IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 8, August 2016

Selective Gene Exchange (SGEX) Mutation Algorithm as a Data Encryption and Decryption Mechanism

Cris Paulo G. Hate[1], Arnel C. Fajardo[2]

Lecturer, Department of Computer Engineering, Technological Institute of the Philippines, Quezon City, Philippines[1]

Professor, School of Graduate Studies, Technological Institute of the Philippines, Quezon City, Philippines[2]

Abstract: This paper describes the mechanism utilizing the alteration of data at a binary level using a binary exchange algorithm. This is to address the need for a flexible and robust form of encrypted data during transmission over networks. The process covers mathematical conversion of data which crossovers and mutates the binary version of a plain text string, and translates the data without the need for a standardized cipher key. The resulting algorithm is also applied to a database driven application, and then exposed to a decryption tool. The data will be passed to a functional database, which in turn will be queried and decrypted.

Keywords: Evolution, Genetic, Cryptography, Mutation, Bitwise, Algorithm, Cipher Text

I. INTRODUCTION

Darwin's theory of evolution is a widely held notion that all life is a variation or a descendant from a common ancestor. This theory presumes that the simplest entities adapts to the environment and phenomenon through mutation. These random mutations occur on the organism's genetic code, and are passed onto further generations as time progresses [6].

Evolution can also be applied in computing, particularly to data security. Algorithms mimic the same concept as living organisms, as security threats remain present in the world, a method for protecting information should also be as robust and as reliable [3].

The ability to secure data from external parties, protection from unauthorized access, usage, disclosure, alteration, and other threats is considered as a priority in almost all system existing today[3]. Techniques in securing data involve overall defence, access controls, and cryptography.

The goal of the study is to develop an encryption algorithm, using an evolutionary approach [12]. To attain the said objective, the project is subdivided into twomajor phases:

- Develop an algorithm that Is capable of encrypting a string of ASCII data using a genetic mutation algorithm without the need for a vignere cipher
- Apply the algorithm to a database driven application, which stores, fetches and decrypts the data using a generated decryption key

The findings of this study will contribute to the enhancement of information security, considering that information leakage is now a significant factor in designing systems.

II. REVIEW OF RELATED STUDIES

The capability to encode information using a mutation algorithm was performed using multiple existing encryption techniques, wherein well-known algorithms are modified for testing [13].

In this said method, plaintext is altered using a particular technique or code, resulting to a virtually unreadable form called the cipher text. This particular cipher text, like any lock, can be retranslated or decrypted back to plaintext using an encryption key.

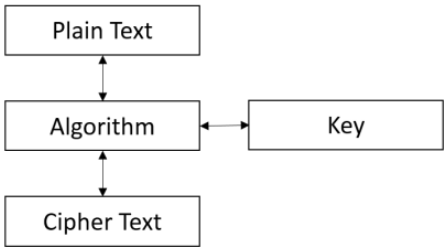


Figure 1- General Encryption Methodology

Methods such as using "hash" is common for most encryption techniques, such as SHA-1 and MD5 methods for databases and other applications shown in above figure 1 above[15]. These methods converts' data to string type information which is unreadable to the user, in a systematic method [2][5]. These standardized encryption techniques share a particular weakness, which causes them to be easily decrypted using scripts and tools existing today.

To address the weakness of a standardized conversion [10], a consistently changing algorithm is to be introduced during the translation process. The technique involves representing data into an array or string, preferably in

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 8, August 2016

binary format, splits the entire string into symmetrical parts and rearranges the binary structure of the information [4]. The technique selects a particular bit in random among the bit string, and alters its structure. This is similar to the Evolutionary theory of Darwin wherein genotypes are mutated to produce a much more robust version of the two parent specie [9]. Applying a mutation, similar to, and usually used by genetic algorithms [1] [7], to alter ciphers is considered to ensure the robustness of data security due to the absence of a standardized method of conversion, and to ensure speed for the encryption process as tested by a similar study[14].

III. METHODOLOGY

The algorithm will be performed on java, and will be tested on data strings of variable length, and consists of any ASCII character available. The algorithm is subdivided into 4 major phases. This will consist of 4 phases, namely:

- Parenting- Mutation algorithms commonly require multiple source data for processing. In the case of the algorithm, both parents originate from the initial string to be encrypted, where Parent 1 is the original string, and P2 is the inverse.
- Pair Selection – The process involves the selection N random sets of binary pairs from parents P1 and P2. In this procedure, an embedded key is generated.
- Mutation – The mutation algorithm chooses the binary pairs and exchanges the position.
- Mating – The procedure involves the joining of both mutated P1 and P2, and the merging of the key generated from the pair selection phase. .

Each phase in the encryption process consist of sub procedures for cryptography, such as data conversion, hexadecimal coding/decoding and concatenations. The algorithm can be summarized as shown in the figure 2.

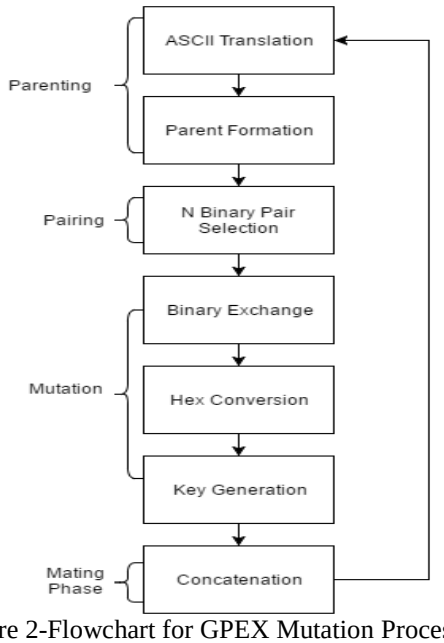


Figure 2-Flowchart for GPEX Mutation Process

Similar to the KBGCT study for cryptography [11] wherein bit-level mutation is introduced randomly, this algorithm mutates binary data at the same degree; however, the difference is the exchange in pairs in the binary stream of data.

IV. TESTING AND RESULTS

The input string is required to be converted to its ASCII equivalent number;thisindicates that the test value “FliP@420”, consisting of 8 ASCII characters, is to be converted to its binary equivalent of 64-bit length.

The algorithm will require the input of a string, which may consist of uppercase and/or lowercase characters, numbers, special characters and no spaces.

A. Parenting

Binary evolution algorithms requiretwo parent strings. Parents for this algorithm will be the original inputted string (P1), and the inverted inputted string (P2).

Both parent strings are to be converted to their ASCII forms, which will then be converted to the respective binary formats.

This will lead to the parent strings to be formatted in the following binary values:

P1

01100110 01101100 01101001 01110000 01000000 00110100 00110010 00110000

P2

00110000 00110010 00110100 01000000 01110000 01101001 01101100 01100110

B. Pair Selection

In this phase, pairs of genes from Parents P1 and P2 that are to be modified are selected at random. An N number of pairs are randomly selected.

In this case, n = 4. Four pairs are selected. P1[52] and P2[0], P1[27] and P2[53],P1[58] and P2[18],P1[26] and P2[62]. Marked pairs are shown below in array format.

P1

01100110 01101100 01101001 01110000 01000000 00110100 00110010 00110000

P2

00110000 00110010 00110100 01000000 01110000 01101001 01101100 01100110

C. Mutation

The genetic mutation phase exchanges the bits from the selected pairs of both parents.

The bit locations are stored on array A as an encryption key. This exchange phase is exhibited in the array below.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 8, August 2016

P1
01100110 01101100 01101001 01110100 01000000 00110100 00110010 00110000

P2
00110000 00110010 00110100 01000000 01110000 01101001 01101100 01100100

Array A containing the bit locations of the mutated pairs will be used as embedded key.

$$A = \begin{bmatrix} 52 & 0 \\ 27 & 53 \\ 58 & 18 \\ 26 & 62 \end{bmatrix}$$

Mutated P1 and P2 are converted into hexadecimal format as the final task in the mutation phase.

P1 - 666C697440343230
P2 –3032344030323440

D. Mating
Converted P1 and P2 are to be mated to produce offspring called child C1. This phase concatenates the mutated Parents P1 and P2, along with the elements of the array A, and N. Final encrypted data is shown below as the string:

C1
666C69744034323030323440303234405200275358182662 4

E. Iteration
The final phase of the algorithm has an arbitrary loop count. Child C1 will replace Parent P1 and its inverse as Parent P2.Encrypted data can be inputted to a database using standard SQL scripting methodologies without any need for interpretation or tools aside from the algorithm itself. An observed issue is the conversion and mutation process consumes a portion of the process.

Testing procedure shows an increase in encrypted data length. Mutation indicates that the length of the candidate data, and the number of genes affect the final length of the encrypted data, as one character is equivalent to 8 bits of data, added to the key containing the N number of pairs.. This can be exhibited in the table shown below.

TABLE I
Effect of the number of gene pairs mutated as compared to the output encrypted data

Trial	Gene Pairs	Encrypted
1	3	466C6B50402412B00C4C2C020A 9636625022150743563
2	7	46DC6D50003C62300C4C2C020A 963662495116353344492161520 85
		710117
3	9	46646951507D12B20C4C2C020A9 6366245621244310447124150354 65640546234489
4	4	4E2C6D50403432300C4C2C020A 96366204382109633537164
5	8	446DE910003432100C4C2C020A9 6366232262516065906370733581 5555563288

Further tests were performed to shows the delay of the algorithm. A single iteration for encryption is performed.

Results are shown for the character of the same value when subjected to a mutation, and then compared to the average time data of the same length and genetic pairs modified.

TABLE II
Results for Data Encryption Process showing average encryption delay with respect to the number of mutated genes

Genes Mutated	Average Time(nanoseconds)
1	3.610789
2	3.828537
3	3.925927
4	3.931224
5	4.068283
6	4.313896
7	4.301630
8	4.643317
9	4.603201

A total of 400 trials per gene count were attempted, an average processing time was derived from the data, wherein a correlation value of 0.976525258 was derived.

This indicates a significant increase in encryption overhead and delay with respect to the amount of genetic pairs mutated. A curve with the below equation is determined from the data collected.

$$y = 0.3126271 + 0.06825133x - 0.02289367x^2 + 0.003473851x^3 - 0.0001742728x^4$$

This curve indicates an increase in overhead and delay in the event of ahigher amount of mutated genes are determined. The uniqueness of mutated data is also considered. As there are m! possible ways of pairing genes, where m is the length of the binary form of the given string.

Given the case of an 8-character data string such as the sample data, when converted to its respective binary level form, acquires 64-bit length. Since both parents P1 and P2

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 8, August 2016

inherit the same length, combinations of data can be determined.
It can be said that the algorithm exhibits a very miniscule probability P of having a duplicate pattern. This can be exhibited in a probability equation exhibited:

$$P = \left(\frac{1}{(m \cdot 8)!}\right)\left(\frac{1}{(m \cdot 8)! - 1}\right)\left(\frac{1}{(m \cdot 8)! - 2}\right)\left(\frac{1}{(m \cdot 8)! - 2}\right) \dots \left(\frac{1}{(m \cdot 8)! - n}\right)$$

where:
Pistheoddsof matching geneticpattern
misthelengthofthedatainASCIIform
nisthenumberofgenestobemutated

The probability equation suggests that the algorithm has total 1.2688693 x 1089 possible combinations of genetic pairs to be mutated. Since 4 unique pairs of genes are to be selected, the possibility of encountering an exactly similar mutation pattern for the encryption method has $1/2.0429x10^{267}$ of probability.

This indicates that an increase in mutated genes decreases the odds of encountering an exact duplicate of the gene pattern.

V. CONCLUSION

String data has been successfully transformed to a fully encrypted string without the need for a vignere cipher and separate key through the concatenation of the key to the code, and stored to a remote database location.

Without a standardized method for decryption, unauthorized access may be reduced and brute force attacks may be rendered inept due to the very miniscule chance of encountering the exact same decryption pattern due to the robust behaviour of the genetic algorithm.

An observation present is the possibility of an increase in overhead and delay when mutated genes are increased. Encrypted data has been inserted to a database and subjected to mechanisms for decryption which resulted to a successful attempt. The Proposed methodology will give the new area of research on cryptography and ASCII algorithms.

This new methodology for encryption and decryption using GPEX algorithm can possibly be an effective method as compared with other cryptography systems in terms of flexibility and robustness.

VI. FUTURE STUDY

An attempt to perform an optimized implementation method will be conducted to lessen the overhead of this dataencryption scheme. Multiple threads for mutation can be considered as a solution for this problem. An application to be used for encryption of image and audio information is being performed to assess viability for Big Data applications of the algorithm.

Testing should involve the usage of decryption mechanisms and scripts to further validate robustness and effectively of algorithm.

ACKNOWLEDGEMENT

This research was supported by the Department of Computer Engineering, the School of Graduate Studies, and the Research and Development Management Office of the Technological Institute of the Philippines. We thank our colleagues from the College of Engineering and Architecture, and the Department of Computer Engineering who provided insight and expertise that assisted the research.

REFERENCES

[1] Afarin, R., &Mozaffari, S. (2013). Image Encryption Using Genetic Algorithm .8th Iranian Conference on Machine Vision and Image Processing (MVIP), 441-445.
[2] Alsharafat, W. S. (2014). Evolutionary Genetic Algorithm for Encryption.IEEE International Conference on Computational Intelligence and Computing Research.
[3] Anderson, R. (2001). Security Engineering : A Guide to Building Depencable Distributed System. Hoboken, New Jersey: John Wiley & Sons.
[4] Bhateja, A., & Kumar, S. (2014). Genetic Algorithm with Elitism for Cryptanalysis of Vigen ere Cipher . International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), 373-377.
[5] Bougrine, M. (2012).New evolutionary tools for a new Ciphering System. IEEE 978-1-4673-2451-9/12, 140-146.
[6] Darwin, C. (1859). On the Origin Of Species by means of natural Selection. London: Royal, Geological, Linnean, Etc. Societies.
[7] Iquebal, M. A. (2000). GENETIC ALGORITHMS AND THEIR APPLICATIONS:. New Delhi, India.
[8] Kerby, F. (2011, July).Understanding Encryption. OUCH! Security Awareness Newsletter.SANS - Securing the Human.
[9] Negnevitsky, M. (2002). Artificial Intelligence: A Guide to Intelligent Systems. Harlow, England: Pearson Education.
[10] Scripcariu, L. (2015). A Study of Methods Used To Improve Encryption Algorithms Robustness .International Symposium on Signals, Circuits, and Systems (ISSCS).
[11] Som, S., &Mandal, N. S. (2011). Key Based Bit Level Genetic Cryptographic Technique (KBGCT) . 7th International Conference on Information Assurance and Security (IAS) 241, 240-245.
[12] Vafaee, F., & Nelson, P. C. (2009). A Genetic Algorithm That Incorporates an Adaptive Mutation Based On an Evolutionary Model. International Conference on Machine Learning and Applications, 102-107.
[13] Yu, L., Wang, W., & Wang, Z. (2012). The Application of Hybrid Encryption Algorithm in Software Security . Fourth International Conference on Computational Intelligence and Communication Networks, 762-765.
[14] Gurpreet Singh, S. (2013). Performance Evaluation of DES and Blowfish AES) for Information Security. International Journal of Computer Applications, 33-38.
[15] VineetSukhraliya, S. C. (2013). Encryption and Decryption Algorithm using ASCII values with substitution array Approach. International Journal of Advanced Research in Computer and Communication Engineering, 3094-3097.

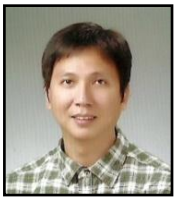
IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 8, August 2016

BIOGRAPHIES

Cris Paulo G. Hate earned his Bachelor's Degree in Computer Engineering at the Technological Institute of the Philippines, Quezon City in 2014, and is currently completing his Master of Engineering, Major in Computer Engineering at the Technological Institute of the Philippines. He is currently a Lecturer at the College of Engineering and Architecture, Computer Engineering Department of the same institution. He is also a Cisco Certified Network Associate in Routing and Switching Technologies. His research interest consists of Artificial Intelligence, Algorithm Analysis,Embedded Systems, Internet of Thingsand Robotics.



Arnel C. Fajardo received the B.S. degree in electrical engineering in 1991 from Mapua Institute of Technology ,the M.S. degree in computer science in 1999 from La Salle University both in Manila, and the Ph.D. degree in computer engineering from Hanbat National University, Daejon, South Korea in 2014. He is a Senior Lecturer at the Technological Institute of the Philippines, Department of Graduate Studies and also an Industry Lecturer at the Department of Computer Engineering at the same institution. He is also a program evaluator/accredit or of information technology and computer science of PAASCU (Philippine Accrediting Association of Schools, Colleges and Universities). He is an author of more than 10 articles in journals and conference proceedings. His research interest includes speech recognition, Artificial intelligence and engineering education.

DOI 10.17148/IJARCCE.2016.58115