International Journal of Scientific Research in Computer Science, Engineering and Information Technology
ISSN : 2456-3307 (www.ijsrcseit.com)
doi :https://doi.org/10.32628/CSEIT1820116

Developing Random Prime Number Substitution Cipher for Cryptography

Dr. G. Sreedhar
Department of Computer Science, Rashtriya Sanskrit Vidyapeetha, Tirupati, India

Article Info

Publication Issue :
Volume 3, Issue 3
March-April-2018
Page Number : 2187-2190
Article History
Accepted: 10/03/2018
Published: 30/03/2018

ABSTRACT

This paper introduces the Random Prime Number Substitution Cipher (RPN-SC), a novel cryptographic technique that employs randomized prime number mappings to encrypt and decrypt textual data. By leveraging the mathematical properties of prime numbers and incorporating randomness, the RPN-SC enhances security against traditional cryptanalytic attacks. The cipher's implementation is demonstrated through a web-based application, providing an interactive platform for users to explore encryption and decryption processes.

Key terms — Decryption, Encryption, Prime Number Randomness, Substitution Cipher

I. INTRODUCTION

Cryptography has undergone significant evolution since the advent of classical ciphers like the Caesar cipher. The introduction of public-key cryptography in the 1970s, particularly through the Diffie-Hellman key exchange and the RSA algorithm, marked a paradigm shift by utilizing the computational difficulty of prime factorization for secure communication. A substitutional approach for cryptography is the basic method that is used in encryption and decryption process of cryptography. Building upon this foundation, the RPN-SC introduces randomness into prime number-based substitution, aiming to bolster security against modern cryptanalytic techniques.

II. RELATED WORK

The proposed research work is carried out by observing various authors contribution in the field of cryptography. The study provided foundation for utilization of prime numbers with randomization property in cryptography systems. The author Jain et al. (2015) proposed a randomized approach to enhance the Caesar cipher by incorporating affine and transposition techniques, thereby increasing the complexity and security of the encryption method. The author Naidu et al. (2014) developed a symmetric key algorithm that employs randomized prime numbers for character-based encryption, directly aligning with the principles of the RPN-SC. The scholars Dey et al. (2010) introduced a data hiding technique that decomposes pixel values into sums of prime numbers, demonstrating the versatility of primes in cryptographic applications.These works underscore the potential of prime numbers and randomization in enhancing cryptographic security.

III. METHODOLOGY

The methodology includes five phases. These are as follows.

1. Prime Number Generation

The generatePrimes(limit) function uses the Sieve of Eratosthenes algorithm to generate all prime numbers up to a specified limit. This method efficiently identifies prime numbers by iteratively marking the multiples of each prime starting from 2.

```
function generatePrimes(limit)
{
 const sieve = Array(limit + 1).fill(true);
 sieve[0] = sieve[1] = false;
 for (let i = 2; i * i <= limit; i++)
 {
  if (sieve[i]) {
   for (let j = i * i; j <= limit; j += i)
   {
    sieve[j] = false;
   }
  }
 }
 return sieve.map((isPrime, number) => isPrime ?
number : -1)
  .filter(number => number !== -1);
}
```

2. Shuffling Primes Once the prime numbers are generated, they are shuffled randomly using the sort(() => Math.random() - 0.5) method. This randomness ensures that each encryption is unique.

```
const shuffledPrimes = [...primes].sort(() =>
Math.random() - 0.5);
```

3. Mapping Letters to Primes

The alphabet letters 'A' to 'Z' are mapped to the shuffled prime numbers. This mapping is stored in two objects:primeMap: Maps each letter to a prime number.reversePrimeMap: Maps each prime number back to its corresponding letter.

```
const primeMap = {};
const reversePrimeMap = {};
for (let i = 0; i < alphabet.length; i++) {
 primeMap[alphabet[i]] = shuffledPrimes[i];
 reversePrimeMap[shuffledPrimes[i]] = alphabet[i];
}
```

4. Encrypting Text

To encrypt a message, the processText('encrypt') function iterates over each character of the input text. If the character is a letter, it appends the corresponding prime number to the output. Spaces and non-alphabetic characters are preserved.

```
function processText(action)
{
 const inputText =
document.getElementById('inputText').value.toUpper
Case();
 let outputText = '';
 for (let i = 0; i < inputText.length; i++) {
  const char = inputText[i];
  if (/[A-Z]/.test(char)) {
   outputText += primeMap[char] + ' ';
  } else if (char === ' ') {
   outputText += ' ';
  } else {
   outputText += char;
  }
 }
 document.getElementById('outputText').value =
outputText;
}
```

5. Decrypting Text

The decryption process involves reversing the mapping. The processText('decrypt') function splits the input by spaces, converts each prime number back to its corresponding letter using reversePrimeMap, and reconstructs the original message.

```
const numbers = inputText.split(' ');
for (let i = 0; i < numbers.length; i++) {
 const num = parseInt(numbers[i]);
 if (!isNaN(num) && reversePrimeMap[num]) {
  outputText += reversePrimeMap[num];
 } else if (numbers[i] === '') {
  outputText += ' ';
 } else {
  outputText += numbers[i];
```

All these five phases are implemented through web application so that any user can understand the

process. The screenshot of encryption is shown in figure 1. The screenshot of decryption is shown in figure2.

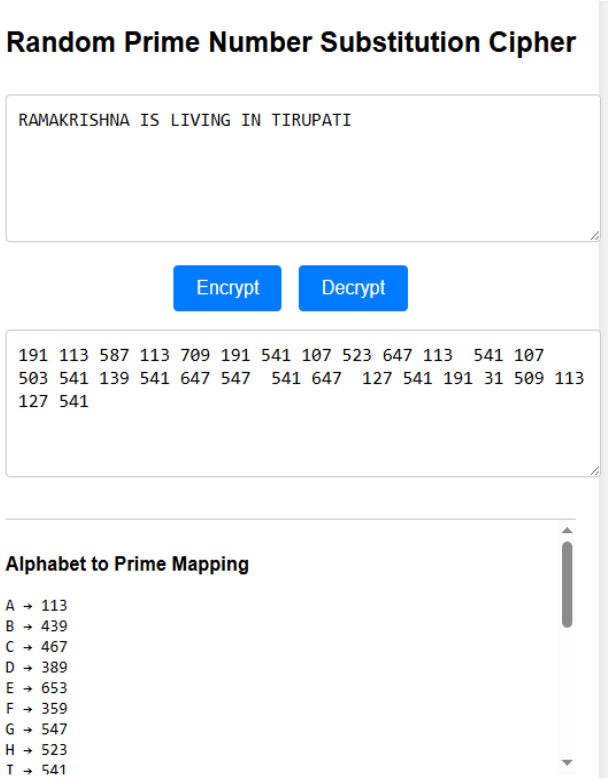

Figure1: Encryption process

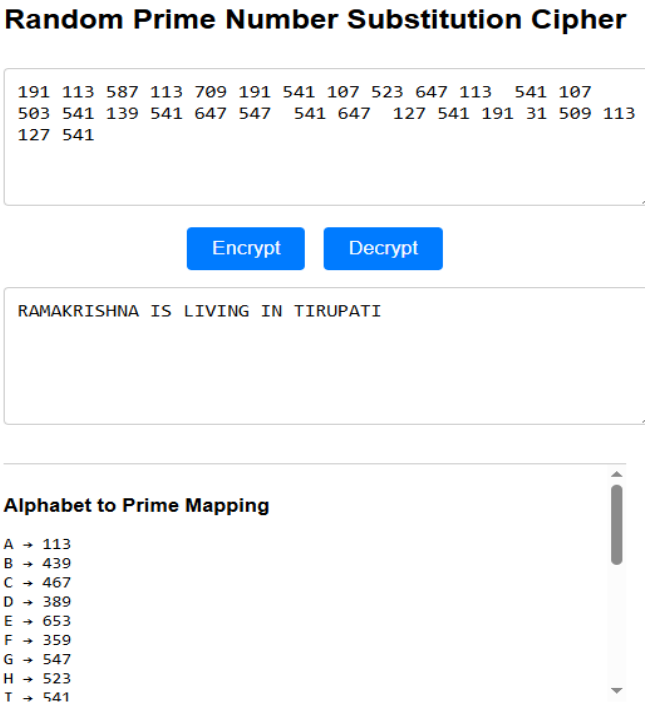

Figure 2: Decryption Process

IV. COMPARATIVE ANALYSIS

Feature	Traditional Substitution Ciphers	Random Prime Number Substitution Cipher
Key Type	Static	Dynamic
Resistance to Frequency Analysis	Low	High
Educational Utility	Moderate	High
Practical Security	Low	Moderate

The RPN-SC offers enhanced security through dynamic key generation and resistance to frequency analysis, making it a valuable tool for educational purposes and experimental cryptographic applications.

V. ADVANTAGES AND DISADVANTAGES

Advantages:

- Enhanced Security: The use of randomized prime number mapping increases resistance to cryptanalytic attacks.
- Educational Value: Serves as an effective teaching tool for understanding cryptographic principles.
- Simplicity: The algorithm's straightforward implementation makes it accessible for educational and experimental purposes.

Disadvantages:

- Limited Practical Security: While secure against basic attacks, it may not withstand advanced cryptanalytic techniques.
- Scalability Issues: The need for a large list of prime numbers can lead to increased computational complexity for larger datasets.

Overall Time Complexity: Considering both components, the overall time complexity of the cipher can be expressed as:

O(nlog⁡log⁡n+m)O(n \log \log n + m)
Where:
n is the upper limit for prime number generation,
m is the length of the input text.

VI CONCLUSION

The Random Prime Number Substitution Cipher presents an innovative approach to encryption by combining the mathematical properties of prime numbers with randomization techniques. While primarily educational, its principles can inform the development of more secure cryptographic systems. Continued research and development are essential to address its limitations and explore its full potential in modern cryptography.

REFERENCES

[1] Jain, A., Dedhia, R., & Patil, A. (2015). Enhancing the security of Caesar cipher substitution method using a randomized approach for more secure communication. arXiv preprint arXiv:1512.05483.

[2] Naidu, K.P., Rao, V.L., & Kumar, A.U. (2014). Character-Based Symmetric Key Algorithm using Randomized Prime Numbers. International Journal of Engineering Research & Technology (IJERT), 3(2).

[3] Dey, S., Abraham, A., & Sanyal, S. (2010). An LSB data hiding technique using prime numbers. arXiv preprint arXiv:1003.5509.

[4] Diffie, W., & Hellman, M. (1976). New directions in cryptography. IEEE Transactions on Information Theory, 22(6), 644-654.

[5] Rivest, R.L., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21(2), 120-126.

[6] Koblitz, N., & Miller, V. (1985). Elliptic curve cryptosystems. Proceedings of the International Conference on Advances in Cryptology, 262-273.

[7] Cocks, C. (1997). An identity-based encryption scheme based on quadratic residues. Proceedings of the IMA International Conference on Cryptography and Coding, 360-363.