International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 04 Issue: 06 | June - 2020 SJIF Rating: 6.049 ISSN: 2582-3930

Enhancing Data Security: A Web-Based Implementation of the Vernam Cipher with Random Numeric Key Generation

Dr. G. Sreedhar

Department of Computer Science, Rashtriya Sanskrit Vidyapeetha, Tirupati

Abstract -This paper presents the design and implementation of a Vernam cipher utilizing a randomly generated numeric key for encryption and decryption processes. The Vernam cipher, a symmetric-key stream cipher, is known for its simplicity and theoretical unbreakability when the key is truly random, as long as the key is as long as the message and used only once. This study explores the practical application of this cipher using modern web technologies, providing an accessible tool for secure communication.

***Key Words*:** Vernam Cipher, Random Number, Encryption, Decryption

1.INTRODUCTION

Cryptography is the cornerstone of secure communication in the digital age. The Vernam cipher, introduced in 1917 by Gilbert Vernam, is a symmetric-key cipher that operates on the principle of bitwise XOR between the plaintext and a key of the same length. When the key is truly random and used only once, the cipher is theoretically unbreakable, a property known as perfect secrecy. In this article, the Vernam Cipher is implemented through a web-based application using HTML, CSS, and JavaScript. The key is generated randomly for each encryption session, ensuring a unique and secure encryption process.

2. BACK GROUND AND RELATED WORK

The concept of perfect secrecy was formalized by Claude Shannon in his 1949 paper "Communication Theory of Secrecy Systems," where he proved that the one-time pad, a variant of the Vernam cipher, is the only cipher with provable security under certain conditions. Modern cryptographic systems, such as AES and RSA, have evolved from these foundational principles. However, the Vernam cipher remains relevant due to its simplicity and the increasing interest in lightweight cryptographic solutions for web applications.

3. METHODOLOGY

3.1 Key Generation

A cryptographically secure random numeric key is generated using the window.crypto.getRandomValues() method in JavaScript. This method provides a strong random number generator suitable for cryptographic purposes.

3.2 Encryption Algorithm

The encryption process involves the following steps:

1. **Input**: Accept the plaintext and key as inputs.
2. **Validation**: Ensure the length of the key equals the length of the plaintext.
3. **XOR Operation**: For each character in the plaintext, perform a bitwise XOR operation with the corresponding character in the key.
4. **Output**: The result of the XOR operation is the ciphertext.

```
Function Encrypt(plaintext, key):
  Ensure length of key equals length of  plaintext
  Initialize empty string ciphertext
  For i = 0 to length of plaintext - 1:
    plaintext_char = plaintext[i]
    key_char = key[i]
    ciphertext_char = XOR(plaintext_char, key_char)
    Append ciphertext_char to ciphertext
Return ciphertext
```

3.3 Decryption Algorithm

The decryption process involves the following steps:

1. **Input**: Accept the ciphertext and key as inputs.
2. **Validation**: Ensure the length of the key equals the length of the ciphertext.
3. **XOR Operation**: For each character in the ciphertext, perform a bitwise XOR operation with the corresponding character in the key.
4. **Output**: The result of the XOR operation is the original plaintext.

Function Decrypt(ciphertext, key):
Ensure length of key equals length of
Initialize empty string plaintext
For i = 0 to length of ciphertext - 1:
ciphertext_char = ciphertext[i]
key_char = key[i]
plaintext_char = XOR(ciphertext_char, key_char)
Append plaintext_char to plaintext
Return plaintext

3.4 IMPLEMENTATION

The application is implemented as a single HTML file containing embedded CSS for styling and JavaScript for functionality. The user interface consists of two sections: one for encryption and another for decryption. The encryption section allows the user to input plaintext and receive the ciphertext along with the generated key. The decryption section enables the user to input ciphertext and the corresponding key to retrieve the original plaintext.

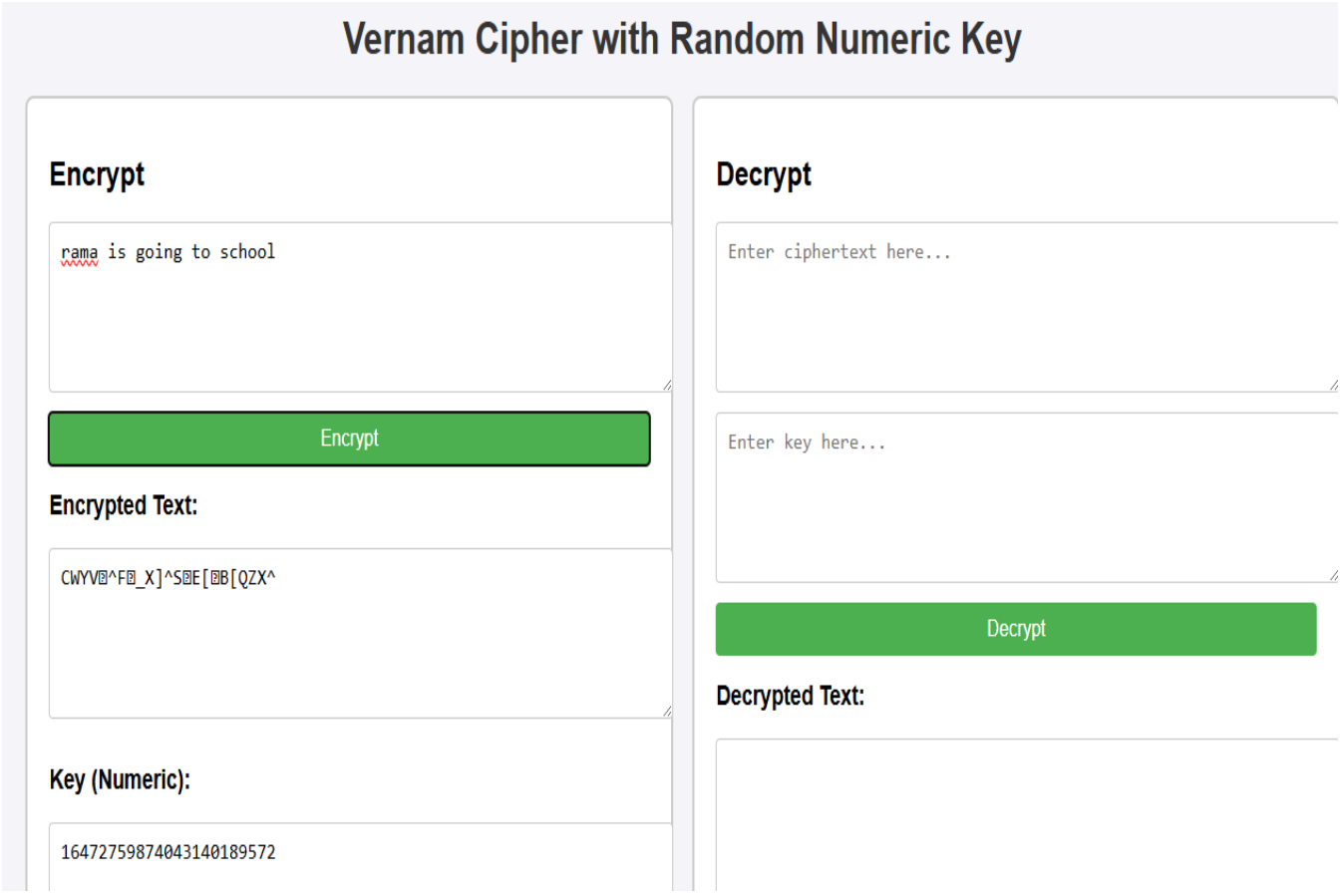

Figure 1: Encryption with random numeric key

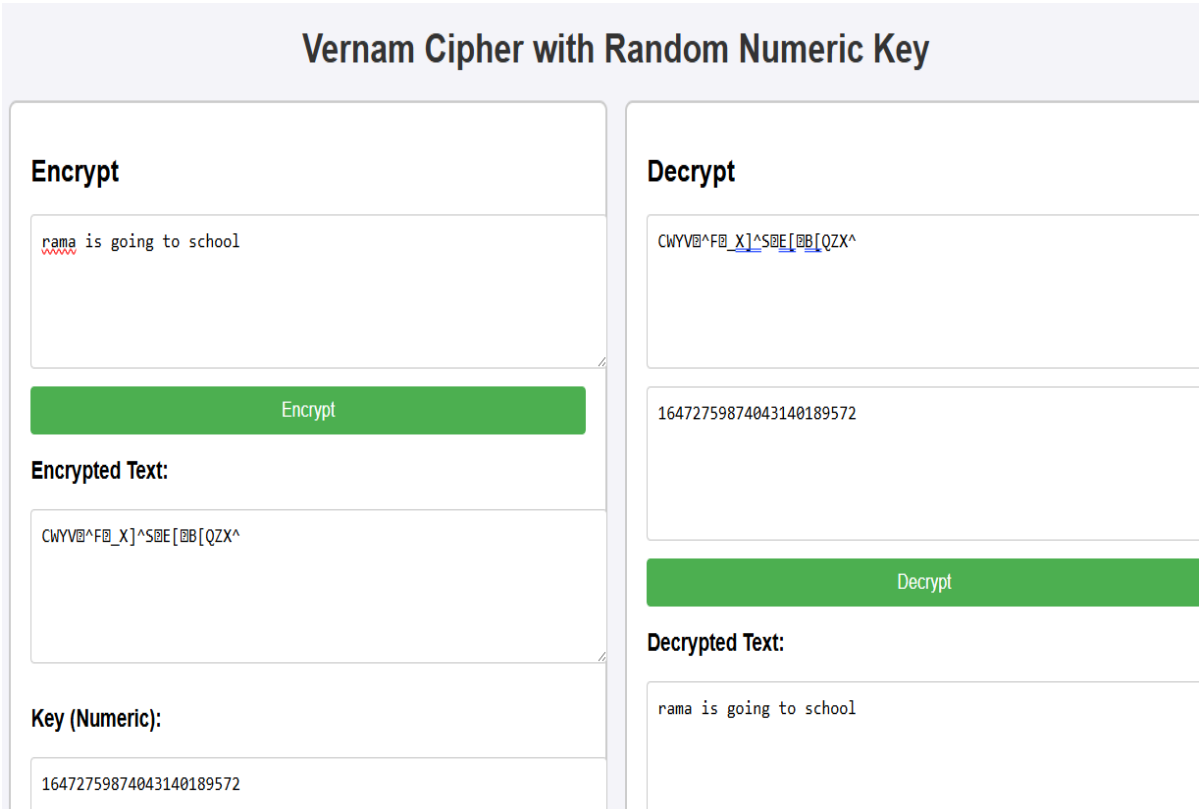

Figure 2: Decryption with random numeric key

4. RESULTS AND DISCUSSION

The application was tested with various plaintext inputs, and the encryption and decryption processes functioned as expected. The randomly generated keys ensured that each encryption session was unique, providing a high level of security. While the Vernam cipher offers perfect secrecy under ideal conditions, practical challenges such as secure key distribution and key management remain. These issues are addressed in modern cryptographic systems through the use of asymmetric key algorithms and secure key exchange protocols.

5. CONCLUSION

This project demonstrates the practical application of the Vernam cipher using modern web technologies. By generating a random numeric key for each encryption session, the application provides a simple yet secure method for encrypting and decrypting messages. Future work could explore integrating this cipher into messaging platforms or developing mobile applications to further enhance secure communication.

REFERENCES

[1]. Shannon, C. E. (1949). "Communication Theory of Secrecy Systems." *Bell System Technical Journal*.
[2]. Ryabko, B. (2013). "The Vernam cipher is robust to small deviations from randomness." *arXiv*. Retrieved from https://arxiv.org/abs/1303.2219
[3]. Leung, D. W. (2000). "Quantum Vernam Cipher." *arXiv*. Retrieved from https://arxiv.org/abs/quant-ph/0012077
[4]. Mirsky, Y., Fedidat, B., & Haddad, Y. (2019). "Physical Layer Encryption using a Vernam Cipher." *arXiv*. Retrieved from https://arxiv.org/abs/1910.08262
[5]. Leung, D. W. (2000). *Quantum Vernam Cipher*. [arXiv:quant-ph/0012077].
[6]. Lee, J. S., & Cleaver, G. B. (2015). *The Cosmic Microwave Background Radiation Power Spectrum as a Random Bit Generator for Symmetric and Asymmetric-Key Cryptography*. [arXiv:1511.02511].
[7]. Mirsky, Y., Fedidat, B., & Haddad, Y. (2019). *Physical Layer Encryption using a Vernam Cipher*. [arXiv:1910.08262].
[8]. Wen, K., Deng, F.-G., & Long, G. L. (2007). *Reusable Vernam Cipher with Quantum Media*. [arXiv:0711.1632].
[9]. Foidl, G. M. (2008). *Vernam encryption/decryption of files*. CodeProject.