability of the proposed encryption-decryption algorithm:

- The accuracy of the decryption process demonstrates the reliability of numerical methods in cryptographic applications.
- The performance comparison highlights the advantages of using iterative methods (Secant and Newton-Raphson) for fast and scalable encryption.
- The robustness of the system ensures secure communication without compromising computational efficiency.

V. Conclusions and Future Scope

A. Conclusion

This research introduces a novel approach to symmetric encryption and decryption by integrating numerical methods with the Diffie-Hellman algorithm. The findings from this study demonstrate the feasibility, efficiency, and robustness of the proposed algorithm for secure communication. Below are the key takeaways and observations from the study:

1) Efficiency of Numerical Methods: Among the numerical methods evaluated, the Secant and Newton-Raphson methods demonstrated superior performance in terms of computational efficiency. These methods required fewer iterations to converge to a solution, making them well-suited for real-time cryptographic applications.
2) Limitations of the Bisection Method: The Bisection method, while robust and simple, proved less effective for longer texts or larger datasets due to its slower convergence rate compared to other methods. This resulted in higher computation times in practical scenarios.
3) Performance Comparison: As shown in Fig. 8, the computation time for the Secant and Newton-Raphson methods was significantly lower than the Bisection method for the same encryption workload, highlighting their scalability and suitability for real-world applications.
4) Alignment with Expectations: The observed results align with theoretical predictions, confirming that iterative methods like Newton-Raphson and Secant are faster due to their higher-order convergence properties. These methods effectively reduce processing time without compromising the accuracy or security of the encryption.
5) Practical Implications: The proposed algorithm provides a robust framework for secure communication by leveraging numerical methods and the Diffie-Hellman algorithm. The results indicate its potential applicability in scenarios requiring high-performance and scalable encryption and decryption.

B. Future Scope

This research paves the way for further exploration and optimization of cryptographic systems through the integration of numerical methods. Several directions for future work are outlined below:

1) Advanced Numerical Techniques: Future research can explore more advanced numerical methods, such as Brent's Method or the Runge-Kutta Method, which offer potential for improved accuracy and efficiency. These methods can be tailored to specific encryption needs, optimizing the performance for larger datasets or complex messages.
2) Integration with Steganography: The algorithm can be extended to combine encryption with steganography, enabling the embedding of encrypted text within other media formats, such as images or videos. This dual-layered approach enhances security by adding an additional layer of obfuscation to the encrypted data.
3) Quantum-Resistant Encryption: With the advent of quantum computing, traditional cryptographic methods face increasing vulnerabilities. Research can focus on adapting the proposed algorithm to quantum-resistant frameworks, utilizing quantum key distribution (QKD) and quantum-safe numerical methods to secure future communications.
4) Functional Encryption: The algorithm can be extended to incorporate functional encryption, allowing specific computations to be performed on encrypted data without decrypting it. This feature is particularly useful in privacy-preserving applications, such as cloud computing and secure data sharing.
5) Dynamic Key Exchange Mechanisms: Beyond the Diffie-Hellman algorithm, exploring dynamic and adaptive key exchange mechanisms can provide enhanced security in scenarios where multiple users or devices are involved. These mechanisms can include elliptic curve cryptography (ECC) or lattice-based methods to ensure robust key sharing.
6) Cross-Platform Implementation and Optimization: The algorithm can be implemented across various platforms, including mobile devices, embedded systems, and cloud environments. Optimizing the algorithm for resource-constrained systems can extend its applicability to Internet of Things (IoT) and edge computing use cases.
7) Performance Benchmarking and Scalability Testing: Extensive benchmarking across diverse datasets and encryption workloads can provide deeper insights into the scalability of the algorithm. Future research can focus on identifying bottlenecks and optimizing the system for high-throughput applications, such as real-time messaging and large-scale data encryption.

The integration of numerical methods into cryptographic systems demonstrates significant promise for enhancing security and performance. While the current study validates the feasibility and efficiency of the proposed approach, the outlined future directions provide a pathway for further advancements in cryptographic research. These developments will contribute to creating more robust, scalable, and future-ready encryption systems capable of addressing evolving security challenges.

REFERENCES

[1] J. Blackledge, "Cryptography Using Steganography: New Algorithms and Applications," Lecture Notes, Centre for Advanced Studies, Warsaw University of Technology, Warsaw, 2011.

[2] A. Elghandour, A. M. Salah, Y. A. Elmasry, and A. A. Karawia, "An Image Encryption Algorithm Based on Bisection Method and One-Dimensional Piecewise Chaotic Map," *IEEE Access*, vol. 9, pp. 1–1, 2021.

[3] T. Nagunwa, "Examining Usage of Web Browser Security Indicators in eBanking: A Case Study," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 4, no. 9, pp. 1–6, Sept. 2014.

[4] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. Boca Raton, FL: CRC Press, 1996.

[5] W. Diffie and M. E. Hellman, "New Directions in Cryptography," *IEEE Transactions on Information Theory*, vol. IT-22, no. 6, pp. 644–654, Nov. 1976.

[6] S. Y. Yan, *Number Theory for Computing*, 2nd ed. Berlin, Germany: Springer-Verlag, 2002.

[7] J. Buchmann, *Introduction to Cryptography*, 2nd ed. New York: Springer-Verlag, 2004.

[8] P. Kandasamy, K. Thilagavathy, and K. Gunavathy, *Numerical Methods*. New Delhi: S. Chand and Co., 2008.

[9] H. C. Saxena, *Finite Differences and Numerical Analysis*. New Delhi: S. Chand and Co., 2008.

[10] A. Ghosh and A. Saha, "Numerical Method Based Encryption Algorithm," Project Report, 2015.