CryptO’Roulette - A Smart Approach To Modern Encryption

Samarth Pralhad Nehe, Varun Mishra, P.G.V Sai Karthik, Atharva Gulhane, Navyaa Sharma

Abstract:** In a symmetric-key cryptosystem, where the secret key is known to both the sender who encrypts the message (let’s call her Alice) and the receiver who decrypts the message (let's call him Bob). Since the secret key is very important so it cannot be disclosed to any third party that is why it should not be exchanged through any public form of communication because if through any means these keys get leaked all the encrypted messages in the future will be compromised [1].We know that the advent of technology has increased the risks of data thefts and compromising data integrity. So, to secretly exchange the key both the parties must hold some sort of private meeting, therefore they need to establish some private communication channel. This is a difficult task practically speaking in terms of internet communications. [6] These days, the utilization of the Internet is developing quickly throughout the planet, and security is turning into a significant public concern. Already security was a significant issue for military applications yet presently the application region has been improved as more correspondence happens on the web. Cryptography is a computer science platform intended to give security to senders and receivers to communicate and recover secret data about an uncertain channel through a cycle called Encryption/Decryption. Cryptography guarantees that the message ought to be sent without change and that only authorized individuals can open and peruse the message [10].In this paper, we tried to randomize the encryption process so that the data becomes more secure and less vulnerable to hacker attacks. We will use the top 4 best existing encryption algorithms like AES, TripleDES, Rabbit, etc., and use them to randomly encrypt the different segments of the message.

***Keywords:** Encryption, Cybersecurity, Decryption, Hacking, Cyber-attacks.*

Manuscript received on 07 March 2022.
Revised Manuscript received on 21 March 2022.
Manuscript published on 30 April 2022.
***** Correspondence Author

Varun Mishra, Department of Computer Science and Engineering, Vellore Institute of Technology, Vellore. (Tamil Nadu), India. E-mail: mishra.varun255@gmail.com

Samarth Nehe*, Department of Computer Science and Engineering, Vellore Institute of Technology, Vellore. (Tamil Nadu), India. E-mail: samarthnehe2000@gmail.com

P.G.V Sai Karthik, Department of Computer Science and Engineering, Vellore Institute of Technology, Vellore. (Tamil Nadu), India. E-mail: karthik0901.pendela@gmail.com

Atharva Gulhane, Department of Computer Science and Engineering, Vellore Institute of Technology, Vellore. (Tamil Nadu), India. E-mail: atharva1610@gmail.com

Navyaa Sharma, Department of Computer Science and Engineering, Vellore Institute of Technology, Vellore. (Tamil Nadu), India. E-mail: navyaasharma2000@gmail.com

I. INTRODUCTION

Since public keys can be shared with anyone, this is a really good option where we cannot securely share keys. Three keys. There are also several issues in using public keys.

1) Possible Performance Disadvantages of Encryption

Public key encryption functions admirably and is extremely secure, yet it depends on complex insights. Along these lines, PCs in the past needed to endeavor to scramble and deal with information utilizing a framework. For applications where you want to work with encoded information, the overhead calculator demonstrates that public key frameworks might be slow [2].

2) Possible Certification Issues

Many public-key systems use a third party to ensure the integrity of the public keys. A third party, called the certificate officer, signs with their public key, converting it into a digital certificate, to ensure its security. However, if the certificate officer is in danger, the criminal may issue false certificates and trick people into sending the wrong information to the wrong place.

3) Potential for Direct Compromise

There are 2 methods for extracting encoded information with a public key system. To start with, observe an opening in the numbers beneath that can be utilized to break the cipher. One more method for breaking the encryption is to figure out the right key. As PCs become more dynamic and as quantum computing, which uses light to make quicker speeds even past conventional supercomputers, it turns out to be more sensible, strong attacks on secret data work effectively [4].

4) False Feeling of Security

No matter how secure your system is, it will only protect the part of the system for which it was designed. For example, if a customer sends his credit card details online, that transfer is protected by a combination of private and public encryption keys and is highly secure. If someone receives your credit card information, or if someone gets open access to your server, they can sit down and download all the securely transferred data attached to your credit card. Here public key encryption will not protect you from that and, as a result, is part of the entire security system [3].

Although we know the algorithms like AES and Triple DES are known as quite efficient algorithms but due to any unforeseen conditions mentioned above in the future, if any hacker gets access to the key then he/she can easily decrypt the message.

Retrieval Number: 100.1/ijeat.B33111211221
DOI: 10.35940/ijeat.B3311.0411422
Journal Website: www.ijeat.org

*Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)
© Copyright: All rights reserved.*

But the algorithm proposed by us even though he gets access to the key he/she won't be able to decrypt the message because they wouldn't know which part of the passage which encryption algorithm has been applied. Several research papers have been published in various international journals in recent years. Here our main intention would be to find out how the algorithms are getting attacked and what are the loopholes in the existing algorithms. To randomly encrypt the message using these algorithms, we need to find the strength of each algorithm. Once the algorithm is encrypted, it is almost impossible for the hacker to figure out the combination in which these algorithms were applied. This will improve the encryption efficiency by a great amount. For the decryption purpose, we are planning to use our identifier to figure out how the encryption algorithms were used and to know what were the segments in which the message was broken into. As the final project outcome, we are planning to publish a conference paper that will highlight our method of random encryption that can be used as a smart approach for improved modern encryption. Apart from the research paper, we are planning to make a user interface in the form of a website that will show the implementation of message encryption and decryption.

II. NOVELTY

With the rapid developments in digital image processing and network communication, electronic publishing, and widespread dissemination of digital multimedia data over the Internet, the protection of digital information against illegal copying and distribution has become extremely important [5]. To meet this challenge, many new encryption schemes have been proposed. Though these algorithms are quite up to the challenges, the core idea behind most of these algorithms is based on the notion of a key. Once the key is compromised the complete data gets compromised. The novelty behind our project lies in adding a protection layer to the whole process through the means of randomizing the selected encryption algorithms used for different segments of the message, indirectly randomizing the complete encryption process. This way even when an external person gets hold of the key, he still won't be able to figure out how to use it. Once the algorithm is encrypted, it is almost impossible for the hacker to figure out the combination in which these algorithms were applied. This will improve the encryption efficiency by a great amount. For the decryption purpose, we are planning to use our identifier to figure out how the encryption algorithms were used and to know what were the segments in which the message was broken into.

III. PROPOSED METHODOLOGY

In contrast to the existing encryption algorithm which can be easily decrypted once a hacker gets access to the key of the encryption. We propose that we can improve the system by dividing the message into fixed-sized segments and applying an algorithm from a predetermined set of algorithms in a random way. We then tend to add some custom/ unique identifiers at the end of these segments for ease of decryption later on. This way even if the hacker is somehow able to get access to the key they still cannot decrypt it as they won't have any idea how to use the key as they have no clue to which algorithm has been used to encrypt the message. This way the encryption becomes much safer compared to the traditional encryption techniques followed.

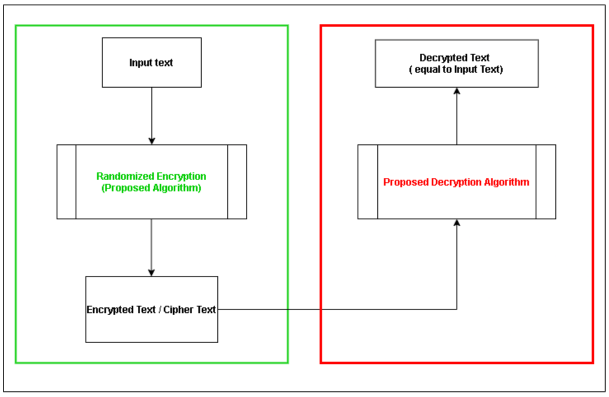

Input text
Randomized Encryption (Proposed Algorithm)
Encrypted Text / Cipher Text
Decrypted Text (equal to Input Text)
Proposed Decryption Algorithm

Fig. 1. Overall System Flowchart

A. Encryption Module:

We begin the encryption process by first taking a text (message) as an input. The next step in the backend is to break this message into fixed-sized segments. Later, we need to initialize an empty string so that it can store our encrypted text. Based on the number of segments we run a loop in order to pick each and every segment and encrypt them individually. Create an array consisting of all the encryption algorithms in our case these are namely AES, Triple DES, Rabbit, RC4; these being the top encryption algorithms used widely nowadays. Now we need to run 2 processes in parallel and those being - selecting a random non-negative integer -to pick an encryption algorithm from the array of algorithm names and apply the encryption algorithm. Immediately after this, we append a unique identifier to the so-far formed string to keep track of which encryption has been used to encrypt this segment so that it will be helpful at the time of decryption. Then we need to repeat this process until all the segments have been encrypted. Once all these segments have been encrypted we get our desired encrypted string.

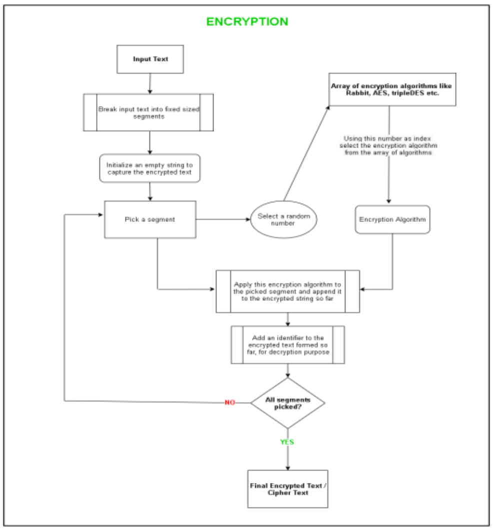

ENCRYPTION
Input Text
Break input text into fixed sized segments
Initialize an empty string to capture the encrypted text
Pick a segment
Select a random number
Array of encryption algorithms like Rabbit, AES, tripleDES etc.
Using this number as index select the encryption algorithm from the array of algorithms
Encryption Algorithm
Apply this encryption algorithm to the picked segment and append it to the encrypted string so far
Add an identifier to the encrypted text formed so far, for decryption purpose
All segments picked?
NO
YES
Final Encrypted Text / Cipher Text

Fig. 2. Encryption Module Flow Diagram

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)
© Copyright: All rights reserved.

B. Decryption Module

Here the input text is not something that has to be manually entered by the user, but it is the encrypted text that we received earlier in the above process (encryption). Similar to encryption, for the final output, we need an empty string that will store the final decrypted message. Initialize a variable called position with value 0. Start to parse the ciphertext until you get an identifier and then we extract the string from the index equal to position value to the current position. Decrypt the string that has been extracted in the above step (with the help of the identifier that we got) and append this decrypted text to the empty string initialized. Now update the value of the position variable to the index of the identifier. Repeat steps 4, 5, and 6 until we reach the end of the string. Return the decrypted string.

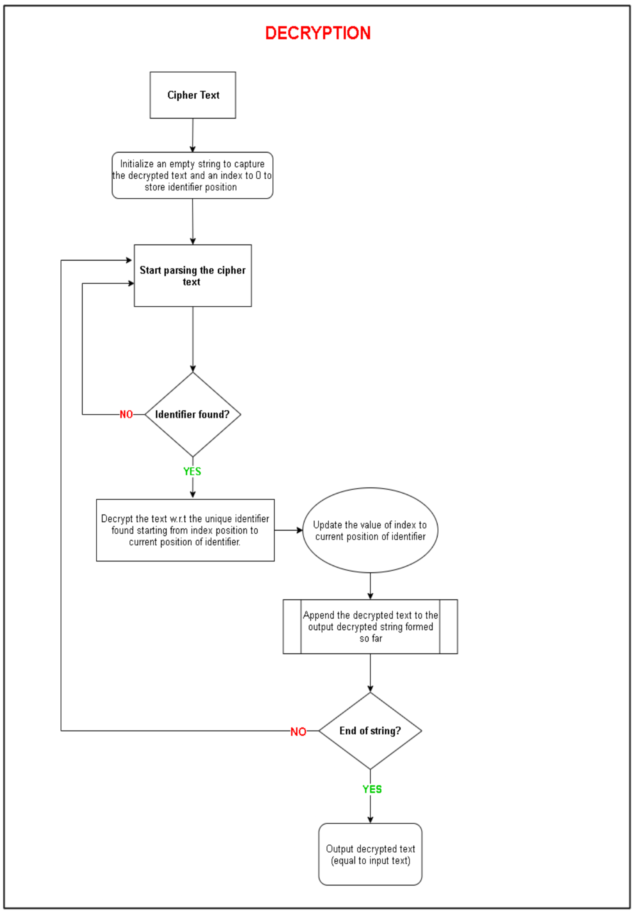

DECRYPTION
Cipher Text
Initialize an empty string to capture the decrypted text and an index to 0 to store identifier position
Start parsing the cipher text
Identifier found?
NO
YES
Decrypt the text w.r.t the unique identifier found starting from index position to current position of identifier.
Update the value of index to current position of identifier
Append the decrypted text to the output decrypted string formed so far
End of string?
NO
YES
Output decrypted text (equal to input text)

Fig. 3. Decryption Module Flow Diagram

IV. PROBLEM STATEMENT

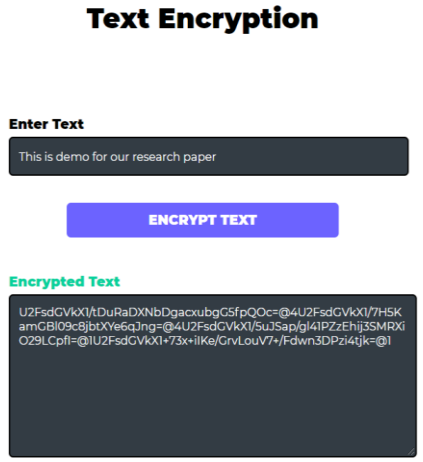
The two primary goals of our work are:

1) **Proposing Algorithm.** To conduct an in-depth study of various encryption algorithms that are used, and to figure out the loopholes in them. By bringing all the existing algorithms together and merging their qualities, we try to bring a solution that makes the encryption system efficient as a whole. Based on our solution, we aim to derive and present a few high-level guidelines, which would be valuable to developers and deployment engineers.

2) **Optimization.** To identify bottlenecks and to draw out action items to improve the overall performance of encryption algorithms. On identifying bottlenecks, our goal is to introduce and implement optimizations to alleviate these bottlenecks.

V. RESULTS AND DISCUSSIONS

Our aim in this paper was to come up with an algorithm and technique that provides a better amount of security and privacy of data through encryption. We have done deep research on the existing encryption algorithms and figured out their pros and cons. Due to the extensive use of the same encryption algorithm for several years, the algorithms have somewhat become extinct and easier to crack and decrypt after noticeable attempts. Taking this point in mind, we decided to use some of the existing algorithms and extract their good qualities to create a hybrid method. We in this paper called it a randomized approach.



Text Encryption

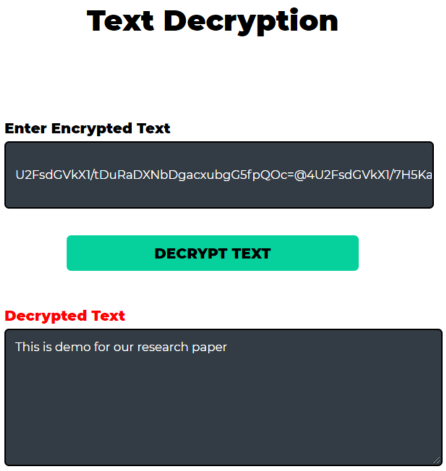
Enter Text

This is demo for our research paper

Encrypted Text

U2FsdGVkX1/tDuRaDXNbDgacxubgG5fpQOc=@4U2FsdGVkX1/7H5KamGBl09c8jbtXYe6qJng=@4U2FsdGVkX1/5uJSap/gl41PZzEhij3SMRXiO29LCpfl=@1U2FsdGVkX1+73x+iIKe/GrvLouV7+/Fdwn3DPzi4tjk=@1

According to our prediction and idea, we assumed the new algorithm to provide better results and be more efficient. We ended up making a platform that would help the user use this algorithm using a proper interface. The results that we got were the proper encryption of the message using randomly chosen algorithms and the application of delimiters and also, we observed the message getting perfectly decrypted with the help of delimiters used.



Text Decryption

Enter Encrypted Text

U2FsdGVkX1/tDuRaDXNbDgacxubgG5fpQOc=@4U2FsdGVkX1/7H5Ka

Decrypted Text

This is demo for our research paper

The main focus of ours was to make the encryption process more secure in such a way that the attackers have no clue as to which technique was used by the encrypt or for the encryption [9]. This way, only the system knows what it has done and how it has to be decoded.

VI. CONCLUSION

In this paper, we studied the performances of various encryption algorithms and their vulnerability to attacks and identified a major bottleneck: Chances of encrypted codes to be broken in the future with modern technology. Toward this, we introduced a new method called randomized encryption [7]. Finally, we introduced a solution that used the positives of other encryption algorithms and works as a hybrid. We know that the advent of technology has increased the risks of Data thefts and compromising Data Integrity. Through our proposed algorithm we have been able to parry the above-mentioned issues. Even though by any chance the Hacker gets the key of the system he won't be able to hack into the system. Also, the algorithms used by our proposed method are among the best encryption algorithms up to date.

ACKNOWLEDGMENT

We thank our faculty, Prof. Manjula R, for providing us this opportunity to write this paper. Along with this, we thank our university for making us select this course that gave us exposure to how research papers are written.

REFERENCES

1. Rao, Sandeep & Mahto, Dindayal & Khan, Danish. (2017). A Survey on Advanced Encryption Standard. International Journal of Science and Research (IJSR). 391. 10.21275/ART20164149.
2. Badugu Nikhil Teja, Shweta Helchel, P.Sai Krishna Pratap Reddy, and V.Seetha Rama Rao. (2019); IMPLEMENTATION OF ADVANCED ENCRYPTION STANDARD (AES) FOR HIGH-SECURITY APPLICATIONS. Int. J. of Adv. Res. 7 (Apr). 498-504] (ISSN 2320-5407).
3. Dawood, Omar A. & Hammadi, Othman. (2017). An Analytical Study for Some Drawbacks and Weakness Points of the AES Cipher (Rijndael Algorithm). 10.25212/ICoIT17.013.
4. Boesgaard, Martin & Vesterager, Mette & Pedersen, Thomas & Christiansen, Jesper & Scavenius, Ove. (2003). Rabbit: A New High-Performance Stream Cipher. 2887. 307-329. 10.1007/978-3-540-39887-5_2
5. P. N., C. S., and S. M. Rehman, "ASIC Implementation of Rabbit Stream Cipher Encryption for Data," 2019 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE), 2019, pp. 1-4, DOI: 10.1109/WIECON-ECE48653.2019.9019903.
6. Jindal, Poonam & Singh, Brahmjit. (2014). RC4 Encryption-A Literature Survey. Procedia Computer Science. 46. 10.1016/j.procs.2015.02.129.
7. Kitsos, Paris & Kostopoulos, Giorgos & Sklavos, Nicolas & Koufopavlou, Odysseas. (2004). Hardware implementation of the RC4 stream cipher. 3. 1363 - 1366 Vol. 3. 10.1109/MWSCAS.2003.1562548.
8. Alanazi, Hamdan & Bahaa, Bilal & Zaidan, A. & Jalab, Hamid & Shabbir, M. & Al-Nabhani, Yahya. (2010). New Comparative Study Between DES, 3DES, and AES within Nine Factors.
9. Ratnadewi, Ratnadewi & Adhie, Roy & Hutama, Yonatan & Ahmar, Ansari & Setiawan, M. (2018). Implementation Cryptography Data Encryption Standard (DES) and Triple Data Encryption Standard (3DES) Method in Communication System Based Near Field Communication (NFC). Journal of Physics: Conference Series. 954. 012009. 10.1088/1742-6596/954/1/012009.
10. Agrawal, Monika & Mishra, Pradeep. (2012). A Comparative Survey on Symmetric Key Encryption Techniques. Int. J. Comput. Sci. Eng.. 4.

AUTHORS PROFILE

Varun Mishra is a creative, dedicated, and enthusiastic Software Engineer at NielsenIQ. He has a degree in B.Tech Computer Science and specializes in Android Application Development and Problem Solving and Programming. He is focused on skill development and has made achievements in areas of Competitive Coding. His main areas of interest in the field of Computer Science include Data Engineering, Cloud Computing, System Design, Application Development, Databases, Data Structures, and Problem Solving. He is a self-motivated and optimistic person who loves to ideate and work on solving complex real-world problems and create solutions to do good to society.
Email: mishra.varun255@gmail.com

Samarth Nehe is a hardworking, passionate and motivated tech enthusiast. He is an upcoming Software Engineer at MakeMyTrip. He has a degree in Computer Science and is specialized in Web Technology and Problem Solving. His major interest lies in Frontend development of web and app, UI/UX, and Competitive Coding. Being a self-motivated person who is always diligent and pragmatic, he is also the Co-Founder and Former Chairperson of C.U.B.E VIT which has more than 200 members at present. He has around 3.5 years of experience in web development and design and has recently gotten into freelancing. He aims to create something big in life with the help of technology. Email: samarthnehe2000@gmail.com

P.G.V Sai Karthik is a hardworking, passionate tech enthusiast and he is a Software Engineer at JP Morgan Chase & Co. He has a degree in B.Tech Computer Science and specializes in Software Development and Problem Solving and Programming. He is ambitious about his learning and career. With his competent skills, he has made achievements in the areas of Competitive Coding and gained good exposure to real-time Problem Solving. His main areas of interest in the field of Computer Science include Web Development, Machine Learning, Artificial Intelligence, Databases, and Problem Solving. He is a self-motivated and optimistic person who loves to ideate and work on solving complex real-world problems. Email: karthik0901.pendela@gmail.com

Atharva Gulhane is a passionate and self-motivated Techie and he is a Software Engineer at JP Morgan Chase & Co. He has a degree in B.Tech Computer Science and specializes in Software Development and Problem Solving and Programming. He is a self-learner and loves to work on skill development and has made achievements in the areas of Competitive Coding. His hard work and knowledge have helped him gain good industry exposure. He aims to create tech-based solutions for people and loves learning about new technologies. His main areas of interest in the field of Computer Science include Web Development, Microservices, DevOps, Databases, Data Structures, and Problem Solving.
Email: atharva1610@gmail.com

Navyaa Sharma is a dedicated, ambitious, and passionate Software Developer. She has a great industrial exposure of working as a Software Developer and Freelancer for over 4 years and is an Associate Consultant at Microsoft. She has a degree in B.Tech Computer Science and specializes in FullStack Software Development and Problem Solving. Her main areas of interest in the field of Computer Science include Cloud Computing, System Design, Web Development, Databases, DevOps, and Data Structures. She prefers to be a team leader and believes in teamwork and optimism. She has experience mentoring people and she likes to create educational content through YouTube that could be helpful to the community.Email: navyaasharma2000@gmail.com