A

SYNOPSIS

of

MINOR PROJECT

on

Using Cryptoghrachy for Image Encryption

Submitted by

Akshay Suthar
(21EGICS007)

Project Guide
Dr. Ajay Kumar Sharma

Head of Department
Dr. Mayank Patel

- **Problem Statement:**

Information Security is the most common word uttered by any man, any device or any peripheral since past two centuries. Protection from malicious sources has become a part of the invention or the discovery cycle. Myriad methods of protection are used ranging from a simple authentication password to most complex Cryptography. The advancements of digital revolution were not achieved without drawbacks such as illegal copying and distribution of digital multimedia documents. In order to provide data security and protection, different encryption methods must be used.

Encryption keys can also be vulnerable to cyber attacks, such as keyloggers, malware, and phishing scams. The frequent changes and updates to encryption standards can make key management more challenging, requiring organizations to regularly update their encryption systems and manage new keys.

These failures include key leakage, software bugs, holes in operating systems, side-channel attacks, phishing attacks, and social engineering. So, it is important to understand and acknowledge that cryptography ≠ security. Nevertheless, when cryptography fails, the consequences can be very severe.

- **Brief Description:**

Image encryption techniques employ mathematical algorithms and cryptographic methods to alter the pixel values or the visual representation of an image. These algorithms convert the original image into a ciphered or scrambled version, rendering it meaningless to anyone without the appropriate decryption key.

Image encryption is a crucial component of Information security that focuses specifically on safeguarding the confidentiality and integrity of digital images. With the increasing prevalence of digital imagery in various domains such as photography, multimedia, medical imaging, and satellite imaging, the need for secure image transmission, storage, and sharing has become paramount.

The primary objective of online image encryption is to transform the visual data within an image into an unreadable and unintelligible form, making it inaccessible to unauthorized individuals. By encrypting images, sensitive information contained within them can be protected from unauthorized viewing, tampering, or interception.

The encryption process ensures that even if an attacker gains access to the encrypted image, they cannot retrieve the original content without the decryption key.

Various encryption algorithms can be used for image encryption, including symmetric key encryption and public key encryption. Symmetric key encryption involves using a shared secret key for both encryption and decryption. Public key encryption, on the other hand, employs a pair of keys: a public key to encrypt pictures and a private key to decrypt the same.

Various online cybersecurity courses, like the CEH v12 training course, detail image encryption.

- **Objective and Scope:**

This is the proposal for the design and development of software named as "Secure Image Encryption Using Algorithm Based On Rubik's Cube Principle". It applies special mathematical algorithm and keys to transform digital image into chiper code before they are transmitted and decrypts using application of mathematical algorithms and keys to get back the original data from the chiper code. The goal is to provide authentication of users and integrity, accuracy and safety of data resources.

The main objective of our project is to provide security of the image-based data with the help of suitable key and protect the image from illegal copying and distribution

By encrypting images, sensitive information contained within them can be protected from unauthorized viewing, tampering, or interception. Image encryption techniques employ mathematical algorithms and cryptographic methods to alter the pixel values or the visual representation of an image.

There are four main goals in cryptography: confidentiality, integrity, authentication, and non-repudiation

1. message confidentiality (or privacy): Only an authorized recipient should be able to extract the contents of the message from its encrypted form. Resulting from steps to hide, stop, or delay free access to the encrypted information.
2. message integrity: The recipient should be able to determine if the message has been altered.
3. sender authentication: The recipient should be able to verify from the message, the identity of the sender, the origin or the path it traveled (or combinations) so as to validate claims from emitter or to validated the recipient expectations.
4. sender non-repudiation: The remitter should not be able to deny sending the message.

- **Methodology:**

Cryptographic algorithms falls into two categories: symmetric and asymmetric. In symmetric or also called as private key cryptography only one key is used to encrypt and decrypt the data. But in asymmetric which is also called as public key cryptography, two separate keys (public and private) are used

1. **Asymmetric key encryption**

First, let's look at asymmetric key encryption with a simple analogy.

Imagine you wanted to send something to your friend, but it was absolutely essential that nobody else, except your friend, could have access to that object. So, your friend buys an indestructible box, fabricated from the strongest metal on the planet, and sends it to you so that you can place the object in it. Your friend also sends you the key that can only be used to lock the box.

Now, this box has one more special property. It has two keyholes. One keyhole to open the box, another to lock the box.

Naturally, this box will also need two keys – one to open and another to lock it.

Both keys are similar, but not identical. As you can see in the image above, for example, the key used to open the box has two prongs while the key used to lock the box has three prongs.

As the sender of the object, all you have is the box to place the object in and a key to lock the box. Only your friend has the key that can unlock the box.

The key used to lock the box is called the public key, and cannot be used to open it, as that requires the private key. If anyone intercepted the package and made a copy of the public key, it could not be used to open the box, only to lock it. Only the person who holds the private key can open the box.

Asymmetric key encryption is used when there are two or more parties involved in the transfer of data. This type of encryption is used for encrypting data in transit, that is encrypting data being sent between two

or more systems. The most popular example of asymmetric key encryption is RSA.

2. **Symmetric key encryption**

Symmetric key encryption uses the same key for encryption and decryption. This makes sharing the key difficult, as anyone who intercepts the message and sees the key can then decrypt your data.

This is why symmetric key encryption is generally used for encrypting data at rest. AES-256 is the most popular symmetric key encryption algorithm. It is used by AWS for encrypting data stored in hard disks (EBS volumes) and S3 buckets. GCP and Azure also use it for encrypting data at rest.

- **Hardware and Software Requirements:**

If you plan to use the RSA option with DFSMSdss to encrypt the data-encrypting key, you must consider the cryptographic hardware that exists at the site that will decrypt the data. Not all types of RSA private keys are supported by all types of cryptographic hardware.

RSA private tokens and required cryptographic hardware for decryption

RSA private key token (internal)	Required cryptographic hardware
RSA private key token 1024 — Modulus-Exponent Internal form	One of the following: – Cryptographic Coprocessor feature – PCI X Cryptographic Coprocessor – Crypto Express2 Coprocessor.
RSA private key token 1024 — Chinese Remainder Theorem Internal form	One of the following: – PCI Cryptographic Coprocessor – PCI X Cryptographic Coprocessor – Crypto Express2 Coprocessor.
RSA private key token 2048 — Chinese Remainder Theorem Internal form	One of the following: – PCI Cryptographic Coprocessor with LIC January 2005 or later, and z/OS® ICSF HCR770B or later – PCI X Cryptographic Coprocessor – Crypto Express2 Coprocessor.

You might be asking yourself whether to use a software or hardware implementation to get the security that you need. The answer depends on the need: how much security does an application really require?

There are still plenty of situations where implementing security with cryptographic hardware makes the most sense. When large volumes of secure data need to be processed as with pay TV or SSL accelerators, high-performance security can best be achieved in hardware.

If the security is implemented with a fast microcontroller, then the data will process faster than with an external software implementation. Hardware blocks can also simplify the design and avoid the problem of

data processed by the core. Processing data simultaneously while the application is executing provides better real-time system performance. Recall, finally, that a hardware implementation costs more than a software implementation.

Here we specify the types of software that is used within the system:-
1. Photoshop
2. Python 3.7
3. SQLite
4. Atom IDE
5. Microsoft Windows 10 OS

- **Technologies:**

Encryption works by encoding “plaintext” into “ciphertext,” typically through the use of cryptographic mathematical models known as algorithms. To decode the data back to plaintext requires the use of a decryption key, a string of numbers or a password also created by an algorithm.
Cryptography ensures confidentiality by encrypting sent messages using an algorithm with a key only known to the sender and recipient. A common example of this is the messaging tool WhatsApp, which encrypts conversations between people to ensure they cannot be hacked or intercepted.

- **Testing Technologies:**

We have design and implement a system which provide dedicated to highly secure data storage for different governmental and non-governmental sector. To implement this system, we have used encryption algorithm based on Rubik's cube principle that encrypted the image into unexpected rough pixel format and generate the key for future decryption and send the generated key into respected email id of an individual's / organization. After the completion of this project, the obtained output is shown as shown in figure below: