Text File Encryption Empowered with Context Sensitive Grammar Approach

Irfan Abbas[1,2], Shahan Yamin Siddiqui[1,3], Munib Ahmad[1], Hamza Iqbal[1], Waqar Ahmad[1], Muhammad Usama Mubashir[1], Abdul Hannan Khan[1], Bilal Shoaib[1], Muhammad Adnan Khan[3]

[1]School of Computer Science, Minhaj University Lahore, Lahore, Pakistan.
[2]Department of Computer Science & IT, University of Central Punjab, Gujrat, Pakistan.
[3]Department of Computer Science, National College of Business Administration and Economics, Lahore, Pakistan.
[3]Department of Computer Science& IT, Lahore Garrison University, DHA Phase-VI, Lahore, Punjab, Pakistan.

Abstract
Data security is very important for every organization and individual. There is a lot of ways to protect data from hackers. Day by day, new techniques, algorithms, methods, and ideas introduced. There is no way to protect data in the field of theory of computations. The absorbing properties of Context-Free Grammar (CFG) and Context-Sensitive Grammar (CSG) use to identify the grammar of a given set of string, so it is easy to generate all strings using this grammar.
Our method is to develop a Context-Sensitive Grammar-based encryption/ decryption method that is used to encrypt/ decrypt text files using a secret key in the theory of computation. CSG based encryption-decryption is a very powerful and efficient technique for data privacy, encoding, and security. A proposed method consisting of a few stages to encrypt as well as decrypt data using a secret key generated by our algorithm. Before this theory of computation, there was no other way to encode text files or data using grammar.

Key Words:
Context-Sensitive Grammar (CSG), Theory of Computation (ToC), Context-Free Grammar (CFG), Formal Languages Encryption (FLE), Secret key (SK), AES, Grammar, Encryption, cryptosystem, Data Security, Decryption.

1. Introduction

In the current era, data communication is very important. Data is a very important thing for different organizations. We send and receive data from different individuals and organizations. They shared information for different purposes. Without data communication, we can't do anything. If we want to do something, then it is necessary for us to communicate, to send data, to receive data [1]. As time going, data communication is gaining more importance. Transferring a large amount of data is now becoming a point of discussion. We have to face different security issues related to our data. All we need is that we want to transfer our data with proper security from one place to another [2]. We must protect our data from hackers because more and more businesses are going digital these days-from small independent start-ups to big multi-national companies. To determine what type of security policy you need to have in place, you will first need to meet key members of your team and discuss exactly what type of data you collect and store, as well as the security that you currently have in place to protect this information. Don't get lazy by using hack-friendly passwords like the user's name, "12345," "password," "ABCDE" or some similar combination. Weak passwords such as these are the dream of a hacker and therefore there is nothing you can do to guard against a breach. Using strong unique passwords instead, and change them every 45-60 days. Establish your programs and systems so that each employee can only access certain data based on what their job requires. Limiting the number of individuals who can access consumer information and what apps can allow you to keep track of where it is going. All computers in the company should have anti-virus and spyware software installed. Any employee attempting to access data from a mobile device should be equipped with firewall tools etc. You should have at least one basic protection program, but for companies that store sensitive electronic information, such as bank accounts and social security numbers, additional intrusion prevention is recommended [3]. For that purpose, we developed different models, techniques, and algorithms to secure our data. Multiple models, techniques, and algorithms have been made for data security. Every one has its pros and cons. We want a model that can maximize the security of our data [4].
When we come to data security, we can say that "Data Security is a process of protecting data files, accounts and databases on communication by taking control, software and methods that identify and verify the importance of different datasets, their sensitivity[5], regulatory acquiescence requirements and then applying suitable protections methods to secure those resources. We have different areas to encounter e.g communication channels, different encryption/ decryption techniques, a security key method, and trusted third party software. Every area has its complexities [6]. We have to overcome all those complications to ensure data security. When one transfers his highly personal data such as bank account credentials, passwords, and military information, etc., he would never

Manuscript received July 7, 2020
Manuscript revised July 20, 2020

want anyone to see his data. So, we have to work on all the areas to ensure data security [7].

Cryptography is a technique to secure data. The word cryptography comes from two Greek words, Krypto and graphein. Krypto means secret and graphein means writing [8]. So, cryptography means secret writing. It is a technique that converts the plain text into a meaningless ciphertext. This process of cryptography is called encryption. It can also convert meaningless cipher text back to plain text. This process is called decryption. So encryption and decryption are two methods that are performed in cryptography to ensure the security of the data [9]. When a person sends data to someone, the data is encrypted and on the receiving side, data is decrypted. Cipher consists of algorithms that are used for encryption and decryption tasks. In detailed, cipher works with two things, algorithm and a key for each instance. This key is very important. If the keys are not used then the encrypted text is easily breakable is less than useful form. Keys and algorithms collectively ensure that only the receiver could see the data [10]. This is how cryptography works. Different researchers are going on to improve the algorithms of encryption. However, it is very much difficult to find a very strong algorithm that can completely help us in encryption because we have to deal with multiple issues like time complexity, space complexity, accuracy, security, and features of that algorithm [11].

The theory of computation is the field of computer science that deals with theoretical problems. It says how to solve computation problems efficiently and effectively. It uses different algorithms and techniques to solve multiple problems.to compute something. TOA has different models, each model used for different purposes like language processing and compiler design. We use the CSG model that is come from the CFG concept to deal with language processing and text encryption [12].

Computational Intelligence approaches like Neural Network [13], Swarm Intelligence & Evolutionary Computing like Genetic Algorithm, Differential Evolutionary [14, 15, 16] Island DE, Deep Extreme Learning Machine [17] are strong candidate solutions in the field of smart city [18, 19] Smart health wireless communication as well as Cryptography etc. Computational Approaches are hot research area which is also used in cryptography. Many other systems are also used for making a strong and powerful encryption system.

2. Grammar

A grammar is a technique that describes how to make strings from alphabets of languages that are valid or invalid according to the syntax of the language [20]. We can say that grammar is a method that defines the meaning of strings according to the rules of the given language. Every compiler and translator should know about syntax], so grammar is used to identify the meaning of alphabets, strings, and sentences of different languages regarding language syntax. Every grammar should be defined in its context. If the grammar defined in their context, then grammar called Context grammar and if the grammar does not define in their context, it is called Context-Free Grammar (CFG). According to the Chomsky classification, there are four types of grammars are as fellows

- **Recursively enumerable grammar**–detectable using the Turing machine [21].
- **Context-sensitive grammar**–detectable using linear bounded automaton .
- **Context-free grammar**-detectable using pushdown automaton .
- **Regular grammar** detectable using finite state automaton

We just use context-sensitive grammar to encrypt the data or data files.

2.1 Context-Sensitive Grammar

A context-sensitive grammar (CSG) is a formal grammar in which the left sides and right sides of any production rules may be enclosed by the context of the terminal and nonterminal symbols. Thus, the CSG is situated between context-free and unrestricted grammars in the Chomsky hierarchy [21, 22, 23, 24]. Many organization uses the one-way translation of cryptographic algorithms to provide security to their data and files against hackers, but still, it is very useful for the different organization. The one-way translation means that it converts the one value to another but not convert back to the original value. If we have a variable Y and find h(Y) then it is very hard to find Y back. In this paper, we proposed a method that uses CSG to encrypt as well as decrypt text or text files [25, 26, 27].

A CSG is a set of iterative rewriting production rules used to generate patterns of a sentence as well as string/ words. A CSG consists of different symbols (Tuples), the different tuples are G = (N, P, S), where

- **Terminal (TM):** Terminal is a set of symbols that are coming from characters of alphabets of strings/words generated by the given grammar.
- **Non-terminal (NL):** Non-terminal is a set of symbols that are coming from non-terminal as well as terminal.
- **Start Symbol (SL):** The production is allowed if the start symbol exists and does not appear on the other (right) side of the production.

- **Production rule (PL):** productions are the rules that are used for rewriting and replacing terminal/non-terminals with the other terminals/non-terminals. [28, 29, 30]

If a CSG generates any language then the language called context-sensitive language. Now, CSG is defined [31].

Lang (CSG's) = {SET | (SET is an element of (TM)*) ^ (SL) =>Graph^{+} SET) }
Equation 1: Context-Free Grammar

Context-sensitive grammar is much powerful then the Regular Expression (R.E), Finite Automata (FA), Non-deterministic Finite Automata (NFA), Context-Free Grammar (CFG) because Any language generated using CSG can be generated by R.E, F.A, NFA and CFG.
Any language generated using R.E, F.A, NFA, and CFG not necessarily generated by CSG.
So, that CSG is more powerful rather than the CFG and other mention techniques [32].

3. Related work

In the world, there are multiple hacking techniques that hackers use to decrypt important data or thief important data from a different organization. They can use different methods to access different organizations as well as individual personal data for an unethical purpose. The simple and straight-forward method is trying to decrypt a message by using every possible key. Many times, it was rejected but one might accept. At that time you can decrypt required data [33].
After that, there is a new method, in which a secret key is used to protect data. Secret key use on both encryptions as well as decryption but many techniques is given in the world that use to decrypt the text, message, and file without knowing the secret key. Even a skilled cryptanalyst can decipher text or data without knowing the encryption algorithm technique and secret key [34].
Another method is the Advanced Encryption Standard (AES). It was used by the United States government to protect credentials. It is a new and progressive method that is used to save data from hackers and data snatchers. It is a symmetric-key block cipher that can repeatedly use a set of keys of 16 bytes, 24 bytes, and 32 bytes and encrypt, decrypt data in 16 bytes. It is a permutation and substitutions based method. Permutations mean that change the sequence of data and substitutions means that change that data unit with the other data unit. It performs all operations using bytes. It takes data in bytes and encrypts using matrices of 4x4 of 16 bytes . It performs an encrypting method called Add-Round-Key (ARK). The ARK performs all operation byte by byte. The older version of AES is Data Standard Encryption (DES). The encryption process of AES consists of four sub-stages or sub-processes. The four processes are Sub-Bytes, Shift-Rows, Mix-Colom, and Add-Round-Key. The sub-processes are as fellows

- **Sub-Bytes:** Sub-Bytes are also known as Bytes Substitutions. It can take 128 bits of data and convert it into 4x4 metrics
- **Shift-Rows:** All rows are shifted. First is not shifted but second is shifted to the one position forward, third is two positions and fourth is three positions. After all. A new matrix found.
- **Mix- Columns**: All columns are mixed using special mathematical functions or format but it does not perform in the last round. It takes 4 bytes as input and produces 4 bytes as an output. It replaces each byte result with the bytes column.
- **Add-Round-Key:** The considered matrix of 16 bytes XORed to the 16 bytes of the round key. If the is the last round the text is encrypted or decrypted otherwise again perform similar round

Some special operations like addition and multiplications, not a usual operation but the mathematical field operations. The above four operations are called inside a loop that executes Nth round times for the size of a given keyless one. The given Rounds numbers for encryption are 10, 12, and 14 and depend on the key size of 128 bits, 192 bits, and 256 bits [36].
The new AES will be the real theorem for all forms of electronic information encryption is called DES. It is impossible to break AES encrypted data and impossible to decrypt encrypted text without using a brute force algorithm where we can use all possible combinations of 256 bits. The AES data encryption length must be equal or less than 256 but the brute force technique provides the facility to break 256 bits data. So, in front of the brute force technique, the AES method well is failed. We are going to a proposed new method that encrypts data and can't be break using a brute force algorithm.

3.1 Encryption of AES

The encryption process of data using the AES method is consisting of different stages. In this method, we take plain text or data that we want to convert into the ciphertext. After taking data, we convert into to 16 bytes and then make different metrics of the data after that shift all the rows of given metrics using a specific method after that mix all columns of these metrics. Repeat rounds for more data otherwise resulted from data are encrypted. All the encryption process is given below

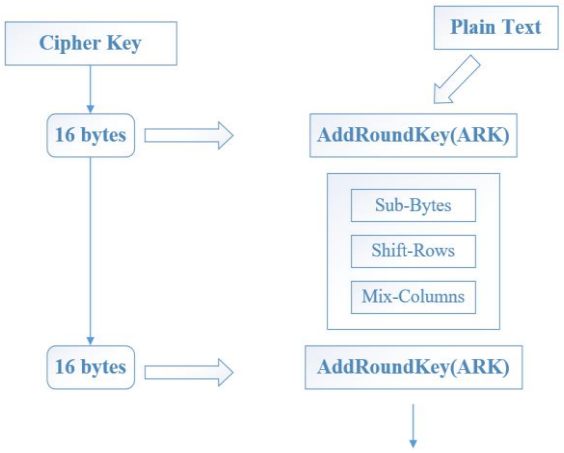

Fig. 1 AES Sub Processes for Encryption

3.2 Decryption of AES

The Decryption process of the AES method is opposite to the encryption. It is also consisting of different phases the same as encryption but these phases are opposite then the encryption. In the process, we take encrypted data and Add-Round-Key on it. After that, we will mix columns. After missing columns, we will shift matrix rows in the same format as were shifted at the time of encryption. We use the same format for rows shifting that we use at the time of encryption. After this, take the given matrix and convert it into 16 bytes or 128 bits. So, this is the actual data that we want for further use. The decryption method is given below

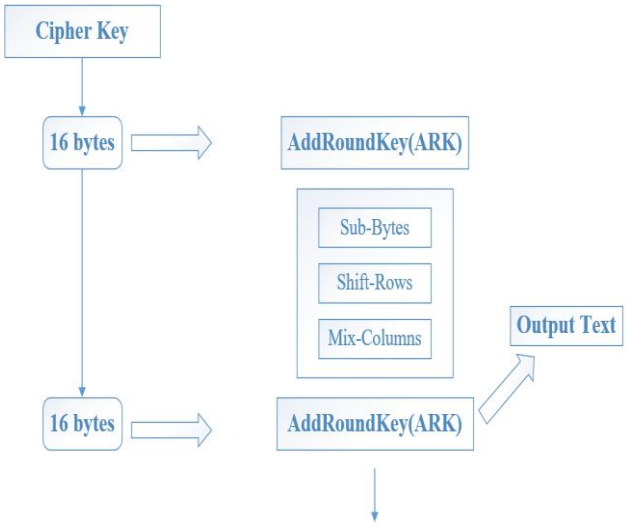

Fig. 2 AES Sub Processes for Decryption

4. Proposed Method

The method that we are going to introduced named cryptosystem use generator to "Generate Random Numbers". It is a very difficult method that encrypts as well as decrypt data. In this method, we use a security key and order of key that we use for plain text encryption. Our method overhead the fixed key method from the user and make more perfect decryption. In our system, we use a single key for both encryption and decryption. The efficiency of our system is defined by the closing key method and encoding text method.

The Context-Sensitive Grammar (CSG) defined the property to generate and verify the different strings form grammar. It is very difficult to find that given strings generated by the specific grammar but using CSG are very easy to identify. The goal of our paper is to provide a CSG based encryption system that is used for encryption as well as decryption that protect our data from virus and security attacks.

In the proposed system, we use a security key for encryption and decryption. By using this key, we convert our text into encrypted data and decrypt it on the other receiver side using a key. The key generation algorithm showed in figure 3. The encryption process as in figure 4.

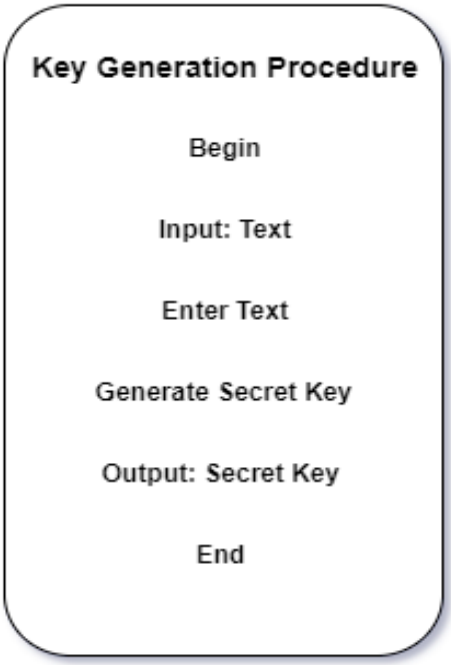

Fig. 3 Key Generation Algorithm

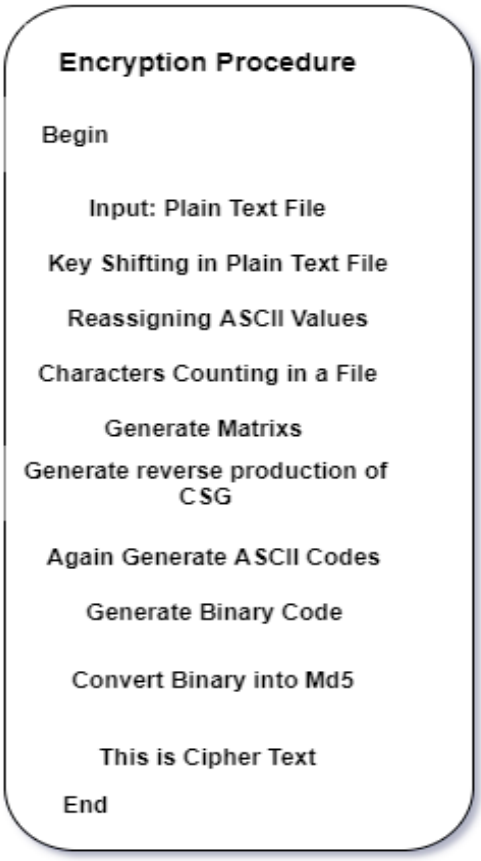

Fig. 3 Encryption Algorithm

In the above algorithm, we just take plain text files that we want to encrypt and a key that is generated by the system. The key generated by the user using a key generator by providing text. When the key is generated then the encryption process will be started.
The Security Key consists of four parts. The first is user-entered text, second is shifting bits, third is the length of user-entered text for further use and the fourth is matrix columns format.

- Take a text file that we want to encrypt.
- Add all shifting bits into the text, after shifting bits, find the ASCII of the resulted text.
- Delete a specific value from ASCII code then again convert it into the text.
- Count the number of text characters and create a matrix of 4x4.
- Apply the matrix columns format that was given in a key. Shifting columns according to the format. After this, convert these matrices into a string.
- Find the reverse CSG of the string.
- Eliminate all the non-terminals
- Take a binary of resulted string.in eight-length

The encryption process of the file consisting of multiple phases. The first phase is the KEY GENERATION are as fellows

4.1 Key Generation

The key generation process is as (Figure 5)
Text file: This is a text to be encrypted
User input text: cipher.
Generated security key: cipher0030063124.
Generated Key consisting of four parts as shown in the diagram.

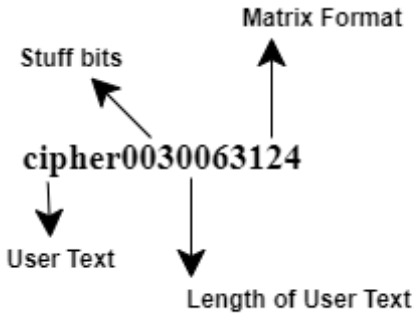

Fig. 4 Key Parts

Cipher is user-entered text, 003 is a stuff bit that indicated after how many characters bits stuffed, 006 is the length of total bits and 3124 is auto-generated matrix format that can be changed during every key generation.

4.2 Stuffing bits

The next phase is to convert all letters into lower cases and stuff bits after the specific length of characters. After converting and stuffing bits, the text file is

$$his\ iis\ ap\ tehxt\ eto\ rbe\ encrypted$$

Equation 2: Text after stuffing bits

4.3 Reassigning ASCII Values

After stuffing bits, find the ASCII value of every bit and subtract specific decimal value (65) from the values of the bits and again convert back into ASCII values. After reassigning, it generates the following string and spaces show in a specific letter ⇒

$$3'(\text{“}2 = ((2 = \Rightarrow / = 3\$'73 = \$3. = 1!\$ = \$ - \text{“}|8/3\$\#$$

Equation 3: Text after reassigning ASCII

4.4 Build the 4x4 Matrices

The next phase is to build 4x4 matrixes. For building 4x4 matrices, first, count the number characters of a file after that take the mod of length of characters with 16 because the total number of characters in one matrix is 16 to find the number of metrics.

Let

$$count = 36$$

$$\sqrt[16]{36}$$

$$36 - (2 * 16) = 4$$

Now the matrixes are

$$Matrix\ Format = (3 \quad 1 \quad 2 \quad 4\ )$$

$$\begin{bmatrix} 3 & ' & (& " \\ 2 & = & (& (\\ 2 & = & \Rightarrow & / \\ = & 3 & \$ & ' \end{bmatrix}$$

$$\begin{bmatrix} 7 & 3 & = & \$ \\ 3 & . & = & 1 \\ ! & \$ & = & \$ \\ - & " & 1 & 8 \end{bmatrix} \begin{bmatrix} / & 3 & \$ & \# \\ < & < & < & < \\ < & < & < & < \\ < & < & < & < \end{bmatrix}$$

By using Matrix format to convert all columns into rows after that the resulted matrix again columns converted into rows, now the result is

$$\begin{bmatrix} = & (& 2 & (\\ = & \Rightarrow & 2 & = \\ . & (& 3 & " \\ 3 & \$ & = & ' \end{bmatrix}$$

$$\begin{bmatrix} . & = & 3 & 1 \\ \$ & = & ! & \$ \\ 3 & = & 7 & \$ \\ " & 1 & - & 8 \end{bmatrix} \begin{bmatrix} < & < & < & < \\ < & < & < & < \\ 3 & \$ & / & \# \\ < & < & < & < \end{bmatrix}$$

After converting all matrices data into string data. The actual data is given below

=(2(=⇒2/'(3"3$='.=31$=!$3=7$"18≪≪≪≪3$/#≪≪

Equation 4: String data

4.5 Take Reverse CSG

In this step, we will take the reverse Context Sensitive Grammar of string is given below

= (**2**(= ⇒ **2**/	â€™ (**3** â€ **3** $ = â€™
A1 → / = A2 2 (	A4 → ' ' A5 = (
= A2 → = ⇒ 2 A3	'A5 → ' $ 3 A6
2 A3 → 2 = (	3 A6 → 3 3 "
. = **31**$ = ! $	**3** = **7**$â€**1** − **8**
A7 → $. A8! =	A10 → 8 3 A11 - =
. A8 → . = 3 A9	3A11 → 3 1 7 A12
3 A9 → 3 $ 1	7 A12 → 7 " $
<<<<<<<<	**3**$/# <<<<
A13 → < < A14 <<	A16 → < 3 A17 < $
< A14 → A15 <<<	3 A17 → 3 < /A18
< A15 → <<<	/ A18 → / < #

We have all the rules in CSG are in the form of a1Aa2→a1Ba2.
After eliminating a1 a2 and non-terminals we get the string

/=2 (⇒ 2=("=($ 3 3" $.!= =3 $/83-=17"$< < < < < < < < < < 3 < $ < / <#

Equation 5: String after Elimination

4.6 Take Reverse CSG

Consider every symbol is ASCII character and convert into binary

00101111 00111101 00110010 00100000 00110010 00111101 00101000 00100111 00100111 00111101 00100000 00100100 00110011 00110011 00100010 00100100 00101110 00100001 00111101 00111101 00110011 00100100 00101111 00111000 00110011 00101101 00111101 00110001 00110111 00100010 00100100 00111100 00111100 00111100 00111100 00111100 00111100 00111100 00111100 00111100 00111100 00110011 00111100 00110011 00100100 00111100 00101111 00111100 00100011

4.7 Take Md5

The last step is to take md5 value of the resulted string

The md5 of given binary code is

caf138d3267de6d7d161be1cdacf14d4

Equation 5: Cipher Text

This is our ciphertext.
Note: Take the reverse process for the decryption process.
It is the same as the encryption but the opposite

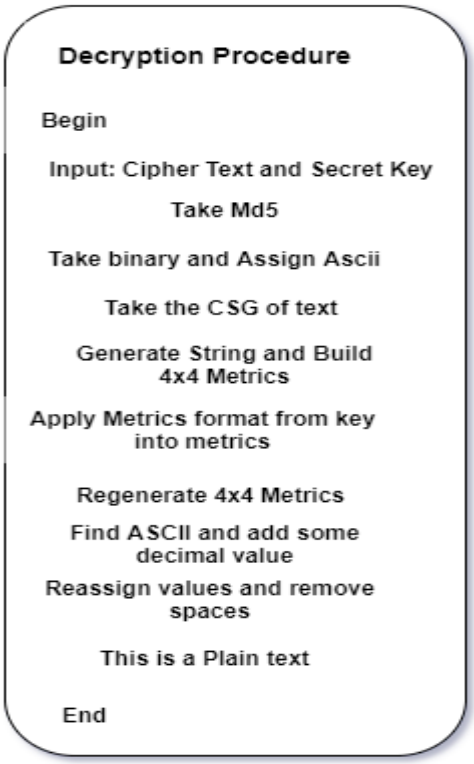

Fig. 5 Decryption Algorithm

5. Design

The grammatical representation of our system for both encryption and decryption are as fellows

5.1 Encryption Process Design

The complete encryption process represented

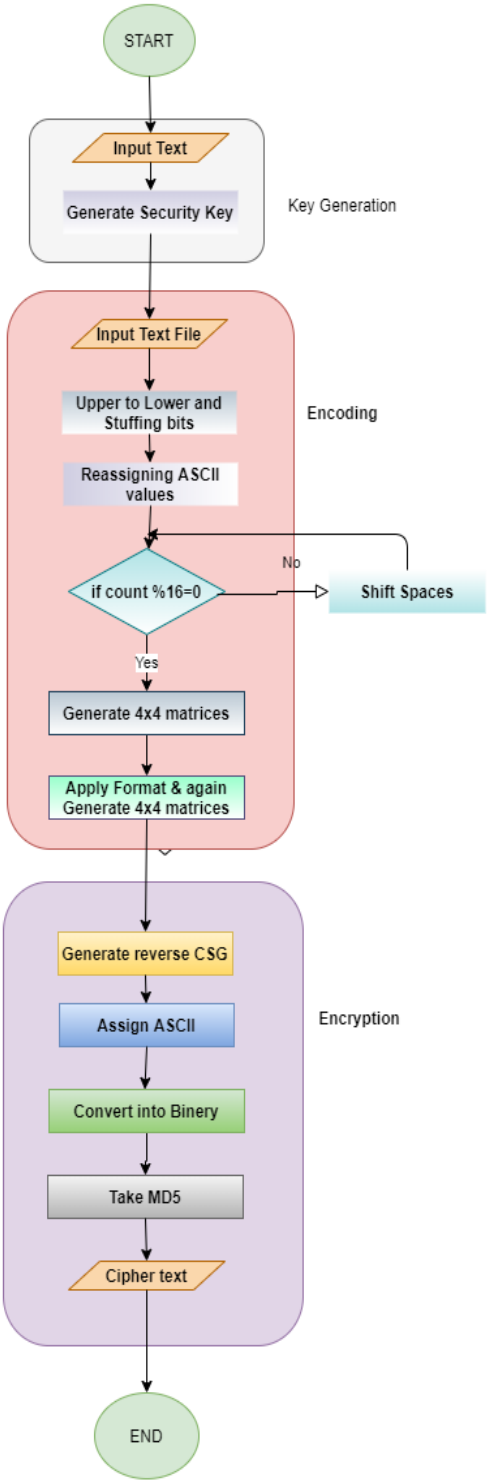

Fig. 7 Encryption Design

5.2 Decryption Design Process

The complete decryption process represented below

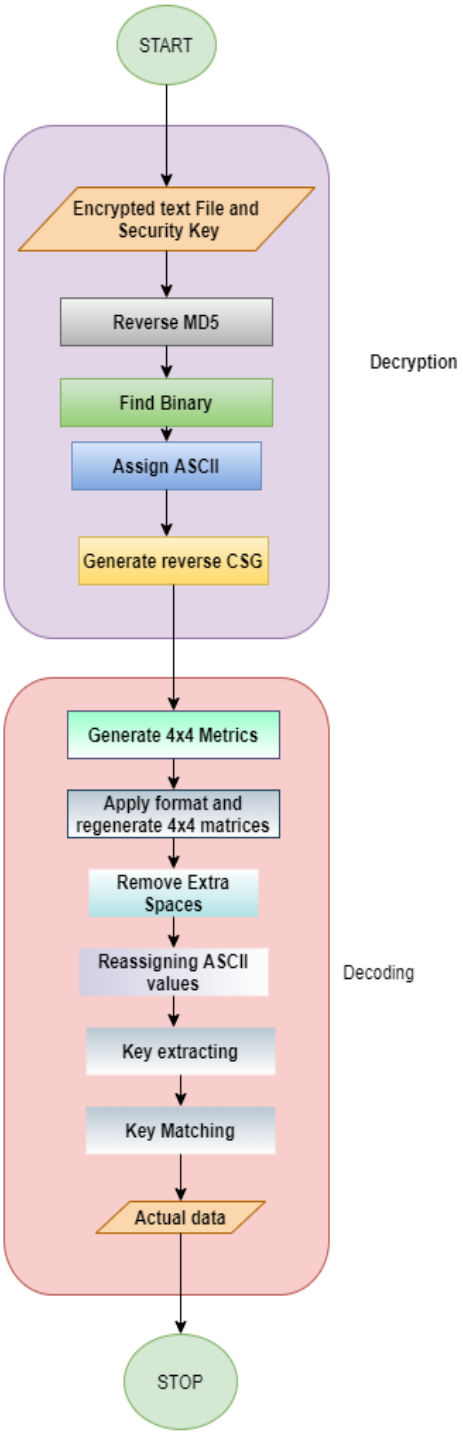

Fig. 6 Decryption Design

6. Security Attacks and Defence

There are multiple attacks in the world that hackers use to decrypt data. The most powerful techniques that are used by hackers are Brute Force, Known text, and Crypt Analysis technique .

6.1 Brute Force Technique

The Brute force is a simple technique that we use for decrypting data. Our system (In theory) has the power to defend against this attack because we are using CSG. It is very difficult to generate a string using CSG and go back using this grammar .

6.2 Known Plaintext Attack

The basic purpose of know plaintext is to find the secret key or find the technique that is used in an algorithm for encryption. So, that in the proposed method we are using a random key generator that can generate random key at every input. So it is very difficult to find the secret key as well as the key generation process [37].

6.3 The Crypt Analysis Technique

The cipher analysis technique is a method that has two final goals. The first goal is to use encrypted text to find the plain text or actual text and the other goal is to use cipher or encrypted text or plain text to find the security key. Both are the most common attacks used by hackers. We use the random key so it is impossible to get back security key and reverse CSG is a very difficult process so it is hard to find the actual text [38].

7. Conclusion

The most perfect and advanced encryption and decryption method proposed in this paper. In this paper, no extra layer required for encryption and decryption. Context-sensitive grammar is used in this system for data security. There is a lot of features of CSG, CSG is very easy to understand, easy to implement, and no extra work. We use CSG for data security. Before this, no system uses grammar for data safety, grammar use for design different programming languages. It is very easy to generate as well as validate data or string but the only disadvantage it is very hard to identify given grammar by the string is generated. Our system does not rely on any other system except md5. It is a very powerful system that is used for the safety of the data files.

References

[1] Hu, C., Li, H., Huo, Y., Xiang, T., & Liao, X. (2016). Secure and efficient data communication protocol for wireless body area networks. IEEE Transactions on Multi-Scale Computing Systems, 2(2), 94-107.

[2] Sundhari, M. R. (2019). Biomedical Image Cryptosystem Based on Visual Cryptography Using Disintegration of Image Blocks. Journal of Medical Imaging and Health Informatics, 9(3), 490-494.

[3] Hai, N. M., Ogawa, M., & Tho, Q. T. (2015, October). Obfuscation code localization based on CFG generation of malware. In International symposium on foundations and practice of security (pp. 229-247). Springer, Cham.

[4] Indrayani, R., Nugroho, H. A., Hidayat, R., & Pratama, I. (2016, October). Increasing the security of MP3 steganography using AES Encryption and MD5 hash function. In 2016 2nd International Conference on Science and Technology-Computer (ICST) (pp. 129-132). IEEE.

[5] Ali, M. N., Khan, M. A., Adeel, M., & Amir, M. (2016). Genetic Algorithm based adaptive Receiver for MC-CDMA system with variation in Mutation Operator. International Journal of Computer Science and Information Security, 14(9), 296.

[6] Luchaup, D., Shrimpton, T., Ristenpart, T., & Jha, S. (2014, November). Formatted encryption beyond regular languages. In Proceedings of the 2014 ACM SIGSAC Conference on Communications Security (pp. 1292-1303).

[7] Lenka, S. R., & Nayak, B. (2014). Enhancing data security in cloud computing using RSA encryption and MD5 algorithm. International Journal of Computer Science Trends and Technology (IJCST), 2(3).

[8] Kumar, P., & Rana, S. B. (2016). Development of modified AES algorithm for data security. Optik, 127(4), 2341-2345.

[9] Alavizadeh, H., Jang-Jaccard, J., & Kim, D. S. (2018, August). Evaluation for combination of shuffle and diversity on moving target defense strategy for cloud computing. In 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE) (pp. 573-578). IEEE.

[10] Shnishah, H. A. H., & Mulvaney, D.(2019, January). Encryption of text file using a user controlled automatically-generated key. In 2019 IEEE 12th International Conference on Global Security, Safety and Sustainability (ICGS3) (pp. 1-6). IEEE.

[11] Singh, G. (2013, December). Modified Vigenere encryption algorithm and its hybrid implementation with Base64 and AES. In 2013 2nd International Conference on Advanced Computing, Networking and Security (pp. 232-237). IEEE.

[12] Qiu, P., Lyu, Y., Zhang, J., Wang, D., & Qu, G. (2017). Control flow integrity based on lightweight encryption architecture. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 37(7), 1358-1369.

[13] Ata, A., Khan, M. A., Abbas, S., Ahmad, G., & Fatima, A. (2019). MODELLING SMART ROAD TRAFFIC CONGESTION CONTROL SYSTEM USING MACHINE LEARNING TECHNIQUES. Neural Network World, 29(2), 99–110. https://doi.org/10.14311/nnw.2019.29.008

[14] Siddiqui, S. Y., Athar, A., Khan, M. A., Abbas, S., Saeed, Y., Khan, M. F., & Hussain, M. (2020). Modelling, Simulation and Optimization of Diagnosis Cardiovascular Disease Using Computational Intelligence Approaches. Journal of Medical Imaging and Health Informatics, 10(5), 1005–1022. https://doi.org/10.1166/jmihi.2020.2996

[15] Ö zkural, E. (2014, August). An application of stochastic context sensitive grammar induction to transfer learning. In International Conference on Artificial General Intelligence (pp. 121-132). Springer, Cham

[16] Shankar, K., & Eswaran, P. (2016). An efficient image encryption technique based on optimized key generation in ECC using genetic algorithm. In Artificial Intelligence and Evolutionary Computations in Engineering Systems (pp. 705-714). Springer, New Delhi.

[17] Naz, N. S., Khan, M. A., Abbas, S., Ather, A., & Saqib, S. (2019). Intelligent routing between capsules empowered with deep extreme machine learning technique. SN Applied Sciences, 2(1). https://doi.org/10.1007/s42452-019-1873-6.

[18] Atta, A., Abbas, S., Khan, M. A., Ahmed, G., & Farooq, U. (2018). An adaptive approach: Smart traffic congestion control system. Journal of King Saud University - Computer and Information Sciences. https://doi.org/10.1016/j.jksuci.2018.10.011.

[19] Yamin Siddiqui, S., Adnan Khan, M., Abbas, S., & Khan, F. (2020). Smart occupancy detection for road traffic parking using deep extreme learning machine. Journal of King Saud University - Computer and Information Sciences. https://doi.org/10.1016/j.jksuci.2020.01.016

[20] Piecha, J., & Staniek, M. (2010, September). The context-sensitive grammar for vehicle movement description. In International Conference on Computer Vision and Graphics (pp. 193-202). Springer, Berlin, Heidelberg.

[21] Brembilla, E., Hopfe, C. J., & Mardaljevic, J. (2018). Influence of input reflectance values on climate-based daylight metrics using sensitivity analysis. Journal of Building Performance Simulation, 11(3), 333-349.

[22] Ding, Z., Zhou, H., Shen, H., & Ge, Q. W. (2014). A public-key cryptosystem based on stochastic Petri net. J Softw, 9(1)

[23] Liu, Y., & Tsyvinski, A. (2018). Risks and returns of cryptocurrency (No. w24877). National Bureau of Economic Research.

[24] Hileman, G., & Rauchs, M. (2017). Global cryptocurrency benchmarking study. Cambridge Centre for Alternative Finance, 33.

[25] Brzozowski, J. (2009). Quotient complexity of regular languages. arXiv preprint arXiv:0907.4547.

[26] Sir, M. Y., Dundar, B., Steege, L. M. B., & Pasupathy, K. S. (2015). Nurse–patient assignment models considering patient acuity metrics and nurses' perceived workload. Journal of biomedical informatics, 55, 237-248.

[27] Fenton, N., & Bieman, J. (2015). Software metrics: a rigorous and practical approach. CRC press.

[28] Afarin, R., & Mozaffari, S. (2013, September). Image encryption using genetic algorithm. In 2013 8th Iranian conference on machine vision

[29] Enayatifar, R., Abdullah, A. H., & Isnin, I. F. (2014). Chaos-based image encryption using a hybrid genetic

algorithm and a DNA sequence. Optics and Lasers in Engineering, 56, 83-93.
[30] Putera, A., Siahaan, U., & Rahim, R. (2016). Dynamic Key Matrix of Hill Cipher Using Genetic Algorithm. Int. J. Secur. Its Appl, 10(8), 173-180.
[31] Tang, Z., Xu, S., Ye, D., Wang, J., Zhang, X., & Yu, C. (2019). Real-time reversible data hiding with shifting block histogram of pixel differences in encrypted image. Journal of Real-Time Image Processing, 16(3), 709-724
[32] Shinge, S. R., & Patil, R. (2014). An encryption algorithm based on ASCII value of data. IJCSIT) International Journal of Computer Science and Information Technologies, 5(6), 7232-7234.
[33] Singh, L. D., & Singh, K. M. (2015). Implementation of text encryption using elliptic curve cryptography. Procedia Computer Science, 54, 73-82.
[34] Moradi, A., Shalmani, M. T. M., & Salmasizadeh, M. (2006, October). A generalized method of differential fault attack against AES cryptosystem. In International Workshop on Cryptographic Hardware and Embedded Systems (pp. 91-100). Springer, Berlin, Heidelberg.
[35] Muthanna, S., Kontogiannis, K., Ponnambalam, K., & Stacey, B. (2000, November). A maintainability model for industrial software systems using design level metrics. In Proceedings Seventh Working Conference on Reverse Engineering (pp. 248-256). IEEE.
[36] Gitanjali, J., Jeyanthi, N., Ranichandra, C., & Pounambal, M. (2014, June). ASCII based cryptography using unique id, matrix multiplication and palindrome number. In The 2014 International Symposium on Networks, Computers and Communications (pp. 1-3). IEEE.
[37] Habboush, A. (2018). Multi-level encryption framework. (IJACSA) Int. J. Adv. Comput. Sci. Appl., 9(4), 130-134.
[38] Waters, B. (2012, August). Functional encryption for regular languages. In Annual Cryptology Conference (pp. 218-235). Springer, Berlin, Heidelberg.