National College of Ireland

Securing passwords storage using image steganography by implementing AES encryption and Argon2 hashing

MSc Research Project
Cyber Security

Shubham Karodimal Parakh
Student ID: X21154376

School of Computing
National College of Ireland

Supervisor: Dr. Rohit Verma

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Shubham Karodimal Parakh
Student ID:	x21154376
Program:	MSc in Cyber Security **Year:** 2022-2023
Module:	MSc Internship Project
Supervisor:	Rohit Verma
Submission Due Date:	15/12/2022
Project Title:	Securing passwords storage using image steganography by implementing AES encryption and Argon2 hashing
Word Count:	7350 **Page Count: 20**

I hereby certify that the information contained in this (my submission) is information about research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use another author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ……

Date: 14/14/2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on the computer.	□

Assignments that are submitted to the Programme Coordinator's Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Securing passwords storage using image steganography by implementing AES encryption and Argon2 hashing

Shubham Karodimal Parakh
X21154376

Abstract

With the development of technology and advancements in data transmission methods, there have also been developments in the methods used to conceal and forge sensitive data such as a password. We thus need more advanced and improved methods for transmitting and storing this password. In the globalized era, storing passwords is a crucial duty, and ensuring their safety and security is as crucial. Both steganography, as well as cryptography, have their own issues, despite being intended to secure passwords. The issue with cryptography techniques is that the cipher data pretends to be ambiguous, so the attacker would either interrupt the communication or carefully examine the data as it travels between the sender and the recipient. The problem with the steganography technique is that even if concealed information is discovered or even suspected to be present the message has already been revealed. The research presented in this paper demonstrates a novel password security method with different system configurations that combine advanced hashing techniques along with a strong encryption mechanism and steganography. This makes the password impossible to crack by an attacker. In the proposed hybrid system the password is first hashed using the Argon2 hashing technique, then encryption is done through Advance Encryption Standard (AES), & lastly saved in the image using steganography. This increases the security of the stored password. The suggested model also generates high-quality steganography pictures and integrates more quickly with efficient encryption and decryption time.

Keywords: *Hashing, AES Encryption, Cryptography, steganography*

1. Introduction

The most crucial problem nowadays is how to store passwords securely. Although technology advances daily, the password remains the most popular method for users to verify their identity. Several methods are involved in securely storing a such password. To enter any private network, a user must pass through the user authentication process. As outlined by (Venukumar and Pathari, 2016) the process of establishing a user's identification may also include verifying something that more than one individuals are aware of, this may include a password or something they own, like an access card, or another technique, like the fingerprint.

1.1 Background

Passwords like text passwords are becoming the most popular form of user identification in recent times. To circumvent it, different users register for the systems by entering their credentials such as username and password to sign in. To verify their identity, people must recall these credentials. It may be challenging for the individual to remember each of these passwords given how many user identities individuals have nowadays due to services like social networking, business emails, and other apps. People then attempt to use the same password across all of their accounts. The usage of several identification mechanisms, including fingerprint, RFID cards as well as other techniques, each of which is unique, endangers the security of these systems. Furthermore, the issue that concerns us is how safely user credentials are kept in the backend servers. Regardless of what security guidelines were followed, every system has a unique method for keeping passwords inside its database. Additionally, it depends on the kind of database management system being utilized. It's a great idea to save confidential information somewhere else. It has sometimes been proven that database systems aren't particularly secure. However, storing this away makes it more difficult for hackers to locate crucial information and launch attacks there. Numerous websites follow appropriate storage procedures, which preferably include encryption technology, strong hashing algorithms, and various measures like password splitting or rather salting. However, keeping and ensuring that in most of the setups, the password file is stored in the database can be a cause of stress. However, several well-known companies, like MySpace, Facebook, Twitter, Under Armour, Whatsapp, eBay, and many more, have experienced attacks and password breaches in recent years as described in (CardConnect, n.d.)

1.2 Motivation

People who aren't entitled to access crucial information from an organization often take advantage of the situation. (Yu and Huang, 2015) the study demonstrates how simple it is to break into a system using the many tools available online to launch cyberattacks against the first line of defense. Examples of attacks include dictionary types of attacks, hybrid kinds of attacks, and also common brute force attacks. Whenever it is believed that it is easy to perform such attacks, it becomes difficult for these companies to safeguard themself from such cyberattacks. Based on the notion that the system may be compromised, it might not be a good idea to rely upon those security measures. Observing how hackers are creating fresh, in-demand methods to compromise the system is also important. There are new techniques to defend the system against them, and many techniques that include cryptography and steganography are been developed to secure data transmissions online. (Forgáč et al., 2021a) explain that as steganography may be used in several circumstances, it is among the most widely utilized methods for data secrecy. We are familiar with the idea that each pixel inside an image, often known as a pixel, contains data that describes the picture. As a result, pixels in photographs include information that is buried, and a slight change made in a pixel cannot

be seen by the human eye. Despite how fantastic it seems, steganography is useless when data is buried in a single picture. Therefore, multiple-picture steganography was adopted, which conceals the data in several images as opposed to simply one. Multiple-picture steganography is considered an excellent approach to communicating secret information without arousing suspicion since it is difficult to locate secret data within a single image. Even if the attacker is aware that certain photographs contain secret information, it will be challenging to determine which particular image contains the data. But one way or another, the attackers continue to enter the system successfully.

1.3 Research Question

The following questions will be answered in the report

- How passwords can be secured by combining cryptography and steganography?
- Why argon2 hashing, AES-256 encryption, and LSB image steganography techniques are better for securing password storage?

1.4 Research Objective

It became crucial for security researchers as well as specialists to provide a more robust solution. Solutions that are challenging enough to be overcome yet adaptable enough to work in many circumstances. The solution must use existing security measures in a totally new way to boost security even further. The solution ought to be such that it complements the existing security measures and enhances them. A type of hybrid password security system and many levels of protection make it feasible.

The methodology used in this study takes into account all of these security measures of the cryptographic system in general. The research topic outlines a safe solution to store passwords utilizing image steganography in addition to cryptographic techniques like encryption and hashing. The proposed research provides the system with a high degree of security.

1.5 Structure of report

- Section 2 discusses the research articles and papers, that are published, what they conducted, and what they discovered about integrating steganography, cryptography, and other techniques like hashing algorithms and encryption strategies. We begin by examining some studies on image authentication mechanisms and steganography in our evaluation of the literature review. After that, there were studies that recommended integrating cryptography and security techniques like steganography. In the next part, we looked at research that compared and evaluated several encryption techniques, including argon2 hashing, and symmetric and asymmetric algorithms in connection to the concept of cryptography.
- Section 3 describes the suggested research method's usage of AES encryption, Argon2 hashing, and LSB-based image steganography to improve the security of the password along with the design in section 4. Section 5 put forward the implementation of these techniques along with their comprehensive evaluation in section 6 based on encryption time, decryption time, avalanche effect, and the throughput that are performed on a different system with a different configuration.
- The report concludes in Section 7 with ideas on how to make the suggested system better in the future.

2. Related Work

The concepts and fundamentals of past studies that researchers have conducted using a variety of approaches and procedures are presented in this section of the literature review. Among most significant portions of the relevant research publications will be discussed in this section, along with comparisons to other researchers' findings. The review will provide a quick overview of the studies conducted by several researchers that have combined image-based steganography techniques with encryption and hashing methods to save passwords.

2.1 Steganography to protect data

Nowadays there is a greater demand for even more secure and confidential methods to store and safeguard digital data due to the expansion of online communication technologies like the internet and cloud hosting. Authors(Torvi et al., 2016) discuss how steganography is regarded as a reliable method of protecting privacy. In order to create a system that transmits and receives encrypted communication that is concealed in text, this study suggests an alternative data security method that uses text steganography. It used the XOR method for encryption. Each communication the user sends will have a unique password that he may establish. Therefore, the method is a highly effective approach to safeguarding data. The proposed concept has a significant downside of requiring more memory plus bandwidth. As a result, (Marwa E. et al., 2016) have discussed the issue with text passwords as well as how steganography and encryption might be used to improve them. In order to address the issues and strengthen the authentication system against brute force attacks and password cracking, this article suggests a novel method of user authentication. By employing steganography, the password text is concealed inside a picture, which is then used to carry an encrypted message. Because of this, a brute-force attack is difficult to succeed. Additionally, this work discusses LSB steganography to conceal textual passwords and ensure the security of the authentication system. The encryption algorithm utilized is computationally and operationally slow. To get around this, (Baby et al., 2015) propose an alternative strategy that makes use of steganography and DWT-based picture security mechanism. In this research, the authors create a novel steganography technique that uses DWT steganography to conceal many RGB pictures inside a single RGB. The Peak Signal Noise Ratio (PSNR) & the Structure Similarity Index (SSIM) measurements are also used by the authors in order to distinguish the quality of both the steganography images with the source images. The authors noted that more data could be embedded through the use of compression and that their findings were superior to those that had previously been reported. However, in this case, the resolution of the photos is a concern since the stego-image that is produced is more warped than the original image. Finally, (Prashanti et al., 2017) carried out a study of the capabilities of LSB-based picture steganography in order to produce a high-resolution steganography image. The authors explore topics like increased resilience and increased embedding capacity, plus information that cannot be discovered as they relate to the advancements that have been made in the study of steganography. Data or personal information are concealed in the cover picture in the initial stage. The second stage involves hiding a concealed grayscale picture inside another grayscale image. These techniques produce numbers that appear to have been drawn at random using four-state tables. These two approaches have superior security since secret information is concealed on LSBs of the picture that are selected at random using pseudo-random integers generated by the table. Therefore, their method is secure and the steganography picture doesn't undergo numerous changes.

2.2 Authentication by using images

(Kumar et al., 2015) demonstrates how to utilize the frame Decomposition approach to conceal a picture on multiple-frame film. To conceal information in multiple films, authors employ the LSB technique. It states that after the steganography is finished, the exact identical picture will be extracted. The LSB method is utilized to conceal communications. The data or the picture remain unharmed and may be recovered in its current state even after various attacks are been made on the block. When used alone, the LSB technique may not be the most secure approach since it is susceptible to steganalysis. As a result, (Alias et al., 2015) develop a safe yet adaptive pixel-matching method that is effective for both color and grayscale pictures. The author also developed a technique for concealing audio, text, and other sorts of information in photographs. The writers encrypt the secret information using a bogus sequence. The authors also contrast some of the more traditional techniques, such as Diamond Encoding with the Least Significant Bit (LSB) replacement. A novel method to encrypt photos that focuses on a faster technique than what is presently utilized is suggested in a study published by (Moumen and Sissaoui, 2017) in order to further increase security. The authors' proposed approach eliminates the step in the encryption process when the secret key is exchanged. Theorems like symmetric encryption, steganography, and asymmetrical encryption method are used by authors. The authors first symmetrically encrypt the picture, then apply an asymmetric technique to encrypt the private key, and then utilize the least significant bit of steganography to hide the secret key inside an encrypted image. According to the author's study, the procedure is not only the fastest but also the most accurate. The authors of (Li et al., 2018) use batch steganography in order to further increase accuracy. The authors implement the answers to the issue, which include the most effective methods for embedding the payload and creating an original method for data slicing. The data is expanded and divided into several portions by the authors using a unique matrix procedure. The strength of the author's developed technique is that even if part of the photos is hijacked while being transmitted, the recipient may still decipher the secret information.

Authors (Preda and Vizireanu, 2015) employ DCT watermarking, which is compatible with JPEG image compression, for picture authentication. The digital authentication code is created using a fictitious random sequence which is based on both a block-dependent characteristic and encryption key. That protects the system against copy-and-paste assaults. Compression is also avoided by the JPEG matrix. It is not accurate enough to detect and thwart cyberattacks when watermarks cannot be seen. The authors (Sikder et al., 2017) developed a semi-fragile technique that utilizes Slant Transform or ST with LU Decomposition to circumvent this. In order to obtain the ST coefficients, the authors first applied ST to each frame. The authors were able to create a high-quality watermark picture that was secure to embed, extract, and validate. Researchers (Wahid et al., 2018)used a similar strategy, but combined MD5 hashing with LSB substitution inside the pixels which are chosen in the photos. Authors may determine a digital signature using MD5 by looking at certain pixels. The above-mentioned technique had several computational errors that jeopardized the system's integrity. In order to solve this problem, the authors (Forgáč et al., 2021b) developed steganography-based picture authentication. To enable it to work, researchers use neural networks, symmetric encryption, plus a strong cryptographic hash algorithm. A crucial component of the proposed system is OM-PCNN. The authors create the neural network parameters using a steganographic base key and encode the authentication data using the AES technique to make the system more secure. Additionally, researchers employ the SHA-2 algorithm using a 512-bit hash to verify the image's integrity.

2.3 Data security using hashing and salting

(Pritesh, 2013) demonstrate how passwords and any other confidential information may be stored using the salt hash approach. The most effective method of data security, according to the researchers, is salting. It also explains how using hashed passwords might reduce the likelihood of dictionary attacks. To defeat hashing, utilize the dictionary attack. One can prevent these issues by using the salt technique. Even though it is given to each individual randomly in this research, the Salt is created only once and then saved as a specified quantity in the database. Thus a result, the Salt is simple to locate and may be used to decipher the password in the event that a hostile person hacks the database system. In order to protect the password from being used even if an attacker gets unauthorized access to the database, the researchers (Jeong et al., 2019) suggest a strategy. The authors claim that as the computers start getting more powerful, the MD5 and even SHA-1 hash algorithms have begun to overlap. The researchers utilized a random Salt as both a solution to safeguard the hashing method and in order to address issue. By including the password at the beginning or just at the end, the Salt algorithm is incorporated to the hash algorithm. The authors draw attention to the problem that if the database gets compromised, it is simple to locate Salt and use it to determine the credentials. Access salts are always changing, making passwords more secure and making it more difficult to determine what the salt is.

2.4 Evaluation of several hashing and encryption methods

When more information is sent over computers or kept there, it's critical to ensure that it cannot be altered or intruded. (Patil et al., 2016) recommend that future studies examine the data security mechanisms of several encryption algorithms, including RSA, DES, triple DES, as well as AES. According to analysis, the avalanche effect, the calculated throughput, and the speed are all best handled by the AES algorithm. The authors get to the conclusion that RSA, ECC, as well as DSA, were slower than symmetric methods. Also, compared to DES and AES, the encryption algorithm that was least secure was the RSA algorithm which is shown in Figure-1. However, combining hashing and encryption makes these techniques considerably safer. In order to construct a mobile application, the author (Levent Ertaul, 2016) analyzed and contrasted the efficiency of hashing algorithms that included PBKDF2, Bcrypt, and Scrypt. This program analyzes the time and difficulty involved in creating a password hash generation. Bcrypt is the slowest since it employs blowfish, although PDKDF2 is the quickest but can be broken. But the authors come to the conclusion that the Bcrypt algorithm or Scrypt is almost hard to break because of its computational strength. Similar research was conducted by (Biryukov et al., 2016) who suggested Argon2 as the forthcoming memory-hard function which utilizes password hashing. According to the study, argon2 uses low entropy and requires a certain amount of system memory to operate quicker than previously used algorithms such as Bcrypt and Scrypt algorithm. Argon2 is the most efficient in terms of speed and security since it has the lowest total ASIC and botnet resilience which is around
0.6 cycles per byte.

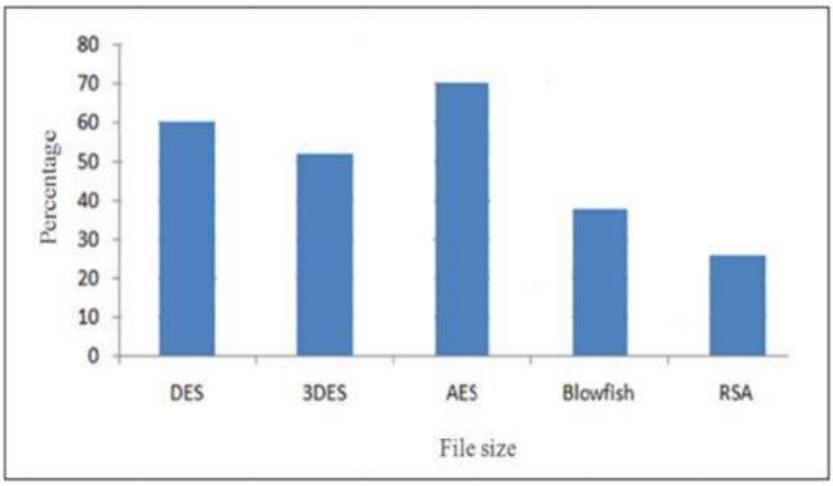

Figure 1- Comparative Study of AES, DES, 3DES, Blowfish and RSA (Patil et al., 2016)

Additionally, the study by (Cordova et al., 2017) compares the effectiveness of a few algorithms, particularly AES (Rijndael), Blowfish, and RSA. The findings indicate that, when tested with different data loads or even memory sizes, AES had a superior time efficiency ratio than Blowfish and RSA. The authors examined key generation time, time for encryption, and time for decryption, and in every test, AES achieved the smallest amount of processing time. Even if the simulation just looked at performance and efficiency, the technique is still highly promising and might be used to protect network and cloud services. The influence of each strategy on the efficiency of the algorithm in national security applications is provided by the cryptanalysis discussion on the AES algorithm carried out by (Kaminsky et al., 2010). According to their study, cryptanalysis over AES is improving. Furthermore, due to the fact that the development is being made in plain view, care is advised. The findings demonstrate that AES is presently exposed to several side-channel attacks. But there are suitable countermeasures that, when used correctly, may get rid of these weaknesses.

2.5 Hybrid Model

Data security is made safer and more impenetrable via hybrid models, which include hashing and encryption techniques. (. et al., 2016)provide a similar technique for protecting cloud storage. For all of this, the authors combine the SHA256 hashing algorithm with two encryption algorithms: RSA for the asymmetric type of encryption and Advance Encryption Standard (AES) for the symmetric type of encryption. The model yields a 32 percent result for encryption algorithm using a mix of RSA and AES, and around 33 percent result for decryption. Additionally, the approach offers non-repudiation, secrecy, and integrity for safe cloud storage. The proposed system by them for encryption phase and decryption phases takes up the majority of its execution time. As a result, Bcrypt and the AES algorithm were used in the new model that (Naved, 2021)and his colleagues developed. According to this research, the Bcrypt algorithm is effective against Brute Force attacks for hybrid characters but not for characters which are both numeric and alphabetic. (Joshi et al., 2022) suggest another study on hybrid data encryption that makes use of batch steganography for safe data transport. The payload within the cover picture is encoded using a password. Information is twice encrypted by making use of XOR technique. As a result, the information is more secure, and its almost difficult to decipher since neither the password nor the encryption algorithm

are known. The info of each pixel in the main picture is encrypted using the LSB method. Data theft and MITM (Man in the Middle attack)become impossible as a result.
According to research, each of the numerous encryption and hashing schemes suggested above for a secure password storage process has its own downsides and limits. Text steganography is used by the researcher (Torvi et al., 2016) to protect data by using the XOR approach, which uses greater memory and bandwidth. Researchers (Prashanti et al., 2017) address the resilience and embedding capability of the model utilizing random numbers, which lengthened the computing time. They employ LSB base approach of image steganography. Although there was a security risk, (Alias et al., 2015) suggested an adaptive strategy to match each pixel to conceal different types of data. In an effort to increase this accuracy, the authors (Li et al., 2018) used the batch steganography technique. Sikder et al. (2017) carried out research that employed the ST and LU-based decomposition techniques to secure transmitted data. However, the method is insufficiently safe to extract and validate the data. Researchers developed picture authentication utilizing the steganography approach and techniques like symmetric encryption since the hashing and replacement used in the work by Wahid et al. (2018) lacked integrity. In another research s acting was found to be not the ideal strategy, and it may also be often used to decipher the password. (Jeong et al., 2019) provide a fresh approach that utilizes MD5 and SHA-256 techniques of hashing to stop this. Argon2 was discovered as the strongest hashing method for securing the data by (Biryukov et al., 2016). Furthermore, the hybrid paradigm was presented for data security, using AES and SHA-256. It was discovered that the best encryption algorithm is AES. Researchers proposed the use of the XOR technique in conjunction with a strategy that produces superior results but is subject to man-in-the-middle attacks.
After going through the literature, the proposed model makes use of Argon2 hashing technique and AES encryption algorithm for safe password storage employing steganography in order to increase both securities of the password-storing process as well as its speed.

3. Research Methodology

This section presents four steps for implementing the suggested approach which involves hashing, encryption algorithm, LSB image-based steganography, and then finally decryption. The password is hashed using the Argon2 hashing method in step one. The hashed password then gets encrypted using the AES-256 technique in step two. The password that is encrypted will be hidden behind the pictures using an extra layer that will be AES-
256 encryption when performing step three, when we will apply LSB-based image steganography. The password will be extracted during the last and fourth procedures by using decryption.
It is abundantly clear from the analysis that was done as the step of the procedure of reviewing the appropriate literature review that the only way through which the security of the password can be increased is by using stronger encryption techniques and hashing algorithms like AES256 as well as Argon2, and finally combining these methods with extra security measures like LSB based steganography. The study focuses on the security of the password storage system by highlighting the hashing, encryption, and LSB image steganography techniques used to store sensitive data.

3.1 Argon2 Hashing

Any sort of data may be converted into string text using a method of cryptography called hashing. Utilizing methods that prevent unauthorized individuals from obtaining passwords is crucial for enhancing security and preventing data breaches. Passwords are protected using a

variety of hashing algorithms, including MD5, SHA-1 hashing, SHA-256, PDKDF2 hashing technique, Bcrypt, Scrypt, and argon2. For systems that consist of multiple core GPUs, PDKDF2 may be parallelized fast, and it is simple to do so on systems with specialized requirements, such as FPGAs. Techniques like BCrypt are more resistant to GPU attacks than PBKDF2, however, you ought to only utilize it on more recent computers since it performs poorly in cases of offline cracking. The research done by (Pritesh, 2013) and (Biryukov et al., 2016) shows that due to its superior architecture, particularly in terms of memory hardness, Argon2 is a better choice than SCrypt and BCrypt. Argon2 comes in two different forms: argon2I and argon2id. The algorithm Argon2i uses a memory access method that is more effective in hashing passwords because it is independent of the contents. Argon2id functions similarly to Argon2i but only in the first half before switching to Argon2d in the second. Due to trade-offs between time and memory, Algorithm Argon2id defends from side-channel attacks and also reduces the chance of brute-force assaults. Therefore, as shown in Figure 2 the methodology suggested in this work, argon2id is utilized to perform hash the passwords.

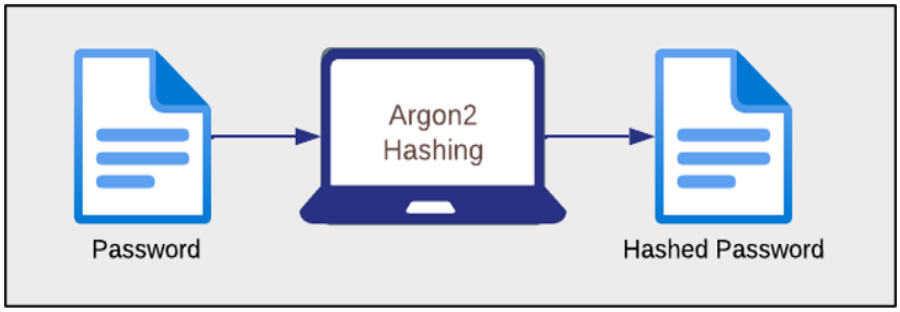

Figure 2- Hashing of password using Argon2

3.2 AES-256 Encryption

The concept of encryption is not new anymore. It is considered to be the most effective and crucial approach to prevent hackers from accessing sensitive data, such as our password (Kaminsky et al., 2010), (Cordova et al., 2017). Password encryption alone, however, is insufficient. The authentication process needs to employ a better algorithm which needs to be more exact. The algorithm approach must be flexible enough to adapt to changing conditions and robust enough to be implemented. Since it has been demonstrated to be the finest option when data integrity and secrecy are taken into account, we adopt the AES-256 encryption technique in our suggested approach. The quantity of rounds is also a crucial feature used by AES and lowers computing costs as well. The number of repetitions in AES is determined by how long your key is which is to be utilized. AES uses keys with various bit counts, including 128 bits keys, 192 bits keys, and 256 bits keys coupled with 10 rounds, 12 rounds, and 14 rounds, accordingly. The DES, 3DES, RSA, and Blowfish encryption algorithms are less effective and less practical than AES. Additionally, it is believed that the only way to defeat AES is with a brute-force assault. So, we used AES-256 encryption on hashed passwords in our proposed model as shown in Figure-3.

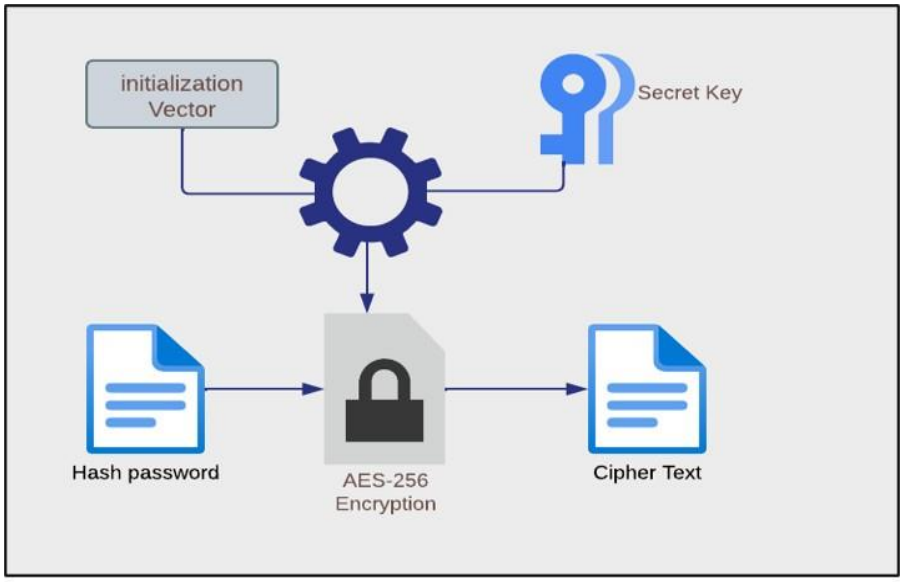

Figure 3– Encryption of hash password using AES-256

3.3 LSB Image Steganography

Steganography's main goal is to deliver sensitive data to the individual or organization it is intended for under the guise of a cover picture file. Many individuals think that photographs are the most efficient medium when steganography is considered. Hence, the proposed system uses LSB base image steganography to store the cipher text (Hash + Encrypted password). As the picture becomes less likely to catch the attention of prospective attackers the more detailed it seems. The proposed approach exclusively uses high-quality photos, which allows it to store more information without sacrificing the image's overall quality.

3.4 Decryption

In the decryption phase as shown in Figure 4 we first extract the stored cipher text from the image using a decoding function from LSB image steganography which pulls out the binary data from the corresponding image. In the next stage, AES-256 decryption is performed on it by using a decrypt function which requires a key and initialization vector (iv) using the AES-CBC method. Finally, this decryption gives us the hash value of the password. This hash password is then verified with the hash password provided by the user using verify function from argon2.

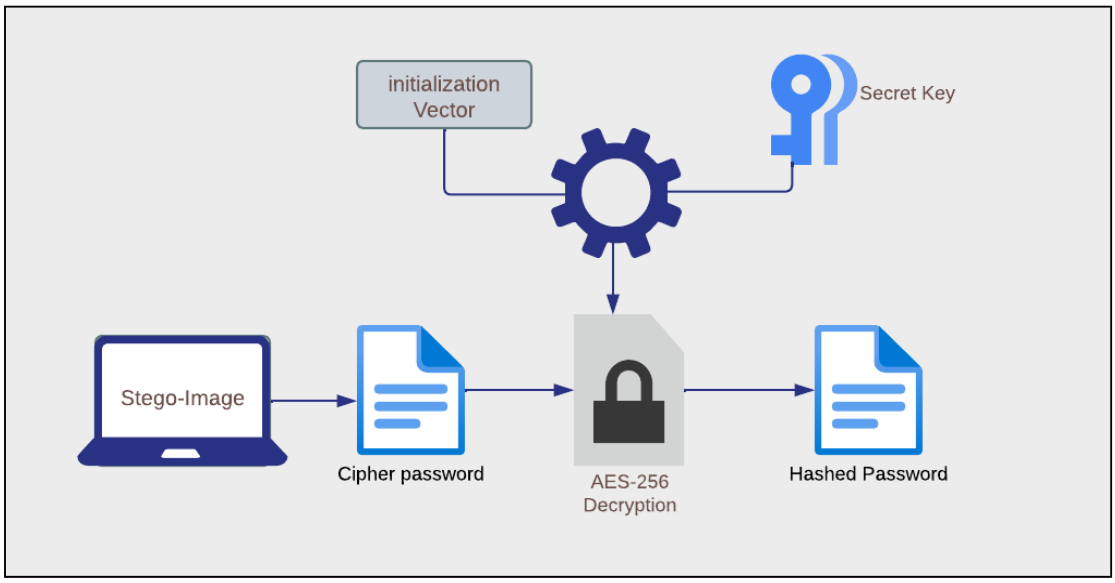

Figure 4 – Decryption Process

4. Design Specification

We did not alter the presently used encryption techniques for the suggested system. Rather, we utilized the same techniques with other algorithms at various program phases to add a further security layer for our developed system and store them at various places as our steganographic pictures. We developed a new security formula from our research: Password Protection = Hash + Encrypt + Hide. The suggested system's architecture is shown in Figure 5 below.

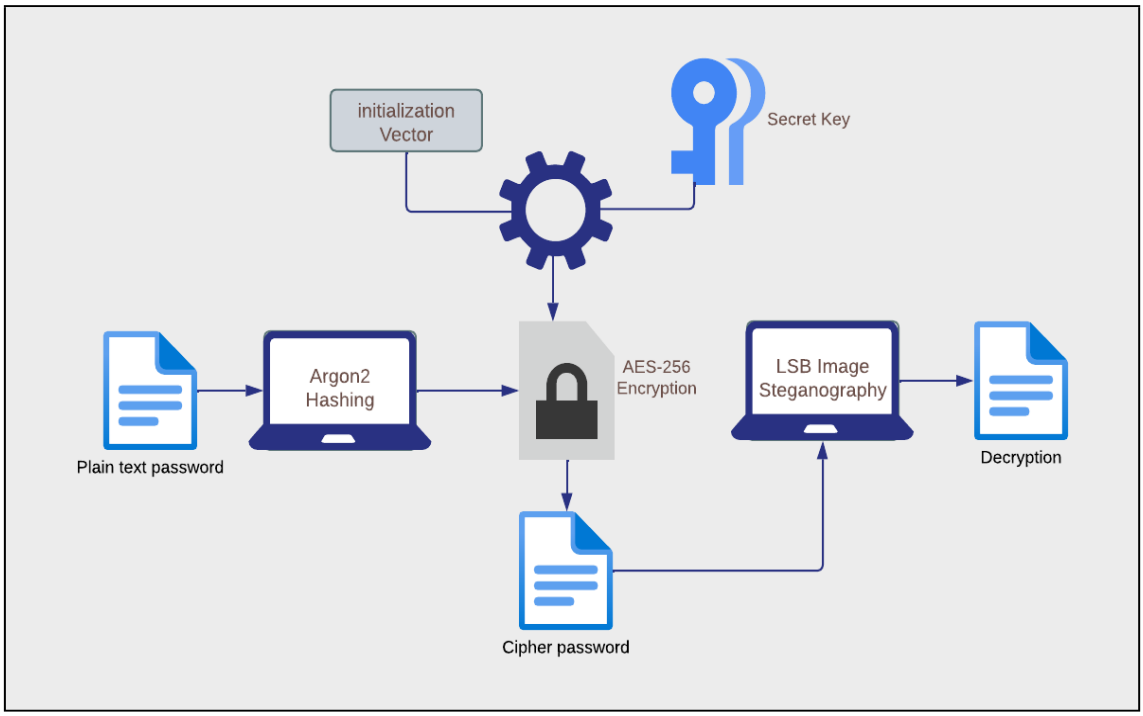

Figure 5- System Architecture

By doing this, we ensure that the system remains secure even if a malicious user is successful in breaking through it. The information is encrypted utilizing the AES-256 algorithm, which indeed makes use of an iterative approach. It also uses a block size which is of 128 bits and a key of 256 bits which requires fourteen rounds for the key, making it extremely difficult for a malicious user to decrypt the information even if they manage to crack the system and gain access to the stego-images. The difficulty of the cryptography methods also grows as the key size does. Additionally, they would still be unable to decipher any password in its entirety. The algorithm we used for the suggested system is shown in the section that follows.

4.1 Algorithm for the proposed System

Step-1 : Start
Step-2 : Take a single password from a list of passwords file.
if a password exists in the file move to Step-3
else,
show message – No password found in the file
Step-3 : Apply argon2 hashing by using the PasswordHasher library of python.
Step-4: Apply AES-256 bit encryption for the password hash in Step-2 using the key and an Initialisation vector (iv) generated randomly.
Step-5: Apply LSB-based image steganography to the hash and encrypted password and store the ciphered password inside an image.
Step-6 : Extract the cipher data from the stego-image using the decode function if data exists in the image move to Step-7
else,
Display message- No data inside the image
Step-7 : Apply AES-256 bit decryption to the extracted cipher data using a key and an initialization vector to get the hash password
Step-8 : Compare the hash password obtained using verify library of argon2
if,
Hash matches Display “Hash Verified ” message
else,
Exit the program
Step-9 : Perform steps 1 till step 8 for the consequent password in the file

Step-10: Exit

5. Implementation

In the following section, we go through the proposed system and its implementation process as well as flow. Finding a method to secure passwords utilizing strong encryption techniques and password-storing algorithms on steganographic pictures was the main objective of the research. We needed Python version 3.10, visual studio code a particular Development Environment in order to display the results of our system in the terminal and show the operations and outcomes at each stage.
A list containing user passwords, as well as a few pictures for steganography, are the two pieces of information needed to put this research into practice. Based on the output that is our steganographic pictures and the ability to decode and combine the encrypted passwords using the right sequence with the right components, the results were evaluated. Stage I of the procedure involves selecting one password from a list of password-containing files. The password is taken out of the password file and hashed using argon2 hashing methods. In Stage II, a key and an initialization vector are used to encrypt the hashed password using

AES-256 algorithm. Additionally, the system automatically salts the password. Stage III involves the use of LSB-based image steganography, in which a hashed and encrypted password is hidden within a stego-picture. The cipher text contained within the stego-image is retrieved in the last step, commonly known as the decryption phase, and then decoded using AES-256 decryption to provide a matching hashed password. We then go on to the next password on the list once this hashed password has been confirmed.

Figure 6: Stage 1 - Argon2 Hashing

Pseudo code for argon2:

Input: userpass
Steps: Argon_hash(userpass)
Passwordhasher()← userpass
Hashvalue ← Passwordhasher.hash(userpass) // perform hash using passwordhash
Output: hash password

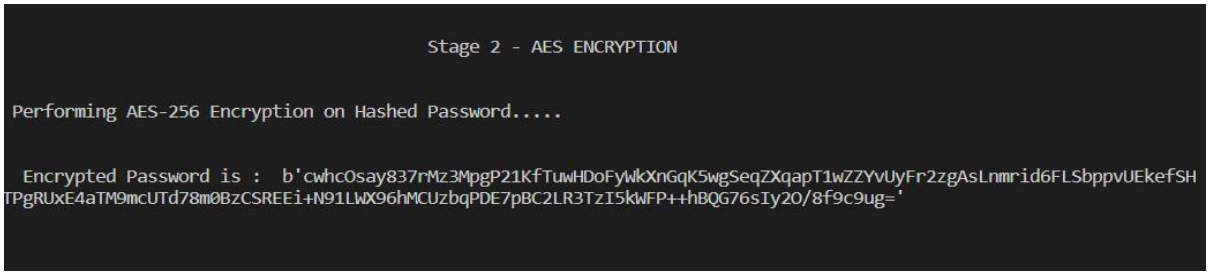

Figure 7: Stage 2 – AES-256 Encryption

Pseudo code for AES-256:

Input: hash_pass
Steps: AESCipher ← aes_key
encrypt() ← aes_key, hash_pass
raw ← pad(hash_pass) //padding for the hash password
iv ←Random_generator().read(AESblock_size) //generating the iv for the block size
cipher ← AES_Encrypt(key, AES_MODE_CBC, iv)
Enc_hash_pass ← cipher
Output: Encrypted hash password

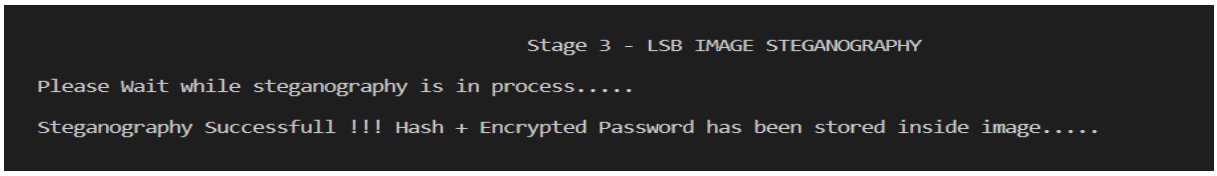

Figure 8: Stage 3 – LSB image Steganography

Pseudo code for Steganography:

Input: Enc_hash_pass
Steps: Stegno_encode() ← Enc_hash_pass
Img_read(), img_open() // reading and opening of the original image
enc_data ← hide data(image, data) // hiding the cipher data
stego_image ← imwrite(enc_img, enc_data) // Stego image/
img ← img1.resize((w, h),Image.ANTIALIAS) //Opening the stego image and resize
Output: Stego_image

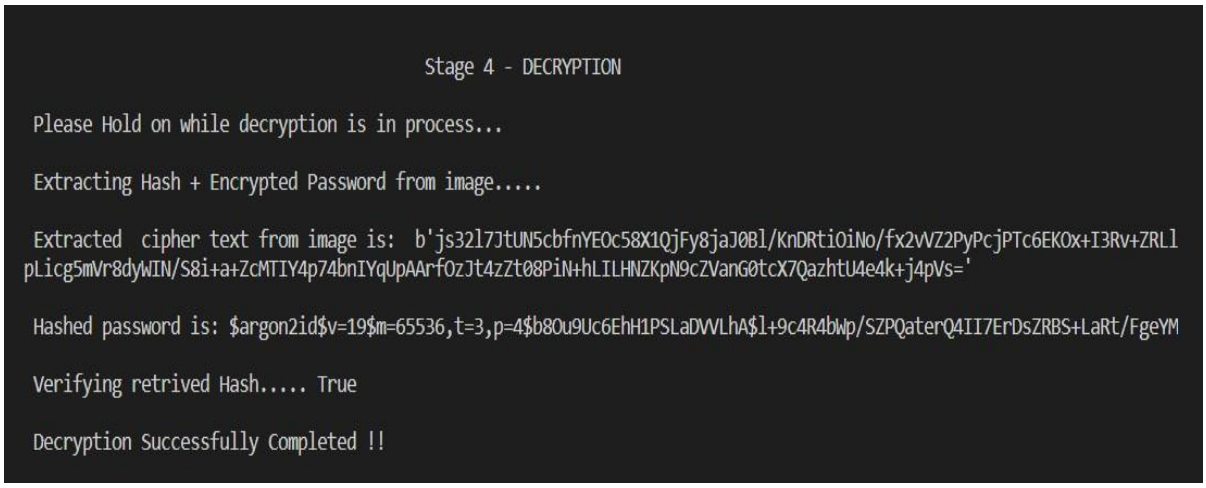

Figure 9: Stage 4 – Decryption

Pseudo code for Decryption:

Input: stego_image
Steps: dataFromSteg = stegno.decode()
msg = find_data(image)
AESCipher ← aes_key
encrypt() ← aes_key, msg
raw ← unpaid(msg) //unpadding for the hash password
iv ← enc(AES.block_size)
hash_password ← AES_Decrypt(key, AES.MODE_CBC, iv)
hash_verify() ← hash_pass // verify hash with original hash
Output: hash_pass

6. Evaluation

The following section put forward the details about the suggested model that considers different the performance metrices. We also applied our suggested model in systems with various configurations and tracked the corresponding performance data to assess it. To achieve it, we take into account different performance metrics like encryption as well as decryption time with varying passwords and key sizes. We are also considering execution time and the Avalanche effect on passwords.

6.1 Experiment 1

In experiment 1 we implemented the model proposed in the system with Intel(R) Core(TM) i7- 8250U CPU@1.80 GHz, 16 GB RAM, Windows 11 Home Single Language and 64-bit OS, with NVIDIA GeForce MX130 graphics card. The performance metrics graphs are shown in figure 10,11 and 12. Here the encryption time is the time required to perform hashing, AES- 256 encryption, and LSB image steganography. The results are mentioned in figure 10,11 and 12 below

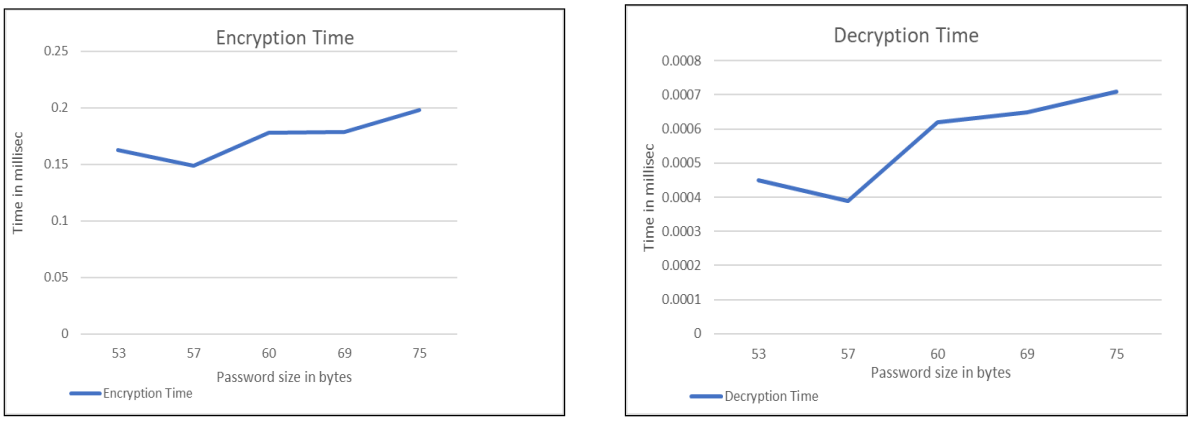

Figure 10- Analysis of Encryption and Decryption Time with varying password sizes

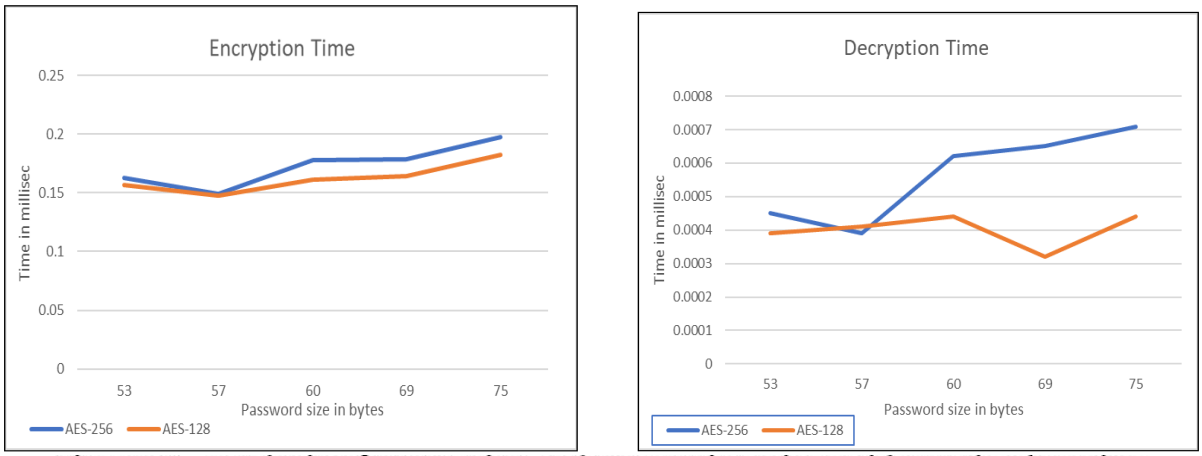

Figure 11- Analysis of Encryption and Decryption Time with varying key sizes

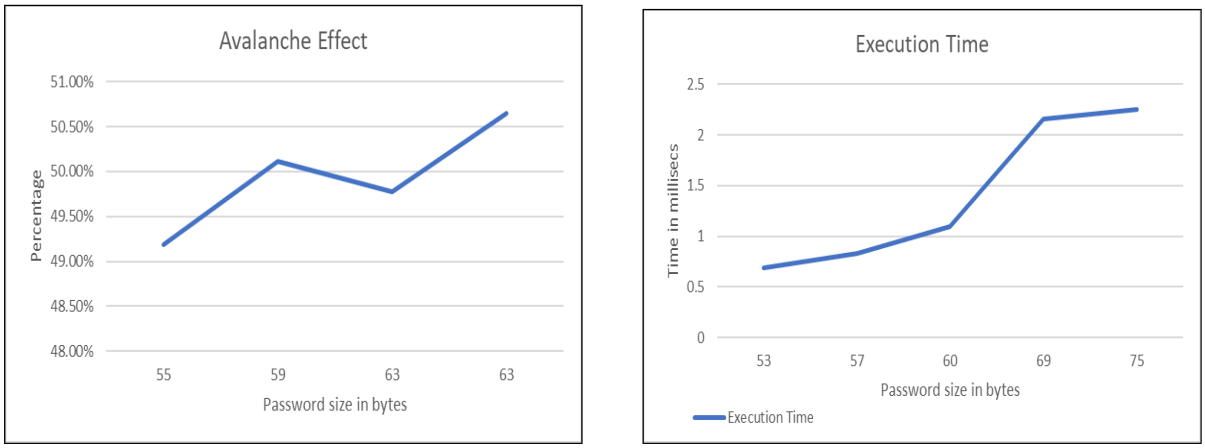

Figure 12 – Analysis of Avalanche effect and execution time on varring password sizes

6.2 Experiment 2

In experiment 2 we implemented the model proposed in the system with Ryzen 5 Hexa Core AMD R5-5600H, 8 GB RAM, Windows 11, 512 GB SSD, with NVIDIA GeForce RTX
3050. For this experiment, we have used same password list that was used for experiment 1. The results thus calculated are shown in below figures 13,14 and 15

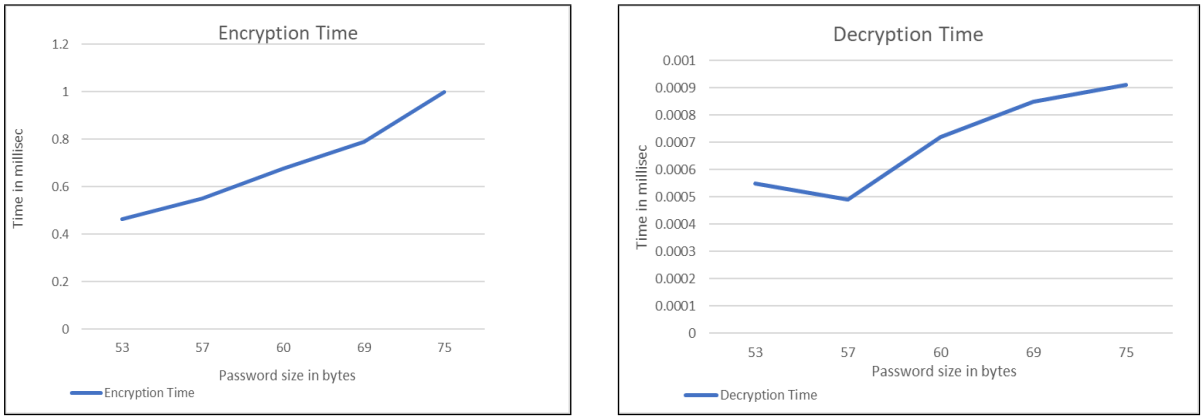

Figure 13- Analysis of Encryption and Decryption Time with varying password sizes

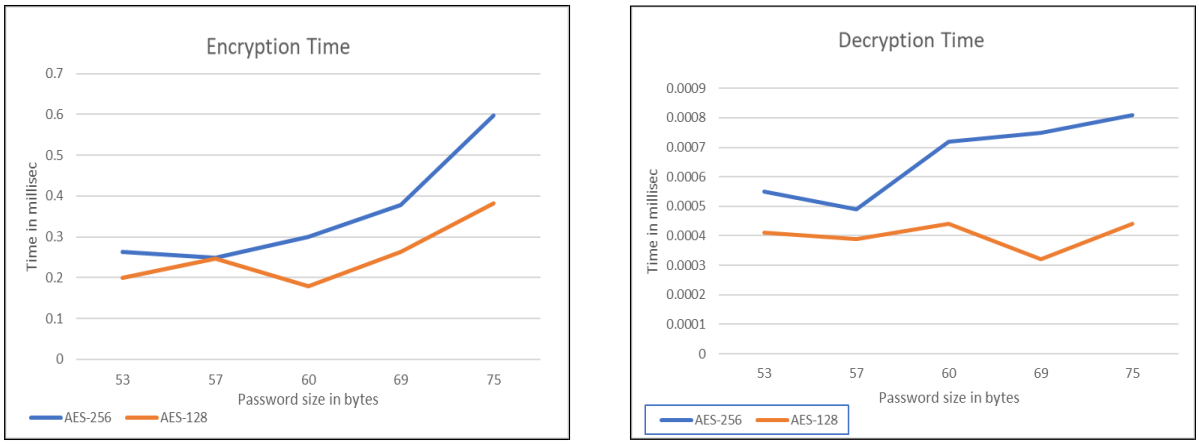

Figure 14- Analysis of Encryption and Decryption Time with varying key sizes

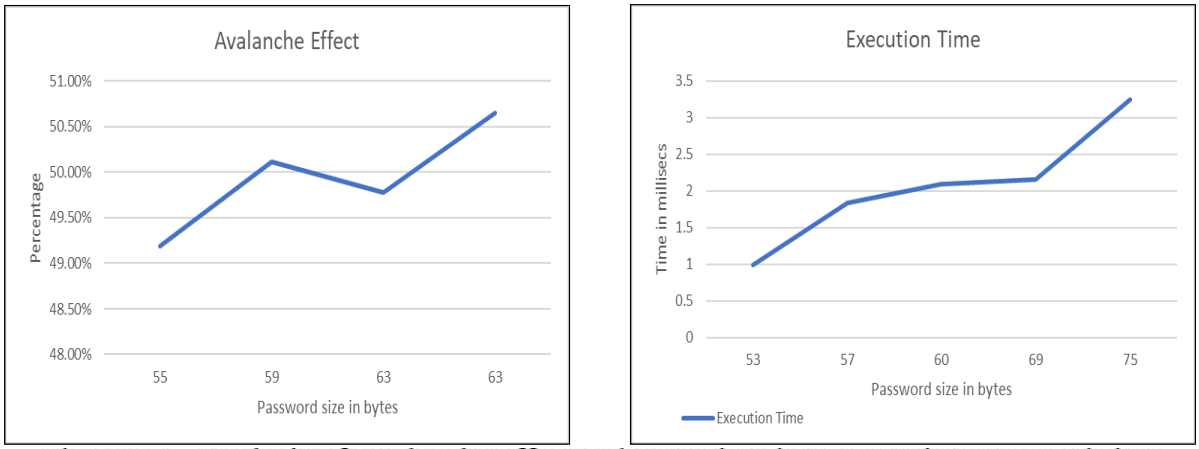

Figure 15 – Analysis of Avalanche effect and execution time on varring password sizes

6.3 Discussion

The evaluation done above on our proposed model found out that integrating Argon2 hashing, AES-256 encryption, and LSB image steganography together gave us better security than the hybrid model that used SHA-256 hashing and AES encryption along with LSB image steganography proposed by (Joshi et al., 2022) in terms of Encryption time and decryption time but it requires multiple attempts to get a precised avalanche effect on the passwords. In the proposed model the encryption time which is the time taken to perform hashing, encryption, and image steganography has obtain better results for different password sizes and key sizes. The varying key size which was AES-128 and AES-256 shows that the lesser the key size the less is the encryption and decryption time taken. But the less key size implies less security and the passwords can thus become more prone to brute force and dictionary attacks as the key space decreases. To prevent this and increase security we store encrypted passwords inside images. In the decryption process, we just compare the obtained hash with the previous hash and hence the decryption time thus obtain is quite less as compared to the encryption time. Additionally, using multi-layered encryption and better encryption techniques like AES-256 makes it more difficult for malicious users to get into the system. Also, the performance of the technique, which implements the proposed solution and executes the operations on high-quality photos, is not directly impacted by different system configurations. The machine needs at least 8GB of RAM and a speedier CPU, such as an intel i5 processor or above. However, experiment 2 demonstrated that the system's performance was adequate even with a lesser configuration only the execution time was better on the system with a higher configuration.

7. Conclusion and Future Work

The major goal of this study was to enhance the security of passwords by combining security techniques like LSB image steganography, AES-256 encryption and Argon2 hashing on private data, in this instance, a password. To improve the security even more we have base64 encoded the key used for encryption. Additionally, using multi-layered encryption and better hashing techniques makes it more difficult for hackers to get into the system. A different system configuration is needed to implement the proposed system and execute the operation in a more efficient way on high-quality images. Different configuration of the system directly influences the overall performance of our proposed model. As a result, it is strongly advised to use computers with better configurations to enhance the performance of the suggested solution. This study comes to the conclusion that further protection for passwords can be improved by using a hybrid model that uses Argon2i, AES and image steganography methods.

In future work, we can integrate the proposed model for real-time systems, we can also consider updating passwords dynamically, the model uses a limited environment setup that can be upgraded, and integrating the proposed model ERP system is also one of the options. Finally, we can consider a huge list of passwords on which the proposed model can be implemented for enhancement.

References

. N., Omara, F., Omran, N., 2016. A Hybrid Hashing Security Algorithm for Data Storage on Cloud Computing. https://doi.org/10.13140/RG.2.1.4103.3844
Alias, T.E., Mathew, D., Thomas, A., 2015. Steganographic Technique Using Secure Adaptive Pixel Pair Matching for Embedding Multiple Data Types in Images, in: 2015 Fifth International Conference on Advances in Computing and Communications (ICACC). Presented at the 2015 Fifth International Conference on Advances in Computing and Communications (ICACC), pp. 426–429. https://doi.org/10.1109/ICACC.2015.60
Baby, D., Thomas, J., Augustine, G., George, E., Michael, N.R., 2015. A Novel DWT-Based Image Securing Method Using Steganography. Procedia Computer Science, Proceedings of the International Conference on Information and Communication Technologies, ICICT 2014, 3-5 December 2014 at Bolgatty Palace & Island Resort, Kochi, India 46, 612–618. https://doi.org/10.1016/j.procs.2015.02.105
Biryukov, A., Dinu, D., Khovratovich, D., 2016. Argon2: New Generation of Memory-Hard Functions for Password Hashing and Other Applications, in: 2016 IEEE European Symposium on Security and Privacy (EuroS&P). Presented at the 2016 IEEE European Symposium on Security and Privacy (EuroS&P), pp. 292–302. https://doi.org/10.1109/EuroSP.2016.31
CardConnect, n.d. The 10 Biggest Data Breaches [With Chart] [WWW Document]. CardConnect. URL https://cardconnect.com/launchpointe/payment-security/10-biggest-data-breaches (accessed 12.2.22).
Cordova, R.S., Maata, R.L.R., Halibas, A.S., Al-Azawi, R., 2017. Comparative analysis on the performance of selected security algorithms in cloud computing, in: 2017 International Conference on Electrical and Computing Technologies and Applications (ICECTA). Presented at the 2017 International Conference on Electrical and Computing Technologies and Applications (ICECTA), pp. 1–4. https://doi.org/10.1109/ICECTA.2017.8252030
Marwa E., ,Fatma F., 2016. Data Security Using Cryptography and Steganography Techniques. ijacsa 7. https://doi.org/10.14569/IJACSA.2016.070651
Forgáč, R., Očkay, M., Javurek, M., 2021a. Steganography Based Approach to Image Authentication, in: 2021 Communication and Information Technologies (KIT). Presented at the 2021 Communication and Information Technologies (KIT), pp. 1–6. https://doi.org/10.1109/KIT52904.2021.9583618
Forgáč, R., Očkay, M., Javurek, M., 2021b. Steganography Based Approach to Image Authentication, in: 2021 Communication and Information Technologies (KIT). Presented at the 2021 Communication and Information Technologies (KIT), pp. 1–6. https://doi.org/10.1109/KIT52904.2021.9583618
Jeong, J., Woo, D., Cha, Y., 2019. Enhancement of Website Password Security by Using Access Log-based Salt, in: 2019 International Conference on Systems of Collaboration Big Data, Internet of Things & Security (SysCoBIoTS). Presented at the 2019 International Conference on Systems of Collaboration Big Data, Internet of Things & Security (SysCoBIoTS), pp. 1–3. https://doi.org/10.1109/SysCoBIoTS48768.2019.9028012
Joshi, R., Bairwa, A.K., Soni, V., Joshi, S., 2022. Data Security Using Multiple Image Steganography and Hybrid Data Encryption Techniques, in: 2022 International Conference for Advancement in Technology (ICONAT). Presented at the 2022 International Conference

for Advancement in Technology (ICONAT), pp. 1–7. https://doi.org/10.1109/ICONAT53423.2022.9725949
Kaminsky, A., Kurdziel, M., Radziszowski, S., 2010. An overview of cryptanalysis research for the advanced encryption standard, in: 2010 - MILCOM 2010 MILITARY COMMUNICATIONS CONFERENCE. Presented at the 2010 - MILCOM 2010 MILITARY COMMUNICATIONS CONFERENCE, pp. 1310–1316. https://doi.org/10.1109/MILCOM.2010.5680130
Pritesh, P., 2013. A Cryptography Application using Salt Hash technique. Avilable at: https://www.ijaiem.org/Volume2Issue6/IJAIEM-2013-06-22-060.pdf
Kumar, S., Yadav, A.K., Gupta, A., Kumar, P., 2015. RGB image steganography on multiple frame video using LSB technique, in 2015 International Conference on Computer and Computational Sciences (ICCCS). Presented at the 2015 International Conference on Computer and Computational Sciences (ICCCS), pp. 226–231. https://doi.org/10.1109/ICCACS.2015.7361355
Li, F., Wu, K., Zhang, X., Yu, J., Lei, J., Wen, M., 2018. Robust Batch Steganography in Social Networks With Non-Uniform Payload and Data Decomposition. IEEE Access 6, 29912–29925. https://doi.org/10.1109/ACCESS.2018.2841415
Moumen, A., Sissaoui, H., 2017. Images Encryption Method using Steganographic LSB Method, AES, and RSA algorithm. Nonlinear Engineering 6, 53–59. https://doi.org/10.1515/nleng-2016-0010
Ertaul, L., 2016. Implementation and Performance Analysis of PBKDF2, Bcrypt, Scrypt Algorithms.Availableat:http://borg.csueastbay.edu/~lertaul/PBKDFBCRYPTCAMREADYICWN16.pdf
Patil, P., Narayankar, P., Narayan D.G., Meena S.M., 2016. A Comprehensive Evaluation of Cryptographic Algorithms: DES, 3DES, AES, RSA, and Blowfish. Procedia Computer Science, 1st International Conference on Information Security & Privacy 2015 78, 617–624. https://doi.org/10.1016/j.procs.2016.02.108
Prashanti, G., Jyothirmai, B.V., Chandana, K.S., 2017. Data confidentiality using steganography and cryptographic techniques, in: 2017 International Conference on Circuit , Power and Computing Technologies (ICCPCT). Presented at the 2017 International Conference on Circuit, Power, and Computing Technologies (ICCPCT), pp. 1–4. https://doi.org/10.1109/ICCPCT.2017.8074276
Preda, R. o., Vizireanu, D. n., 2015. Watermarking-based image authentication was robust to JPEG compression. Electronics Letters 51, 1873–1875. https://doi.org/10.1049/el.2015.2522
Sikder, I., Dhar, P.K., Shimamura, T., 2017. A semi-fragile watermarking method using slant transform and LU decomposition for image authentication, in: 2017 International Conference on Electrical, Computer and Communication Engineering (ECCE). Presented at the 2017 International Conference on Electrical, Computer and Communication Engineering (ECCE), pp. 881–885. https://doi.org/10.1109/ECACE.2017.7913027
Naved, M., 2021. A novel data classification-based scheme for cloud data security using various cryptographic algorithms. Available at: https://www.researchgate.net/publication/354825262_A_novel_data_classification-based_scheme_for_cloud_data_security_using_various_cryptographic_algorithms
Torvi, S.D., ShivaKumar, K.B., Das, R., 2016. An unique data security using text steganography, in: 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom). Presented at the 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), pp. 3834–3838.
Venukumar, V., Pathari, V., 2016. Multi-factor authentication using threshold cryptography, in: 2016 International Conference on Advances in Computing, Communications, and Informatics (ICACCI). Presented at the 2016 International Conference on Advances in

Computing, Communications and Informatics (ICACCI), pp. 1694–1698. https://doi.org/10.1109/ICACCI.2016.7732291
Wahid, M., Ahmad, N., Zafar, M.H., Khan, S., 2018. On combining MD5 for image authentication using LSB substitution in selected pixels, in: 2018 International Conference on Engineering and Emerging Technologies (ICEET). Presented at the 2018 International Conference on Engineering and Emerging Technologies (ICEET), pp. 1–6. https://doi.org/10.1109/ICEET1.2018.8338621
Yu, F., Huang, Y., 2015. An Overview of Study of Password Cracking, in: 2015 International Conference on Computer Science and Mechanical Automation (CSMA). Presented at 2015 International Conference on Computer Science and Mechanical Automation (CSMA), pp. 25–29. https://doi.org/10.1109/CSMA.2015.12