Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Symmetric Key Encryption Using Modular Traversal of Ananta-Graphs

Vidyashree H R[1], Lakshminarayana S[1]

[1]School of Applied Sciences, Department of Mathematics, REVA University, Bengaluru, 560 064, Karnataka, India

ARTICLE INFO

Received: 24 Dec 2024

Revised: 12 Feb 2025

Accepted: 26 Feb 2025

ABSTRACT

In this paper, we propose a novel symmetric key encryption scheme inspired by modular traversal of Ananta-Graphs. The system utilizes an iterative traversal mechanism over modular arithmetic to generate a shared secret key between communicating parties without directly exchanging private information. By leveraging the predictable yet secure path generation properties of the modular traversal, the shared key is derived independently by both participants, ensuring confidentiality and integrity. The encryption and decryption processes employ lightweight operations mapped into readable alphabetic ciphertext. Experimental analysis demonstrates consistent key synchronization, efficient encryption-decryption, and resilience against cryptographic attacks, particularly when using large prime moduli. This study highlights the potential of Ananta-Graph inspired modular traversal techniques for lightweight and secure symmetric cryptographic systems.

Keyword: Cryptography, Collatz Conjecture, Key Exchange Mechanisms, Ananta- Graphs, Public Key Cryptosystem.

Classifications: 94A60, 68R10, 68W40, 11Y16.

1 Introduction

Cryptography plays a crucial role in ensuring secure communication in today's interconnected world. Traditional symmetric key encryption methods rely heavily on shared secret keys established through secure channels or sophisticated key exchange protocols. The complexity and computational overhead of existing cryptographic methods have led researchers to explore lightweight, efficient alternatives, especially those inspired by mathematical structures such as graphs and modular arithmetic [1, 2, 3].

In this work, we propose a novel symmetric key encryption scheme based on the modular traversal of Ananta-Graphs [4, 5]. Ananta-Graphs are specially constructed dynamic graphs influenced by iterative function application, resembling trajectories found in number theory problems [5, 6]. Specifically, our traversal mechanism draws inspiration from the famous Collatz conjecture, which revolves around sequences generated by the function $3n+1$. Although we do not

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

use the full structure of Ananta-Graphs in this paper, the modular variant of the Collatz-like traversal forms the core of our key generation strategy[6, 7].

Our scheme consists of two main components: (1) a lightweight key exchange protocol based on modular graph traversal, and (2) symmetric key encryption and decryption processes that utilize the generated shared secret. We implement the proposed method in Python using a user-friendly Tkinter interface, allowing users to specify parameters such as the base node, modulus (a prime number), and private keys [8, 9].

The security of the method is enhanced by the unpredictability of modular traversal and the inherent difficulty in reversing the modular transformation without knowledge of the private keys. Furthermore, by leveraging properties of modular arithmetic and prime moduli, the system offers resistance to various cryptographic attacks while maintaining computational efficiency [4, 7].

2 Preliminaries

2.1 Ananta-Graphs and Traversal Mechanism

Ananta-Graphs are a class of dynamic graphs constructed by iteratively applying a mathematical function to generate vertices and edges. Each node represents an integer, and directed edges signify transitions from one node to another based on a defined function. These graphs are inherently infinite in nature, reflecting dynamic behaviors over the integer set [6].

In this work, we focus on a modular version of Ananta-Graphs inspired by the Collatz conjecture. The classical Collatz sequence is generated by repeatedly applying the function $f(n) = 3n + 1$ (followed by a division by 2 for even numbers), leading to complex, yet structured trajectories over integers. However, in our system, we simplify the traversal by considering only the modular form of $3n + 1$, defined as:

$$f(n) = (3n + 1) mod\ m$$

where m is a chosen prime modulus.

This modular traversal results in a finite cyclic graph structure where nodes and edges are bounded within the modular space. By selecting different starting nodes (base nodes) and applying the modular function iteratively for a number of steps determined by private keys, unique traversal paths are established. These paths ultimately define public keys and shared secret keys in our key exchange mechanism [5, 9, 10].

The use of modular Ananta-Graph traversal introduces nonlinearity and unpredictability in key generation, making it challenging for an external observer to predict or reverse engineer the keys without knowledge of the private traversal steps [11].

2.2 Modular Arithmetic Basics

Modular arithmetic involves performing calculations where numbers ”wrap around” after reaching a certain value known as the modulus. For integers a, b, and modulus m, the congruence relation is defined as:

$$a \equiv b\ (mod\ m)$$

Meaning, a and b leave the same remainder when divided by m.

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Modular operations are fundamental in cryptography due to their cyclic behavior and the difficulty of reversing certain operations, especially when large prime moduli are used. In this scheme, modular arithmetic ensures that traversals remain bounded and unpredictable, enhancing the cryptographic strength [3, 5].

2.3 Symmetric Key Cryptography Concepts

Symmetric key cryptography employs a single secret key for both encryption and decryption processes. The security of the system depends on keeping the shared secret undisclosed to adversaries.

Typically, symmetric systems offer:

- High-speed encryption and decryption.
- Lower computational overhead compared to asymmetric systems.
- Dependence on secure key exchange mechanisms to initiate communication.

In the proposed system, symmetric encryption is used after securely deriving a shared secret through Ananta-Graph traversal. The encryption process involves lightweight operations like XOR combined with modular transformations, and the output ciphertext is encoded into readable alphabetic characters for ease of transmission [11].

Proposed Methodology

This section describes the methodology used to achieve secure symmetric key encryption using modular traversal of Ananta-Graphs. The processes of key generation, encryption, and decryption are detailed, along with an overall protocol outline in pseudocode.

2.4 Key Generation Using Modular Traversal

Key generation relies on the traversal of an implicit Ananta-Graph using a modular transformation. The steps involved are:

- A public base node v_0 and a prime modulus m are agreed upon between the two parties.
- Each party (Alice and Bob) selects a private integer (a for Alice, b for Bob) as their private key.
- Alice and Bob independently compute their public keys:

$$v_A = f^{(a)}(v_o) \quad and \quad v_B = f^{(b)}(v_o)$$

where $f^{(k)}$ denotes applying the traversal function k times.

- Upon exchanging public keys, the shared secret key K is computed:

$$K = f^{(a)}(v_B) = f^{(b)}(v_A)$$

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

This ensures that both Alice and Bob independently compute the same shared key without exposing their private keys.

Algorithm 1 Key Generation Using Modular Traversal of Ananta-Graph

Input: Base node v_0, prime modulus m, Alice's private key a, Bob's private key b
Output: Shared secret key K
Alice computes public key: $v_A = f^{(a)}(v_0) \mod m$
Bob computes public key: $v_B = f^{(b)}(v_0) \mod m$
Alice receives v_B, Bob receives v_A
Alice computes shared key: $K = f^{(a)}(v_B) \mod m$
Bob computes shared key: $K = f^{(b)}(v_A) \mod m$
Return K

2.5 Encryption Process

Once the shared key K is established:

- The plaintext message is treated as a sequence of characters.
- Each character is converted to its ASCII value.
- An XOR operation is performed between each ASCII value and the shared key.
- The resulting cipher values are mapped back to printable alphabetic characters using modular encoding.

This transformation ensures that the ciphertext appears as readable text while maintaining confidentiality.

Algorithm 2 Encryption Process Using Shared Key

Input: Plaintext message M, shared secret key K
Output: Ciphertext C
for each character c in message M **do**
    Convert c to ASCII value
    Compute encrypted value: $e = \text{ASCII}(c) \oplus K$
    Convert e back to a character and append to ciphertext
end for
Return Ciphertext C

2.6 Decryption Process

The decryption process involves reversing the encryption steps:

- Each character from the received ciphertext is mapped back to its corresponding cipher value.
- An XOR operation with the shared key is performed to retrieve the original ASCII values.
- The ASCII values are converted back into the original plaintext characters.

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Since the XOR operation is symmetric, applying it twice with the same key restores the original message.

Algorithm 3 Decryption Process Using Shared Key

1: **Input:** Ciphertext C, shared secret key K 2:
Output: Decrypted plaintext message M 3:
for each character e in ciphertext C **do**
4: Convert e to ASCII value
5: Compute decrypted value: $d = \text{ASCII}(e) \oplus K$
6: Convert d back to a character and append to message
7: **end for**
8: **Return** Decrypted message M

2.7 Overall Protocol Description

The complete protocol for secure communication can be described by the following pseudocode:

Algorithm 4 Symmetric Encryption Using Modular Traversal of Ananta-Graphs

1: Agree on public values: base node v_0 and prime modulus m
2: Alice chooses private key a, Bob chooses private key b
3: Alice computes $v_A = f^{(a)}(v_0)$ and sends to Bob 4:
Bob computes $v_B = f^{(b)}(v_0)$ and sends to Alice 5:
Alice computes shared key $K = f^{(a)}(v_B)$
6: Bob computes shared key $K = f^{(b)}(v_A)$
7: Alice encrypts her message M using K and sends ciphertext C to Bob
8: Bob decrypts ciphertext C using K to retrieve message M

3 Implementation Details

3.1 Python Tkinter Application Overview

The proposed cryptographic system is implemented using Python with the Tkinter library to design a simple Graphical User Interface (GUI). The application provides input fields for users to enter the base node, prime modulus, and private keys for Alice and Bob. Buttons are provided to compute the shared secret key, encrypt a given message, and decrypt an encrypted message. The shared key is derived using modular traversal based on the Ananta-Graph transformation, and encryption/decryption is performed using simple XOR operations between the key and the ASCII values of the message characters. The application ensures ease of use and quick visualization of each stage in the encryption process.

3.2 Sample Inputs and Outputs

Below are sample values used during the operation of the application:

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

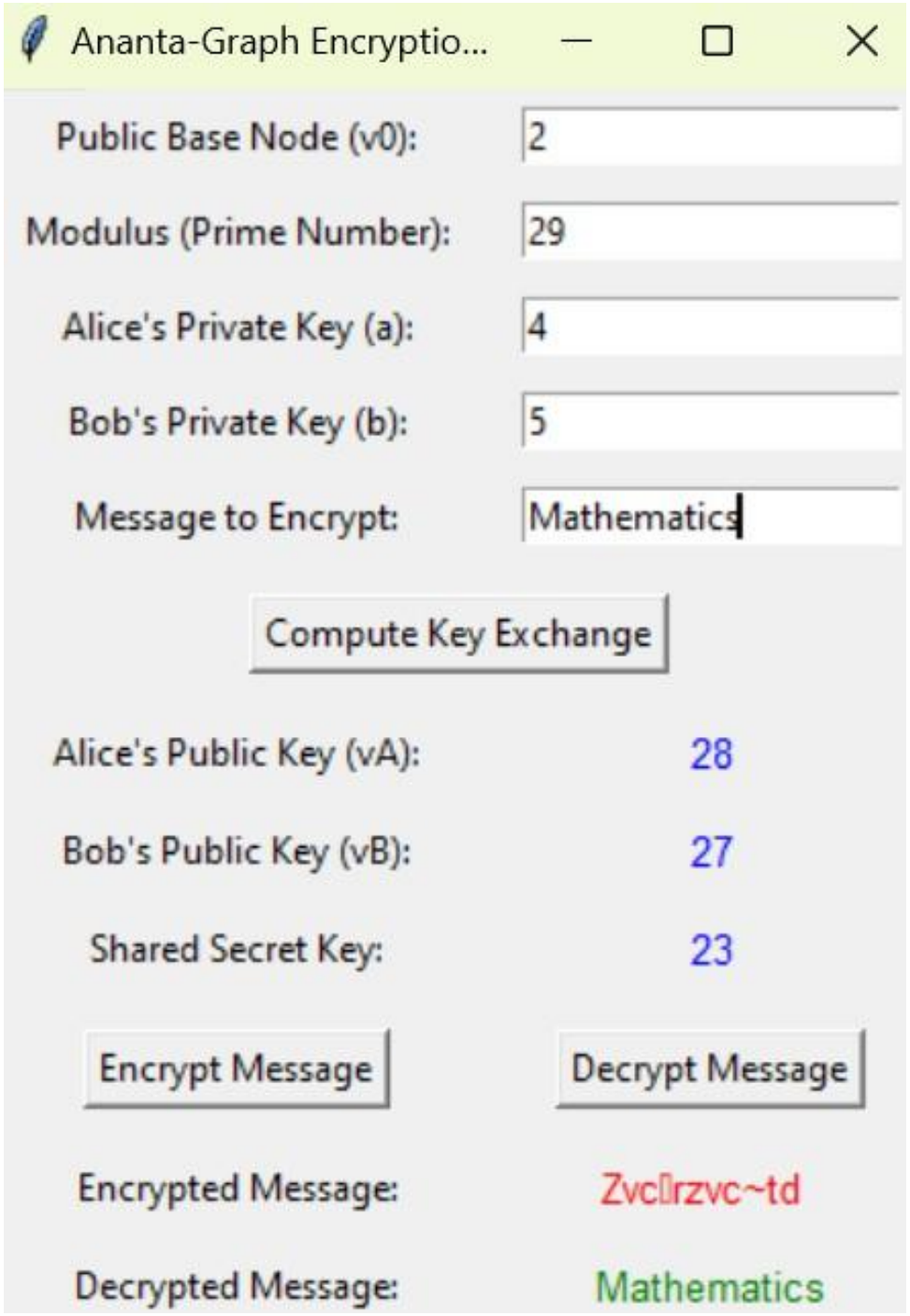

Figure 1: Encryption and Decryption Process.

3.3 Discussion on Parameters: Base Node, Modulus, Private Keys

The security and correctness of the scheme depend heavily on the chosen parameters:

- **Base Node (v_0):** Should be a small positive integer to ensure computational efficiency.

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

- **Modulus (m):** Must be a large prime number. A larger prime enhances the security of the key generation process by ensuring a greater traversal space and unpredictability.

- **Private Keys (a, b):** Should be randomly selected and kept confidential. Larger private keys lead to deeper traversals in the graph, increasing the difficulty of brute- force attacks.

Careful selection of these parameters ensures that the key exchange remains secure and resistant to cryptanalytic attacks while maintaining efficient encryption and decryption operations.

4 Security Analysis

4.1 Resistance to Eavesdropping

The security of the proposed key exchange and encryption scheme relies on the difficulty of predicting the traversal outcomes over the Ananta-Graph. Even if an attacker intercepts the public information, namely the base node (v_0), the modulus (m), and the public keys (v_A and v_B), they cannot easily determine the private keys (a and b) or the shared secret key (K) without solving the underlying modular traversal problem, which is computationally difficult for large moduli. This ensures that the system is resistant to passive eavesdropping.

4.2 Resistance to Brute Force Attacks

Brute force attacks would involve an adversary attempting all possible private keys to derive the shared secret key. The number of possible private keys grows with the size of the modulus. If a sufficiently large prime modulus is used (e.g., a 128-bit prime), the number of possible keys becomes practically infeasible to exhaustively search through, providing strong protection against brute-force attacks. Additionally, since the traversal transformation is non-linear and modular, guessing the key becomes even more complex than traditional discrete logarithm-based systems.

4.3 Importance of Choosing a Large Prime Modulus

The choice of the modulus is critical to maintaining security. A small modulus would lead to a small traversal space, making it easier for an attacker to perform an exhaustive search or pattern analysis. By choosing a large prime number as the modulus:

- The traversal space is significantly expanded.

- Collisions and repetitions in traversal are minimized.

- Reverse engineering the traversal sequence becomes computationally infeasible.

Therefore, for practical deployment, a prime modulus of at least 256 bits is recommended to meet modern cryptographic standards.

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

5 Comparison with Existing Methods

5.1 Comparison with Diffie-Hellman Key Exchange

The Diffie-Hellman key exchange protocol is one of the most established methods for secure key generation between two parties over an insecure channel[2]. It relies on the difficulty of solving the discrete logarithm problem in a finite field. In contrast, the proposed Ananta-Graph based key exchange uses modular traversal through a non-linear transformation inspired by the structure of the Ananta-Graph.

Key differences include:

- **Mathematical Foundation:**
  - Diffie-Hellman relies on exponential functions and discrete logarithms.
  - The proposed method relies on iterative modular traversal using a simple non-linear function.
- **Computation Complexity:**
  - Diffie-Hellman involves modular exponentiation, which can be computationally intensive.
  - Modular traversal operations are computationally simpler and faster, making the proposed scheme lightweight and suitable for low-resource environments.
- **Security Assumptions:**
  - Diffie-Hellman's security is well-studied and depends on the hardness of the discrete logarithm problem.
  - The proposed method's security stems from the difficulty of predicting traversal sequences without knowledge of private keys.

5.2 Advantages and Limitations

Advantages:

- **Simplicity:** The use of simple modular arithmetic operations makes the algorithm easy to implement and understand.
- **Efficiency:** Lightweight operations enable fast execution, even on devices with limited computational power.
- **Adaptability:** The method can be adapted to different modular functions to enhance security.

Limitations:

- **Security Maturity:** Unlike Diffie-Hellman, the security assumptions of the proposed method are less studied and require formal cryptanalysis.
- **Parameter Sensitivity:** Choosing inappropriate parameters (small modulus, predictable private keys) could compromise security.

Journal of Information Systems Engineering and Management
2025, 10(44s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

- **Scalability:** For extremely high-security needs (e.g., post-quantum cryptography), further enhancements or hybrid schemes may be necessary.

6 Conclusion

In this paper, we propose a lightweight symmetric key cryptography scheme based on modular traversal of Ananta-Graphs. The methodology integrates a secure key exchange protocol using simple yet effective modular operations derived from the traversal of an initial base node. An encryption and decryption mechanism was built upon the generated shared secret key, demonstrating the practicality of the approach through a Python Tkinter application. Security analysis highlights that with appropriate parameter choices, the scheme offers reasonable resistance against eavesdropping and brute-force attacks while maintaining computational efficiency.

References

[1] W. Stallings, *Cryptography and Network Security: Principles and Practice*, 7th ed., Pearson, 2017.

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

[3] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996.

[4] R. C. Merkle, "Secure Communications over Insecure Channels," *Communications of the ACM*, vol. 21, no. 4, pp. 294–299, 1978.

[5] D. Burton, *Elementary Number Theory*, 7th ed., McGraw-Hill Education, 2010.

[6] H. R. Vidyashree, S. Lakshminarayana, and S. Ajith, "Visualizing and Proving the Collatz Conjecture: Ananta-Graph Approach," *Communications on Applied Nonlinear Analysis*, vol. 32, no. 6s, pp. 16–25, 2025.

[7] J. Buchmann, E. Dahmen, and M. Schneider, "Graph-Based Cryptography," in *PostQuantum Cryptography*, Springer, 2009, pp. 177–202.

[8] A. Poschmann, "Lightweight Cryptography: Cryptographic Engineering for a Pervasive World," Ph.D. dissertation, Ruhr University Bochum, Germany, 2009.

[9] P. K. Srimani and B. Pushpalatha, "A New Public Key Cryptosystem based on Collatz Problem," *International Journal of Network Security & Its Applications (IJNSA)*, vol. 5, no. 2, pp. 89–97, Mar. 2013.

[10] B. Pushpalatha and P. K. Srimani, "Secure Key Exchange using Collatz-like Problem," *International Journal of Computer Applications*, vol. 58, no. 8, 2012.

[11] R. Mallikarjuna and B. Pushpalatha, "A Symmetric Cryptographic Algorithm using Modified Collatz Function," *International Journal of Computer Applications*, vol. 92, no. 17, 2014.