ISSN: 3067-2392
Research Article
Planetary Journal of Social Sciences & Humanities Research

Cryptographic Solutions for Protecting Text and Image Data in Digital Communication

Danail Runchevski[1*], and Ustijana Rechkoska-Shikoska[2]

[1]*Faculty of Communication Networks and Security University for Information Science and Technology "St. Paul the Apostle" Resen, North Macedonia*

[2]*Faculty of Computer Science and Engineering University for Information Science and Technology "St. Paul the Apostle" Ohrid, North Macedonia*

[*]Corresponding author: Danail Runchevski, Faculty of Communication Networks and Security University for Information Science and Technology "St. Paul the Apostle" Resen, North Macedonia.

Submitted: 27 June 2024 **Accepted:** 04 July 2024 **Published:** 12 July 2024

https://doi.org/10.63620/MKPJSSHR.2024.1003

Citation: *Runchevski, D., & Rechkoska-Shikoska, U. (2024). Cryptographic Solutions for Protecting Text and Image Data in Digital Communication. Planetary J Soc Sci & Hum Res, 1(2), 01-05.*

Abstract

This research paper introduces a Python-based script for secure message and image encryption and decryption. Users can input data for encryption and obtain the encrypted output, with decryption requiring a specific key. Leveraging Python's versatility and cryptographic libraries, our script provides a user-friendly, yet robust, solution for data protection. This paper outlines the script's functionality and discusses its relevance in modern data security, emphasizing the importance of encryption in the digital age. Our comprehensive evaluation covered various aspects of the user experience in encryption and decryption. The results indicated a favorable response to the design, usability, efficiency, and time-saving features of the application. These insights inform our ongoing endeavors to improve encryption and decryption applications, ensuring they align with user expectations.

Keywords: Security, Encryption, Decryption, Data Protection, Cryptography

Introduction

Encryption and decryption are essential tools for protecting sensitive data in transit and at rest. In this paper, we present a Python-based encryption and decryption application that allows users to encrypt and decrypt text messages and images using an encryption algorithm. The application consists of two main components: an encryption module and a decryption module. The encryption module takes a plaintext message or image as input and produces a ciphertext message or image as output. The decryption module takes a ciphertext message or image as input and produces the original plaintext message or image as output. The application is implemented in Python, a popular open-source programming language. Python is well-suited for this task because it provides a number of libraries and functions for encryption and decryption. We chose to develop this application because we believe that encryption and decryption are important tools for protecting sensitive data. We hope that our application will be useful to researchers and practitioners who need to encrypt and decrypt data in a secure and efficient manner [1, 2].

Related Work

Security plays a pivotal role in the storage and transmission of information across networks, ensuring that data is exchanged in a secure manner. Secure communication stands as a fundamental requirement for all network transactions. Cryptography, a cornerstone of secure communication, serves to facilitate the transmission of information while offering essential security services such as confidentiality, data integrity, access control, authentication, and non-repudiation.

This field provides a means to safeguard sensitive information by transforming it into an unintelligible format, only accessible to authorized recipients who can convert it back to its original form. The process of converting plaintext into ciphertext with the aid of a key is known as encryption, while the reverse operation is termed decryption. The optimal design of cryptographic algorithms seeks to be secure, efficient, cost-effective, with a minimal memory footprint, easy to implement, and adaptable across various platforms.

While a vast array of applications have been developed to bolster the security of cryptographic algorithms using diverse mathematical techniques, it remains a challenge to create entirely secure encryption algorithms. This challenge stems from the persistent efforts of cryptanalysts who continuously strive to breach any available cryptographic systems [3-7].

Cyber Seciruty

Cybersecurity is the proactive endeavor of safeguarding interconnected systems on the internet, including hardware, software components, and sensitive data, to shield them from the inherent risks associated with cyberthreats. This strategic approach is widely adopted by individuals and organizations to forestall any unauthorized entry into their data centers and computer systems. A robust cybersecurity strategy assumes a pivotal role in establishing a resilient security posture against malicious attacks aimed at infiltrating, manipulating, erasing, compromising, or coercing an entity's systems and valuable data assets. Furthermore, it serves as a formidable deterrent against attacks designed to disrupt or disable the normal operations of a system or device.

The significance of cybersecurity is magnified in the contemporary enterprise landscape characterized by a burgeoning user base, an array of interconnected devices and applications, and an unprecedented deluge of data, often of a sensitive or confidential nature. The challenge is compounded by the escalating sophistication and volume of cyber attackers and their evolving tactics.

The cybersecurity domain can be dissected into various specialized areas, each of which necessitates meticulous coordination within organizations to fortify the overarching cybersecurity framework. These facets encompass application security, data security, network security, disaster recovery/business continuity planning, operational security, cloud security, critical infrastructure security, physical security, and end-user education.

Sustaining robust cybersecurity in the face of a perpetually evolving threat landscape is an enduring challenge for all entities. The traditional reactive approach of concentrating resources on defending against well-known threats, while neglecting lesser-known ones, has proven inadequate. To effectively counter dynamic security risks, a proactive and adaptive approach is indispensable. Esteemed cybersecurity advisory bodies, such as the National Institute of Standards and Technology, advocate the adoption of continuous monitoring and real-time assessments as integral components of a risk assessment framework, offering a resilient defense against both recognized and unforeseen threats.

Encryption

In the realm of cryptography, encryption plays a vital role in securing information. This process involves transforming the original data, known as plaintext, into an alternative format called ciphertext. The goal is to make it comprehensible only to authorized individuals while keeping it unintelligible to potential interceptors. Encryption itself doesn't thwart interception but rather safeguards the content.

For practical reasons, encryption typically employs a pseudo-random encryption key generated by an algorithm. Decrypting the message without possessing the key is theoretically possible but demands significant computational resources and expertise. Authorized recipients can easily decipher the message using the key provided by the sender, while unauthorized users remain locked out.

Throughout history, a variety of encryption techniques have been used, especially in military communications. As technology advanced, new methods, including public-key and symmetric-key cryptography, have become commonplace in modern computing. These techniques are considered secure because contemporary computers face significant challenges when attempting to crack encryption [8, 9].

Decryption

Decryption is the pivotal process of reverting encrypted data, which has been made unreadable, back to its original and intelligible form. During decryption, the system takes scrambled data and converts it into clear text and images, making it easily comprehensible for both the user and the system. This transformation can be achieved manually or automatically and typically necessitates specific keys or passwords.

In the realm of cybersecurity, decryption serves as a formidable technique that obstructs unauthorized access to sensitive information, making it more challenging for hackers to intercept and decipher data. It is the process of reversing encryption, which obfuscates data to render it unreadable to unauthorized parties. Only individuals possessing the matching decryption keys can access and understand the data.

While encryption ensures data security, decryption requires the appropriate decoding tools and methods for data access. Decryption can be executed manually or automatically, facilitated by robust decryption software, unique keys, passwords, or codes. This process translates previously unreadable or indecipherable data back into its original formats, encompassing text files, email messages, images, user data, and directories, thereby enabling users and computer systems to readily read and interpret the information.

Image Encryption

Image encryption employs an innovative approach by manipulating consecutive or random pixel bits within an image. These bits are collectively processed using logic, resulting in a new set of pixels that differ from the original arrangement. This transformation introduces a unique mode of information transmission. To bolster security, an additional layer of complexity can be incorporated through the use of the Cipher Block Chaining (CBC) method. With CBC, the plain text is intricately woven into the encrypted image, enhancing data transfer security. Furthermore, the introduction of a key fortifies the image's resilience against external threats. The CBC method involves a relatively simple process: it takes an N-bit block of plain text and combines it with an M-bit block known as an initialization vector through a process called XOR. This XOR operation results in a binary bit pattern that bears no resemblance to the original message. This process generates a block of data, often denoted as M×N. Subsequently, a secret key is introduced into the amalgamation of plain text and initialization vector using Block Cipher Encryption. This step culminates in the creation of an encrypted cipher text or image [10-13].

Encrypt Image – BS Scenario

Using a Hex-Editor we can check the contents of an image as represented in its bytes state, which is presented in “Fig. 1”.

Figure 1: Bytes arragement before encoding

Encdoing the image by XOR-ing each byte, is presented in "Fig. 2" and "Fig. 3".

```
image = bytearray(image)
```

Figure 2: Reading the image as an array of bytes

```
for index, values in enumerate(image):
        image[index] = values ^ key
```

Figure 3: XOR-ing each byte of the image

After encoding each byte, the final results are generated and shown in "Fig. 4".

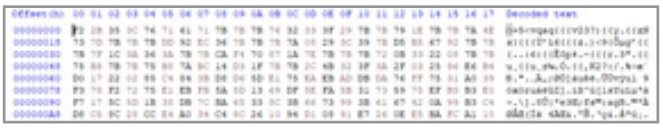

Figure 4: Bytes arragement after encoding

In summary, this code takes an image file, applies XOR-based encryption by bitwise XOR-ing each byte of the image data with an encryption key, and then writes the encrypted image data back to the same file. XOR is a bitwise operation that is reversible when you have one of the operands (either the original data or the encrypted data) and the result of the XOR operation.

Image Decryption

In order to decrypt the encrypted image, all we need to do is use the same key that was used for encryption and XOR the array of bytes that make up the image.

```
fin = open(path, 'rb')

image = fin.read()
fin.close()
image = bytearray(image)
for index, values in enumerate(image):
image[index] = values ^ key

fin = open(path, 'wb')

fin.write(image)
fin.close()
```

Figure 5: User satisfaction for encryption

In "Fig. 5", `fin` is a file object that is opened for reading in binary mode ('rb'). This means that the code will read the file in binary mode, which is suitable for reading non-text files like images. The read method is used to read the entire content of the file into the image variable. This content is read as a sequence of bytes. After the reading, the file is closed to free up system resources using the close method. The image is converted into a bytearray. This is done so that individual bytes in the image data can be modified later using indexing. Following is the loop iterating through each byte in the image bytearray. For each byte, it performs a bitwise XOR (^) operation with a "key" (the decryption key). The result of this XOR operation is then stored back in the same byte position in the image bytearray. The write method is used to write the modified image data (contained in the image bytearray) back to the file – decrypting it.

In summary, this code reads an image file, XORs each byte of the image data with a specified "key" for decryption, and then writes the modified data back to the same file. The "key" is a crucial parameter for this process, and it should be known in order to decrypt the image.

Without Key

Decryption without a key can be done in two ways:

- If you have the original data: You can XOR the original data with the encrypted data (which you already have) to recover the encryption key. Once you have the encryption key, you can use it to decrypt the entire image.
- If you don't have the original data: You can try XORing the encrypted data with various values until you find a combination that results in meaningful content. This is essentially a brute-force approach and can be very time-consuming and impractical for larger files or complex data.

Quality of Experience of Image Encryption and Decryption

Quality of Service (QoS) encompasses the technical and operational aspects of a service, including factors like response time, capacity, and data transport capabilities. On the other hand, Quality of Experience (QoE) gauges the variance between users' expectations and the actual service they receive. Evaluating QoE proves valuable in assessing users' perceptions of a service's quality, considering factors like usability, accessibility, reliability, and data integrity [9]. QoE means overall acceptability of an application or service, as perceived subjectively by the end user and represents multidimensional subjective concept that is not easy to evaluate. In our work, we have used QoE evaluation in order to measure the quality of the image encryption and decryption scenarios.

In the following figures, "Fig. 6 – Fig. 11" results according QoE evaluation performed University representatives from the Faculty of Communication Networks and Security are presented, for image encryption and decryption respectfully.

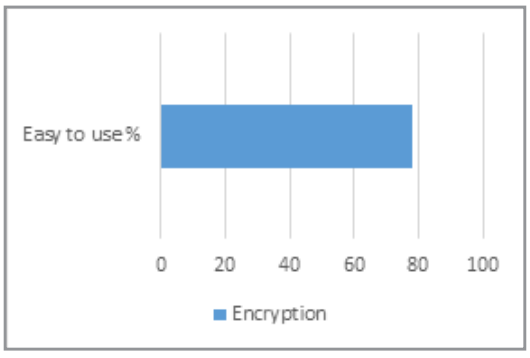

Figure 6: User satisfaction for encryption

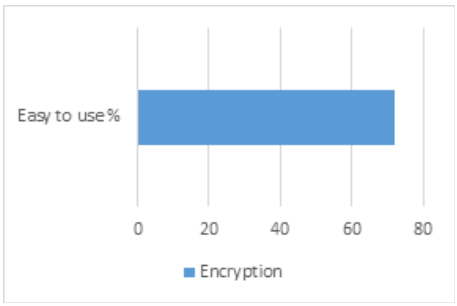

Figure 7: Easy to use for encryption

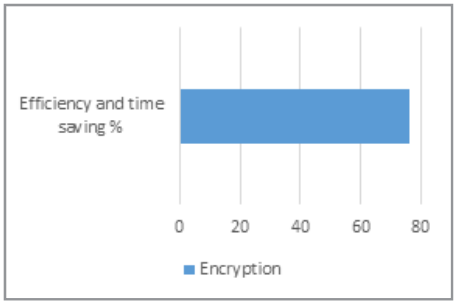

Figure 8: Efficiency and time saving for encryption

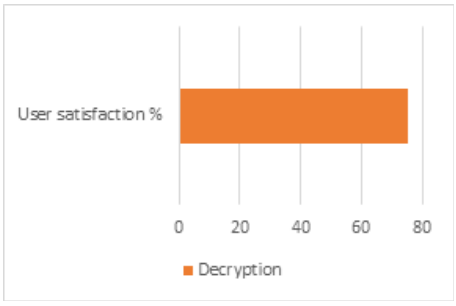

Figure 9: User satisfaction for decryption

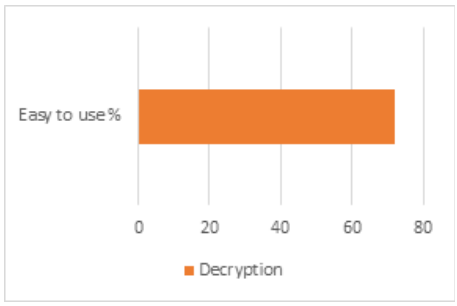

Figure 10: Easy to use for decryption

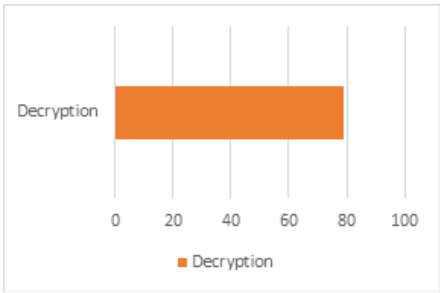

Figure 11: Efficiency and time saving for decryption

The results presented in the following chart were obtained according the Quality of Experience evaluation performed with the University representatives from the Faculty of Communication Networks and Security. The image encryption and decryption are tested and evaluated by different scenarios. The survey questions were answered by a group of representatives of the University that participated in the image encryption and decryption. Analyzing the answers from the image encryption and decryption has provided with the summary given in histogram presented in "Fig. 12".

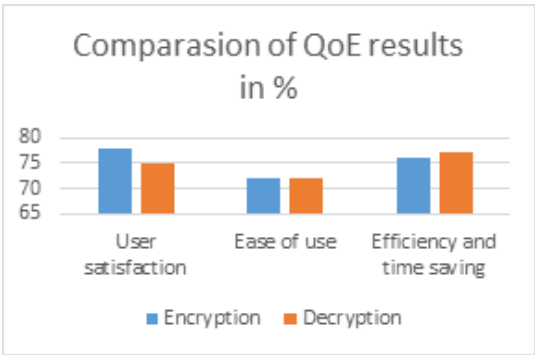

Figure 12: Comparasion of the resutls

This feedback analysis serves as a critical component of our commitment to enhancing the usability and effectiveness of our tools. Based on the reports for our encryption and decryption tools, focusing on user satisfaction, ease of use, efficiency, and time-saving aspects we came to the following conclusions.

User Satisfaction

In the context of user satisfaction, our encryption tool garnered a substantial 78% approval rating, reflecting the high level of contentment among our users. Our decryption tool received commendable feedback as well, with a notable 75% user satisfaction rating. These favorable responses underscore the alignment of our tools with the users' expectations.

Ease of Use

Our encryption tool demonstrated a noteworthy 72% rating for ease of use, indicating that a significant portion of users found the tool to be highly user-friendly. Similarly, our decryption tool was deemed user-friendly by an impressive 72% of respondents. These statistics underscore our commitment to offering intuitive and accessible interfaces to facilitate user interactions.

Efficiency and Time Saving

Efficiency and time saving capabilities are paramount in our tools. In this regard, our encryption tool received commendation from 76% of users, acknowledging its efficacy in securing data while optimizing time management. The decryption tool excelled in this aspect, receiving a remarkable 79% rating, signifying its ability to swiftly restore encrypted data. We acknowledge the significance of time in the digital realm and are pleased to offer tools that not only ensure data security but also expedite data recovery.

In summary, our evaluation encompassed multiple facets of the encryption and decryption user experience. Findings revealed a positive reception of the encryption application's design, us-

ability, efficiency, and time-saving attributes. Users expressed varying preferences based on the application's orientation, highlighting the importance of offering versatile solutions to cater to diverse user needs. These insights contribute to our ongoing efforts to enhance the quality of encryption and decryption applications and their alignment with user expectations.

Conclusion

In conclusion, this paper has delved into the critical concepts of encryption, decryption, and their role in data security. The Python-based encryption and decryption application presented here empowers users to protect their text messages and images. Cybersecurity is a proactive defense against a multitude of digital threats, safeguarding interconnected systems and sensitive data. Encryption ensures data confidentiality, rendering information inaccessible to unauthorized parties. In today's evolving digital landscape, data protection remains paramount. Encryption and decryption are fundamental tools for securing data, as showcased by our Python application, offering practical solutions for users and organizations [14, 15].

Future Work

Today’s federal agencies are generating, analyzing, and transporting data at an unprecedented, exponentially accelerating rate. To keep all that data safe whether it’s at rest, in use, or on the move the government must not only employ today’s most reliable encryption technologies, it must be ready to adopt tomorrow’s as well. As insider and adversarial threats grow and advance, promising new approaches to encryption like post-quantum cryptography, quantum key distribution, and homomorphic encryption will be key to maintaining the nation’s information security.

Most modern encryption systems are key based. To keep data usable only to those with permission to access it, an encryption key uses an algorithm to encode readable data into unreadable data. When a credentialed person or machine is ready to access that data, a decryption key is used to make it readable again.

As cyber adversaries become more sophisticated and enterprise mobility, cloud, and Internet of Things technologies are more broadly and thoroughly embraced by federal agencies, more powerful encryption methods will be necessary.

Organizations will need to prepare for the adoption of superior emerging encryption technologies as they become available for practical use.

References

1. Sharma, N., Er, P., & Kaur, H. (2017). A Review of Information Security using Cryptography Technique. International Journal of Advanced Research in Computer Science, 8(4).
2. Mandal, A. K., Prakash, C., & Tiari, A. (2012). Performance evaluation of cryptographic algorithms: DES and AES. In Proceedings of the IEEE Students' Conference on Electrical, Electronics and Computer Science: Innovation for Humanity (SCEECS).
3. Kumar, N., & Chaudhary, P. (2016). Performance evaluation of encryption/decryption mechanisms to enhance data security. Indian Journal of Science and Technology, 9, 1-10.
4. Katz, J., & Lindell, Y. (2008). Introduction to Modern Cryptography. Taylor & Francis Group, LLC.
5. da Silva, N. B. F., Pigatto, D. F., Martins, P., & Branco, K. C. (2016). Case studies of performance evaluation of cryptographic algorithms for an embedded system and a general-purpose computer. Journal of Network and Computer Applications, 60, 130-143.
6. Callas, J. (2007). The Future of Cryptography. Information Systems Security, 16, 15-22.
7. Maqsood, F., Ali, M. M., Ahmed, M., & Shah, M. A. (2017). Cryptography: A comparative analysis for modern techniques. International Journal of Advanced Computer Science and Applications, 8(6).
8. Mel, H. X., & Baker, D. M. (2001). Cryptography decrypted (1st ed.). Addison-Wesley.
9. Preneel, B. (2015). Cryptography and Information Security in the Post-Snowden Era. In IEEE/ACM 1st International Workshop on Technical and Legal aspects of data privacy and security, Florence.
10. Piper, F., & Murphy, S. (2002). Cryptography: A Very Short Introduction. Oxford University Press.
11. St Denis, T., & Johnson, S. (2007). Cryptography for Developers. Syngress Publishing Inc.
12. Dooley, J. F. (2013). A Brief History of Cryptology and Cryptographic Algorithms. Springer.
13. Orman, H. (2014). Recent Parables in Cryptography. IEEE Internet Computing.
14. Dworkin, M. (2005). Recommendation for Block Cipher Modes of Operation: the CMAC Mode for Authentication.
15. Wu, W., Arefin, A., Rivas, R., Nahrstedt, K., Sheppard, R., et al. (2009). Quality of Experience in Distributed Interactive Multimedia Environments: Toward a Theoretical Framework. In MM '09 Proceedings of the 17th ACM international conference on Multimedia (pp. 11-45).