iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 6 **Issue:** III **Month of publication:** March 2018

DOI: http://doi.org/10.22214/ijraset.2018.3032

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue III, March 2018- Available at www.ijraset.com

Generic Approach for Encryption using Reverse Context Free Grammar Productions

Aishwarya R Parab[1], Melina Maria Afonso[2]
[1, 2] *Department of Computer Engineering, Goa College of Engineering, Farmagudi-Ponda, Goa*

Abstract: ***The paper introduces a new symmetric key cryptosystem. This technique makes use of context free grammar as it presents an incredible cryptographic property that it is easy to generate and validate strings from a given grammar; however it is hard to identify a grammar given only the strings generated by it. The proposed idea consists of four modules: Encode, Encrypt, Decrypt and Decode. Firstly the file is encoded using the encoding techniques to obtain an intermediate text. The intermediate text is then encrypted to get the cipher text file using context free grammar along with the secret key. The secret key generated using random number generation algorithm. At the receiver side, the cipher text file is then decrypted using context free grammar followed by decoding techniques to obtain the original file.***
Index Terms: Grammar, Cryptosystem, Symmetric, CFG, Encryption, Decryption.

I. INTRODUCTION

Data security is a demanding issue of data communications today that touches many areas such as secure communication channel, strong data encryption technique and trusted third party to maintain the database. The rapid development in information technology, the secure transmission
of confidential data together with this gets a great deal of attention. The straight methods of encryption can only maintain the data security. The information can be accessed by an unauthorized user for malicious purpose. Therefore, it is necessary to apply effective encryption/decryption methods to enhance data security [1]. Cryptographically strong or strong cryptography are general terms applied to any cryptographic systems or mechanisms that are considered highly resistant to cryptanalysis.
Transmission of sensitive data over the communication channel has emphasized the need for fast and secure digital communication network to achieve the requirements for secrecy, integrity and non-reproduction of the exchanged information. Cryptography provides a method for securing and authenticating the transmission of information over secure channels. It enables us to store sensitive information or send it across insecure network so that unauthorized persons cannot read it.
Cryptography refers to encryption that is the process of converting plaintext into unintelligible cipher text Decryption is the reverse, moving from unintelligible cipher text to plaintext. A cipher is a pair of algorithms which creates the encryption and the reverse that is the decryption. The exhaustive operation of a cipher is controlled both by the algorithm and, in each instance; by a key .This is a secret parameter for a specific message exchange context. Keys are considered important, as ciphers without variable keys are trivially breakable and therefore less than useful for most purposes. Traditionally, ciphers were often used for encryption or decryption, without supplementary procedures such as authentication or integrity checks. New technologies and new applications bring new threats and, with the ever-increasing growth of data communication, the need for security and privacy has become a necessity.

II. GRAMMAR

A. Context-Free Grammars

Many cryptographic algorithms use one-way functions to provide their security against adversaries, but still be useful for authorized parties. A one-way function is a function that given x, it is easy to find f(x). However, given f(x) it is hard to find x. An algorithm that uses context free grammars is proposed in this paper A CFG consists of the following components:

1) A set of *terminal symbols*, which are the characters of the alphabet that appear in the strings generated by the grammar.
2) A set of *nonterminal symbols*, which are placeholders for patterns of terminal symbols that can be generated by the nonterminal symbols.

3) A set of *productions*, which are rules for replacing (or rewriting) nonterminal symbols (on the left side of the production) in a string with other nonterminal or terminal symbols (on the right side of the production).

4) A *start symbol*, which is a special nonterminal symbol that appears in the initial string generated by the grammar. Context-free grammars are strictly more powerful than regular expressions.

5) Any language that can be generated using regular expressions can be generated by a context-free grammar. There are languages that can be generated by a context-free grammar that cannot be generated by any regular expression.

III. CURRENT SYSTEM

The most straight-forward attack on an encrypted message is simply to attempt to decrypt the message with every possible key. Most of these attempts will fail. But one might work. At which point you can decrypt the message.

Most encryption algorithms can be defeated by using a combination of sophisticated mathematics and computing power. The results are that many encrypted messages can be deciphered without knowing the key. A skilled cryptanalyst can sometimes decipher encrypted text without even knowing the encryption algorithm.

AES is a cryptographic algorithm that can be used to protect electronic data. Specifically, AES is an iterative, symmetric-key block cipher that can use keys of 128, 192, and 256 bits, and encrypts and decrypts data in blocks of 128 bits(16 bytes). Unlike public-key ciphers, which use a pair of keys, symmetric-key ciphers use the same key to encrypt and decrypt data. Encrypted data returned by block ciphers have the same number of bits that the input data had. Iterative ciphers use a loop structure that repeatedly performs permutations and substitutions of the input data[6].

AES is the successor to the older Data Encryption Standard (DES).The AES algorithm is based on permutations and substitutions. Permutations are rearrangements of data, and substitutions replace one unit of data with another. AES performs permutations and substitutions using several different techniques

The AES encryption routine begins by copying the 16-byte input array into a 4×4 byte matrix named State. The encryption algorithm performs a preliminary processing step that's called AddRoundKey in the specification. AddRoundKey performs a byte-by-byte XOR operation on the State matrix using the first four rows of the key schedule, and XORs input State[r,c] with round keys table w[c,r].

The main loop of the AES encryption algorithm performs four different operations on the State matrix, called Sub Bytes, Shift Rows, Mix Columns, and AddRoundKey in the specification. The AddRoundKey operation is the same as the preliminary Add Round Key except that each time AddRoundKey is called; the next four rows of the key schedule are used. The SubBytes routine is a substitution operation that takes each byte in the State matrix and substitutes a new byte determined by the Sbox table. ShiftRows is a permutation operation that rotates bytes in the State matrix to the left. The Mix Columns operation is a substitution operation that is the trickiest part of the AES algorithm to understand. It replaces each byte with the result of mathematical field additions and multiplications of values in the byte's column.

The addition and multiplication are special mathematical field operations, not the usual addition and multiplication on integers. The four operations Sub Bytes, Shift Rows, Mix Columns, and Add Round Key are called inside a loop that executes Nr times—the number of rounds for a given key size, less 1. The number of rounds that the encryption algorithm uses is 10, 12, or 14 and depends on whether the seed key size is 128, 192, or 256 bits [6].

The new AES will certainly become the de facto standard for encrypting all forms of electronic information, replacing DES. AES-encrypted data is unbreakable in the sense that no known cryptanalysis attack can decrypt the AES cipher text without using a brute-force search through all possible 256-bit keys.

IV. PROPOSED SYSTEM

The paper introduces a new technique for encrypting and decrypting files. The proposed algorithm is a theoretical study based on the context free grammar which provides the security to the system. The algorithm makes cryptanalysis even more difficult because of the use of “Random Number Generator” function which further decides order of encryption rounds and keys to be used to encrypt the plain text. This eliminates the overhead of defining a fixed key by the user and makes algorithm even secure. With secret key cryptography, a single key is used for both encryption and decryption. The key selection mechanism and the encoding method express the efficiency of the cipher text generated.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue III, March 2018- Available at www.ijraset.com

Context free grammars present the desirable cryptographic property that it is easy to generate and validate strings from a given grammar; however it is hard to identify a grammar given only the strings generated by it [3]. The project aims at developing a context free grammar based cryptosystem that will encrypt a file to protect it from various security attacks.

This cryptosystem will use a symmetric algorithm that will have a secret key. The text file will be converted into a cipher text which will be sent to the receiver who will decrypt it.

```
Procedure Key Generation ()
Input: text
Output: secret key
Begin
Enter text
Generate secret key
End
```

Figure 1: Algorithm for key generation

```
Procedure Encryption()
Input: plain text file, secret key
Output: ciphertext file
Begin
Generate Secret key
Add key to plaintext file
Generate matrices
Generate reverse productions
Generate ASCII and Binary
Generate Ciphertext
End
```

Figure 2: Algorithm for encryption

```
Procedure Decryption()
Input: Ciphertext file, secret key
Output: plain text file
Begin
Generate ASCII and Binary
Generate reverse productions
Generate matrices
Key extraction
Key matching
If secret key = = key in text file
Display plain text
Else
Display garbage value
End
```

Figure 3: Algorithm for decryption

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue III, March 2018- Available at www.ijraset.com

A. Advantages of Proposed System

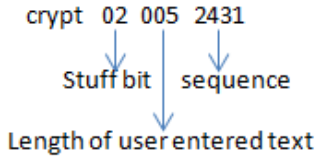
One of the advantages of the proposed system is the time complexity. The proposed algorithm is faster as compared to AES as it does not make use of a round function. AES makes use of a loop structure that repeatedly performs permutations and substitutions of the input data. Since AES works on a fixed block size, and takes approximately the same size independent of the input, thus its complexity is O(1). Another advantage of the proposed algorithm is irreversibility. The use of context free grammar makes the algorithm even more robust because of its cryptographic property. Even if the attacker gets the context free grammar productions it will be very difficult to obtain the grammar, this provides added security to the algorithm. The use of random number generator for the generation of secret key is another advantage of the proposed algorithm. Therefore a user will not have to define a fixed key each times it wants to encrypt a message. The proposed algorithm is a theoretical study; the experimental results will be compared on implementing the new technique.

B. Comparison With the Existing System

Most of the existing cryptosystems take an input of 128 bit block of plaintext including AES and make use of feistel network for the encryption process. The proposed algorithm does not make use of a round function. More the number of rounds, more secure the system but also more inefficient and slow encryption and decryption. AES makes use of block cipher which requires more memory. A small error in one symbol may corrupt the entire block. The proposed algorithm makes use of stream cipher which is more efficient than block cipher.

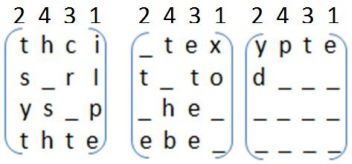
C. Example

Considering an example
Plain text file: this is the text to be encrypted
User entered text: crypt
Generated key (secret key): crypt020052431
The first level is Key Generation where the following processes will occur:
-user will enter text
-the secret key will be generated
The secret key is divided into 4 parts

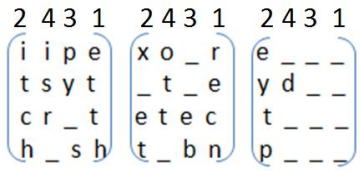


02 in the secret key indicates after how many places to stuff the key, 005 is the length of the user entered text, 2431 is he matrix sequence which is randomly generated.
The next level is ENCODE where the following processes will occur:
The user will input the text file
Adding the key into the file
count the number of characters in the text file
generate 4 X 4 matrices
After stuffing the key we get:
Thc isr iy sp tth e text to be encrypted
It wll then count the number of characters in the text file
Let ‘c’ be the count
c = 37
Now compute c mod 16
37 mod 16 = 5
16-5=11
Thus, 11 padding characters will be added in the text file.

The next step is to generate the 4 x 4 matrices:



(Note: "_" is used to indicate the padding character in the above matrices)
Shuffling of matrices:



Hence we get the string:
etthitchpy_sisr_recnx_et__ebott_____eytp_____d__
The third level is the ENCRYPT level which begins by generating reverse productions using context free grammar (CFG)

etthitch	py_sisr_	recnx_et
A1->heA2ct	A4->_pA5 ry	A7-> trA8ee
A2-> ttA3ih	A5->s_A6is	A8-> _eA9xh
A3-> E	A6-> E	A9-> E
__ebott_	____eytp	_____d__
A10->__A11t_	A13-> p_A14t_	A16-> __A17__
A11-> teA12 ob	A14-> y_ A15e_	A17-> d_ A18__
A12-> E	A15-> E	A18-> E

After eliminating the non-terminals we get the string:
hectttih_prys_istree_exh__t_teobp_t_y_e_____d___
The algorithm then generates the ASCII and binary values of each character in the text file.
Therefore the cipher text file will contain all 0's and 1's. :
01100010010000011111101001100010011000110100110001111001010110100111100101101100011000010111100101001101011000010110001001100011010111110100010010110111011110000011011010100000011111101011110000110001101010111010001010100010101111000011110010111100001000100011110000111100001111001011110000111100101111001011110010111100101111001111000

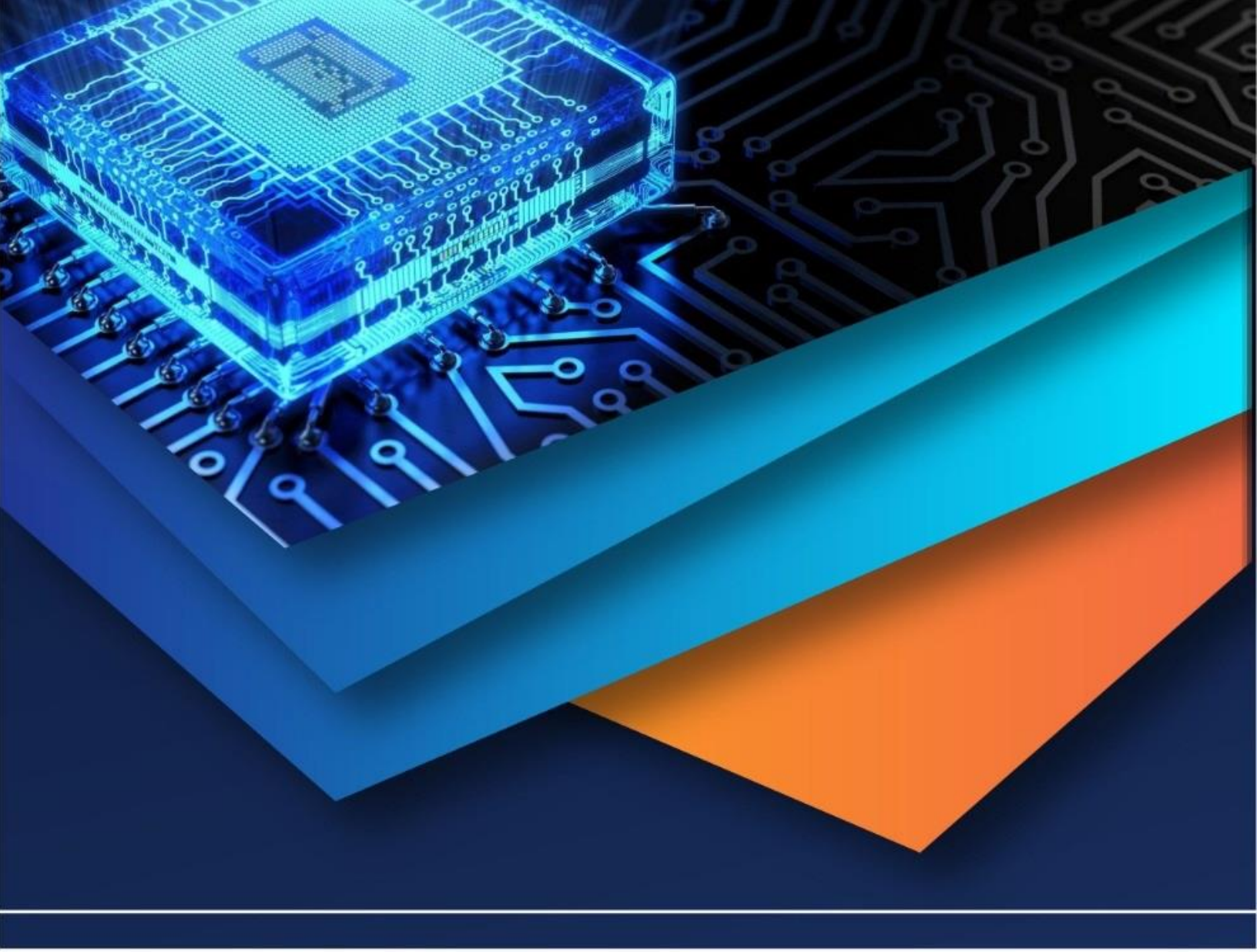
This cipher text file will then be sent to the receiver who will decrypt it using the same secret key.
The decryption process consists of the DECRYPT and the DECODE level which is exactly opposite to encryption except that key matching will occur at the end of the algorithm. If the key is matched only then the plain text will be displayed to the receiver.

V. CONCLUSION

A powerful cryptosystem based on Context free grammar has been proposed in this paper. The system discussed provides security without requiring additional layer of encryption like SSL and also it does not rely on any other cryptographic protocol. The salient features of the proposed algorithm include three step protocol, no large overheads, user friendliness and independent of any other cryptographic protocol. This paper presents and analyzes the protocol with respect to its robustness against malicious attacks.
The described cryptosystem makes use of interesting issues of context free grammars that until now have only been used to design programming languages. It also makes use of random number generation algorithm for secret key generation. Tests were then conducted to determine, given a string from a language how difficult it is to generate another string which belongs to the same language. As the size of the file increases, percentage of accepted strings generated after breaking the string decreases. Hence the chances of guessing the key and the data in the file become nearly impossible.

REFERENCES

[1] Sumedha Kaushik, Ankur Singhal, "Network Security Using Cryptographic Techniques", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 12, December 2012
[2] Abhishek Singh, Andre L M dos Santos, "Context Free Grammar for the Generation of a One Time Authentication Identity"
[3] Abhishek Singh, Andre L M dos Santos, "Grammar Based Off line Generation of Disposable Credit Card Numbers
[4] Dripto Chatterjee, Joyshree Nath, Suvadeep Dasgupta, Asoke Nath, "A new Symmetric key Cryptography Algorithm using extended MSA method: DJSA symmetric key algorithm", 201
[5] Vikrant M. Adki, Prof. Shubhanand S. Hatkar, " A Survey on Cryptography Techniques", Volume 6, Issue 6, June 201
[6] Raj Jain,"Advanced Encryption Standard (AES)" Washington University in Saint Louis, MO 63130
[7] Sandeep Rao, Dindayal mahto, Danish Ali khan "A Survey on Advanced Encryption Standard" International Journal of Science and Research (IJSR) · January 201
[8] Pierre L'Ecuyer1 ,"Random Number Generation" Departement d'Informatique et de Recherche Op´erationnelle, Universit´e deMontr´eal, C.P. 6128, Canada
[9] Adi A. Maaita, Hamza A. A. Al_Sewadi "Deterministic Random Number Generator Algorithm for Cryptosystem Keys" International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol:9, No:4, 201
[10] Cryptography and Network security 4 ed. William Stallings PEA, ISBN:978-81-7758-774-6
[11] Udit Khanna, Neha Gupta "Modified Symmetric Key Algorithm for Improving Data Security" MIT International Journal of Computer Science and Information Technology, *Vol. 5, No. 2, August 2015.*
[12] Hala Bahjat AbdulWahab, Hala Bahjat AbdulWahab, Nedhal A. Al-Saiyd "Proposed New Algorithm to Generate Cryptography Session Keys Based on CFG and Huffman Code" European Journal of Scientific ResearchISSN 1450-216X Vol.78 No.4 (2012), pp.546-545. 2012.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY