

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**Cryptography and Network Security – CSE555**

**Non-CIE Component Report**

**5th Semester ‘A’ Section**

Submitted by

**Aryan Anchalia 1MS21CS030**

**Aneesh M Somayaji 1MS21CS016**

**Anirudh Sanal Kumar 1MS22CS017**

**Ankit Kumar Singh 1MS21CS020**

Under the guidance of

Dr. Sangeetha V

Associate Professor, Dept. of CSE

**RAMAIAH INSTITUTE OF TECHNOLOGY**

**(Autonomous Institute, Affiliated to VTU)**

**BANGALORE-560054**

[www.msrit.edu](http://www.msrit.edu/),

January 2024



# 

**CERTIFICATE**

This is to certify that the Project work carried out by **Aryan Anchalia(1MS21CS030)**, **Aneesh M Somayaji(1MS21CS016), Anirudh Sanal Kumar(1MS21CS017), Ankit Kumar Singh(1MS21CS020)** as a 20-mark component for the course Cryptography and Network Security (CSE555), V semester B.E, CSE during the academic year October 2023 - January 2024 satisfies the academic requirements for awarding the marks.

### Signature of the Faculty

#### **Content**

| **Sl.No.** | **Description** | **Page No.** |
| --- | --- | --- |
| **1** |  |  |
| **2** |  |  |
| **3** |  |  |

**CIPHERCRAFT: A RANDOMIZED ENCRYPTION/DECRYPTION ADVENTURE**

# Abstract:

Encryption of data is done in order to change the readable and comprehendible form of a plaintext message to an unreadable one, that is an “encrypted” form, by which the information concealed cannot be read by malicious and unauthorized entities. This encrypted message is often referred to as the ciphertext. At the recipient’s side, that is at the other end of the data-transmission process, the encrypted message is converted back to its original plaintext form so that the one the message was directed towards, can see the information the message holds for what it truly is. This is what decryption deals with. Cryptography refers to the study of encryption and decryption methods, and it involves creating secret codes with the help of which the enciphering and deciphering are carried out. There are various types of encryption or ciphering techniques that are employed, and this paper aims to provide the analysis of namely seven of them which are; Affine, Autokey, Keyed Transpositional, Vigenere, Keyless Transpositional, Playfair and Hill ciphers. The paper notes their computational time for various plaintext sizes and compares them with the help of graphical representations. This is done in order to understand and evaluate each one of their performances, first against smaller plaintext sizes, and noting how the trend changes as we gradually increase the plaintext sizes.

*Keywords: Affine cipher, Autokey cipher, Vigenere cipher, Keyless Transpositional cipher, Hill cipher, Playfair cipher, Keyed Transpositional cipher, Analysis, Encryption, Decryption*

**INTRODUCTION**

During the course of data transmission, it is of utmost importance to secure the plaintext message being transmitted so that external parties and entities cannot determine or comprehend the contents of what the message holds. If in case a successful attack has been carried out and the information determined, both the sender as well as the receiver of the plaintext message will be put at a degree of risk and vulnerability, if the data breached is sensitive content, which can be used against both the parties and also for more attacks to be carried out on further transmissions. This is where we make use of the art of cryptography, wherein through the implementation of certain codes pertaining to the specific type of ciphering used, attacking to decode the plaintext message becomes difficult, and the plaintext contents can only be deciphered, read and altered by the concerned party, on its delivery. This code used to encrypt and decrypt the messages is referred to as a cipher key, with its identity only known to the concerned parties.

Encryption of the plaintext is carried out at the sender’s end and the corresponding decryption, at the receiver’s end. The techniques employed to carry out the enciphering and deciphering, are broadly classified into two categories; Symmetric encryption and Asymmetric encryption. Symmetric encryption has one set of key to encrypt and decrypt data, while Asymmetric encryption employs two sets of keys, the public key to encrypt data and the private key to decrypt data. Examples of the latter include RSA and Data Signature Algorithm (DSA). Symmetric encryption, which is what has been dealt with in this paper, is divided into two types which are Substitution ciphers and Transposition ciphers. While the former replaces plaintext characters with other numerals, characters and symbols, unchanging the position of each character, the latter deals with only changing the positions of the plaintext characters without replacing any of them. Keyed and Keyless ciphers are the two forms of Transposition cipher. The first uses a key to change the character positions of the plaintext, while the second does not make use of one to do so.

Depending on the cipher that has been used, the Substitution classification is further divided into Monoalphabetic and Polyalphabetic ciphers. Monoalphabetic ciphers have a single valued alphabetic key to encrypt and decrypt messages. Additive, Multiplicative and Affine ciphers are examples of this. On the other hand, Polyalphabetic ciphers make use of a multivalued key to do so, with each plaintext character being able to be mapped to multiple characters of the ciphertext. Examples include Autokey, Playfair, Vigenere and Hill ciphers. This research aims to compare the mentioned Symmetric encryption techniques against designated sizes of plaintext messages, to deduce their execution time and hence their performance, also displaying the overall trend showcased by each of these methods.

**LITERATURE REVIEW**

[1] present a groundbreaking attack method that significantly elevates the cryptanalytic efficiency for general affine ciphers compared to existing techniques. Their approach hinges on exploiting "weak keys" within the cipher, those susceptible to attack due to specific mathematical properties of the key coefficients. The researchers meticulously delve into various scenarios based on the key space and field size, meticulously illustrating the attack's effectiveness across a broad spectrum of cipher parameters. This in-depth analysis underscores the crucial role of selecting strong keys for robust affine cipher implementations and offers invaluable insights for securing systems that rely on this encryption technique.

The authors of [2] cracked the code on a historically challenging cipher variant: the autokey. Their innovative approach departs from traditional cryptanalysis methods, instead wielding the cipher's inherent statistical properties as a weapon. By meticulously scouring the encrypted text for telltale patterns and strategically injecting known plaintext fragments (think common words), they unravel the keystream like a magician pulling a rabbit from a hat. This breakthrough attack represents a major leap forward in autokey cryptanalysis, exposing a hidden vulnerability and rendering even seemingly secure messages vulnerable to decryption.

The authors of [3] presented an efficient attack on row transposition ciphers when the key length is known. They utilize frequency analysis and pattern matching to identify repeated sequences in the ciphertext, effectively revealing the transposition pattern and recovering the plaintext. This research highlights the vulnerability of row transposition ciphers even with limited information about the key, highlighting the importance of employing additional security measures like key diversification when using such ciphers.

In 2009 Courtois and Maze brought about a groundbreaking transformation in Hill cipher cryptanalysis by leveraging the power of "Gröbner bases," a sophisticated mathematical tool derived from abstract algebra. Through adeptly tackling the key matrix equations, they successfully decrypted the cipher, thereby underscoring its susceptibility to advanced mathematical attacks. This seminal work underscores the imperative for exploring alternative encryption schemes or implementing resilient key selection strategies when employing Hill ciphers.

In the exploration of Vigenere cipher decryption, a breakthrough method emerged through the analysis of repeating letter patterns. This innovative approach by the researchers of [5] efficiently identified key lengths by scrutinizing the distances between repetitions, allowing for the subdivision of the cipher into manageable "sub-ciphers." Named after a Polish cryptanalyst, this technique became integral to Vigenere cryptanalysis and set the stage for future statistical attacks on polyalphabetic ciphers.

In the assessment of the Rail Fence cipher's security, the researchers of [6] uncovered vulnerabilities in this apparently simple transposition technique. By distributing plaintext characters across multiple "rails," the cipher, although providing basic obfuscation, is susceptible to advanced ciphertext-only attacks, particularly with longer ciphertexts. This emphasizes the importance of recognizing security implications in seemingly straightforward encryption methods within the cryptographic research landscape.

A statistical attack on the Playfair cipher was introduced by researchers of [7] who exploited letter frequencies and digraph probabilities. Through the analysis of ciphertext, they identified repetitive patterns and letter pairings, uncovering insights into the key square configuration. This approach facilitated the deduction of key information and the subsequent decryption of the message. Notably, vulnerabilities in the Playfair cipher were exposed, especially in cases involving short keywords or predictable key generation methods. The study emphasized the critical need for robust key selection and the exploration of alternative techniques to ensure secure encryption.

**METHODOLOGY AND IMPLEMENTATION**

**PLAYFAIR CIPHER:**

In this cipher the plain text is divided in blocks of size 2. If the length of the plain text is odd then append X in the end of the plaintext. For a given key a 5x5 key matrix is generated by first writing all the characters in the key text (without repetition) and then the remaining characters from A-Z that are not present in the plaintext.

From the key matrix generated the ciphertext is obtained by using the following rules:

1. If both the characters of the block are in the same row, then the next character from the same row is taken.

2. If both the characters of the block are in the same column, the the next characters from the same column is taken

3. If both the characters in the block are in different rows and columns, say first character is in (i,j) position and second character is in (m,n) then take the character present in (i,n) and (m,j) respectively.

Eg: Let the plain text be “CRYPTOGRAPHY” and the key “ENCRYPT”. The equivalent plain text in blocks and the key matrix is:

P = CR YP TO GR AP HY

For the block CR the cipher block will be RY (Rule 1) and for YP it will be ED (Rule 2) Similarly, the CipherText would be C = RY ED AM IN BT IC

**AFFINE CIPHER:**

The Affine Cipher is a type of monoalphabetic substitution cipher, a method of encryption where each letter in an alphabet is mapped to its numeric equivalent, encrypted using a simple mathematical function, and converted back to a letter. Specifically, the Affine Cipher combines modular arithmetic with a pair of keys, typically denoted as k1​ and k2​.

The general encryption formula for the Affine Cipher is:

C≡(P⋅k1​+k2​) mod 26

Where:

C is the ciphertext, P is the plaintext (numeric value of the letter), k1​ is the multiplicative key (it should be chosen such that it is coprime to the size of the alphabet, usually 26), k2​ is the additive key, The result is taken modulo 26 to stay within the bounds of the alphabet.

The decryption formula is:

P≡k1 inverse x (C−k2​)mod26

Where:

P is the plaintext (numeric value of the letter), C is the ciphertext, K1 inverse is the modular multiplicative inverse of k1 modulo 26, K2​ is the additive key, The result is taken modulo 26 to obtain the original numeric value.

Example:

Encryption: Assuming multiplicative key (k1​) is 5 and additive key (k2​) is 8:

Take the word "HELLO".

Convert each letter to a numeric value: H=7, E=4, L=11, L=11, O=14.

For each letter:

Multiply the numeric value by 5.

Add 8 to the result.

Take the result modulo 26.

The encrypted message is "PAIIM".

Decryption: Assuming multiplicative key (k1​) is 5, additive key (k2​) is 8, and the modular multiplicative inverse of 5 is 21:

Take the encrypted message "PAIIM".

For each letter:

Subtract 8.

Multiply by 21 (the modular multiplicative inverse of 5).

Take the result modulo 26.

The decrypted message is "HELLO".

**AUTOKEY CIPHER:**

Autokey cipher is a polyalphabetic substitution cipher that uses a key to encrypt or decrypt a message. In this cipher, the key is combined with the plaintext to generate the ciphertext. The key can be a word, phrase, or series of characters. The key is used to encrypt the message itself and is also used to extend the key for subsequent characters in the message.

Rules for Ciphertext Generation using Autokey:

1. Choose a key (a word or phrase).
2. Align the key with the plaintext message.
3. For Encryption, Combine the key and the plaintext character by character.
4. Use a predefined method (e.g., alphabetical addition) to determine the ciphertext character.
5. Update the key by appending the newly generated ciphertext character.
6. For Decryption, Combine the key and the ciphertext character by character.
7. Use the same method as encryption to determine the original plaintext character.
8. Update the key by appending the original plaintext character.

Example:

Key: "KEY" Plaintext: "HELLO"

Encryption:

H + K = O (since H is the 8th letter, and K is the 11th letter, 8 + 11 = 19, which corresponds to S)

E + E = U

L + Y = Y

L + O = X

O + O = U

Ciphertext: "OSYXU"

Decryption:

O + K = H

S + E = E

Y + Y = L

X + L = L

U + O = O

Decrypted Plaintext: "HELLO"

**HILL CIPHER:**

The Hill Cipher is a polyalphabetic substitution cipher based on linear algebra. It encrypts blocks of plaintext letters at a time and uses matrix multiplication as the primary operation. The key for the Hill Cipher is a square matrix, and the size of the matrix determines the block size. It was developed by Lester S. Hill in 1929.

Rules:

1. The key matrix must be invertible (have a modular multiplicative inverse) modulo the size of the alphabet.
2. The block size (number of letters in each block) is determined by the size of the key matrix.
3. The plaintext is divided into blocks, each of which is converted to a numeric vector.
4. The key matrix is applied to each vector through matrix multiplication.
5. The resulting ciphertext vectors are converted back to letters.

Algorithm:

Key Generation:

* Choose a key matrix K of size n ×n, where n is the block size.
* Ensure that K is invertible modulo the size of the alphabet.

Encryption:

* Divide the plaintext into blocks of size n.
* Convert each block to a numeric vector, where each letter is represented by its position in the alphabet.
* Multiply each vector by the key matrix K (modulo the size of the alphabet).
* Convert the resulting vectors back to letters to get the ciphertext.
* The encryption formula: C=P×K mod alphabet size.

Decryption:

* Invert the key matrix K (modulo the size of the alphabet) to get the inverse matrix K inverse.
* Multiply each ciphertext vector by the inverse key matrix K inverse (modulo the size of the alphabet).
* Convert the resulting vectors back to letters to get the plaintext.
* The decryption formula: P= C×K inverse mod alphabet size.

**VIGENERE CIPHER:**

The Vigenère Cipher is a method of encrypting alphabetic text using a simple form of polyalphabetic substitution. It uses a keyword to shift letters in the plaintext by varying amounts, creating a repeating key pattern. The Vigenère Cipher provides a more secure alternative to the Caesar Cipher.

Rules:

* The key (keyword) is repeated to match the length of the plaintext.
* Each letter in the plaintext is shifted according to its position in the keyword.
* The shift is performed using a Caesar Cipher-like mechanism, where each letter in the key determines the amount of shift.
* Non-alphabetic characters (numbers, punctuation, etc.) in the plaintext are typically ignored or preserved unchanged.

Algorithm:

Key Expansion:

* Choose a keyword, e.g., "KEY".
* Repeat the keyword to match the length of the plaintext, creating the key stream: "KEYKEYKEY...".

Encryption:

* Align each letter in the plaintext with the corresponding letter in the key stream.
* Use a Caesar Cipher shift determined by the key letter to encrypt each plaintext letter.
* Preserve non-alphabetic characters unchanged.
* The encryption formula for each letter Pi​ in the plaintext: Ci​=(Pi​+Ki​) mod26, where Ki​ is the numerical value of the corresponding key letter.

Example:

Key: "KEY"

Plaintext: "HELLO"

Key Stream: "KEYKE"

Encryption:

H (7) + K (10) = 17 (R)

E (4) + E (4) = 8 (I)

L (11) + Y (24) = 9 (J)

L (11) + K (10) = 21 (V)

O (14) + E (4) = 18 (S)

Ciphertext: "RIJVS"

**KEYED TRANSPOSITIONAL CIPHER:**

The Keyed Transposition Cipher is a type of transposition cipher that uses a keyword to determine the arrangement of characters in the ciphertext. It involves rearranging the order of the characters based on the positions of letters in the keyword.

Rules:

* The key (keyword) is used to determine the order of rearrangement.
* Each unique letter in the keyword represents a column in the transposition grid.
* The columns are arranged based on the order of letters in the keyword.
* The plaintext is then filled into the grid row by row.
* The ciphertext is obtained by reading the columns in the order determined by the keyword.

Algorithm:

Key Arrangement:

* Choose a keyword, e.g., "KEY".
* Remove duplicate letters to form the ordered key: "KEY".

Encryption:

* Write the plaintext into a grid row by row.
* Arrange the columns of the grid based on the ordered key.
* Read the columns in the order determined by the key to obtain the ciphertext.
* The encryption involves rearranging the characters based on the columns determined by the ordered key.

Example:

Key: "KEY"

Plaintext: "HELLOKEYEDTRANSPOSITION"

Ordered Key: "KEY"

Ciphertext: "KHYEDOLTORNNEESAPIT"

In this example, the ordered key "KEY" is used to determine the arrangement of columns for the transposition. The plaintext "HELLOKEYEDTRANSPOSITION" is then filled into the grid row by row, and the ciphertext is obtained by reading the columns in the order specified by the key, resulting in "KHYEDOLTORNNEESAPIT."

**KEYLESS TRANSPOSITIONAL CIPHER:**

The Keyless Transposition Cipher is a variant of the transposition cipher that does not use a specific keyword for the arrangement of characters in the ciphertext. Unlike the Keyed Transposition Cipher, which relies on a keyword to determine the column order, the Keyless Transposition Cipher employs a predefined pattern or rule for rearranging the characters.

Algorithm:

Encryption

* Define a rearrangement pattern: Determine a specific pattern or rule for rearranging the characters in the plaintext. This could involve specifying the number of columns in the transposition grid or using a predetermined arrangement algorithm.
* Write the plaintext into a grid: Divide the plaintext into blocks of a specified length and write them into a grid row by row. The number of columns may be determined by the rearrangement pattern.
* Rearrange the columns: Apply the predefined pattern or rule to rearrange the columns of the grid.
* Read the columns to obtain ciphertext: Read the columns of the rearranged grid in a specific order to obtain the ciphertext.

Example:

Plaintext: "HELLOKEYLESSTRANSPOSITION"

Procedure: Divide the plaintext into blocks of 5 characters each:

HELLO KEYLE SSTRA NSPOS ITION

Rearrange the columns of the grid based on a specific pattern:

LEHOL ESLEY RNPOT ASSTO IISIN

Read the columns in a specific order to obtain the ciphertext: "LEHOLESLEYRNPOTASSTOIISIN"

Ciphertext: "LEHOLESL EYRNPOTASSTOIISIN"

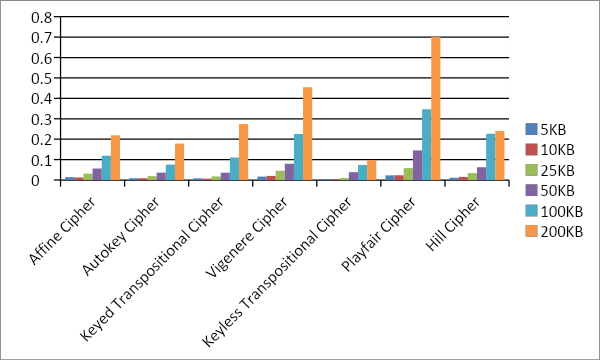
In this example, the plaintext "HELLOKEYLESSTRANSPOSITION" is divided into blocks and rearranged without using a specific keyword. The resulting ciphertext is "LEHOLESL EYRNPOTASSTOIISIN."

**RESULTS**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Affine Cipher** | **Autokey Cipher** | **Keyed Transpositional Cipher** | **Vigenere Cipher** | **Keyless Transpositional Cipher** | **Playfair Cipher** |
| 5Kb | 0.013866424560546875 | 0.008700370788574219 | 0.008182764053344727 | 0.01687765121459961 | 0.004744529724121094 | 0.022762060165405273 |
| 10KB | 0.012279033660888672 | 0.00862741470336914 | 0.006684780120849609 | 0.020046472549438477 | 0.0037069320678710938 | 0.023189067840576172 |
| 25kb | 0.03120589256286621 | 0.018835067749023438 | 0.01757073402404785 | 0.0453801155090332 | 0.009336471557617188 | 0.05815458297729492 |
| 50kb | 0.05621075630187988 | 0.036012887954711914 | 0.035796165466308594 | 0.07955574989318848 | 0.03836417198181152 | 0.14502358436584473 |
| 100K | 0.11905694007873535 | 0.0753471851348877 | 0.11019039154052734 | 0.2253570556640625 | 0.0737762451171875 | 0.34656381607055664 |
| 200kB | 0.2188255786895752 | 0.1783742904663086 | 0.2737092971801758 | 0.4541034698486328 | 0.09699535369873047 | 0.7000575065612793 |

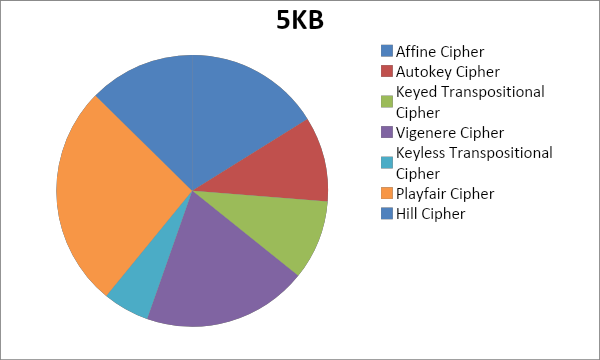
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 5KB | 10KB | 25KB | 50KB | 100KB | 200KB |
| AFFINE CIPHER | 0.013866424560546875 | 0.012279033660888672 | 0.03120589256286621 | 0.05621075630187988 | 0.11905694007873535 | 0.2188255786895752 |
| **Auto key Cipher** | 0.008700370788574219 | 0.00862741470336914 | 0.018835067749023438 | 0.036012887954711914 | 0.0753471851348877 | 0.1783742904663086 |
| **Keyed Transposition Cipher** | 0.008182764053344727 | 0.006684780120849609 | 0.01757073402404785 | 0.035796165466308594 | 0.11019039154052734 | 0.2737092971801758 |
| **Vigenere Cipher** | 0.01687765121459961 | 0.020046472549438477 | 0.0453801155090332 | 0.07955574989318848 | 0.2253570556640625 | 0.4541034698486328 |
| **Keyless Transposition Cipher** | 0.004744529724121094 | 0.0037069320678710938 | 0.009336471557617188 | 0.03836417198181152 | 0.0737762451171875 | 0.09699535369873047 |
| **Play fair Cipher** | 0.022762060165405273 | 0.023189067840576172 | 0.05815458297729492 | 0.14502358436584473 | 0.34656381607055664 | 0.7000575065612793 |
| **Hill Cipher** | 0.010853052139282227 | 0.015153884887695312 | 0.0340120792388916 | 0.06257271766662598 | 0.22638440132141113 | 0.24071264266967773 |

Fig 1 represents the execution time of all the different encryption techniques used. From the graph, we can conclude that Keyless Transposition cipher has the least variation in the execution time when the size of the plaintext is increased.



*Fig 1 Execution time of all the encryption techniques*

Fig 2 is a pie chart for the execution time of all the ciphers for the plaintext of size 5KB. Keyless Transposition Cipher takes the least time to encrypt and decrypt the plain text whereas Play fair Cipher takes the most time. The cipher techniques arranged on the ascending order of their execution time for a 5KB plaintext is Keyless Transposition Cipher < Keyed Transposition cipher < Auto key Cipher < Hill Cipher<  Affine Cipher < Vigenere Cipher < Play fair Cipher.



*Fig 2 Execution time for plaintext of size 5KB*

Fig 3 is a pie chart for the execution time of all the ciphers for the plaintext of size 5KB. Auto key Cipher takes the least time to encrypt and decrypt the plain text whereas Play fair Cipher takes the most time.

A pie chart with text

Description automatically generated

*Fig 3 Execution time for plaintext of size 50KB*

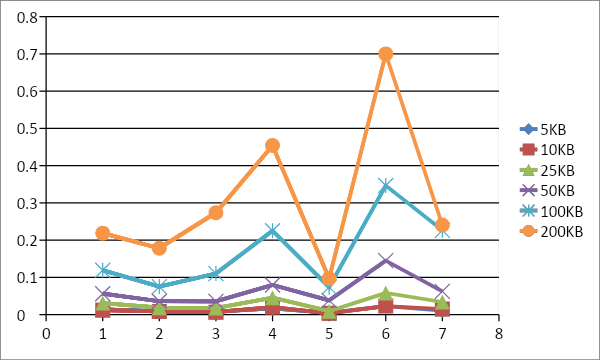
Fig 4 is a pie chart for the execution time of all the ciphers for the plaintext of size 5KB. Keyless Transposition Cipher takes the least time to encrypt and decrypt the plain text whereas Play fair Cipher takes the most time. The cipher techniques arranged on the ascending order of their execution time for a 5KB plaintext is Keyless Transposition Cipher < Auto key Cipher < Affine Cipher < Hill Cipher < Keyed Transposition Cipher < Vigenere Cipher < Play fair Cipher.

A pie chart with text and numbers

Description automatically generated

*Fig 4 Execution time for plaintext of size 200KB*

From Fig 5, we can conclude that the variation in Vigenere and Play fair Cipher’s are the most compared to the other techniques for 100KB and 200KB plaintext. Keyless Transposition cipher takes the least time whereas the variation in Keyed Transposition Cipher, Auto key Cipher, Hill Cipher and Affine Cipher is not much.



*Fig 5 Variation in execution time of different cipher’s*

In figure 5, The x-axis is labelled as:

1. Affine Cipher
2. Auto key Cipher
3. Keyed transposition Cipher
4. Vigenere Cipher
5. keyless transposition Cipher
6. Play fair Cipher.
7. Hill Cipher

**CONCLUSION**

The examinaion of the computation times for encryption and decryption for each of the ciphers, against different plaintext sizes have revealed that, overall the keyless transposition cipher takes the least amount of time and hence showcases the best performance among all the ciphering techniques, throughout the observation carried out. On the contrary, playfair cipher takes the most amount of execution time to carry out the encryption and decryption processes, thereby showcasing the worst of the performances constantly. It is repeatedly followed by vigenere cipher. It is also noted that for the shorter plaintext sizes, keyed transposition cipher shows the second best performnace performing slightly better than hill and autokey ciphers, and taking less execution time as compared to affine cipher. But as the plaintext size gradually increases, the keyed transposition cipher starts taking more computation time and its performance eventually becomes worse than that of hill and autokey ciphers. As we move towrads the largest of the plaintext sizes, keyed transposition cipher’s performance even degrades below that shown by affine cipher, compared to which it had a notibly better performance earlier on.

**REFERENCES**

[1]. Petit, C., & Seurin, Y. (2014). A practical attack on the general affine cipher with modular inverses. In International Workshop on Selected Areas in Cryptography (pp. 206-222). Springer, Berlin, Heidelberg.

[2]. Sage, D., & Slamanig, D. (2017). Breaking autokey ciphers with known plaintext. In International Conference on Selected Areas in Cryptography (pp. 427-444). Springer, Cham.

[3]. Guan, J., Zhou, X., & Guo, J. (2019). Cryptanalysis of row transposition ciphers with the knowledge of key length. In International Conference on Network Security and Applications (pp. 653-664). Springer, Singapore.

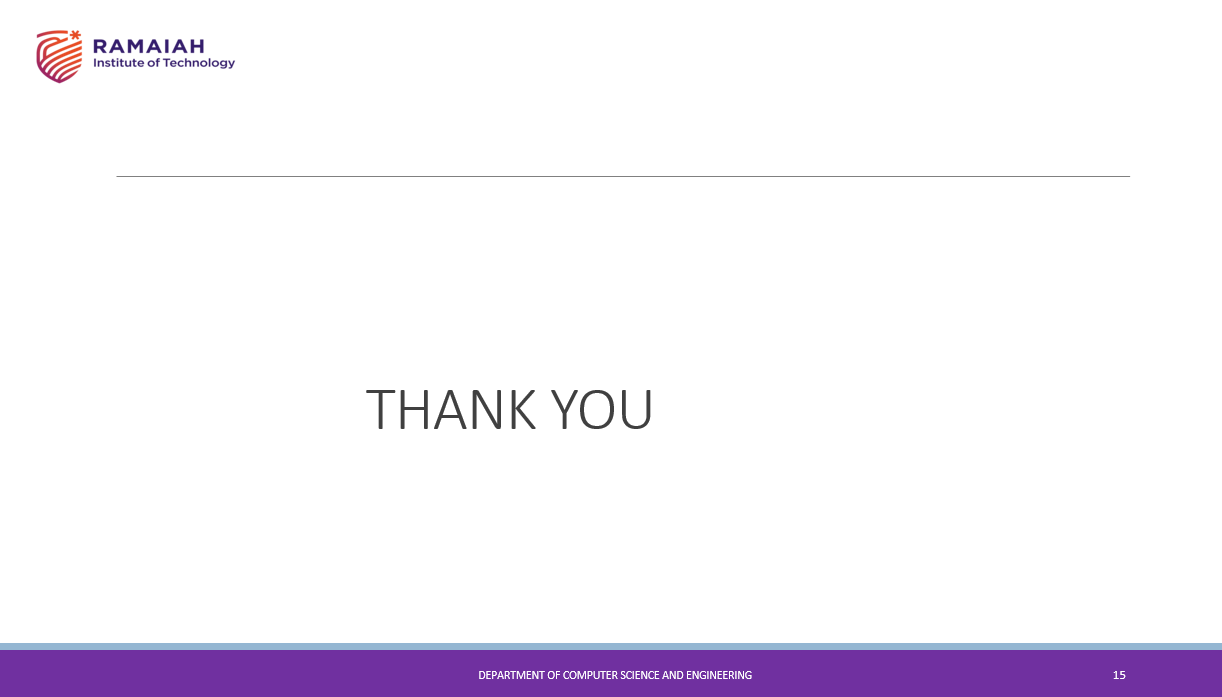
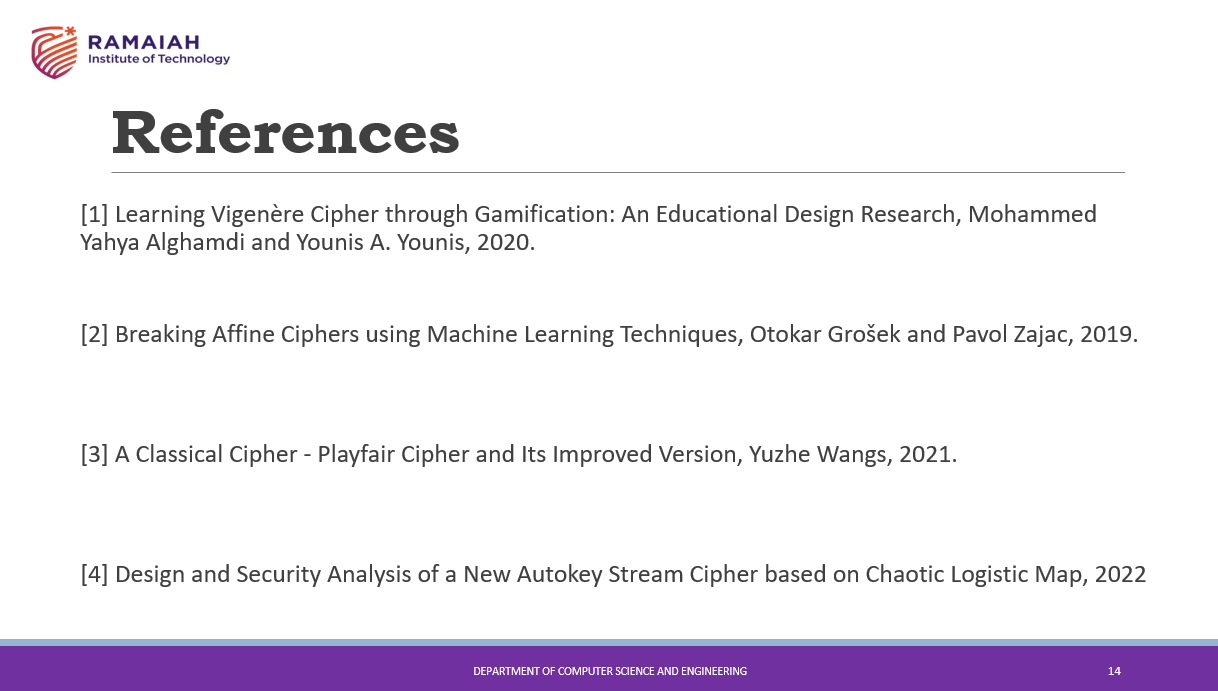
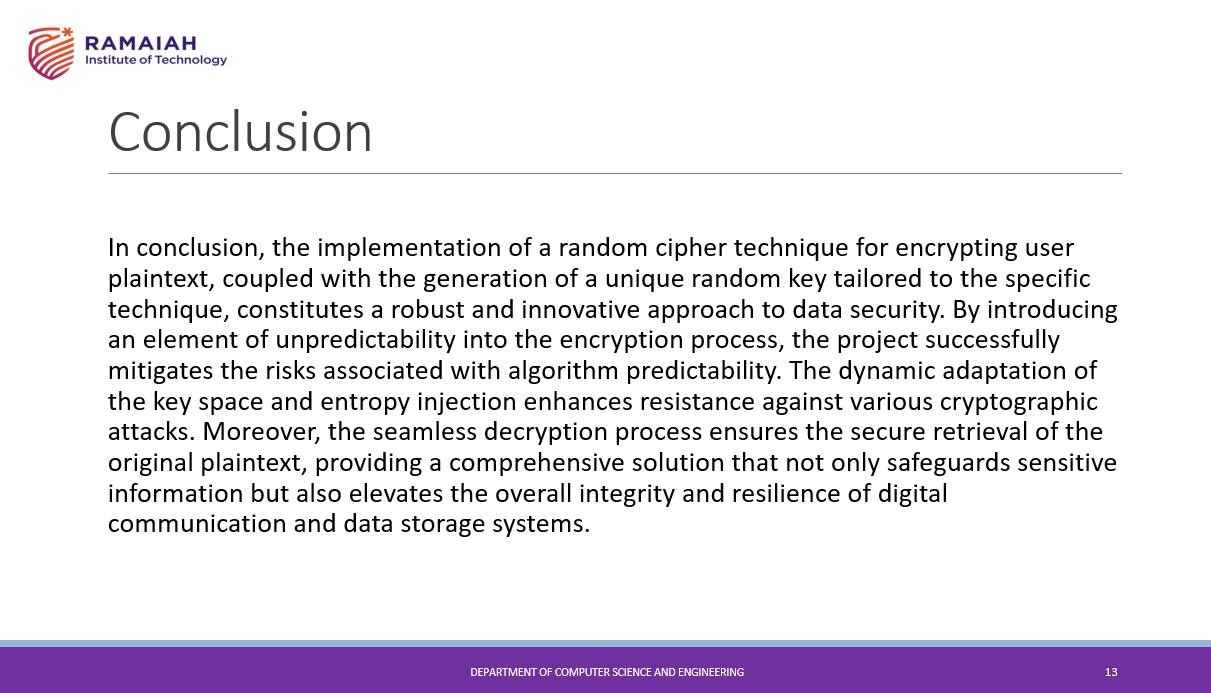
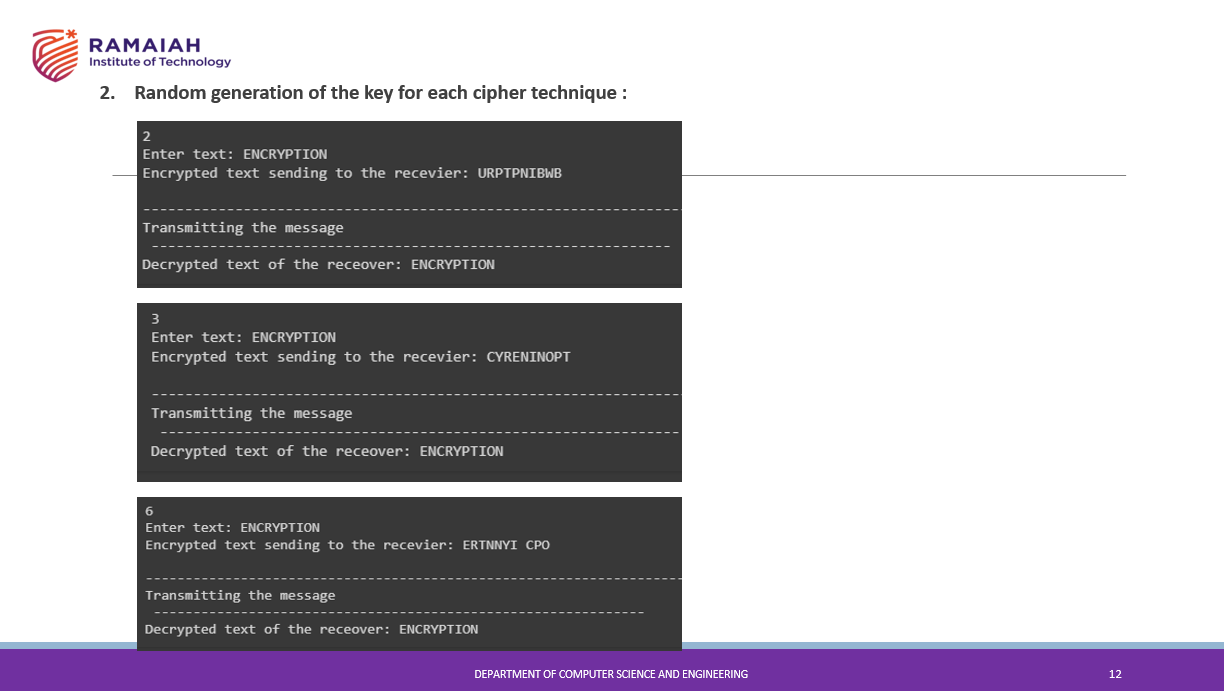
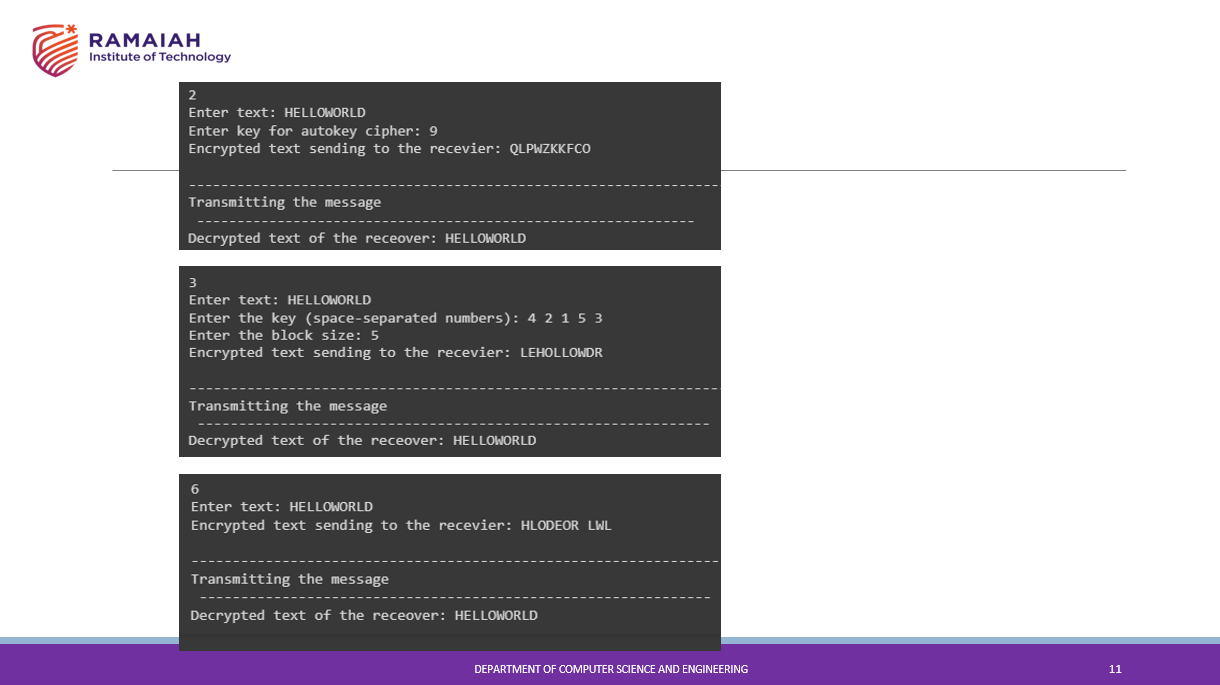
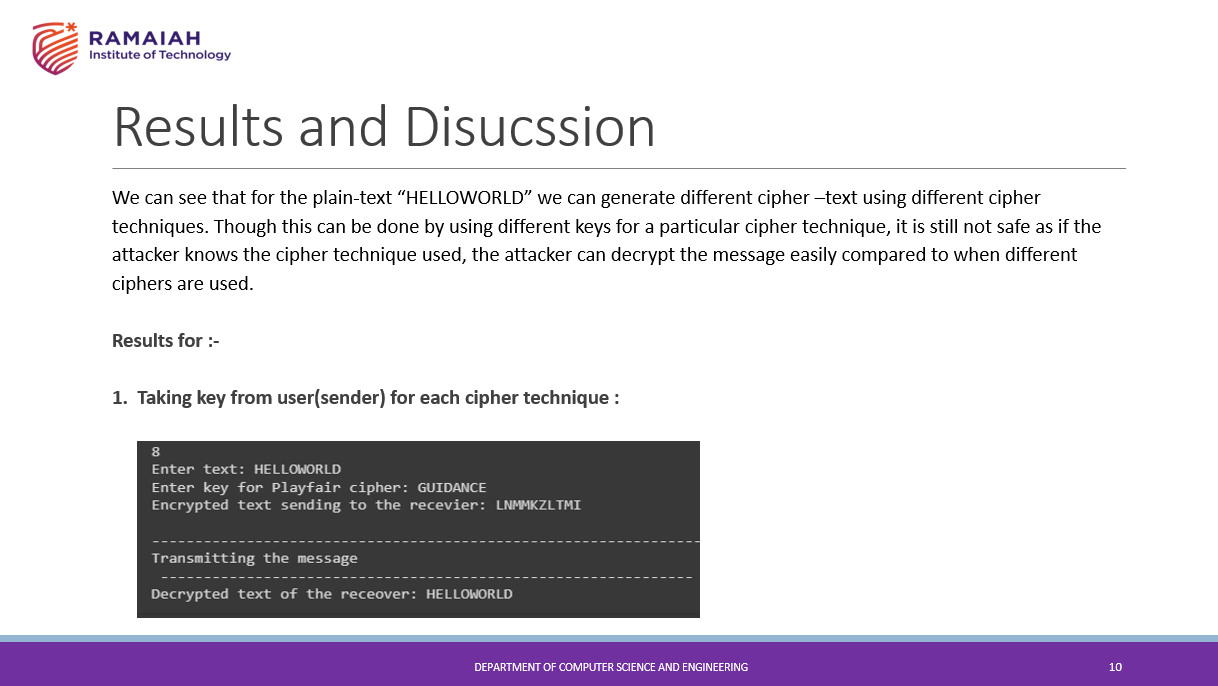
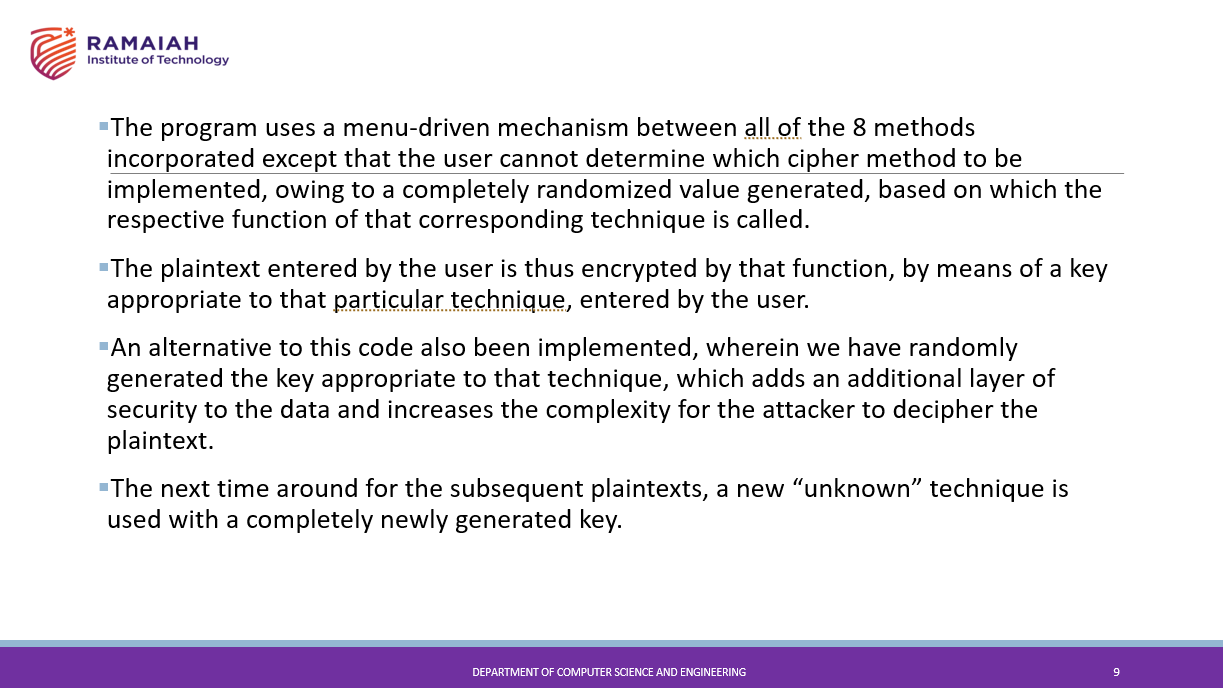
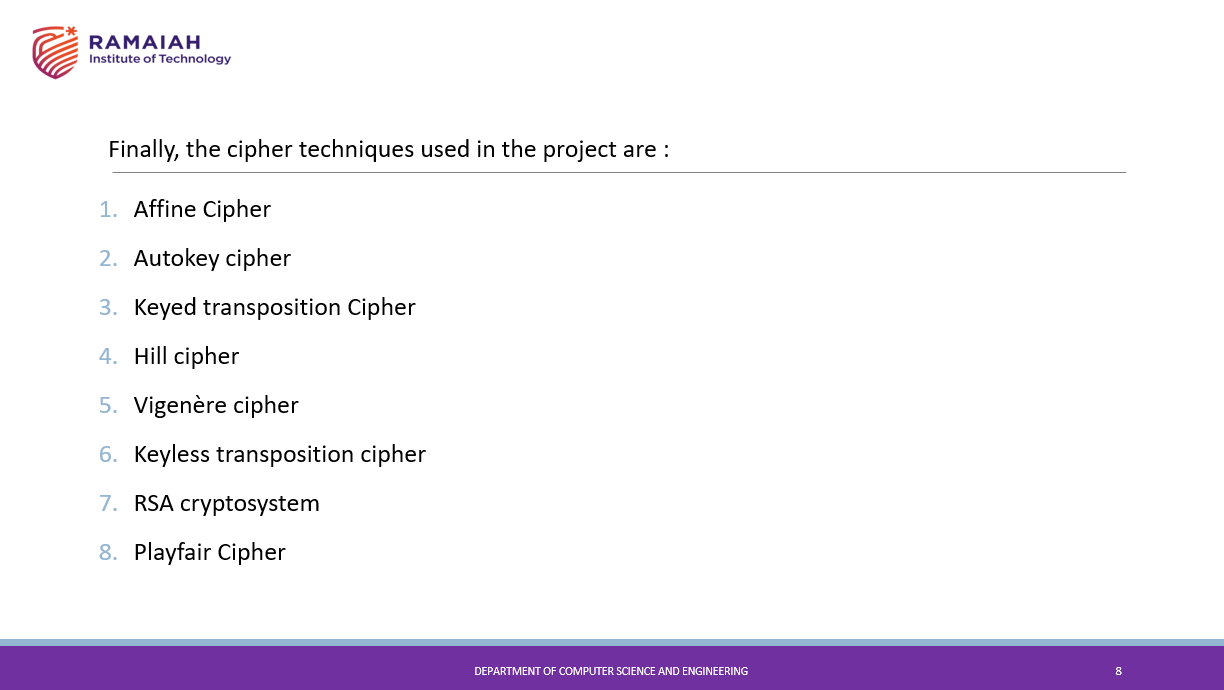
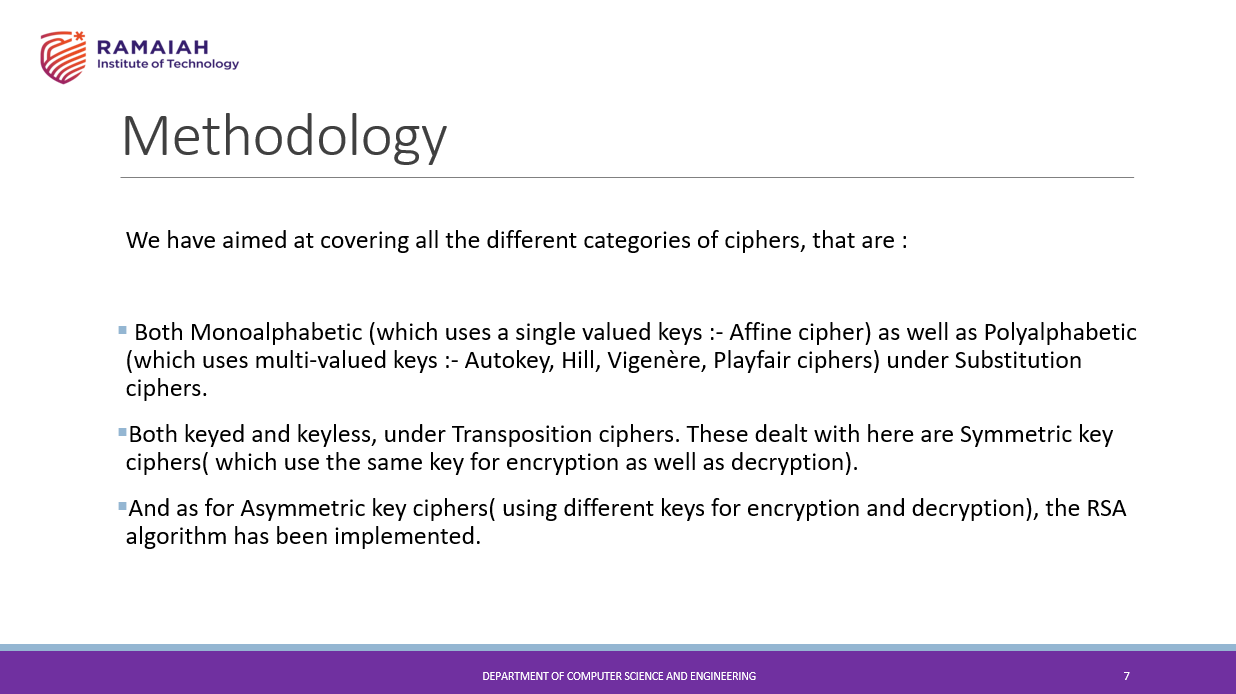
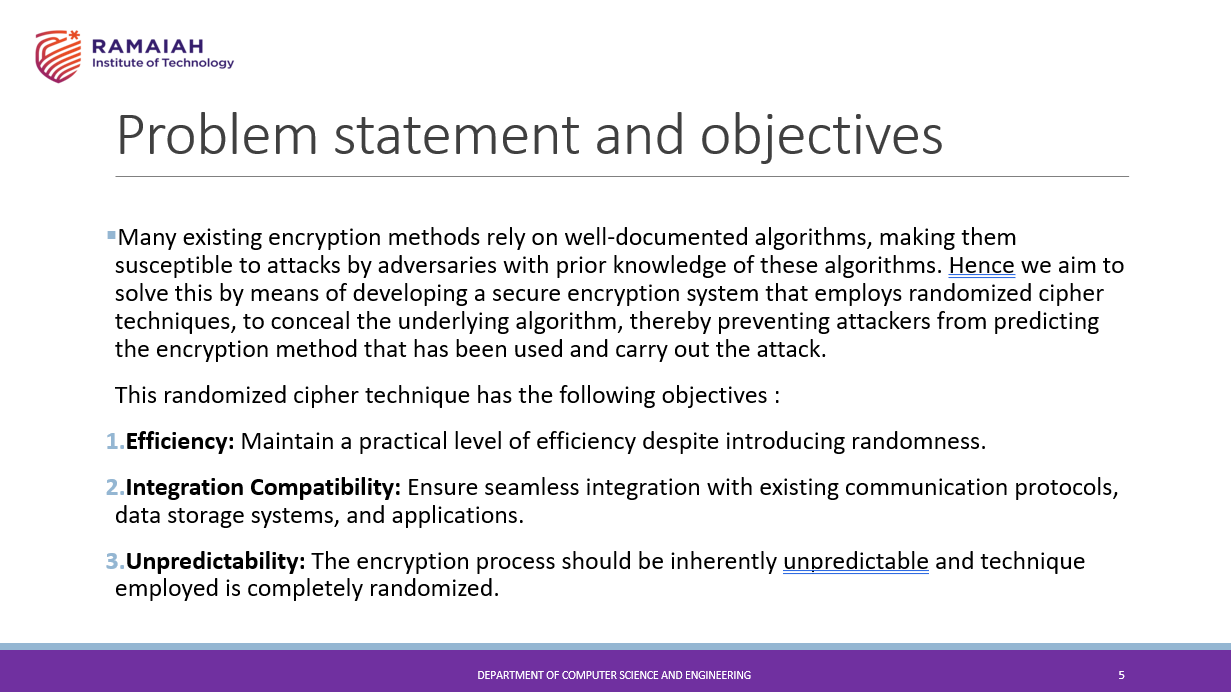
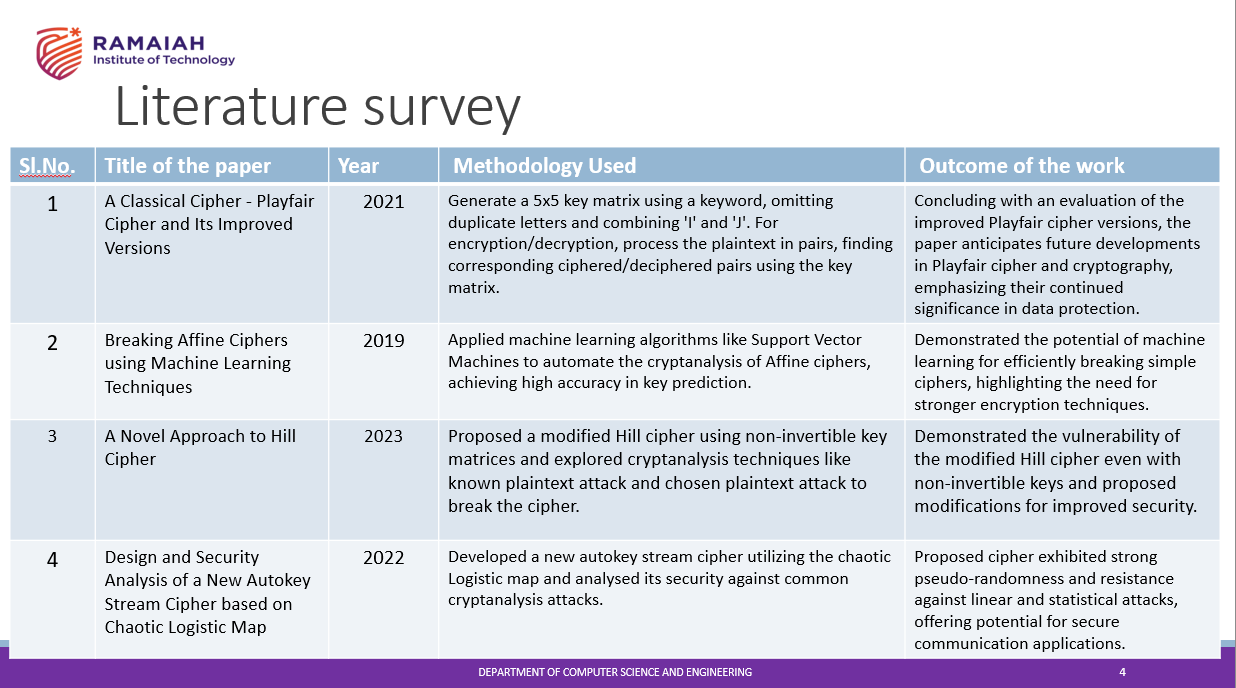
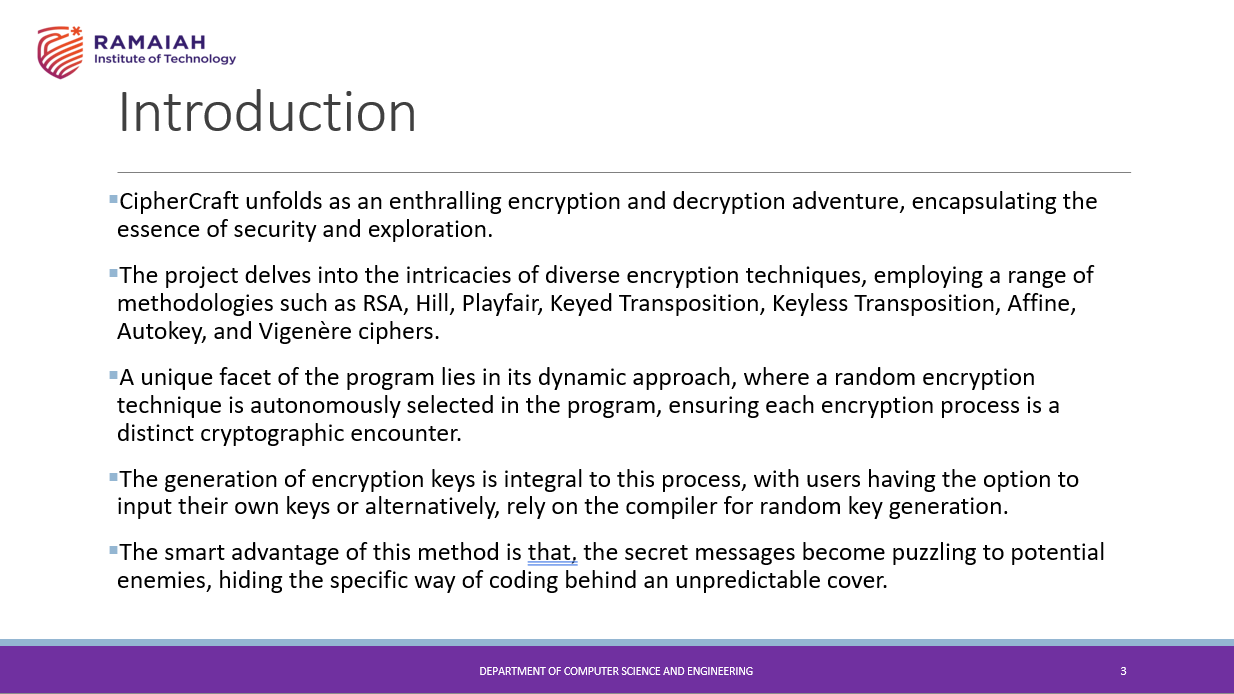
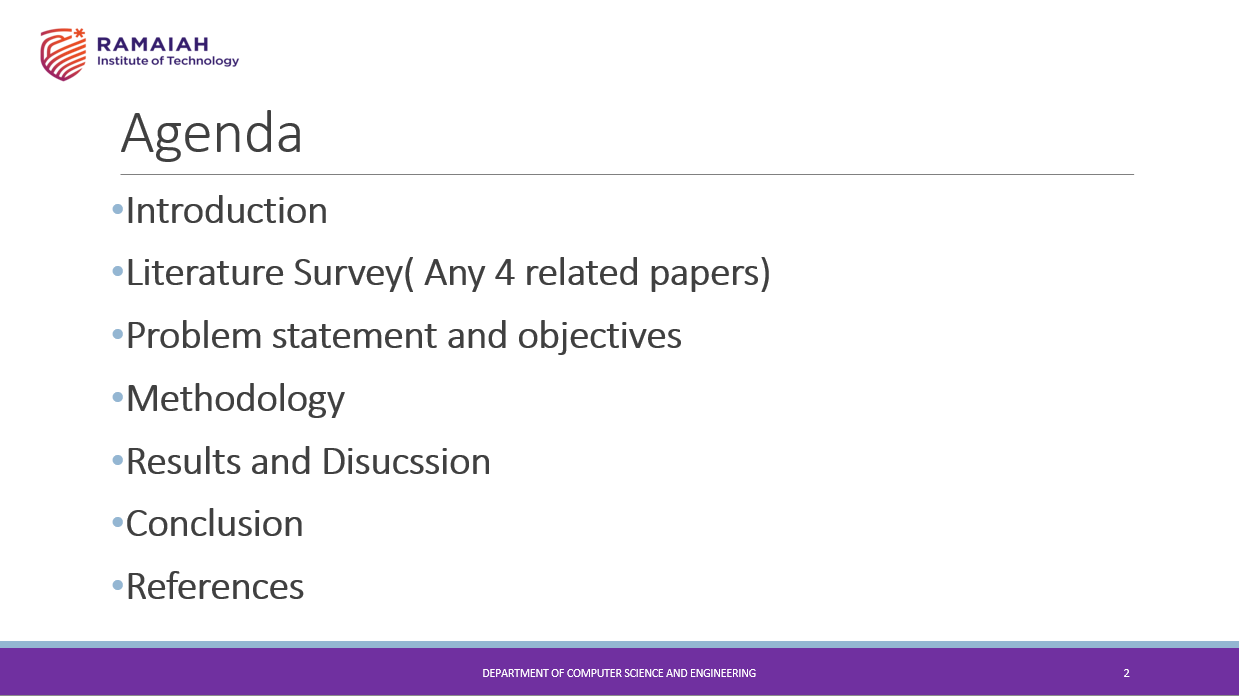
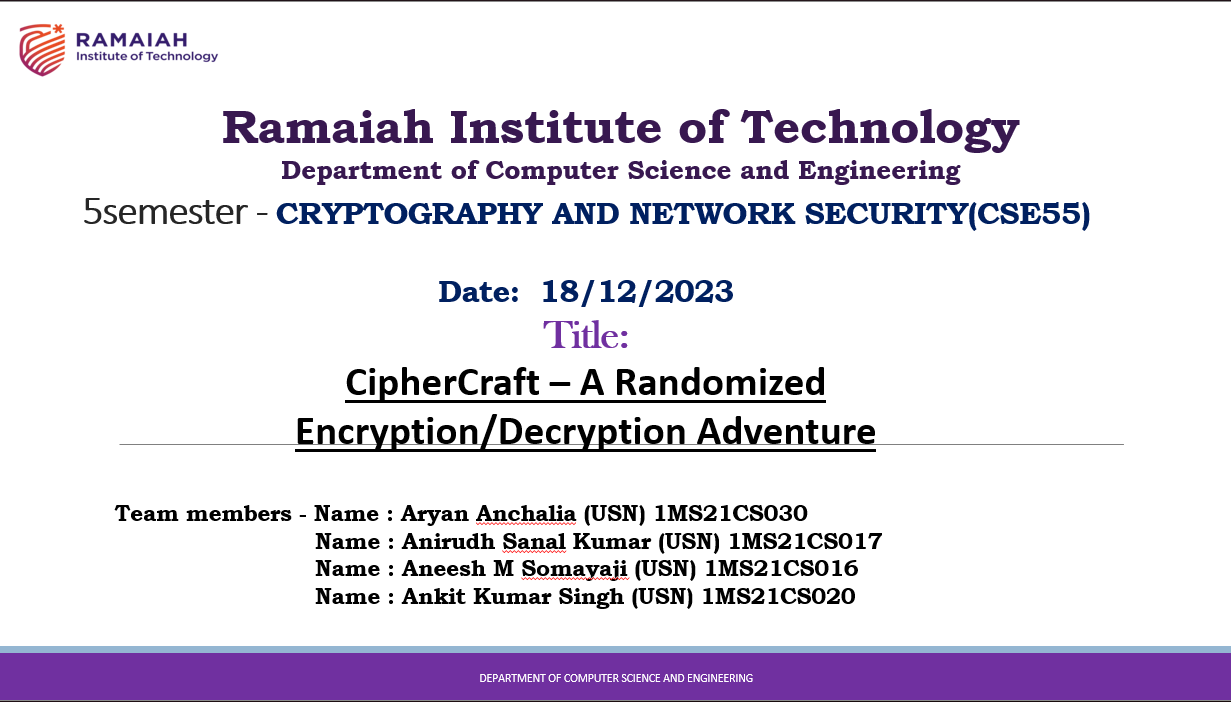
[4]. Courtois, N., & Maze, G. (2009). Efficient hill cipher cryptanalysis using gröbner bases. In Annual International Conference on the Theory and Applications of Cryptographic Techniques (pp. 392-407). Springer, Berlin, Heidelberg.

[5]. Friedman, W. F. (1937). Kasiski examination: A critical appraisal. Technical Report, War Department Signal Intelligence Service, Arlington Hall, Virginia.

[6]. Bernstein, D. J., & Lange, T. (2014). On the security of the rail fence cipher. In International Workshop on Selected Areas in Cryptography (pp. 301-313). Springer, Berlin, Heidelberg.

[7]. Knudsen, L. R., & Stinson, D. R. (2000). A statistical attack on the playfair cipher. Designs, Codes and Cryptography, 16(3), 203-214.

**PROJECT PRESENTATION SLIDES**

****