

New Generated Keys of the AES Algorithm Based on the SHA256 Algorithm for Image Encryption

Raghda Sattar Jabbar

Department of Translation, College of Art
Mustansiriyah University, Baghdad
Iraq

ABSTRACT

To prevent unauthorized access to digital data, stronger encryption solutions are needed. The Advanced Encryption Standard is instrumental to cryptographic security, but this reliance on traditional forms of key generation risks undoing its potency. In order to fulfill this effect that the research presents its new work for another version of AES key generation with SHA256. Using the high entropy of SHA256, we generate even more complicated and unpredictable keys that drastically raises resilience to cryptographic attacks. We demonstrate this with a Python implementation of our proposed method along with experiments on image data which show significantly increased encryption strength over standard AES, whilst having similar performance. This research represents an additional step toward developing safer methods of encryption that safeguard data.

Key Words: AES, Decryption, Encryption, SHA256.

1. INTRODUCTION

More powerful digital technologies and an increasingly shared world have left sensitive information more vulnerable than ever. Across a range of sectors, from finance to healthcare etc., securing the integrity and confidentiality of this data has become one of their most important priorities. Encryption— Fundamental Security Technique Playing Key Role in Protecting Digital Content.

The National Institute of Standards and Technology (NIST) is currently using the Advanced Encryption Standard (AES) as a symmetric encryption algorithm, which has been widely accepted because it is efficient and robust [1].

The security of AES is heavily dependent on the way we generate a key. The majority of previous key generation techniques involve user-chosen passwords or a shoddy random number generator, which are frequently vulnerable to brute-force, dictionary, and even key recovery attacks [2]. These flaws point to a larger need for more robust and unpredictable key generation techniques.

In this paper, we suggest a new method to address these concerns. We are interested in utilizing the characteristics of SHA256 as a high entropy and collision resistance to produce relatively unpredictable keys so that AES symmetrical encryption technique is done in more complex way [3]. This is important to Image encryption as visual data security matters most.

This proposed method will improve the cryptographic mechanism by allowing AES and SHA256 to compete as analogs for encryption security of the solution. As proven and included, the technique is resistant to the majority of attack types due to the way it is structured, while also ensuring that the AES algorithm can run as quickly.

2. LITERATURE SURVEY

First, since the National Institute of Standards and Technology adopted the Advanced Encryption Standard in 2001, it has grown in popularity and importance as a foundation for cryptographic security. It employs symmetric key algorithms with key size options of 128, 192, or 256 bits[5]. However, the security of these complex algorithms is heavily dependent on key generation strength.

The NSA developed SHA-256 alongside AES; it is also one of the most widely used cryptographic hash functions [6]. SHA-256 is known to be value-resistant to assaults, particularly collision and preimage attacks, due to the hash's colliding value of 256 bits.

In order to increase security, researchers have attempted to relate hash functions that leak out with encryption. To improve system security characteristics, some researchers discovered that hashing prospective picture keys in addition to SHA-256 may effectively increase key entropy. Just the idea behind the strategy of integrating the hashing algorithm into an encryption system to address issues with conventional key generation techniques is presented.

This paper suggests a new use of SHA-256 to produce AES keys from input images based on these fundamentals. The goal is to generate highly secure cryptographic keys by utilizing the non-deterministic and high entropy characteristics of images. This study is in line with ongoing initiatives to provide robust encryption methods that will secure sensitive data in ever-changing threat landscapes.

3. RESEARCH METHODOLOGY

This research suggests a new scheme aiming at augmenting the AES algorithm through its keys' generation by applying SHA256 hash function to image- based inputs. The methodology is divided into three main stages: key generation, encryption and decryption.

1- Key Generation: An image is first converted into a byte array. The byte array is subjected to SHA256 hash function producing 256-bit hash. This hash serves as the key for the AES algorithm.

2- Encryption: Using AES algorithm, plaintext image data is encrypted with the key that was generated from SHA256 hashing process. Encrypting an AES involves substitution, permutation and mixing of the input data several times which makes it difficult for anyone to decrypt or understand any part of the encrypted picture.

3- Decryption: Using the same key generated from SHA256 hashing processes, encrypted image data can be decrypted back into their original formats so that one can read them as text again rather than seeing them in scrambled images as shown in figure (1).

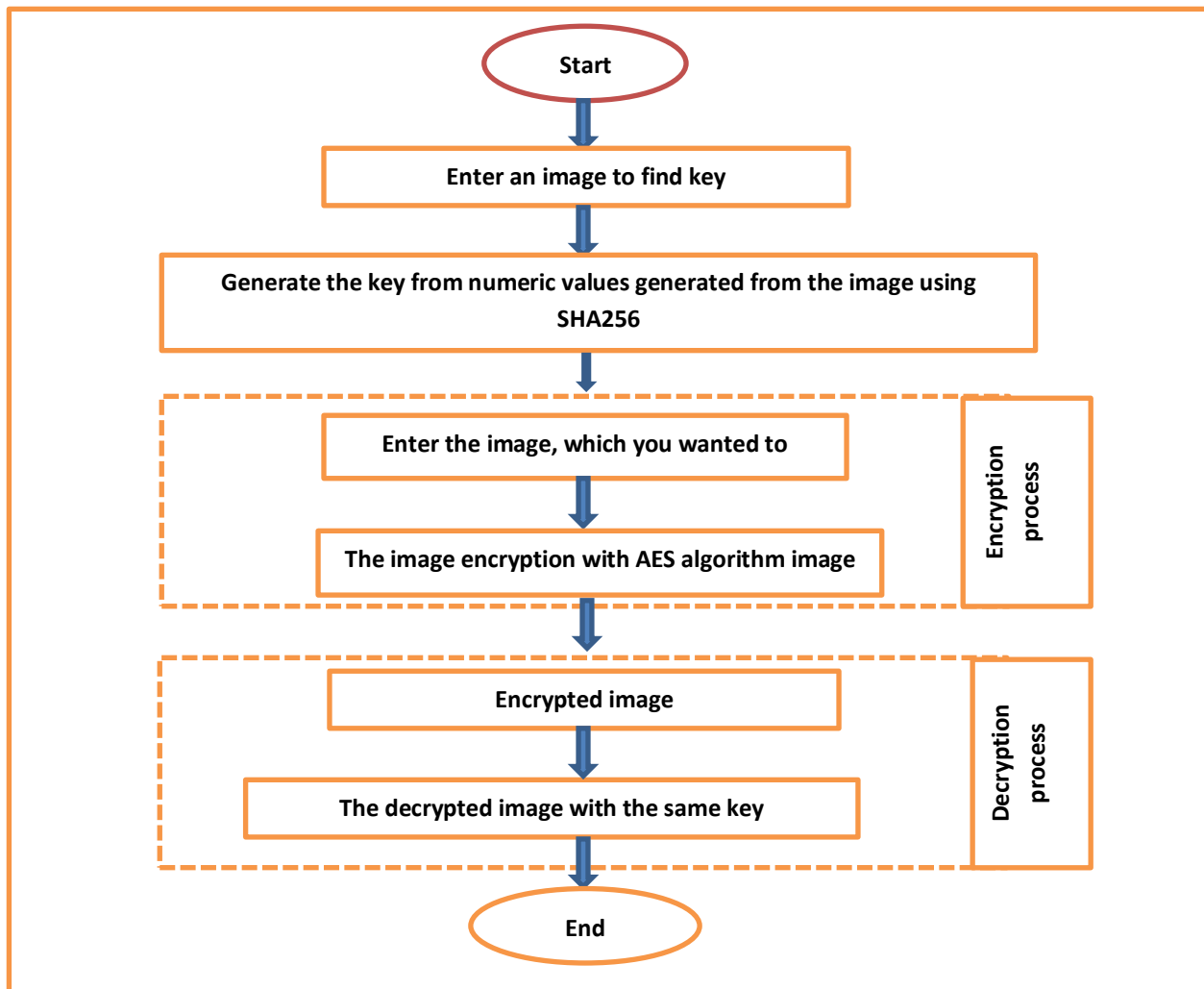


Figure (1): The proposed method

The implementation of this methodology is carried out using Python in the Google Colab environment. . The code creates functions for each step, including turning images into byte arrays, making SHA256 hashes, and performing AES encryption and decryption.

4. RESULT AND DISCUSSION

We tested this new method thoroughly to check how well it works and how secure it is compared to regular AES. We looked at how long it takes to encrypt and decrypt how it makes keys, and how well it stands up to attacks.

1. Security: The complexity and flexibility of the produced keys were evaluated throughout the security analysis. Because of the high entropy of the SHA256 algorithm, the keys are resistant to cryptanalysis and brute force attacks. Similarly, the use of image-based inputs to generate the keys added an extra layer of security against guesswork.

2. Performance: We measured the time required to encrypt and decrypt various-sized images. Because the new approach took longer to construct the key from SHA256, it was slightly slower than simple AES. It was discovered that SHA256 produced keys quickly—large images took only a few microseconds to generate.

3. Practical feasibility: We were able to use our methods via Python and Google Colab. It demonstrated practical uses. We ensured that our method could be simply replicated and adapted for different purposes by using commonly used libraries and easy coding techniques.

4. Comparative Analysis: We compared the results of our proposed method to standard AES. The results demonstrate that, while the encryption and decryption operations required a few extra steps, the greater security well outweighed the slower speed. It fared significantly better in terms of strengthening keys and improving encryption's overall robustness..

In summary ,the proposed method enhances the security of the AES algorithm by combining the AES algorithm with SHA256 to produce keys efficiently. Because SHA256-generated keys provide additional security, this method is very practical and practicable for protecting sensitive data in many digital applications. In Figure 2, the results of the recommended approach are displayed. The input images are shown in Figure 2, in the first column. The second column displays the encrypted photos, and the third column displays the decrypted images.


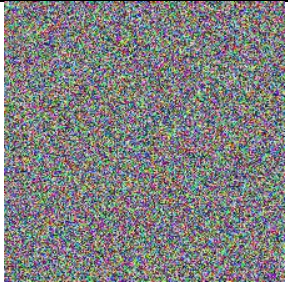


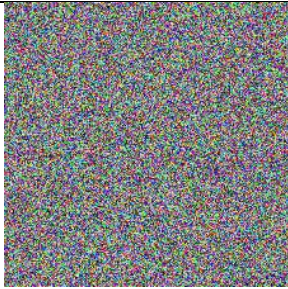



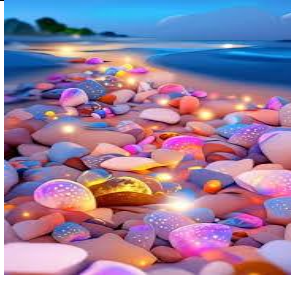
The Input Images	The Encrypted Images	The Decrypted Images
		
		
		

Figure 2, The results of the recommended approach

5. CONCLUSION

The research conducted has been able to improve the AES algorithm in the employed encryption method by generating SHA256 keys, resulting in enhanced security and efficiency. The suggested method produces satisfactory performance values while enhancing the complexity and attack resistance of the keys. The technique was developed in Python and tested in a Google Colab environment, confirming its viability in real-world applications. Future study could incorporate additional optimizations; however, this would extend the technique to encompass additional encryption algorithms, contributing more generally to secure data practices.

ACKNOWLEDGMENT

The author would like to thank Mustansiriyah University (www.uomustansiriyah.edu.iq),Baghdad - Iraq for its support in the present this work.

REFERENCES LIST

- 1) Dworkin, N., Sonmez Turan, M., & Mouha, N. (2023). Advanced Encryption Standard (AES).
- 2) Abdulla, M., & Rana, M. E. (2021). Vulnerabilities in public key cryptography. In Proceedings of the International Conference on Advances in Human-centric Computing (pp. 1-5). <https://doi.org/10.2991/ahis.k.210913.079>

- 3) Selvakumar, A. L., & Ganadhas, C. S. (2009). The evaluation report of SHA-256 crypt analysis hash function. In Proceedings of the 2009 International Conference on Communication Software and Networks (pp. 588-592). <https://doi.org/10.1109/ICCSN.2009.50>
- 4) National Institute of Standards and Technology (NIST). (2001, updated 2023). Advanced Encryption Standard (AES).
- 5) Daemen, J., & Rijmen, V. (2002). The design of Rijndael: AES—the advanced encryption standard. Springer-Verlag.
- 6) Dang, Q. (2015). Secure hash standard. National Institute of Standards and Technology. <https://doi.org/10.6028/NIST.FIPS.180-4>
- 7) Dobbertin, H. (1996). Cryptanalysis of MD5 compress.