LONDON METROPOLITAN UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CC5009NI Cyber Security in Computing

Assessment Weightage & Type

40% Individual Coursework 01

Year and Semester

2024 -25 Autumn Semester

Student Name: Bibhuti Sigdel

London Met ID: 23047458

College ID: np01nt4a230128

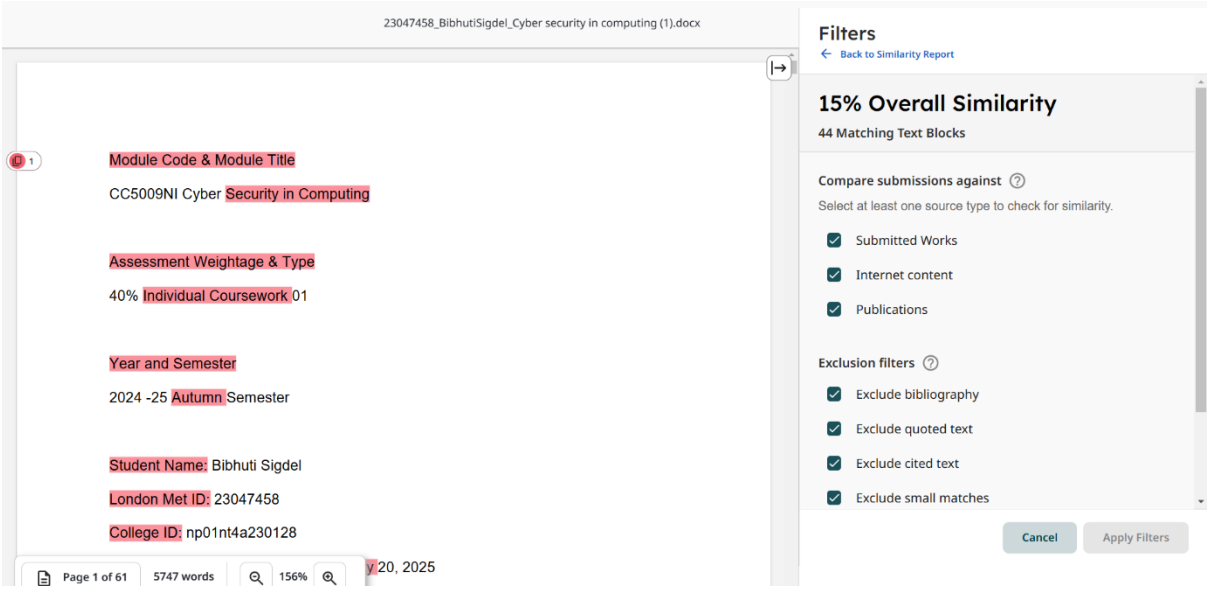
Assignment Due Date: Monday, January 20, 2025

Assignment Submission Date: Monday, January 20, 2025

Word Count (Where required):4728

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Turnitin Similarity Report



Acknowledgement

I am truly thankful to everyone who supported and guided me in the preparation of this report. First and foremost, I owe my deepest gratitude to my instructors and mentors for their invaluable guidance and insights. Their expertise was instrumental in helping me analyse the challenges and potential solutions surrounding third-party risk management, laying a strong foundation for this report.

I am also sincerely grateful to my peers and colleagues for their constructive feedback and unwavering encouragement. Their suggestions and support inspired me to refine my ideas and enhance the overall quality of my work, greatly contributing to the outcome.

Lastly, I would like to extend my appreciation to the organizations and researchers whose case studies, findings, and resources provided valuable input for this report. Their work significantly broadened my perspective and deepened my understanding of the complexities of vendor and supplier security.

Table of Contents

Table of Figures

Table of Tables

Abstract

Cryptography plays a crucial role in safeguarding data by ensuring its confidentiality, integrity, and availability in today's digital world. This report delves into the basics of cryptography, tracing its historical evolution and highlighting its importance in securing communications. Although the Caesar Cipher is one of the earliest and simplest encryption methods, its predictability makes it highly vulnerable to attacks.

To address these vulnerabilities, this report introduces an enhanced cryptographic algorithm that builds on the Caesar Cipher by adding extra layers of complexity. By incorporating cyclic shifts and XOR operations, the new algorithm strengthens security, offering better resistance against brute-force and frequency analysis attacks.

The development of this advanced algorithm is thoroughly explained, along with its advantages over the classic Caesar Cipher. These improvements include a more intricate encryption process, greater disruption of frequency patterns, and enhanced protection for sensitive data. This work contributes to the field of cryptography, emphasizing the need for continuous innovation in encryption techniques to meet the evolving challenges of modern cybersecurity.

1. Introduction

Cryptography is a technique of securing information and communications using codes so that only those persons for whom the information is intended can understand and process it. Thus, preventing unauthorized access to information. The prefix “crypt” means “hidden” and the suffix “graphy” means “writing”. In Cryptography, the techniques that are used to protect information are obtained from mathematical concepts and a set of rule-based calculations known as algorithms to convert messages in ways that make it hard to decode them. These algorithms are used for cryptographic key generation, digital signing, and verification to protect data privacy, web browsing on the internet and to protect confidential transactions such as credit card and debit card transactions. (Geeksforgeeks, 2024)

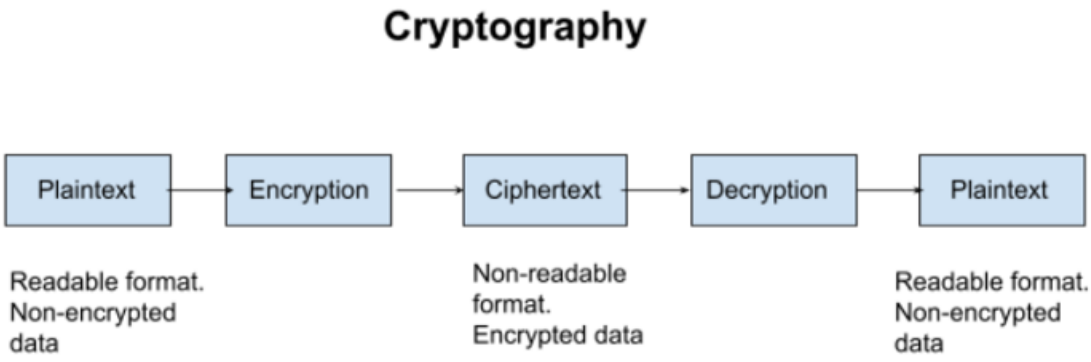


Figure 1: Process of cryptography

Key terminologies of to understand cryptography:

- **Encryption**: The process of converting plaintext into ciphertext using an algorithm and a key, making the data unreadable to unauthorized users.
- **Decryption:** The opposite of encryption, in which the right key is used to transform ciphertext back into its original plaintext.
- **Plaintext:** Data or a message in its original, readable form prior to encryption is known as plaintext.
- **Ciphertext:** The incomprehensible encrypted form of plaintext following the use of cryptographic techniques.

- **Key:** A key is an amount of data used in encryption and decryption procedures, typically a string of characters. It may comprise of a public key and a private key, or it may be symmetric (shared key).

1.1. Aims

To explore, design, create, and evaluate a new cryptographic system that tackles the challenges of ensuring data confidentiality and integrity.

1.2. Objectives

- Design a cryptographic system to prevent unauthorized access and ensure data confidentiality.
- Develop mechanisms to protect data from tampering and maintain its integrity.
- Evaluate the system's effectiveness in addressing confidentiality and integrity challenges.
- Create a secure solution for reliable data storage and transmission.

1.3. Fundamental of Security

- **Security**

Security is the act of safeguarding assets, systems, and information against damage, theft, or unauthorized entry. It includes things like physical security, cybersecurity and information security to make sure we are safe, our privacy is being respected and that we can trust the people we are dealing with. In my experience, all aspects of security need to ensure steer their work to protect processes, prevent intrusions, and preserve the integrity of individual, organizational, and societal processes. (Bacon, 2024)

- **CIA Triad**

  Information security is built on three key principles: **Confidentiality, Integrity, and Availability**, often called the **CIA Triad**. Together, they provide a solid framework for protecting data and systems from various threats. Each principle has its own unique role, but they all work together to ensure that operations are secure, reliable, and able to withstand challenges.

Figure 2: CIA Triad

- **Confidentiality:** Maintaining confidentiality involves protecting private information from unauthorized access. By controlling access to specific data to those who are authorized, this approach protects critical company information and individual privacy. Since data breaches can result in serious financial, legal, and reputational harm, confidentiality measures are especially crucial in industries like healthcare, finance, and government. Confidentiality can be maintained by encrypting sensitive data, restricting access from unauthorized user, using multi-factor authentication. **For example,** to prevent unwanted access, a customer's login and password are encrypted, when they log into their online banking account. Additionally, to make sure that only the authentic account holder has access to their data, banks use multi-factor authentication (MFA), which requires the usage of a second verification technique, such as a fingerprint scan or one-time code. (Hashemi-Pour, 2024)

Figure 3: Confidentiality

- **Integrity:** Integrity guarantees that data is reliable, accurate, and consistent for the duration of its existence. To make sure that data hasn't been deliberately or accidentally manipulated or corrupted, integrity must be maintained. System failures, poor decision-making, and financial loss can result from inaccurate or compromised data. **For Example,** Banks use hashing algorithms during online transactions to make that transaction data, such as a Rs 1 lakh transfer, doesn't change during transmission and hasn't been changed by mistakes or hackers. Additionally, digital signatures are employed to confirm the accuracy and reliability of important interaction between the consumer and the bank. (Hashemi-Pour, 2024)

Figure 4: Integrity

- **Availability:** Availability ensures that, when needed, authorized users can access data, systems, and applications. This principle deals with the necessity of continuous operations in businesses, where lost productivity, income, and customer trust can result from important system failures or inaccessibility. In sectors where real-time information access is critical, such as banking, healthcare, and e-commerce, availability is especially crucial. **For Example**, to ensure the high availability of online banking services, banks use load balancing, cloud-based design, and redundant servers. To minimize disruptions, the system automatically switches to a backup server if one fails. They also employ regular maintenance and 24/7 monitoring to guarantee continuous availability, in addition to DDoS mitigation techniques to guard against attacks that can result in outages. (Hashemi-Pour, 2024)

Figure 5: Availability

- **Important of CIA Triad:**

When combined, the CIA Triad provides a balanced and practical approach to information security by focusing on privacy, accuracy, and accessibility. It helps organizations operate smoothly and securely in today's fast-paced digital world, where threats are constantly evolving. By protecting sensitive data, ensuring its accuracy, and keeping systems accessible when needed, businesses can build trust with customers, safeguard their reputation, and stay compliant with regulations. At the same time, the CIA Triad helps organizations stay ahead of cyber threats, system failures, and human mistakes, creating a strong foundation for secure and reliable operations in an increasingly complex environment.

1.4. History of Cryptograph

Ancient Cryptography

Cryptography began as early as **1900 BC**, when Egyptians used unusual hieroglyphs in tombs to hide information. Around **1500 BC**, people in Mesopotamia encoded recipes for ceramic glazes on clay tablets to keep them secret. In **650 BC**, the Spartans created the *Scytale cipher*, where a leather strip was wrapped around a wooden staff to form a readable message; the correct staff size acted as the key. Later, in **100-44 BC**, Julius Caesar used the *Caesar Cipher*, which shifted letters by a set amount to scramble messages, making it an early example of coded communication (Sidhpurwala, Red Hat, 2023).

Medieval Cryptography

The medieval era saw big steps forward in cryptography. In **800 AD**, Arab mathematician **Al-Kindi** invented *frequency analysis*, a way to break ciphers by studying patterns in how often letters or words appear. This technique was especially useful against simpler ciphers. By **1466**, Leon Battista Alberti improved encryption by using multiple alphabets, making codes harder to break. Around the 1500s, the *Vigenère Cipher* was introduced, a more advanced method that used several alphabets to make messages more secure. (IBM, 2024)

Modern Cryptography

Modern cryptography advanced quickly, especially during wartime. In **WWI (1913)**, cryptography became essential for military communications. In **1917**, **Edward Hebern** invented a machine that used rotors to automatically encode messages, and in **1918**, **Arthur Scherbius** improved on this with the *Enigma Machine*, heavily used by Germany in WWII. However, Allied codebreakers, including **Alan Turing**, cracked the Enigma during **1939-45**, which was a huge turning point in the war. In the 1970s, cryptography shifted toward securing digital communications. IBM introduced the *Data Encryption Standard (DES)* in **1975**, the first encryption system approved for government use. Then in **1976**, the *Diffie-Hellman key exchange* introduced a way to securely share keys without needing a private key. A year later, the RSA algorithm made it possible to encrypt data using very large numbers, and it's still widely used today. Finally, in **2001**, the *Advanced Encryption Standard (AES)* replaced DES with stronger security, becoming the global standard for protecting data. (IBM, 2024)

Types of Cryptography

- **Symmetric Key Cryptography:** Symmetric Key Cryptography is a straightforward encryption system where both the sender and the receiver share a single key to lock and unlock messages. It's fast and simple, making it a popular choice, but the tricky part is securely sharing the key between parties. For instance, if you encrypt an email with a unique key and send it to your friend Tom, he'll need that same key to decrypt and read it. Well-known examples of this type of encryption include the Data Encryption Standard (DES) and the Advanced Encryption Standard (AES). (Geeksforgeeks, 2024)

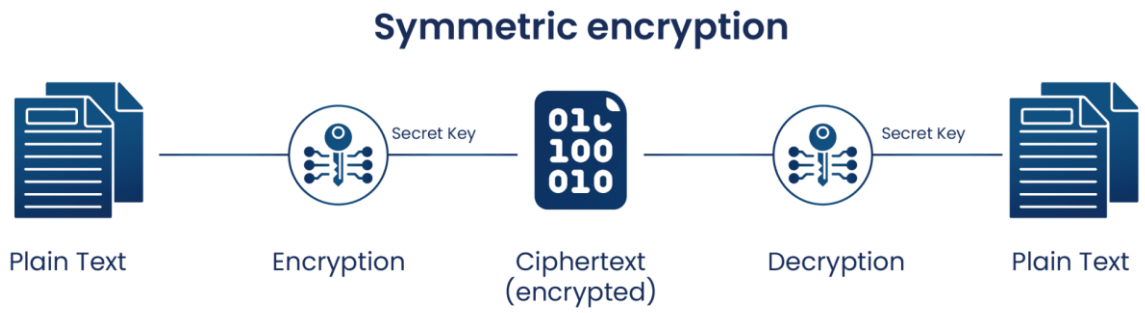


Figure 6: Symmetric Encryption

- **Asymmetric Key Cryptography:** Asymmetric Key Cryptography was designed to solve the key-sharing problem in symmetric encryption. Instead of using a single key for both encrypting and decrypting data, it uses a pair of keys: a public key and a private key. The sender encrypts the message with the receiver's public key, and only the receiver can decrypt it using their private key. Since the public key is meant to be shared openly, there's no risk to the private key, which remains secure and known only to the receiver. This makes communication much safer and more practical. One of the most well-known algorithms for this type of encryption is the RSA algorithm. (p., 2021)

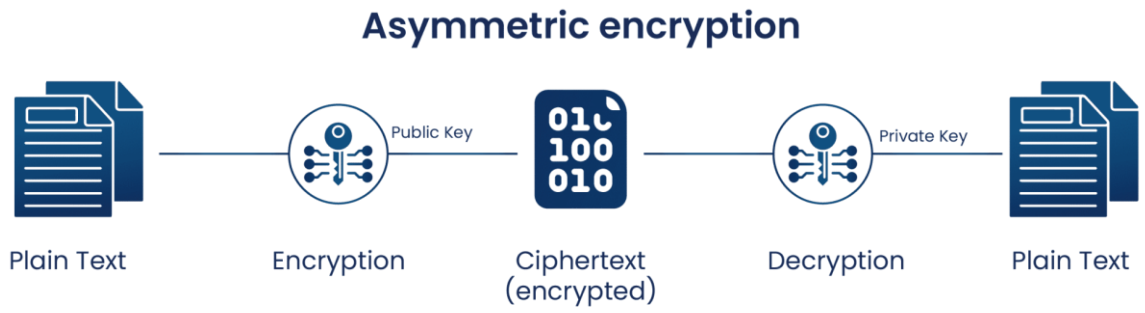


Figure 7: Asymmetric Encryption

2. Background

Around 100 BC, Julius Caesar used a simple form of encryption to send secret messages to his generals on the battlefield. This method, now known as the **Caesar cipher**, is one of the earliest examples of cryptography. The Caesar cipher is a type of **substitution cipher**, where each letter of the original message (the plaintext) is swapped with another letter to create an encrypted message (the ciphertext).

In Caesar's case, the cipher involved shifting each letter of the alphabet by three places. For example, the letter **A** would be replaced by **D**, **B** by **E**, and so on. If the shift went beyond **Z**, it would wrap around to the start of the alphabet—so **X** becomes **A**, **Y** becomes **B**, and **Z** becomes **C**. This simple method helped Caesar send messages that were difficult for enemies to understand, if the enemy didn't know the shift value.

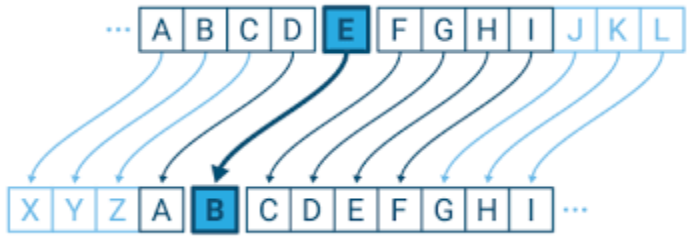


Figure 8: Caesar Cipher

However, the Caesar cipher was far from perfect. Its security relied more on keeping the **shift value** secret rather than having a complex encryption method. Once the shift was known, decrypting the message was easy. Plus, the cipher could be cracked using **frequency analysis**, a technique that takes advantage of the fact that certain letters appear more often than others in a language. For example, in English, **E** is the most common letter. So, by looking at the frequency of letters in the encrypted message, an attacker could figure out the shift and decode the message.

Despite its simplicity and weaknesses, the Caesar cipher was an important step in the evolution of cryptography. It showed the importance of keeping communication secure and sparked the development of more complex and secure encryption methods in the future. (Sidhpurwala, 2023)

2.1. Pros and cons of Ceaser Cipher

Pros

- The Caesar cipher is easy to understand and implement, making it ideal for beginners learning about encryption.
- It can be physically applied using tools like rotating disks or a scytale (a simple tool made of cards), which can be useful in certain scenarios.
- It requires only a small amount of shared information between the sender and receiver, making it straightforward to use.
- It can be modified for greater security by using multiple shift values or adding keywords to the encryption process.

Cons

- The Caesar cipher is highly insecure by today's standards and can be easily broken using modern decryption methods.
- It's vulnerable to known-plaintext attacks, where an attacker can break the encryption if they have access to both the encrypted message and its original version.
- With a limited number of possible keys, attackers can quickly try all options (brute force) to decode the message.
- It's not ideal for encrypting long messages, as its simplicity makes it easy to crack.
- Its lack of complexity makes it unsuitable for secure communication in any serious context.
- The cipher does not provide essential security measures like confidentiality, integrity, or authenticity for messages.

3. Development

3.1. Encryption Algorithm

Step 1: Input a plaintext.

Step 2: Apply a Caesar Cipher and if the alphabet exceeds Z, restart at A and shift it by shifting each letter forward by 5 positions.

Step 3: Now shift each letter to the left by 2 positions in the alphabet and take the encrypted text.

Step 4: Execute an XOR operation between the binary values of the encrypted letters and the binary value (11001).

Step 5: Convert the resulting XOR binary values back into letters using the reverse conversion table.

Step 6: Combine all the letters to form the final encrypted message.

Procedure:

- **Plaintext:** “security”
- **Apply Caesar Cipher:**

Taking this table as reference for encryption:

A	B	C	D	E	F	G	H	I	J
1	2	3	4	5	6	7	8	9	10
K	L	M	N	O	P	Q	R	S	T
11	12	13	14	15	16	17	18	19	20
U	V	W	X	Y	Z				
21	22	23	24	25	26				

Table 1: Index of alphabet for Caeser Cipher

Shift each letter forward by 5 positions

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E

Table 2: Letter after shifting by 5 positions

Values after shifting by 5 positions:

S – X, E – J , C – H , U – Z , R – W , I – N ,T – Y , Y – D

Result after Caesar cipher: **"XJHZWNYD"**

- Changing the encrypted text (XJHZWNYD) by left cyclic shift value 2.

X	J	H	Z	W	N	Y	D
H	Z	W	N	Y	D	X	J

Table 3: Left cyclic shift

Result after Left Shifting: **"HZWNYDXJ"**

- Now use XOR to make encryption stronger.

  Table to show binary value for encrypted text.

Encrypted text	**Binary Value**
H	00111
Z	11001
W	10110
N	01101
Y	11000
D	00011
X	10111
J	01001

Table 4: Converting encrypted text in binary

Table for XOR

Binary value of Encrypted text	**XOR**	**XOR Result**
00111	11001	11110
11001	11001	00000
10110	11001	01111
01101	11001	10100
11000	11001	00001
00011	11001	11010
10111	11001	01110
01001	11001	10000

Table 5: XOR with key 11001

This table is reference for changing XOR binary value in encrypted text.

Alphabet	**Binary Value**
A	00000
B	00001
C	00010
D	00011
E	00100
F	00101
G	00110
H	00111
I	01000
J	01001
K	01010
L	01011
M	01100
N	01101
O	01110
P	01111
Q	10000
R	10001
S	10010
T	10011
U	10100
V	10101
W	10110
X	10111
Y	11000
Z	11001
Space	11010
!	11011

.	11100
?	11101
*	11110
£	11111

Table 6: Binary table with modification

Binary Value	**Text formed after binary value**
11110	*
00000	A
01111	P
10100	U
00001	B
11010	Space
01110	O
10000	Q

Table 7: Final value after XOR

Result: *APUB OQ

3.2. Decryption Algorithm.

Step 1: Input the encrypted message.

Step 2: Convert each letter into its binary value using the reference table.

Step 3: Reverse the XOR operation by performing XOR between the binary values and **11001**.

Step 4: Use the reverse conversion table to convert the binary values back into their corresponding letters in the alphabet.

Step 5: Reverse the Left cyclic shift by moving each letter 2 positions to the right in the alphabet.

Step 6: Reverse the Caesar Cipher by shifting each letter 5 positions backward, restarting at Z if needed, to retrieve the original plaintext.

Procedure:

- **Cyphertext:** "*APUB OQ "
- Convert each letter into its binary value using the reference table.

Text formed after binary value	Binary Value
*	11110
A	00000
P	01111
U	10100
B	00001
Space	11010
O	01110
Q	10000

Table 8: Converting encrypted letter into binary value

- To reverse the XOR operation by performing XOR between the binary values and **11001**.

Binary value of Decrypted text	XOR	XOR Result
11110	11001	00111
00000	11001	11001
01111	11001	10110
10100	11001	01101
00001	11001	11000
11010	11001	00011
01110	11001	10111
01001	11001	01001

Table 9: Converting binary value with XOR key 11001

- Use the reverse conversion table to convert the binary values back into their corresponding letters in the alphabet.

Binary Value	Decrypted text
00111	H
11001	Z
10110	W
01101	N
11000	Y
00011	D
10111	X
01001	J

Table 10: Convert binary value into corresponding alphabets

- Reverse the Left cyclic shift by moving each letter 2 positions to the right in the alphabet.

H	Z	W	N	Y	D	X	J
X	J	H	Z	W	N	Y	D

Table 11: Reverse left cyclic shift

- Reverse the Caesar Cipher by shifting each letter 5 positions backward, restarting at Z if needed, to retrieve the original plaintext.
  X (24-5 = 19) – S, J (10-5 = 5) – E , H (8-5 = 3) – C , Z (26-5 = 21) – U, W (23-5 = 18) – R , N (14-5 = 9) – I , Y (25-5 = 20) – T , D(4-5 = -1 i.e. we need to repeat from Z) – Y
  Final Result: **“SECURITY”**

3.3. Improvement of Caeser Cipher according to new algorithm

- **More Layers of Security**: Adding cyclic shifts and XOR operations makes the encryption much more secure compared to a single Caesar Cipher.
- **XOR Operations**: Using XOR with binary values adds an extra layer of complexity and randomness, making it harder to decode.
- **Nonlinear Transformation**: The XOR operation introduces nonlinearity, which is tougher to crack than the simple linear shift of Caesar Cipher.
- **Double Shifts**: Shifting letters forward and then doing a cyclic left shift scrambles the text even more, making patterns harder to spot.

- **Handles More Characters**: The custom binary table allows encryption of symbols, spaces, and letters, unlike the Caesar Cipher, which only works with alphabets.
- **Better Against Frequency Analysis**: Multiple transformations disrupt letter frequencies, making it harder to guess based on common letters like ‘E’ or ‘A’.
- **Larger Key Space**: The XOR operation creates more possible keys, unlike Caesar’s limited 26 shifts, making brute-force attacks way harder.
- **Requires Multiple Keys to Decrypt**: Without knowing all the keys (XOR value, shift values), decryption is almost impossible.

4. Flowchart

4.1. Encryption Flowchart

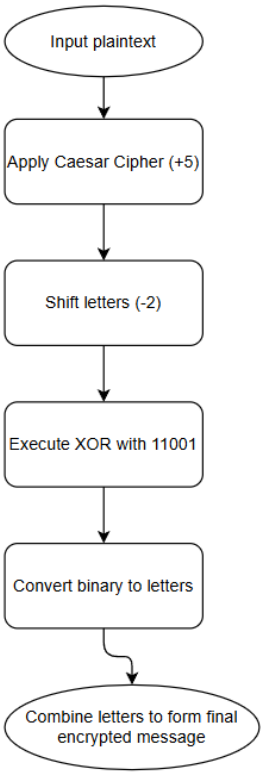


Figure 9: Encryption flowchart

4.2. Decryption Flowchart

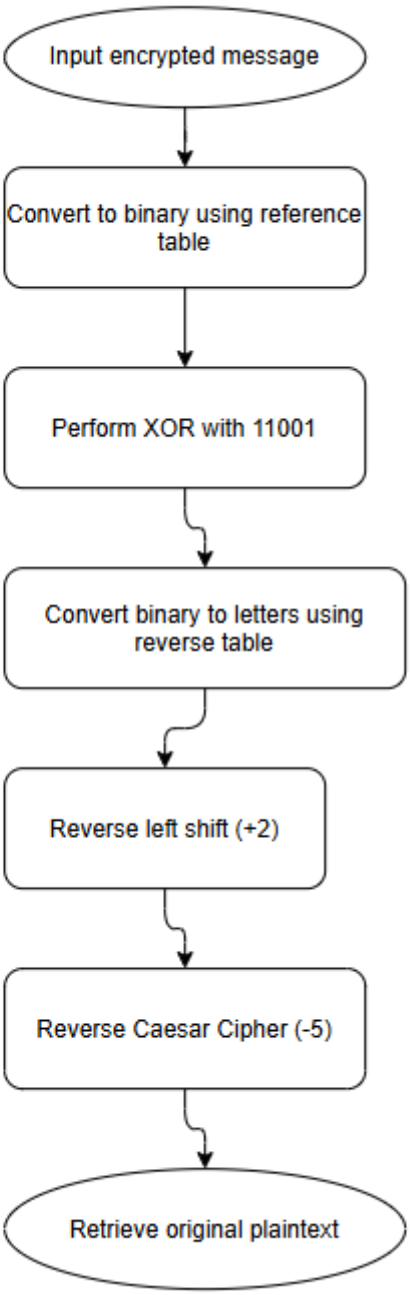


Figure 10: Decryption flowchart

5. Testing

5.1. Test 1: Encrypt and decrypt the word “THE”

Encryption:

- Input “THE” as plaintext; Apply Caesar Cipher and use shift value by 5.
  T (20+5 = 25) = Y
  H (8+5 = 13) = M
  E (5+5 = 10) = J
- Changing the encrypted text (YMJ) by left cyclic shift value 2.

Y	M	J
J	Y	M

Table 12: Left shift by 2 in word "THE"

- Applying an XOR operation between the binary values of the encrypted letters and the binary value (11001).

Encrypted Text	**Binary value**	**XOR**	**XOR Value**
J	01001	11001	10000
Y	11000	11001	00001
M	01100	11001	10100

Table 13: Converting text into binary and doing XOR with key 11001.

- Converting the resulting XOR binary values back into letters using the above table mentioned in encryption process.

XOR Value	**Text formed after doing XOR**
10000	Q
00001	B
10100	U

Table 14: Text after XOR

Final Encrypted Text: “QBU”

Decryption:

- Convert text which is formed after XOR into its binary value using the reference table.

Encrypted Text	Binary value
Q	10000
B	00001
U	10100

Table 15: Reversing encrypted text into binary

- Reverse the XOR operation by performing XOR between the binary values and **11001**.

Binary value	XOR	Decrypted binary value	Decrypted text after XOR
10000	11001	01001	J
00001	11001	11000	Y
10100	11001	01100	M

Table 16: Reverse the XOR operation by performing XOR between the binary values and 11001

- Reverse the Left cyclic shift by moving each letter 2 positions to the right in the alphabet.

J	Y	M
Y	M	J

Table 17: Reverse cyclic shift

- Reverse Caesar Cipher by shifting each letter by 5 positions.
  Y (25-5 = 20) = T
  M (13-5 = 8) = H
  J (10-5 = 5) = E

Final Decrypted Text: “THE”

5.2. Test 2: Encrypt and decrypt the word “CAT”

Encryption:

- Input “CAT” as plaintext; Apply Caesar Cipher and use shift value by 5.
  C (3+5 = 8) = H
  A (1+5 = 6) = F
  T (20+5 = 25) = Y
- Changing the encrypted text (HFY) by left cyclic shift value 2.

H	F	Y
Y	H	F

Table 18: Left cyclic shit in word by shift value 2

- Applying an XOR operation between the binary values of the encrypted letters and the binary value (11001).

Encrypted Text	**Binary value**	**XOR**	**XOR Value**
Y	11000	11001	00001
H	00111	11001	11110
F	00101	11001	11100

Table 19: Doing XOR in Test 2

- Converting the resulting XOR binary values back into letters using the above table mentioned in encryption process.

XOR Value	**Text formed after doing XOR**
00001	B
11110	*
11100	.

Table 20: Text formed after XOR in test 2

Final Encrypted Text: “B*.”

Decryption:

- Convert text which is formed after XOR into its binary value using the reference table.

Encrypted Text	**Binary value**
B	00001

*	11110
.	11100

Table 21: Reversing into binary value in test 2

- Reverse the XOR operation by performing XOR between the binary values and **11001**.

Binary value	**XOR**	**Decrypted binary value**	**Decrypted text after XOR**
00001	11001	11000	Y
11110	11001	00111	H
11100	11001	00101	F

Table 22: Reversing XOR and forming decrypted text

- Reverse the Left cyclic shift by moving each letter 2 positions to the right in the alphabet.

Y	H	F
H	F	Y

Table 23: Reversing left cyclic in test 2

- Reverse Caesar Cipher by shifting each letter by 5 positions.
  H (8-5 = 3) = C
  F (6-5 = 1) = A
  Y (25-5 = 20) = T

  Final Decrypted Text: “CAT”

5.3. Test 3: Encrypt and decrypt the word “HAS”

Encryption:

- Input “HAS” as plaintext; Apply Caesar Cipher and use shift value by 5.
  H (8+5 = 13) = M
  A (1+5 = 6) = F
  S (19+5 = 24) = X

- Changing the encrypted text (MFX) by left cyclic shift value 2.

M	F	X
X	M	F

Table 24: Left cyclic shift in Test 3

- Applying an XOR operation between the binary values of the encrypted letters and the binary value (11001).

Encrypted Text	**Binary value**	**XOR**	**XOR Value**
X	10111	11001	01110
M	01100	11001	10101
F	00101	11001	11100

Table 25: Doing Xor in Encrypted text in Test 3

- Converting the resulting XOR binary values back into letters using the above table mentioned in encryption process.

XOR Value	**Text formed after doing XOR**
01110	O
10101	V
11100	.

Table 26: Converting XOR value in text in Test 3

Final Encrypted Text: “OV.”

Decryption:

- Convert text which is formed after XOR into its binary value using the reference table.

Encrypted Text	**Binary value**
O	01110
V	10101
.	11100

Table 27: Reversing into binary in Test 3

- Reverse the XOR operation by performing XOR between the binary values and **11001**.

Binary value	**XOR**	**Decrypted binary value**	**Decrypted text after XOR**
01110	11001	10111	X
10101	11001	01100	M
11100	11001	00101	F

Table 28: Reverse the XOR operation in test 3

- Reverse the Left cyclic shift by moving each letter 2 positions to the right in the alphabet.

X	M	F
M	F	X

Table 29: Reverse cyclic shift in test 3

- Reverse Caesar Cipher by shifting each letter by 5 positions.
  M (13-5 = 8) = H
  F (6-5 = 1) = A
  X (24-5 = 19) = S

  Final Decrypted Text: "HAS"

5.4. Test 4: Encrypt and decrypt the word "BIG"

Encryption:

- Input "BIG" as plaintext; Apply Caesar Cipher and use shift value by 5.
  B (2+5 = 7) = G
  I (9+5 = 14) = N
  G (7+5 = 12) = L
- Changing the encrypted text (GNL) by left cyclic shift value 2.

G	N	L
L	G	N

Table 30: Left cyclic shift in test 4

- Applying an XOR operation between the binary values of the encrypted letters and the binary value (11001).

Encrypted Text	Binary value	XOR	XOR Value
L	01011	11001	10010
G	00110	11001	11111
N	01101	11001	10100

Table 31: Applying XOR in test 4

- Converting the resulting XOR binary values back into letters using the above table mentioned in encryption process.

XOR Value	Text formed after doing XOR
10010	S
11111	£
10100	U

Table 32: Final encrypted text in test 4

Final Encrypted Text: "S£U"

Decryption:

- Convert text which is formed after XOR into its binary value using the reference table.

Encrypted Text	Binary value
S	10010
£	11111
U	10100

Table 33: Reversing encrypted text in binary in test 4

- Reverse the XOR operation by performing XOR between the binary values and **11001**.

Binary value	XOR	Decrypted binary value	Decrypted text after XOR

10010	11001	01011	L
11111	11001	00110	G
10100	11001	01101	N

Table 34: Reversing XOR in test 4

- Reverse the Left cyclic shift by moving each letter 2 positions to the right in the alphabet.

L	G	N
G	N	L

Table 35: Reverse cyclic shift of test 4

- Reverse Caesar Cipher by shifting each letter by 5 positions.
  G (7-5 = 2) = B
  N (14-5 = 9) = I
  L (12-5 = 7) = G

  Final Decrypted Text: "BIG"

5.5. Test 5: Encrypt and decrypt the word "fun".

Encryption:

- Input "FUN" as plaintext; Apply Caesar Cipher and use shift value by 5.
  F (6+5 = 11) = K
  U (21+5 = 26) = Z
  N (14+5 = 19) = S
- Changing the encrypted text (KZS) by left cyclic shift value 2.

K	Z	S
S	K	Z

Table 36: Left cyclic shit in test 5

- Applying an XOR operation between the binary values of the encrypted letters and the binary value (11001).

Encrypted Text	**Binary value**	**XOR**	**XOR Value**
S	10010	11001	01011
K	01010	11001	10011

Z	11001	11001	00000

Table 37: Applying XOR in test 5

- Converting the resulting XOR binary values back into letters using the above table mentioned in encryption process.

XOR Value	**Text formed after doing XOR**
01011	L
10011	T
00000	A

Table 38: Final word after encryption in Test 5

Final Encrypted Text: "LTA"

Decryption:

- Convert text which is formed after XOR into its binary value using the reference table.

Encrypted Text	**Binary value**
L	01011
T	10011
A	00000

Table 39: converting in binary

- Reverse the XOR operation by performing XOR between the binary values and **11001**.

Binary value	**XOR**	**Decrypted binary value**	**Decrypted text after XOR**
01011	11001	10010	S
10011	11001	01010	K
00000	11001	11001	Z

Table 40: Reversing XOR in test 5

- Reverse the Left cyclic shift by moving each letter 2 positions to the right in the alphabet.

S	K	Z
K	Z	S

Table 41: Reverse cyclic shift in Test 5

- Reverse Caesar Cipher by shifting each letter by 5 positions.

  K (11-5 = 6) = F

  Z (26-5 = 21) = U

  S (19-5 = 14) = N

  Final Decrypted Text: “FUN”

6. Conclusion

This report outlines the journey from exploring the historical context of the Caesar Cipher to creating a more advanced cryptographic algorithm that addresses its weaknesses. While the original Caesar Cipher was groundbreaking in its time, it lacked the ability to offer strong security due to its simple linear shifts and fixed key lengths.

To tackle these limitations, the new algorithm introduces more sophisticated techniques, like cyclic shifts and XOR operations, which significantly boost the complexity and unpredictability of the encryption process. These enhancements help mitigate the main vulnerabilities of the Caesar Cipher, making the algorithm more resistant to brute-force and frequency analysis attacks.

The upgraded system provides better data confidentiality and integrity, making it suitable for educational purposes or low-security environments. However, its reliance on fixed parameters could be a drawback for high-security applications. Future work could focus on incorporating dynamic key generation, adjustable shifts, or hybrid encryption approaches to improve its performance.

Exploring the integration of modern techniques like public-key cryptography and advanced hashing could further elevate the system, making it more aligned with current cybersecurity standards. By building on the core principles of encryption, ongoing research has the potential to lead to more secure and efficient cryptographic solutions.

In summary, this project demonstrates how traditional cryptographic methods can be updated to address modern security challenges. It highlights the importance of innovation in cryptography and offers a foundation for further advancements in secure communication technologies.

7. Reference

Bacon, M. (2024, December 2). *TeachTarget*. Retrieved from TechTarget: https://www.techtarget.com/searchsecurity/definition/security

Consulting, E. (2024, December 2). *Encryption Consulting*. Retrieved from Encryption Consulting: https://www.encryptionconsulting.com/education-center/what-is-cryptography/#history

Geeksforgeeks. (2024, november 18). *Geeksforgeeks*. Retrieved from Geeksforgeeks: https://www.geeksforgeeks.org/cryptography-and-its-types/

Hashemi-Pour, C. (2024, November 30). *TechTarget*. Retrieved from TechTarget: https://www.techtarget.com/whatis/definition/Confidentiality-integrity-and-availability-CIA

IBM. (2024, January 5). *IBM*. Retrieved from IBM: https://ibm.com/think/topics/cryptography-history

p., N. (2021, june 15). *Blog*. Retrieved from Blog: https://preyproject.com/blog/types-of-encryption-symmetric-or-asymmetric-rsa-or-aes#:~:text=Asymmetric%20and%20symmetric%20encryption%20are,a%20private%20key%20for%20decryption

Sidhpurwala, H. (2023, January 12). *Red Hat*. Retrieved from Red Hat: https://www.redhat.com/en/blog/brief-history-cryptography

Sidhpurwala, H. (2023, January 12). *Red Hat*. Retrieved from Red Hat Blog: https://www.redhat.com/en/blog/brief-history-cryptography