

Encryption and Decryption Using ASCII Values

A Detailed Report on Encryption and Decryption Techniques

Prepared by:

Lakhan Sharma

Ayush Tiwari

Under the Guidance of:

Dr. Shivang Tripathi

*A Project Report Submitted in Fulfillment of the
Requirements for Learning Encryption Techniques*

Abstract

Encryption and decryption are essential processes in securing data, ensuring that information remains protected against unauthorized access. This report highlights a Python-based implementation of encryption and decryption using ASCII values, a simple yet insightful method to understand the basics of cryptographic principles.

In this project, encryption converts plain text into encoded data by manipulating the ASCII values of characters, appending a user-defined key for additional security. Decryption reverses the process, validating the key and reconstructing the original message. While this approach demonstrates the core mechanics of encoding and decoding, it is not suitable for safeguarding sensitive information due to its simplicity and susceptibility to brute-force attacks.

The report outlines the underlying logic of the Python scripts, explaining the steps for both encryption and decryption processes. The implementation serves as a foundation for students and beginners to explore fundamental encryption techniques. Furthermore, the report discusses potential applications and limitations, emphasizing the importance of advanced algorithms like AES or RSA for real-world security needs.

This work illustrates the transformation of text using numerical representations, bridging the gap between theoretical cryptography and practical implementation.

Contents

1	Introduction	3
2	Understanding ASCII	3
3	Encryption Process	3
4	Decryption Process	3
5	Applications and Limitations	4
6	Conclusion	4

1 Introduction

Encryption is the process of converting plain text into an encoded format to protect it from unauthorized access. Decryption, on the other hand, is the reverse process of converting the encoded text back to its original form. ASCII (American Standard Code for Information Interchange) provides a numeric encoding standard that can be leveraged for text manipulation. This report examines a project where encryption and decryption are implemented by performing mathematical operations on ASCII values.

2 Understanding ASCII

ASCII assigns a numeric value to each character, enabling systematic encoding and decoding. For example:

- The ASCII value of 'A' is 65.
- The ASCII value of 'a' is 97.
- Numbers and symbols are also represented, such as '1' with a value of 49 and '@' with a value of 64.

By leveraging these numerical representations, characters can be transformed during encryption and recovered during decryption.

3 Encryption Process

The encryption process involves:

1. Reading the content of the input text file.
2. Converting each character into its ASCII value using the `ord()` function in Python.
3. Appending a delimiter (e.g., '—') between ASCII values for separation.
4. Adding a user-specified key to the encoded text for authentication during decryption.
5. Writing the resulting encrypted data back to the file.

This is achieved by adding the key to the ASCII value of each character and converting the result back to a character.

4 Decryption Process

The decryption process involves:

1. Reading the encrypted file.
2. Splitting the encoded string into individual ASCII values using the delimiter.

3. Removing the appended key after verifying it with the user-provided key.
4. Reconstructing the original message by converting ASCII values back to characters using the `chr()` function.
5. Writing the decrypted data back to the file.

The key is subtracted from the ASCII value of each character to recover the original text.

5 Applications and Limitations

Applications

- Educational demonstrations of encryption and decryption principles.
- Encoding non-sensitive text files for basic obfuscation.

Limitations

- Vulnerable to brute-force attacks due to the simplicity of the method.
- Lacks the complexity and security features of modern encryption algorithms like AES or RSA.
- The approach is not suitable for securing sensitive or confidential data.

6 Conclusion

The project demonstrates a simple yet effective way to understand encryption and decryption concepts using ASCII values. While the method is educational, it underscores the importance of using robust cryptographic algorithms for real-world applications where security is paramount.

References

- Python documentation on `ord()` and `chr()` functions.
- ASCII reference table: <https://www.asciitable.com/>