IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Vol. 08, Issue 8 (August. 2018), ||V (VI) || PP 29-41

Framework for Multi-Round Security Algorithm

Khaitul Abeez, Sandeep Sangwan, Majid Zaman, Muheet Ahmed,
[5]Junaid Rasool
[1,2]*Swami Devi Dyal Institute of Engineering and Technology, Haryana, India*
[3,4,5]*University of Kashmir*

***Abstract**: Any and every information that we have taken a certain form. In raw terms we can say that the information that we store in the form of files on our computers and network servers takes a broader shape and we call it data. This data is stored in various manners governed by certain specifications and purposes. This data takes into account all the kind of information that one works with. So the nature of this information is dynamic when looking at different aspects of it. Some part of the information is intended for everyone to access and work on but there is some part of information that is intended for only a select few. That is where data security comes into play. This concept is employed with the intention that unauthorized access to any information does not happen. This research paper proposes a multi-round encryption/decryption algorithm, aiming at developing a security system that will enhance file and database security by administering the algorithm to encrypt and decrypt files, thus fulfilling the very aim of data security. In simpler terms we can say this algorithm takes the unauthorized access to any sort of data or sensitive information out of the picture.*

***Keywords:** Encryption, Decryption, Cipher, Key, Rounds.*

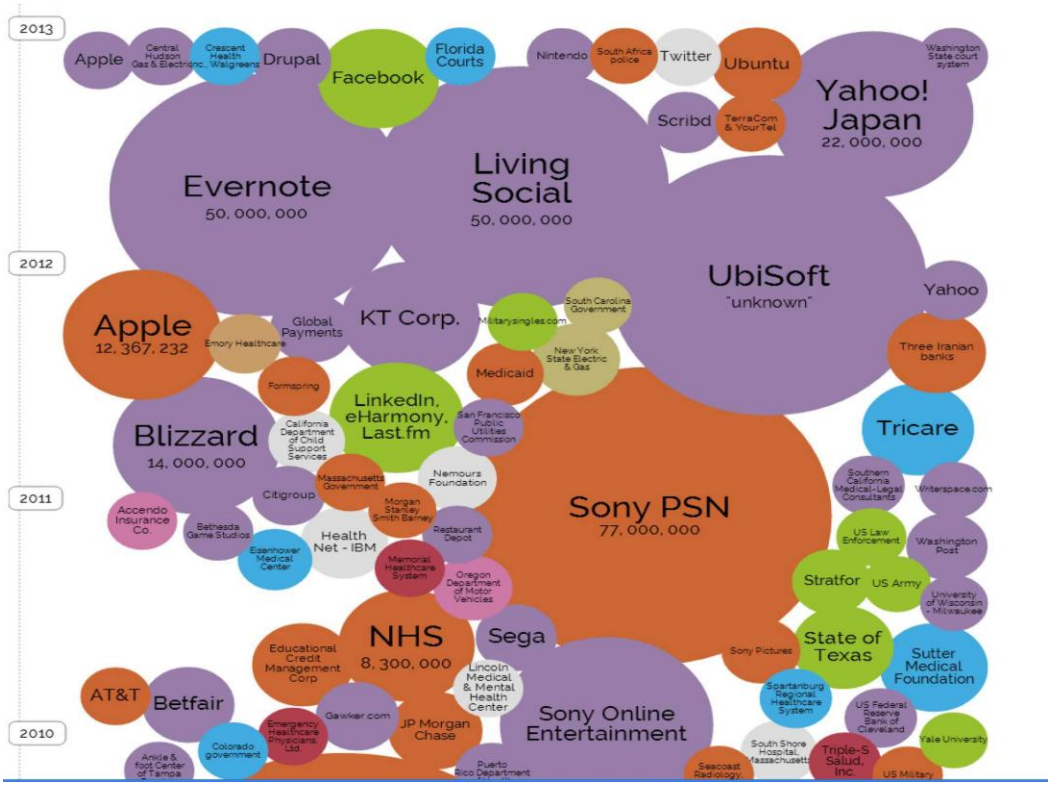
Date of Submission: 26-08-2018 Date of acceptance: 06-09-2018

I. Introduction

Data is the new oil. Production of data has crossed expectations and some would argue that it has crossed imagination as well. When we look at the amount of data that is being created, many issues come into existence. First came the storage and researchers started working on it but in recent times, when we have almost successfully tackled the storage issue, we are struck with yet another issue i.e Privacy. The proverb, "knowing is owning" is something that is being taken very seriously now. The amount of data that people have and create on a daily basis makes an individual vulnerable to data thefts and thus resulting in an alarming risk to privacy.

Thus data security is not a luxury anymore. It is the need of the hour. Through data security we not only aim to ensure privacy but also protection of personal as well as corporate data. This is being done using various mechanisms involving software as well as hardware.

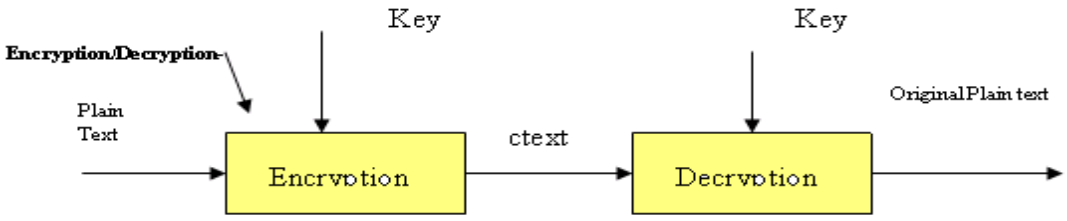
The internet is a giant vault where people store some of their most private information, trusting that the company holding on to it can keep it all safe. That's not always the case, as this info graphic of data breaches in recent history reveals viruses, hacks, lost computers, accidental publishing, inside jobs and more have all been sources of major leaks over the last 9 years. The info graphic identifies breaches by amount of information stolen, type of organization that was breached, year of theft, and the sensitivity of information lost or stolen. An intriguing upshot: By showing major breaches over time, the info graphic illustrates how internet use has changed over the past decade. AOL had the first major breach in 2004, healthcare providers dominate leaks around 2009, and gaming companies had the major data losses in 2012.



Data Encryption and Decryption

Encryption in earlier times was the use of a code word to make the information useless to the unauthorized person. In the electronic world, it is the act of altering electronic data into an unreadable state by using algorithms and ciphers so as to keep the integrity of the data intact or in simpler terms , to make it secure. Before encryption entered the mainstream public, it was used by military organizations and governments to keep the sensitive information from falling into the wrong hands. Now with the advent of internet, the mainstream public has also entered the domain, thus making data security a widespread necessity.

Thus the process of concealing information so as to hide the subject matter is encryption. It is done using a key. The data that we obtain after encryption is called cipher-text. This cipher-text undergoes a process called decryption which is getting back the information in its original form.

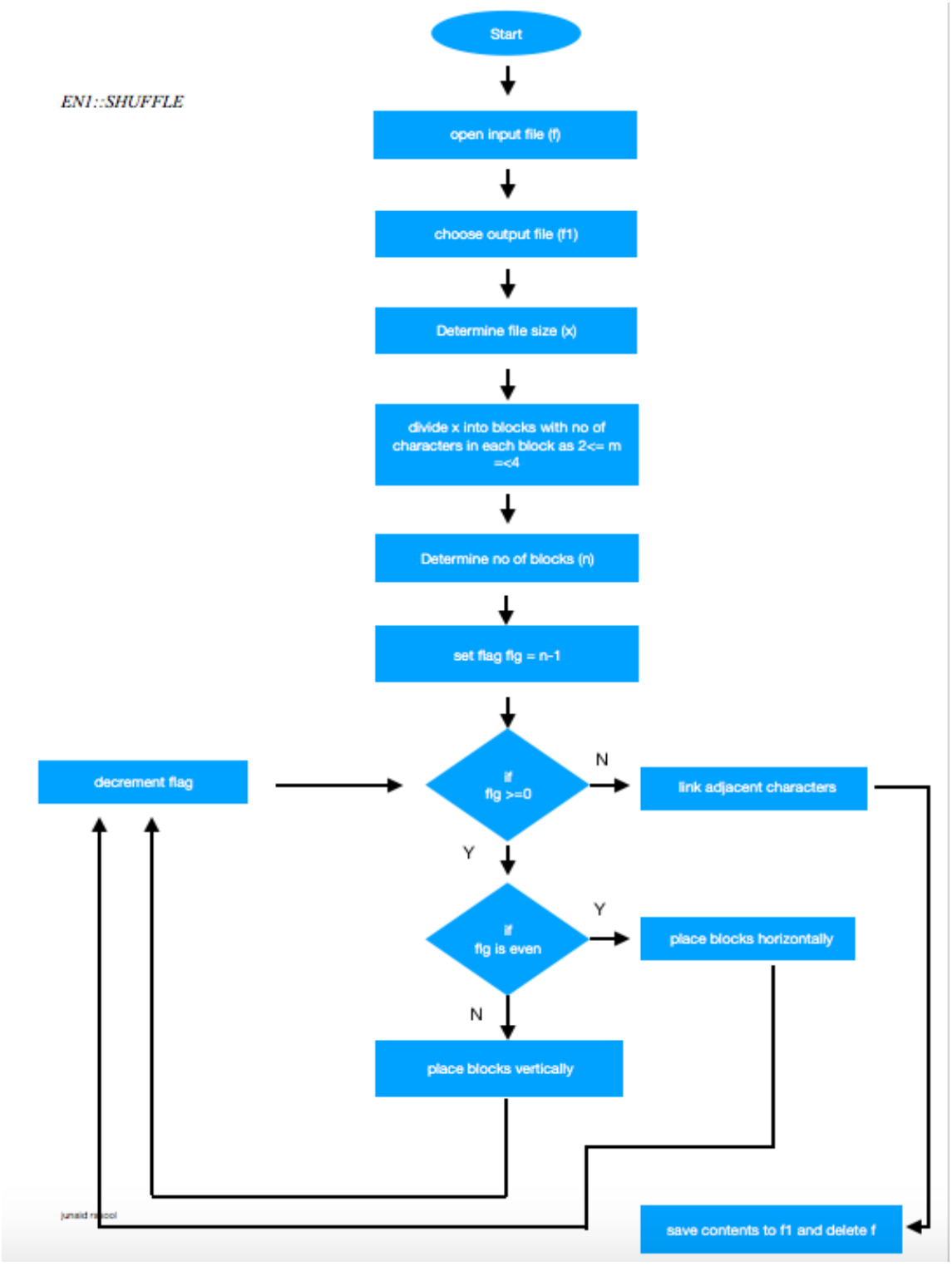


Proposed Algorithm For Encryption

The proposed research aims to perform an encryption technique for an examination data related students which could include the registration details, photograph and marks of a student in a faster and in a memory efficient manner. In this regard below is graphical description of Encryption algorithm. It is a multi-level algorithm. The key step in this algorithm is that the source file is deleted at every level, thus what remains is the cipher-text.

A. MULTI-LEVEL ENCRYPTION

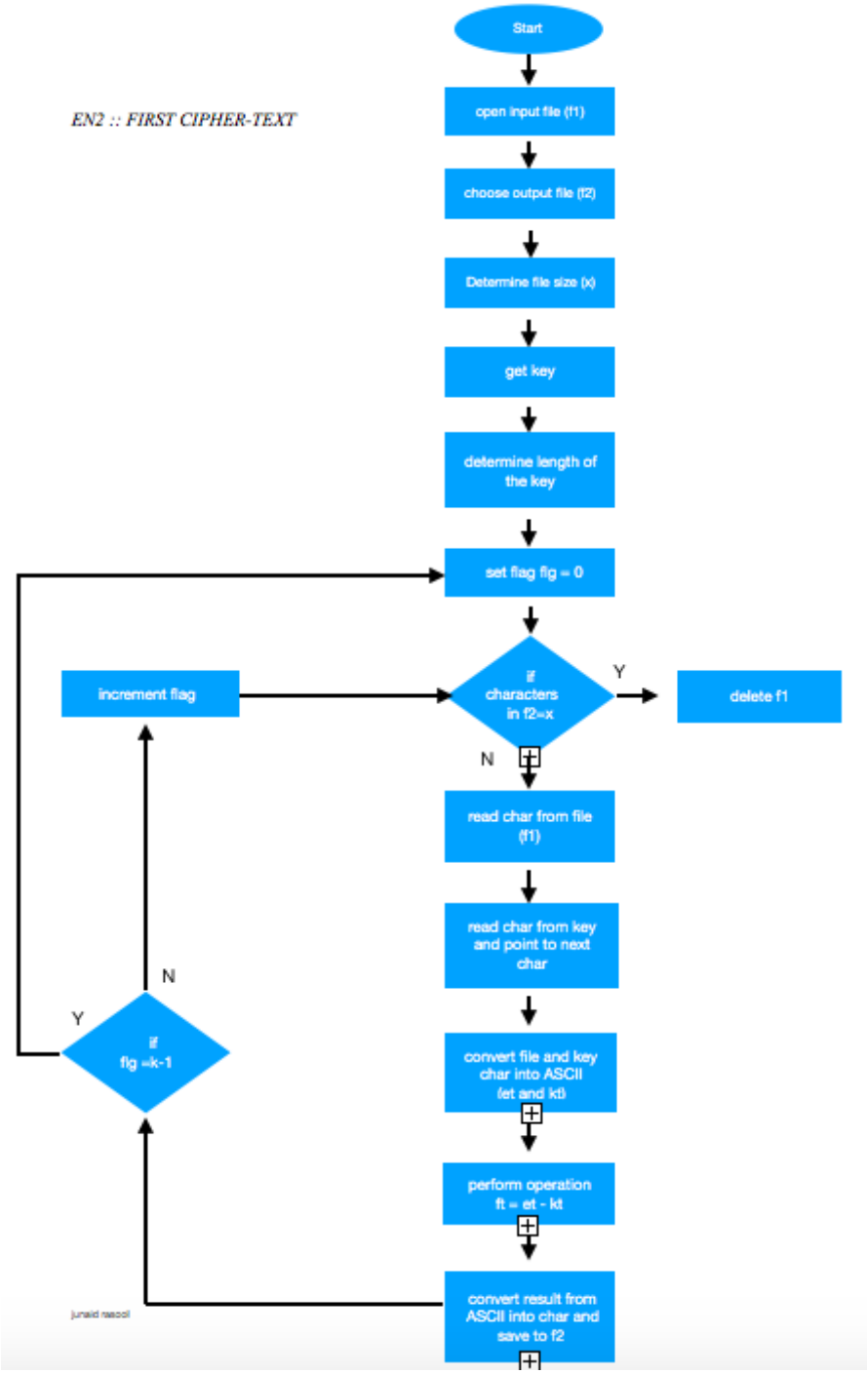
A.1. ENCRYPTION LEVEL 1: SHUFFLE



This is the first level of the encryption. At this level we don't use the key. We determine the size of the file and accordingly divide them into blocks having small chunks of data. These blocks can vary in size i.e different blocks can hold varying no of characters. We determine the no of blocks (n) that have been formed. To simplify the process, we limit the no of characters in each block between 2 and 4. So if the number of characters in any block is denoted by say m , then m has a limit defined by 2<= m =<4 and by obtaining m for each block we determine the no of blocks formed .i.e n. The basic operation that is being done at this level is shuffling these

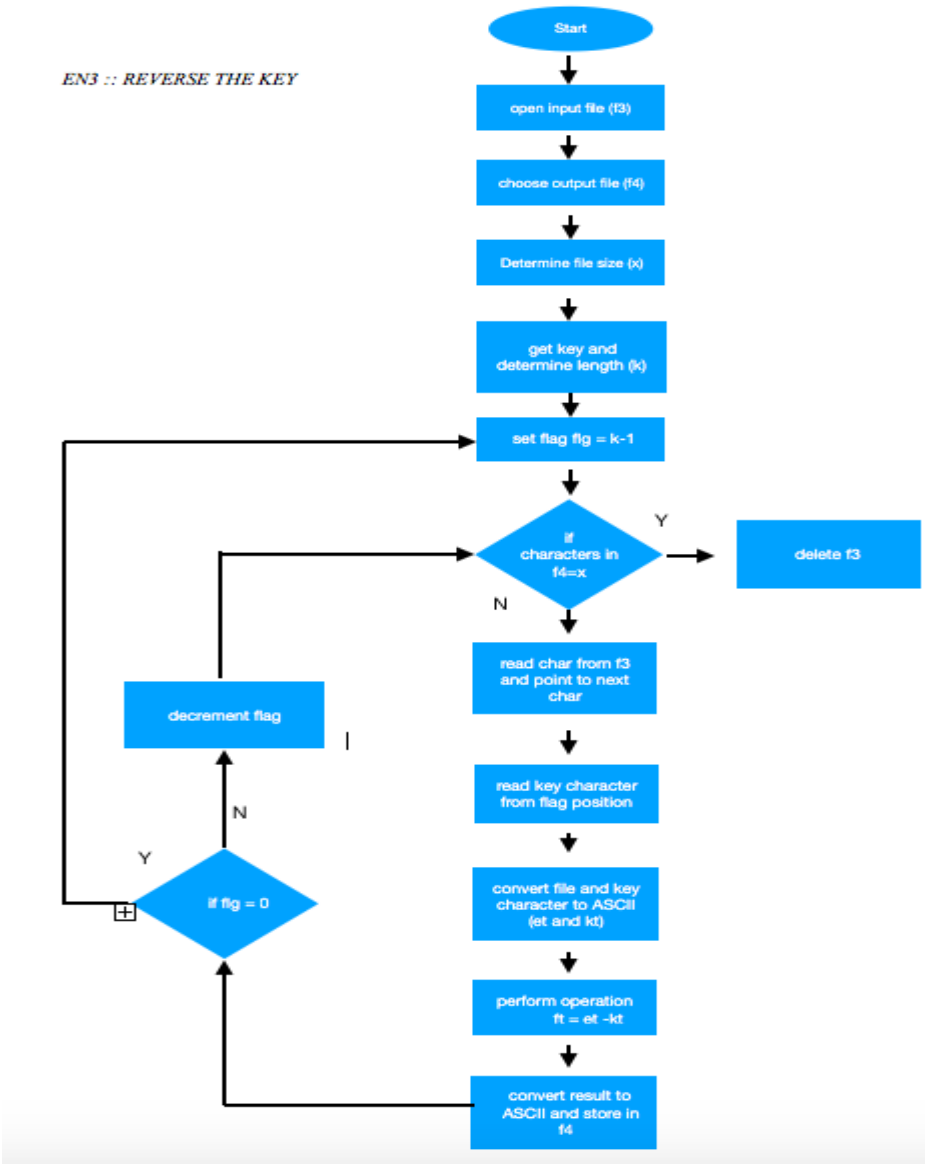
blocks in a specified manner which in turn means shuffling the actual characters in the file. These blocks are numbered starting the index at 0 till n-1. We set the flag position flg= n-1 and start the shuffling. Shuffling is done in a manner that the block placed at even positions of flag are placed horizontally and blocks placed at odd positions of flag are placed vertically. This is done using a loop. Initially the flag is placed at the last block. Thus that block is placed horizontally or vertically depending on whether n is even or odd. Next, the flag is decremented which moves the flag position to n-2 i.e the second last block. If n-2 is even, the block will be placed vertical to the first block. This loop runs till flag position reaches 0. Once the blocks are shuffled, they are concatenated i.e the last character of the first block is linked to the first character of the adjacent block, thus giving it a continuous form. These characters are then saved to another file and the original file is deleted.

A.2. Encryption Level 2: First Cipher-Text



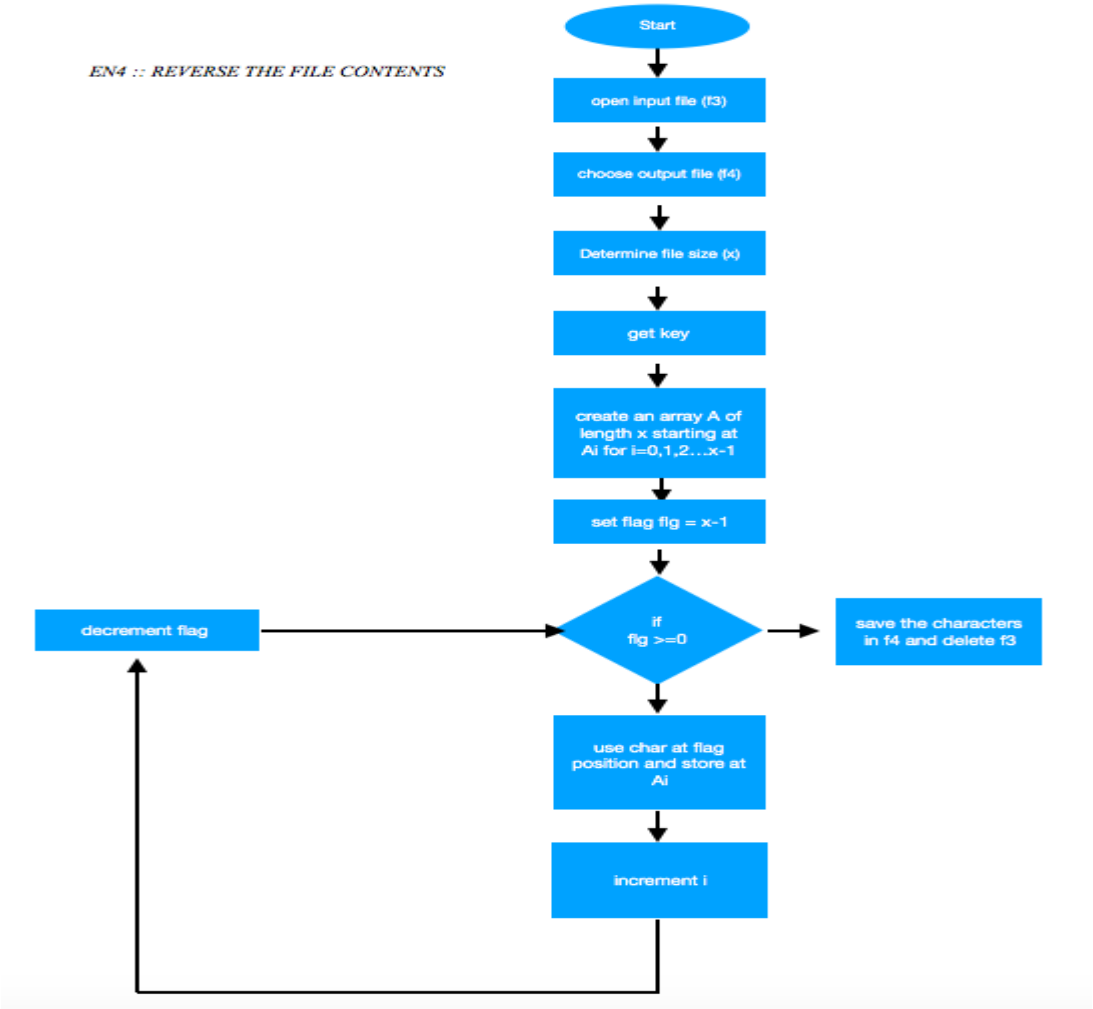
At this level we take the file obtained at the previous level and treat that as the input file. We choose an output file where we would like to save our cipher-text obtained at this level. We determine the size of the file i.e no of characters in the file say (x). We analyses the key used to encrypt the file and determine its size. We set a flag at flg = 0 and run a loop against it. We read the first character from the file (stmp) and convert it into its ASCII (et). Then the next step of the loop reads the key character (ktmp) and converts it into its ASCII (kt) and points to the next character. These 2 ASCII values are then operated upon using a specific arithmetic. The value obtained is then converted back to the character form and then saved in the output file that we chose at the start. The flag is incremented then and the same process runs. When the flag reaches flg=k-1, it is set back to 0 and the process continues. The loop ends when the characters in the output file=x. At this point the original file is deleted.

A.3. Encryption Level 3: Reverse Key



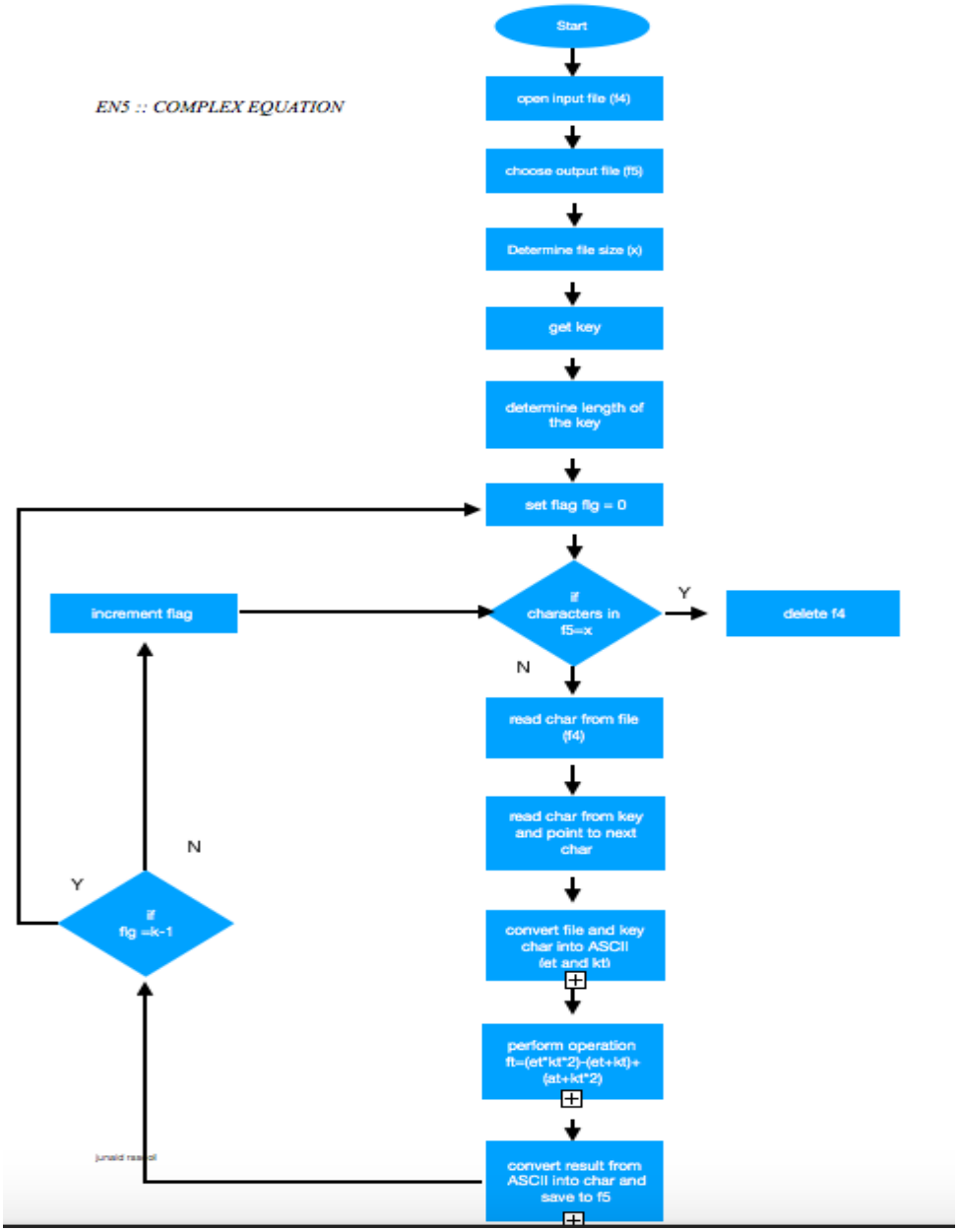
This level runs the same process as EN2 with the only difference of changing the form of the key. The same key is used but in the reverse manner i.e the first character is encrypted using the last character of the key and so on. Firstly the file size (x) is determined and then the key is obtained and the length is determined (k). The main process at this level is reversing the key which is done using a flag at flg = k-1 encrypting the file character with the key character at flag position. A mathematical operation is performed between the ASCII values of file and key character. the result is then converted back to character and saved in output file. The flag is then decremented. This process continues till flg = 0 at which point it is set to flg = k-1 and the process follows the trend. When the characters in the output file reach x, it exits the loop and deletes the input file.

A.4. Encryption Level 4: Reversing File Contents



This level encrypts the contents of the file in yet another manner. Here we don't use the key. We simply reverse the characters in the file i.e we reverse the data stream. This is done using an array. We create an Array (A) of length equal to the length of the file (x). Flag is set at flg = x-1 i.e the last character of the file and the value at that position is stored as first element in the Array. This process continues till all the characters are stored in the array in reverse order. The Array is saved in the output file and the source file is deleted.

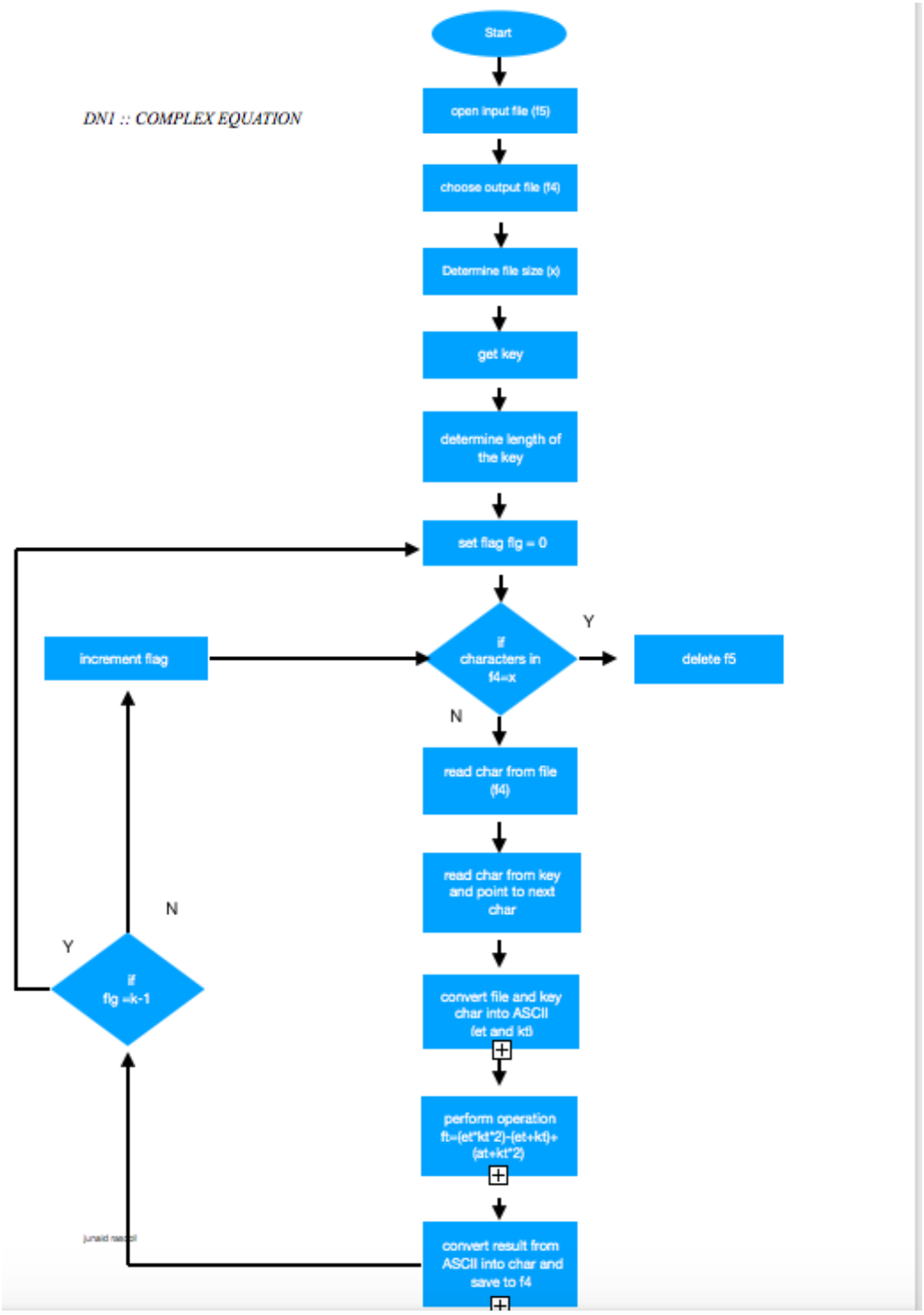
A.5. Encryption Level 5: Complex Equation



At this level we take the file obtained at the previous level and treat that as the input file. We choose an output file where we would like to save our cipher-text obtained at this level. We determine the size of the file i.e no of characters in the file say (x). We analyse the key used to encrypt the file and determine its size. We set a flag at flg = 0 and run a loop against it. We read the first character from the file (stmp) and convert it into its ASCII(et). Then the next step of the loop reads the key character(ktmp) and converts it into its ASCII(kt) and points to the next character. These 2 ASCII values are then operated upon by a complex equation. The value obtained is then converted back to the character form and then saved in the output file that we chose at the start. The flag is incremented then and the same process runs. When the flag reaches flg=k-1, it is set back to 0 and the process continues. The loop ends when the characters in the output file=x. At this point the original file is deleted.

B. Multi-Level Decryption

B.1. Decryption Level 1: Complex Equation

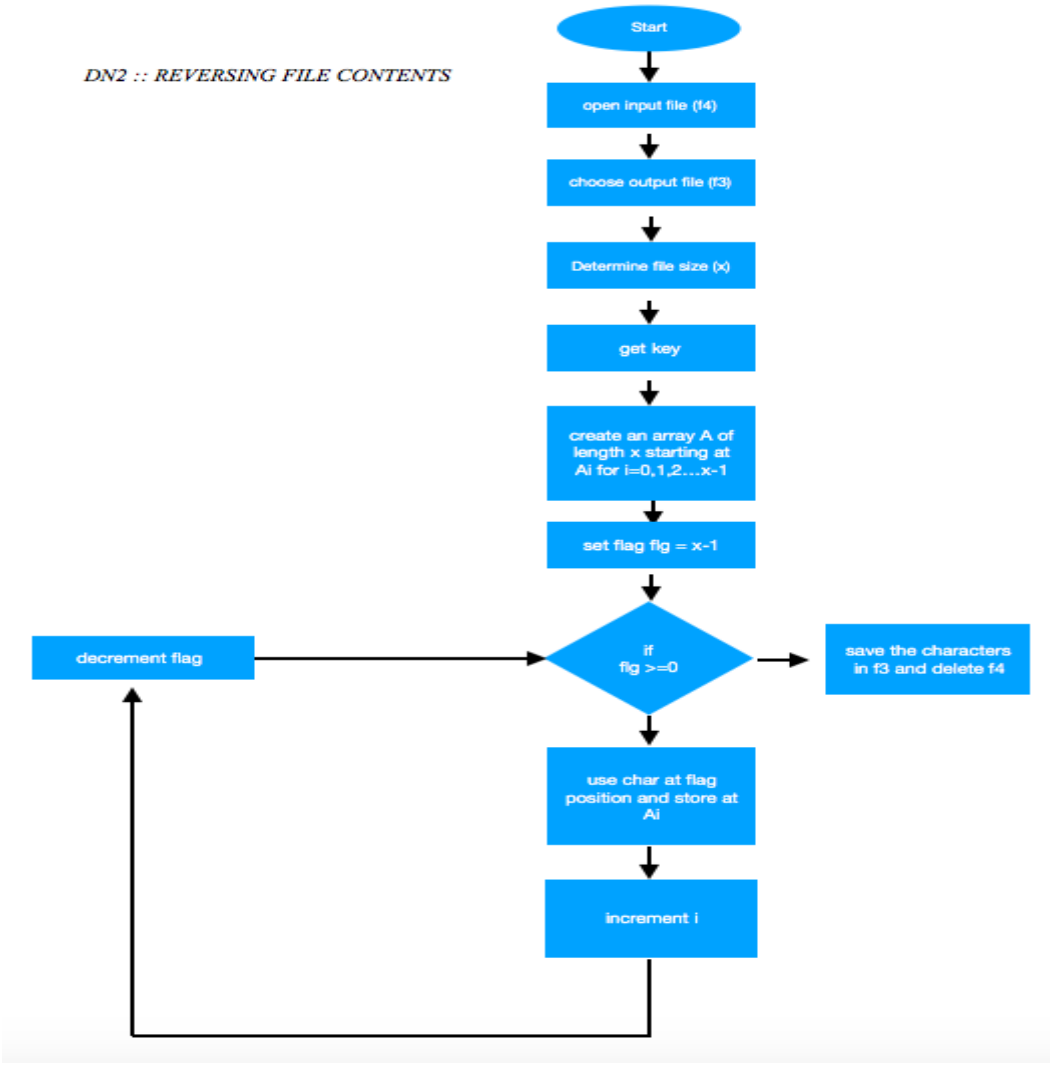


At this level we take the file to be decrypted and treat that as the input file. We choose an output file where we would like to save our data obtained at this level. We determine the size of the file i.e no of characters in the file say (x). We analyse the key used to decrypt the file and determine its size. We set a flag at flg = 0 and run a loop against it. We read the first character from the file (stmp) and convert it into its ASCII(et). Then the next step of the loop reads the key character(ktmp) and converts it into its ASCII(kt) and points to the next character. These 2 ASCII values are then operated upon by a complex equation. The value obtained is then converted back to the character form and then saved in the output file that we chose at the start. The flag is

incremented then and the same process runs. When the flag reaches flg=k-1, it is set back to 0 and the process continues. The loop ends when the characters in the output file=x. At this point the original file is deleted.

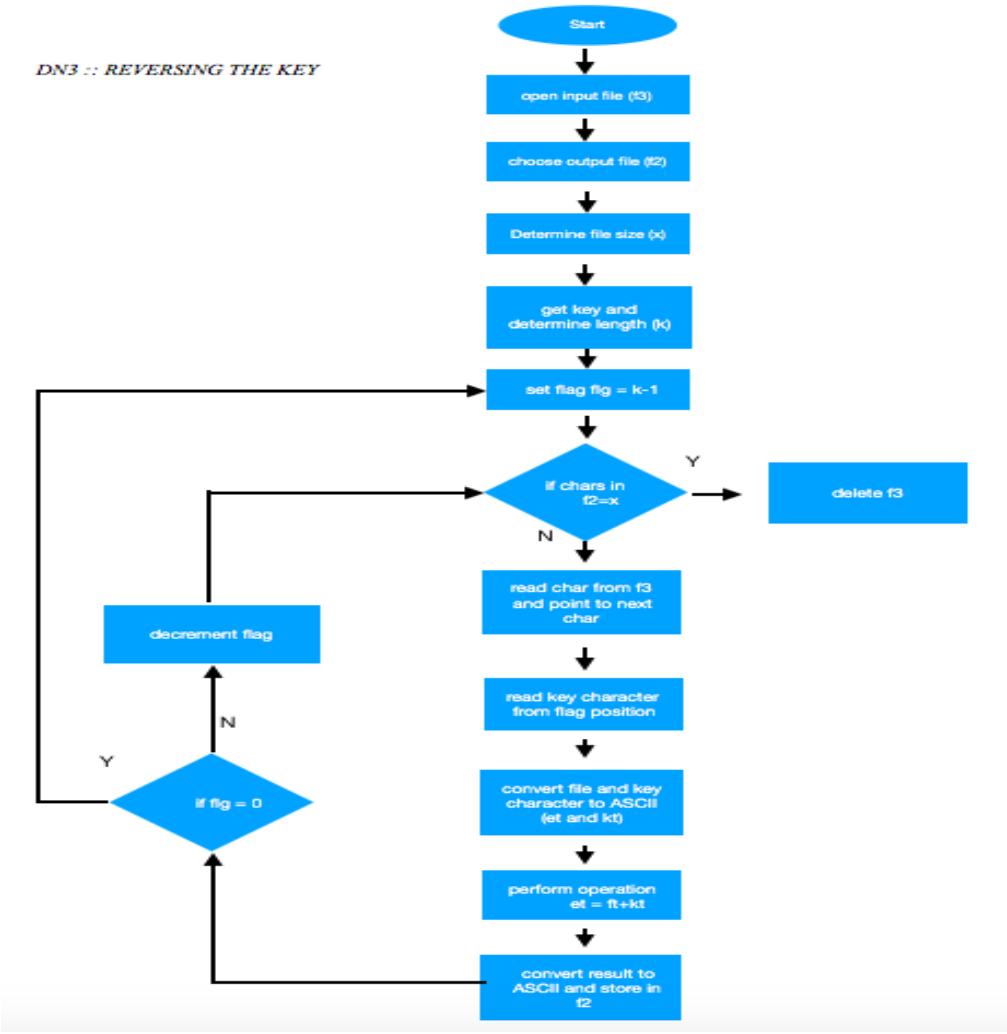
Decryption Level 2: Reversing File Contents

1.



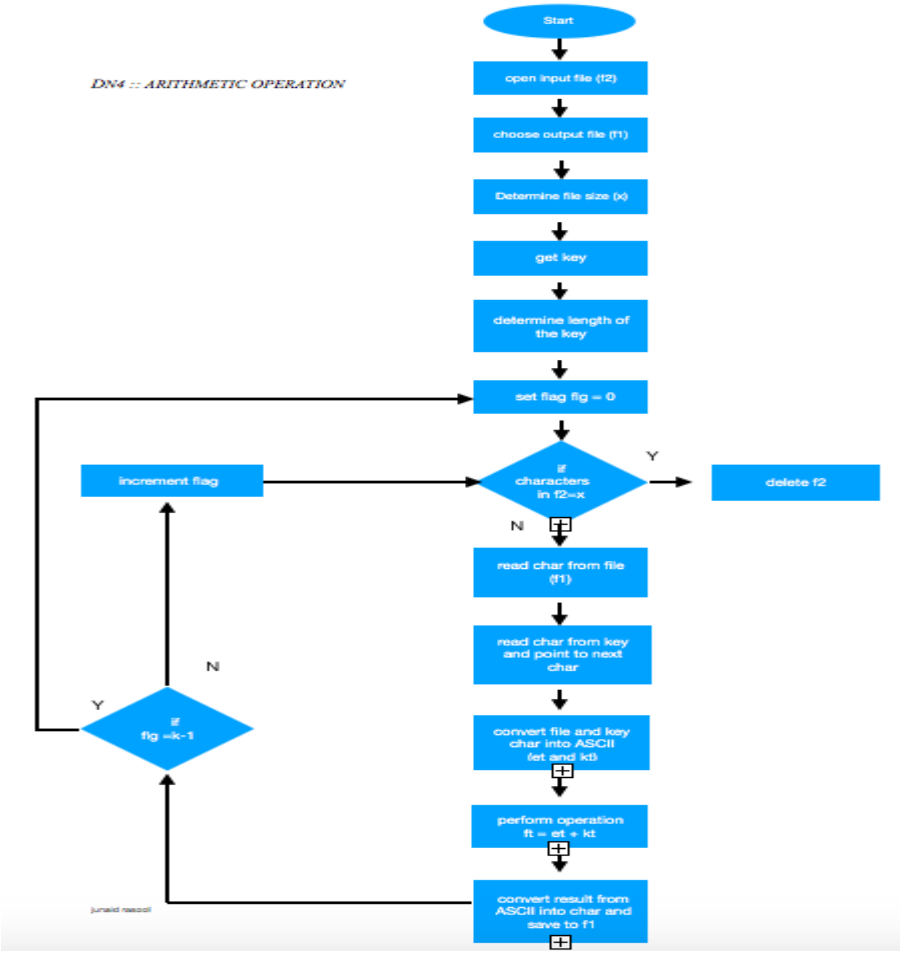
This level decrypts the contents of the file in yet another manner. Here we don't use the key. We simply reverse the characters in the file i.e we reverse the data stream. This is done using an array. We create an Array (A) of length equal to the length of the file (x). Flag is set at flg = x-1 i.e the last character of the file and the value at that position is stored as first element in the Array. This process continues till all the characters are stored in the array in reverse order. The Array is saved in the output file and the source file is deleted.

B.3. Decryption Level 3: Reverse Key



This level runs the same process as DN1 with the only difference of changing the form of the key and the operation used. The same key is used but in the reverse manner i.e the first character is decrypted using the last character of the key and so on. Firstly the file size (x) is determined and then the key is obtained and the length is determined (k). The main process at this level is reversing the key which is done using a flag at flg = k-1 decrypting the file character with the key character at flag position. A mathematical operation is performed between the ASCII values of file and key character. The result is then converted back to character and saved in output file. The flag is then decremented. This process continues till flg = 0 at which point it is set to flg = k-1 and the process follows the trend. When the characters in the output file reach x, it exits the loop and deletes the input file.

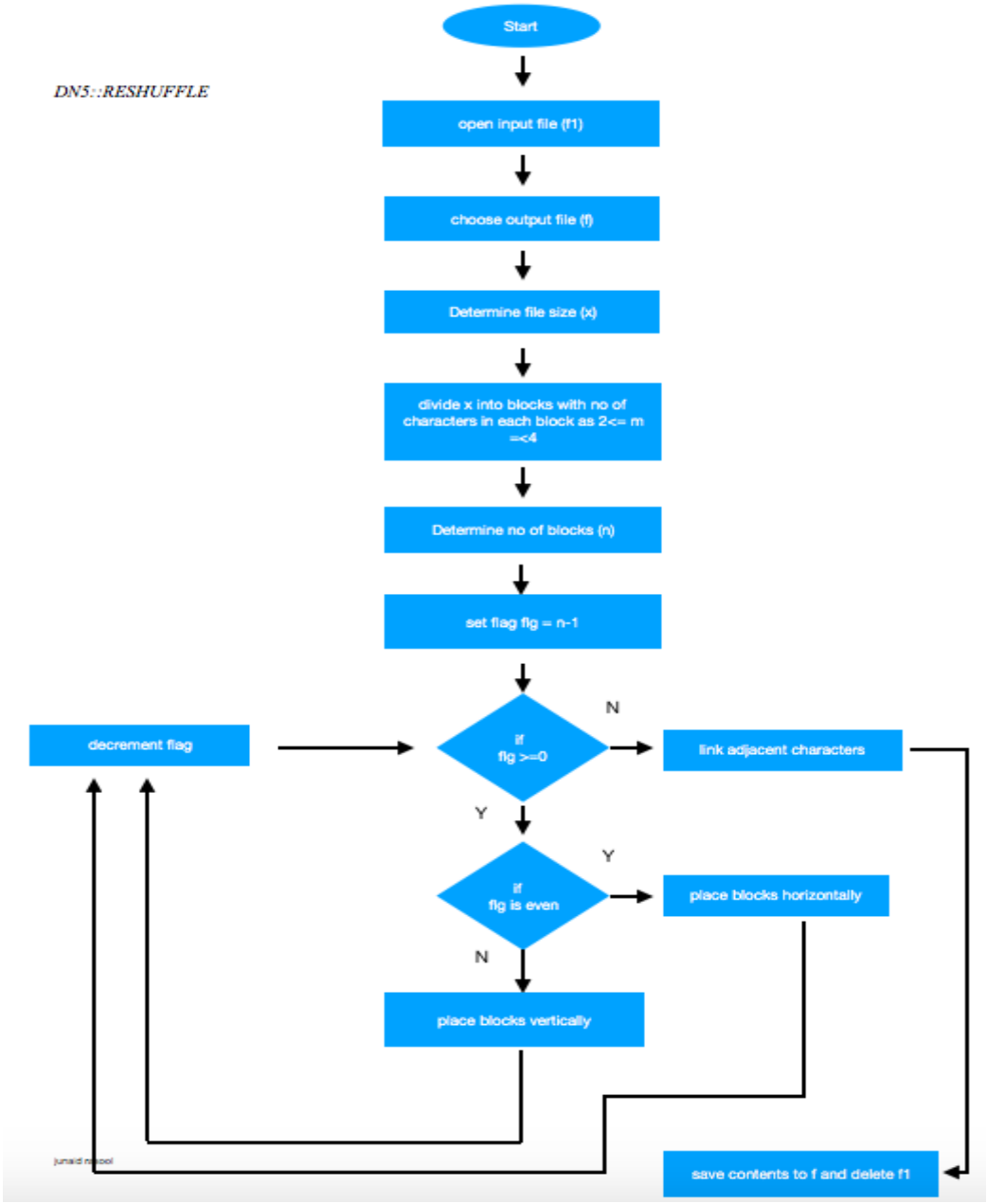
B.4. Decryption Level 4: Arithmetic Operation



At this level we take the file obtained at the previous level and treat that as the input file. We choose an output file where we would like to save our data obtained at this level. We determine the size of the file i.e no of characters in the file say (x). We analyse the key used to encrypt the file and determine its size. We set a flag at flg = 0 and run a loop against it. We read the first character from the file (stmp) and convert it into its ASCII(et). Then the next step of the loop reads the key character(ktmp) and converts it into its ASCII(kt) and points to the next character. These 2 ASCII values are then operated upon using a specific arithmetic. The value obtained is then converted back to the character form and then saved in the output file that we chose at the start.The flag is incremented then and the same process runs. When the flag reaches flg=k-1, it is set back to 0 and the process continues. The loop ends when the characters in the output file=x. At this point the original file is deleted.

B.5. Decryption Level 5 : Reshuffle

This is the final level of the decryption. At this level we don't use the key. We determine the size of the file and accordingly divide them into blocks having small chunks of data. These blocks can vary in size i.e different blocks can hold varying no of characters. We determine the no of blocks(n) that have been formed. To simplify the process, we limit the no of characters in each block between 2 and 4. So if the number of characters in any block is denoted by say m , then m has a limit defined by 2<= m =<4 and by obtaining m for each block we determine the no of blocks formed .i.e n. The basic operation that is being done at this level is reshuffling these blocks in a specified manner which in turn means reshuffling the actual characters in the file. These blocks are numbered starting the index at 0 till n-1. We set the flag position flg= n-1 and start the reshuffling. Reshuffling is done in a manner that the block placed at even positions of flag are placed horizontally and blocks placed at odd positions of flag are placed vertically. This is done using a loop. Initially the flag is placed at the last block. Thus that block is placed horizontally or vertically depending on whether n is even or odd. Next, the flag is decremented which moves the flag position to n-2 i.e the second last block. If n-2 is even, the block will be placed vertical to the first block. This loop runs till flag position reaches 0. Once the blocks are shuffled, they are concatenated i.e the last character of the first block is linked to the first character of the adjacent block, thus giving it a continuous form. These characters are then saved to another file which is the actual plaintext or the original file and the input file to this level is deleted.



II. Conclusion

Attempt has been made to propose, devise and implement next generation security algorithm. Strength lies in the algorithm further key is not stored at any instance of time. However if the key is lost file will become irrelevant and this is where lies the success in data security.

REFERENCES

[1] Dr. Prerna Mahajan & Abhishek Sachdeva, "A Study of Encryption Algorithms AES, DES and RSA for Security" Global Journal of Computer Science and Technology Network, Web & Security Volume 13 Issue 15 Version 1.0 Year 2013

[2] Kale, Karbhari Viswanath," Advances in Computer Vision and Information Technology", IK International Pvt Ltd, 2008 [Book Chapter]

[3] Butt, Muheet Ahmed. "Implementing ICT Practices of Effective Tourism Management: A Case Study." Journal of Global Research in Computer Science4.4 (2013): 192-194.

[4] Denning, D. E., Denning, P. J., Schwartz, M.D. 1979. The tracker: a threat to statistical database security. ACM Trans. Database Syst. 4(1): 76-96

[5] Zaman, Majid, S. MK Quadri, and Muheet Ahmed Butt. "Generic Search Optimization for Heterogeneous Data Sources." International Journal of Computer Applications 44.5 (2012): 14-17.

[6] Butt, Muheet Ahmed, and Majid Zaman. "Assessment Model based Data Warehouse: A Qualitative Approach." International Journal of Computer Applications 62.10 (2013).
[7] D.Ferraiolo, R.Chandramouli, R.Kuhn "Role Based Access Control", Artech House, 2003
[8] Butt, Muheet Ahmed, and Majid Zaman. "Assessment Model based Data Warehouse: A Qualitative Approach." International Journal of Computer Applications 62.10 (2013).
[9] S.Oliveira, O. Zaiane "Privacy Preserving Frequent Itemset Mining", Proc. IEEE ICDM Workshop, 2002
[10] MaqboolRao, Nouman, et al. "Distributed Data Warehouse Architecture: An Efficient Priority Allocation Mechanism for Query Formulation."
[11] Zaman, Majid, and Muheet Ahmed Butt. "Warehouse Creator: A Generic Enterprise Solution." International Journal of Engineering Science (IJES) 2.11.
[12] Butt, Muheet Ahmed. "COGNITIVE RADIO NETWORK: SECURITY ENHANCEMENTS." Journal of Global Research in Computer Science 4.2 (2013): 36-41.
[13] Butt, M. A., and M. Zaman. "Data Quality Tools for Data Warehousing: Enterprise Case Study." IOSR Journal of Engineering 3.1 (2013): 75-76.
[14] Butt, Er Muheet Ahmed, and Er Majid Zaman. "Data Quality Tools for Data Warehousing: Enterprise Case Study."
[15] Mr. Gurjeevan Singh, Ashwani Singla, K S Sandha, "Cryptography Algorithm Comparison for Security Enhancement In Wireless Intrusion Detection System", International Journal Of Multidisciplinary Research Vol.1 Issue 4, August 2011, Issn 2231 5780
[16] Zaman, Majid, and Muheet Ahmed Butt. "Enterprise Data Backup & Recovery: A Generic Approach." International Organization of Scientific Research Journal of Engineering (IOSRJEN) (2013): 2278-4721.

Khaitul Abeez "Framework for Multi-Round Security Algorithm" IOSR Journal of Engineering (IOSRJEN), vol. 08, no. 8, 2018, pp. 29-41.