

“Atlanta: Data Encryption Algorithm”

Anshu Vashisth*, Gagandeep kaur, Bhupinder kaur

Lovely Faculty of Technology and Sciences

Lovely Professional University, Phagwara, Punjab.

anshu.23500@lpu.co.in, gagandeep.23625@lpu.co.in,

bhupinder.23626@lpu.co.in

Abstract: In Today's technical world, An Individual has two intrinsic needs one to impart the data and second to share data, to impart specifically. This give rises an idea about art of coding the information in such a way that only the intended person should access that data. This paper proposed an algorithm for encrypting the information and shows the result using different software such as ENGIMA-1, ENGIMA-II, MINNU. The algorithm uses the random key of 41 digits where encryption is performed on the Unicode value of the character by setting the initial shift with value zero. The block size that has been used is having 171 characters.

Keywords: Encryption, decryption, cypher text, secret key, plain text.

1. INTRODUCTION

Security measures are considered as a major challenge in the computer systems specially when these systems are potential targets for malicious attacks. The proposed system particularly made for the critical systems which needs confidentiality, integrity or authentication. The designed system to maintain the confidentiality, integrity of information that may belong to individuals or to any organizations. The unapproved individuals couldn't gain admittance to any data, regardless of whether the mixed messages fell in their grasp. The workmanship and study of hiding the messages to present mystery in data security is perceived as cryptography [1]. The word 'cryptography' was blend of two Greek words, 'Krypto' which means covered up and 'graphene' which means composing. For the encryption of message, there are two strategies:

- Symmetric Encryption (private key encryption)
- Asymmetric Encryption (public key encryption).

Symmetric encryption is the traditional technique [2]. A private key which might be a number, word or string, is used to change the data or message. The encryption and decryption can be done by sender and receiver respectively as they both are aware about the private key.

The designed algorithm converts the human readable text to the unreadable format called cypher text. That cypher text can be decrypted by the recipient side which will be authorised person. The proposed algorithm provides security in two layers by shuffling the letters of information and then use a key to encrypt it. The two-layer security makes it more difficult for the unauthorized person to get the cypher text.

But, the issue comes when secret keys have trade over the Internet or when a huge system may fall into an inappropriate hand. In the event if the mystery key is known to anybody, message can be unscrambled effectively. For this issue one arrangement is asymmetric encryption, in which a key pair is utilized and these keys are connected strongly. These are public and private key. The public key can be accessed by anyone who wants to send the message. A second, private key which is additionally called secret key and stayed quiet. Any message may contain content, double records, or archives that are scrambled with the assistance of the general population key, it must be unscrambled by applying a similar calculation, yet with the assistance of private key. In proposed algorithm, the private key is being used to generate the secret key that will be used to encrypt message and can be decrypted only if the key generated on receiver finds the exact secret key. The objective of paper is that intruder unable to identifies the message even if he succeeds in calculating one or some of the parts of the key. In proposed algorithm we have combined the ideology of substitution cipher and transposition cipher by implementing the both methods simultaneously. Data generated at sender's end will be shuffled and encrypted in a such a way that a secure transmission can take place. The DEFFIE HELLMAN algorithm and a SHUFFLE ALGORITHM used to generate the secure transmission of data. The proposed algorithm has implemented deffie-hellman algorithm for key exchange between the users. And the data is shuffled by using shuffle algorithm to generate the shuffled text. The key required for deshuffling of the blocks, is encrypted within the shuffled text such that the attacker can't determine the actual plain text.

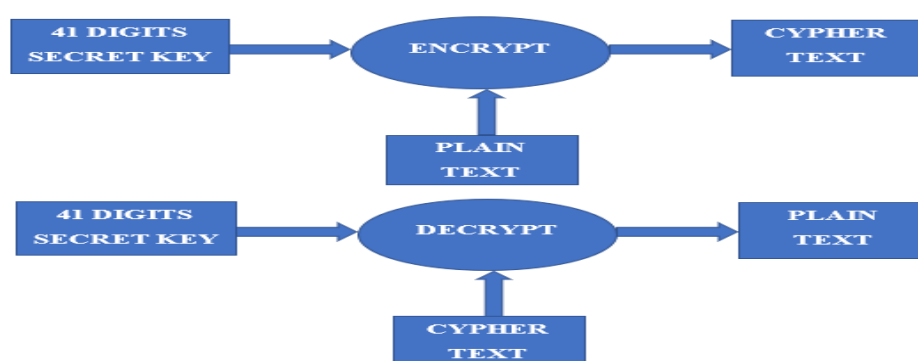


Figure 1: Simple Process for Encryption and Decryption

2. EXISTING ALGORITHMS

Perfect Shuffle Algorithm for Cryptography: Perfect Shuffle Crypto Algorithm (PSCA) is a transposition method or change procedure. PSCA used the asymmetry key encryption where it utilizes a couple of keys, that are an open key for encoding information, and a relating private mystery key for unscrambling the information [6]. For the direct plaintext length of $N=2n$, it will take $O(N \log N)$ to finish both scrambling plaintext and unscrambling figure content. The PSCA is sensibly secure, particularly for jaunty content just assault.

Fisher-Yates Shuffle: Fisher Yates Shuffle algorithm shuffles the text randomly by picking up a random no. k and then assign the position of the character accordingly. This algorithm is based on permutation of the shuffled text formed and the complexity of the algorithm is $O(N!)$.

Modified Caesar Cipher: The Modified Caesar Cipher algorithm is based on implementation of a symmetric key at both sender and receiver end to encrypt as well as decrypt the data. The Key value is added to each character and thus we obtain out cipher text [3]. But this technique has its own drawbacks which made it to be less implemented and used in the current scenario [4]. The key value can't exceed the value of '25' and hence this makes easy for the unauthorized person to access the data easily after assuming the correct value of the key used [5].

Table 1: Pitfalls in Existing Algorithms

Algorithm Name	Limitation of Algorithms
DES	• DES used weak keys and vulnerable to Linear Cryptanalysis attack. • It can be easily cracked by brute force attack [8].
TDES	• It is vulnerable to different man-in-middle attacks. • It is vulnerable to differential and related key attacks.
CAST 128	• CAST 128 can be exposed by 2^{17} chosen plaintexts. • It is exposed to differential related-key attack due to key size as 64 bit.
BLOWFISH	• Solidity of Blowfish is destroyed with large number of poor keys. • The first 4 rounds of process is exposed to 2nd order of

	differential attacks.
IDEA	• IDEA is helpless against Collison assault. • The initial 3 rounds among the 8 rounds of procedure are presented to key-plan assaults and key-related differential planning assaults [7].
RC6	• If the key size is small, it is vulnerable to brute force attack. • Time consumption is high for the process.
AES	• The joined boomerang and square shape assault with related-key differentials utilizes the shortcoming of few non-straight changes in key-plan calculations and can break some decreased round renditions of AES[9]. • It can break 192-piece, 9 rounds AES by utilizing 256 diverse related keys.
RSA	• Due to small and close prime number which can be factorised easily. • Due to more encryption time and memory usage, the speed of the algorithm become very slow.

3. PROPOSED ALGORITHM

Key Generation: This module generates keys. The keys can be generated randomly, or it can be specified by the user. If the keys are being generated randomly, the user does not need to check the key length. But if the keys are entered manually, the user needs to be sure of the number that the number should be prime. When the keys are entered manually, the test for prime number is done.

Encryption: The encryption is performed based on the Unicode value of the character. The encrypted code for every character applied is calculated by the encryption formulae i.e.

$SHIFT = ((KEY * (UNICODE_value)) + SHIFT) \% STRING_len;$

$ENCRYPT_char = STRING[shift] + SHIFT;$

where the initial shift value is set to be zero. Then swap the ENCRYPT_char at the new SHIFT value.

Decryption: The last shift value at the receiver end is used for decrypting the text. Compute the original plain text by the decryption formula i.e.

$\text{Sub_String}[\text{shift}] = \text{Sub_String}[\text{shift}] - \text{shift};$

$\text{Pre_Computed_Value} = (\text{Key} * \text{ASCII Value of Sub_String}[j]) \% \text{Sub_String.length} - \text{shift};$

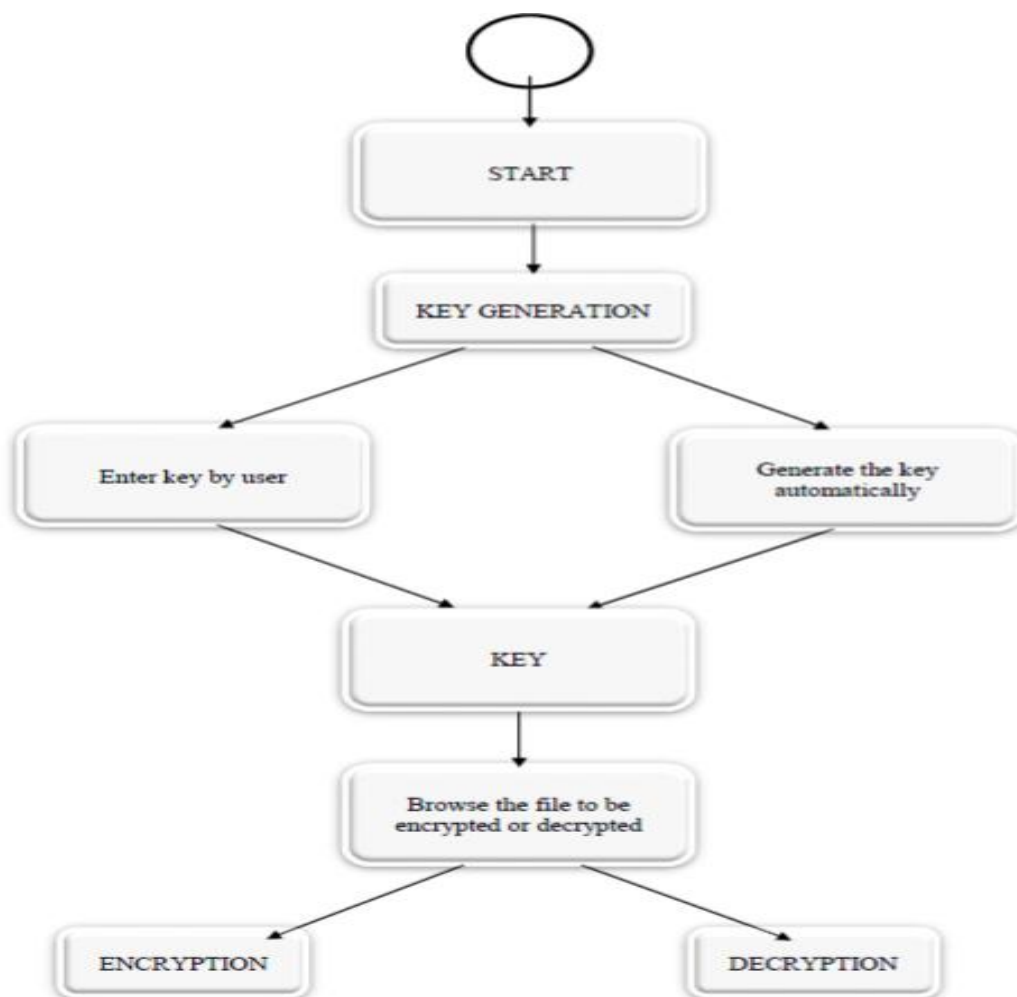


Figure 2: Flowchart of proposed algorithm

Steps of Algorithm:

Step 1:

Generate a secret key that count 41-digits through the key generator.

Step 2:

Initialize the value of initial shift = '0', and compute the no. of blocks in the text file based on the string length. The block size comprises of the length of 169 characters in the file.

Step 3:

Compute the encrypted code for every character applying the encryption formulae i.e.

$SHIFT = ((KEY * (UNICODE_value)) + SHIFT) \% STRING_len;$ ENCRYPT_char = STRING[shift] + SHIFT;

Step 4:

Apply the swap function to swap the ENCRYPT_char at the new SHIFT value.

Step 5:

The last shift value is inserted into the cipher text file so the receiver can avail it and decrypt it obtains the shift value.

Step 6:

For the decryption we need similar 41- digit key to avail the decrypted cipher text and we need to shuffle it back to avail the plain text.

Assumptions of Algorithm

- The key pair generated for every user through Key Pair Generator function will be unique.
- The secret key generated for pair of sender and receiver will be generated through the combination of shared key and their private key respectively. Thus, nullifying chances of getting similar secret key of 41 digits at any instance.
- The algorithm assumes the initial shift value as zero and hence can also be changed by administrator to any no. up to the string length.
- The shift value computed will probably not be like previous shift value.
- The block size is set to be of 169 characters and in the file, length increases more than 169 character then new blocks will be generated.
- We assume adding last shift value to the block will increase the block by 2 characters thus, resulting into a block of 171 character.

4. GUI (Graphical User Interface)

GUI provides an efficient way through which user can interact with the system and works accordingly to fulfil the task. Through GUI user can select the file that is to be encrypted or decrypted. User can generate keys manually by entering them as large prime no. or can be generated by the system. At every window a “back” button is providing that lets the user to go back to the previous window. Thus, it provides a user-friendly environment.

5. RESULTS

ENGIMA: I

Purpose of the software: To secure the file of the single user in their system. Software

Requirement: PYTHON 3.6 integrated with PYQt4

Description: This is a simple application to secure the files and documents for the user on their devices using a key that is only known to the user. This application helps user to store their files and documents securely over any system and will be able to decrypt the text only when

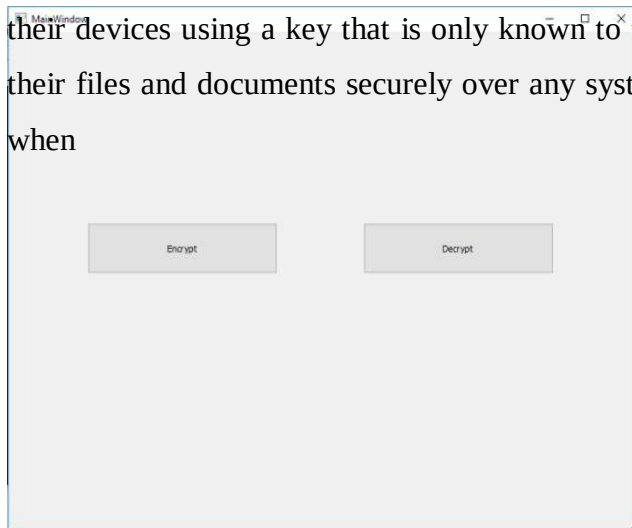


FIG 3: MAIN WINDOW

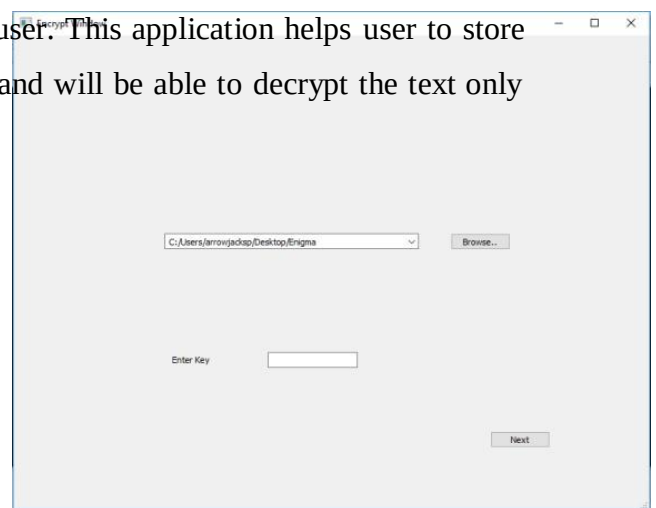


FIG 4: ENCRYPTION WINDOW

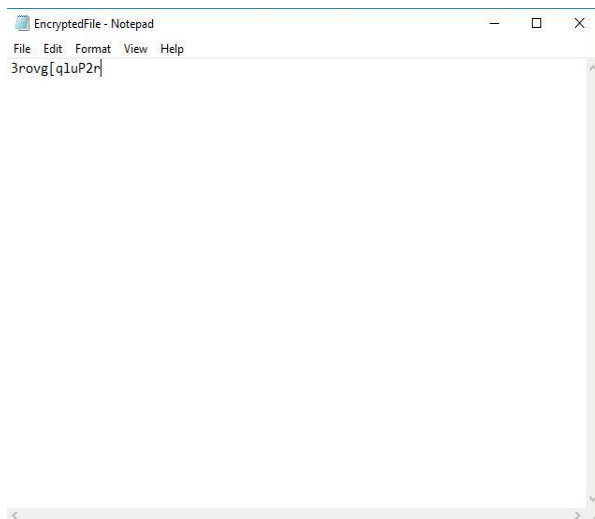


FIGURE 5: DECRYPTION WINDOW

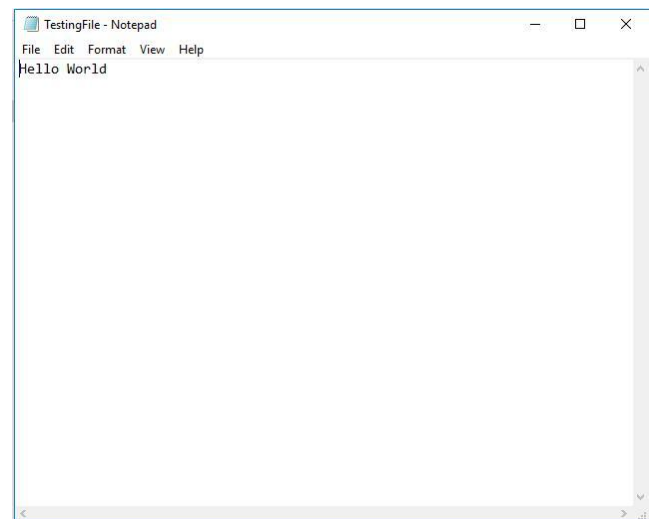


FIGURE 6: PLAIN TEXT

The working of the application is based on entering any prime no. as the key for encryption and browse the path of the file to be encrypted. Then click on encrypt button and the encrypted file will be created. Similarly, for decryption the same key is inserted and the encrypted file path is browsed, the decrypt button decrypts the file.

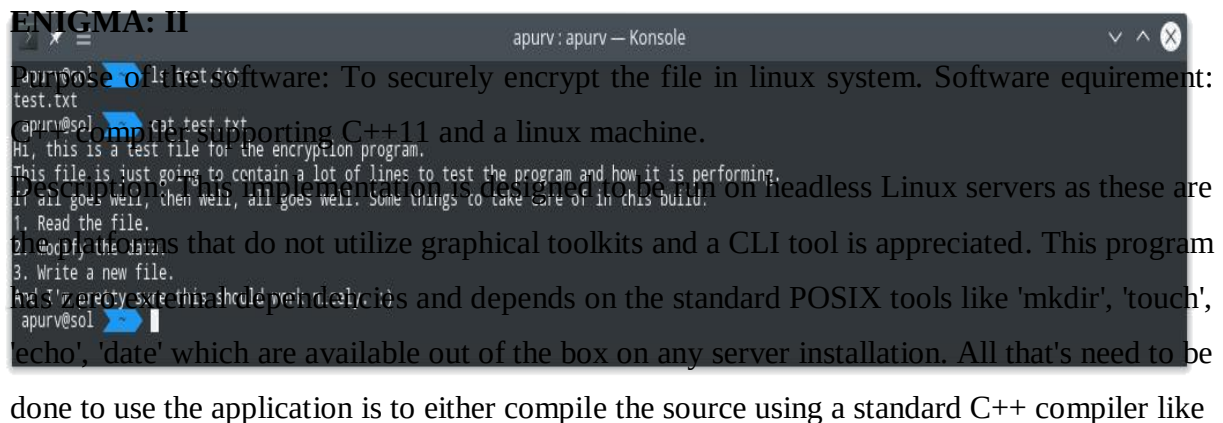


Fig:7

To safeguard the user's details his password and username are stored in a configuration file with hashed values of the user details, that way the program achieves an extra layer of security from prying eyes. SHA-512 is used to generate the hash keys of username and password GCC that supports C++11, or you can just put the binary in a directory pointed out by your PATH variable and just call the program like any other system program. The '-h' option gives a basic idea about how to use the application and can be referred at any time for help. To compile it from source, all you need to do is change into directory and run the command 'make'. It will build a binary in the 'build' directory. We can create multiple users for the application and only a user who encrypts the file can decrypt it because the key used to encrypt/decrypt the text file depends on the user's password and a special number that's generated for the first time whenever the program is begin.

FIGURE 8: PLAIN TEXT

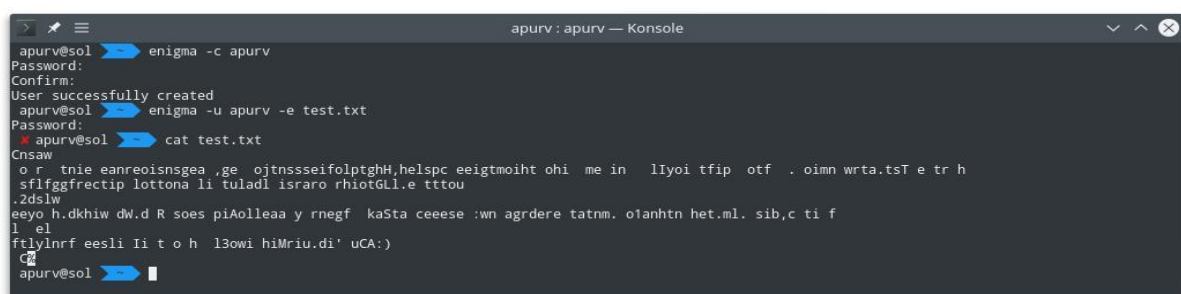
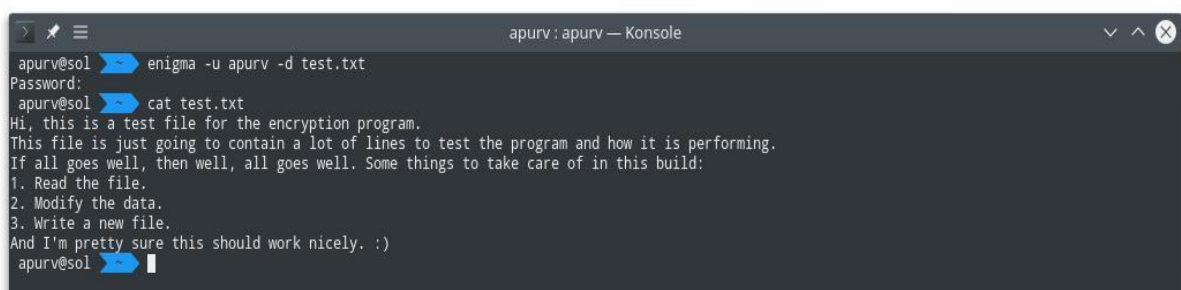


FIGURE 9: USER CREATION AND ENCRYPED TEXT



MINNU

Purpose of the software: It's a chat application that sends the data using the encryption technique. Software requirement: An android based device.

Description: The main functionality of the application is secure communication using the encryption technique.

The user when registers in the application a pair of private key and public key is generated for the user. When the user initiates conversation a shared key is generated using the deffie-hellman algorithm to securely exchange the key between the sender and receiver. As the secret key of 41 digit is set at both the end, this key is used for encryption as well as decrypting the cipher text.

The algorithm ATLANTA is used as the encryption algorithm to securely transmit the data between the sender and the receiver in a secure/unsecure channel.

The user registers themselves in the application and begins the conversation with other user through the application.

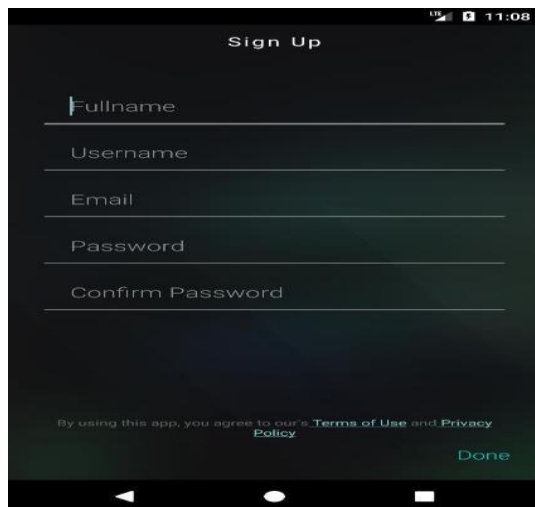


FIGURE 10: SIGN UP WINDOW

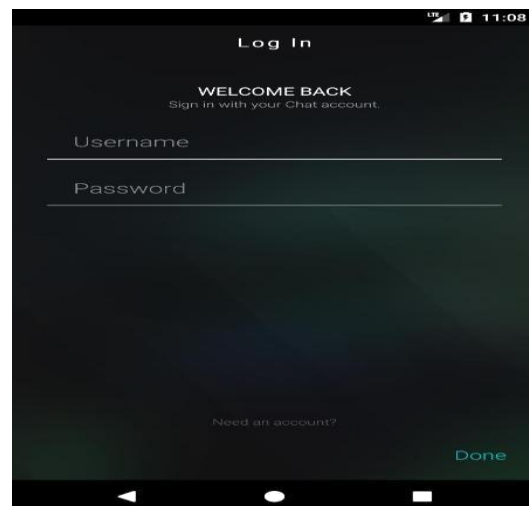
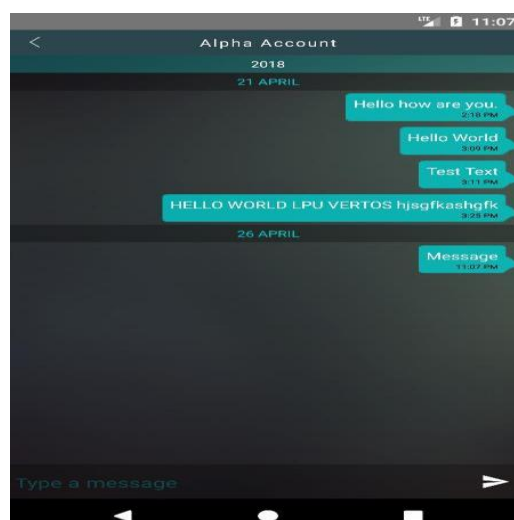


FIGURE 11: LOG-IN WINDOW

FIGURE 12: LOG-IN WINDOW



6. CONCLUSION & FUTURE SCOPE

With the continuous development of Internet technology and scale, the scope and field of these application gradually increases. Especially in the financial, e-commerce, database areas, the security requirements are higher. On the basis of higher demand, with the traditional method to solve the security problem, there are huge drawbacks. More and more users are aware of the importance of data confidentiality. Data sensitivity, confidentiality will have a means or method to solve, so the use of data and dissemination of encryption is very important, must greatly guarantee their encryption reliability and security. But security is a combination of technology, management and regulation, and absolute security is not the only way to minimize the potential threat of the network through all efforts. So encryption is the core of network security, file security, e-commerce security. Encryption algorithm, key management, field type processing problem is an important research topic of network database encryption and file encryption. In this proposed system the data confidentiality technology is studied under such premise, and the key technologies of data encryption, encryption method, data encryption technology and data encryption are analysed. After analysis and comparison, the advanced encryption technology is made using transposition and substitution to encrypt the data. This will maximize the protection of our data security.

7. FUTURE SCOPE

In addition to the improvements and techniques presented in this algorithm, further improvements can be made. Time spent in generation of key can be reduced by using the timestamp values. Parity bits can be inserted to ensure the integrity of the algorithm. This will ensure that the data sent is exactly the same what is to be received at the receiver's end. The algorithm can also be implemented over hardware for greater security purpose.

References:

- [1] D. Bucerzan, "A cryptographic algorithm based on a pseudorandom number generator," *Proc. 2008 10th Int. Symp. Symb. Numer. Algorithms Sci. Comput. SYNASC 2008*, pp. 453–456, 2008.
- [2] M. R. Ogiela, L. Ogiela, and U. Ogiela, "Cryptographic techniques in advanced information management," *Proc. - 2014 8th Int. Conf. Innov. Mob. Internet Serv. Ubiquitous Comput. IMIS 2014*, pp. 254–257, 2014.

- [3] C. Tan and Q. Ji, "An approach to identifying cryptographic algorithm from ciphertext," *Proc. 2016 8th IEEE Int. Conf. Commun. Softw. Networks, ICCSN 2016*, pp. 19–23, 2016. [1] A. O.
- [4] Mulani and P. B. Mane, "Area optimization of cryptographic algorithm on less dense reconfigurable platform," *2014 IEEE Int. Conf. "Smart Struct. Syst. ICSSS 2014*, pp. 86–89, 2015.
- [5] V. Jain, P. Sharma, and S. Sharma, "Cryptographic algorithm on multicore processor: A review," *Conf. Proceeding - 2015 Int. Conf. Adv. Comput. Eng. Appl. ICACEA 2015*, pp. 241–244, 2015.
- [6] M. A. Velciu and V. V. Patriciu, "Methods of reducing bio-cryptographic algorithms computational complexity," *CINTI 2014 - 15th IEEE Int. Symp. Comput. Intell. Informatics, Proc.*, pp. 153–158, 2014.
- [7] P. Semwal and M. K. Sharma, "Comparative study of different cryptographic algorithms for data security in cloud computing," *Proc. - 2017 3rd Int. Conf. Adv. Comput. Commun. Autom. (Fall), ICACCA 2017*, vol. 2018-January, pp. 1–7, 2018.
- [8] C. Algorithms, K. Lengths, and A. E. Standard, "Data encryption standard (DES)," *SpringerReference*, vol. 3, 2012.
- [9] H. Duan, W. Huang, and G. Zhu, "Research on Data Encryption Technology," pp. 10–16, 2017.
- [10] HOW IS THE RSA ALGORITHM USED FOR PRIVACY IN PRACTICE? <https://india.emc.com/emc-plus/rsa-labs/standards-initiatives/rsa-algorithm-used-for-privacy-in-practice.htm>
- [11] A Review of the Diffie-Hellman Algorithm and its Use in Secure Internet Protocols <https://www.sans.org/reading-room/whitepapers/vpns/review-diffie-hellman-algorithm-secure-internet-protocols-751>