FILE ENCRYPTION USING GLITCH VIDEO

Ms. C. Jesifica Cinthamani[1], Mr. T. Paul Wilson[2], Mr. U. Vikram[3], Mr. D. Anukesan[4]

Professor[1], UG Scholar[2],[3],[4], Department of Cyber Security,

SRM Valliammai Engineering College

***ABSTRACT* - In this project, we changing a plain text file into a "Glitched Video" using Steganography method so that we can get a new encryption method. Using this method we can get an additional option of encrypting the data and also an extra layer of security for the data. All files are made of bytes and bytes can be interpreted as numbers ranging from 0-255. This number can be represented with pixels using one of two modes: RGB or binary. Continue to evolve, necessitating advanced and integrated solutions to protect against them, In the realm of digital communication and artistic expression, the convergence of glitch art and encryption techniques has given rise to a fascinating domain known as glitched video encryption. This abstract explores the conceptual framework and practical implications of this emerging field. At their intersection lies glitched video encryption, a fusion of creative expression and data protection. By introducing controlled glitches into encrypted video streams.**

KEYWORDS-Steganography,H264 algorithm, Pixel Embedding, Image to video converter

I. INTRODUCTION:

1. In this project, we changing a plain text file into a "Glitched Video" using H 264 Algorithm so that we can get a new encryption method using this method we can get an additional option of encrypting the data and also an extra layer of security for the data.

2. Using H 264 Algorithm we can achieve file embedded into a video format ("Glitched Video") format. The H 264 algorithm supports motion estimation on blocks from 16✕16 to 4✕4 pixels.

3. Using the above algorithm, we can convert the files into a glitched video format which hides your data and other sensitive files (or) data from the attackers (or) any other third party platforms. In the Glitch video, the blacks and whites are consider as Binary codes such as blacks are 0's and whites are 1's.

4. Pixel embedding is a technique used to conceal information within the pixels of digital images. It's a method of hiding data in an image by subtly altering the color values of individual pixels in a way that's imperceptible to the human eye but detectable by specialized software

5. With this technical standpoint, glitched video encryption presents a unique set of challenges and opportunities. Innovations in **steganography**, error correction coding, and cryptographic primitives are leveraged to seamlessly integrate glitches into encrypted video data without compromising security or visual fidelity. Furthermore, advancements in machine learning and computational creativity empower artists and cryptographers alike to explore new frontiers in glitched video encryption.

6. In today's digital age, the intersection of art and technology has become a fertile ground for innovation and experimentation. Among the myriad forms of creative expression, glitch art has emerged as a captivating medium, disrupting conventional notions of visual aesthetics with its deliberate distortion and manipulation of digital media. Concurrently, the imperative to safeguard sensitive information in an increasingly interconnected world has propelled advancements in encryption technologies, underpinning the foundations of cybersecurity.

7. This paper delves into the captivating realm where these two seemingly disparate domains converge: glitched video encryption. At its core, glitched video encryption represents a synthesis of artistic expression and data security, weaving together the chaotic beauty of glitches with the robustness of cryptographic techniques. By introducing controlled distortions into encrypted video streams, practitioners navigate the delicate balance between obfuscation and integrity, creating a novel paradigm that challenges conventional approaches to both art and encryption.

8. When applied to glitched video encryption, **steganography** plays a pivotal role in embedding cryptographic payloads within the video data stream. By exploiting the redundancy and imperceptible variations inherent in digital media, **steganography** techniques enable the seamless integration of encrypted information into the visual and auditory components of the video file. These hidden payloads may consist of cryptographic keys, authentication tokens, or other metadata necessary for decrypting the video content.

9. In parallel, the **H.264** algorithm serves as the cornerstone of modern video compression and transmission. Developed by the Joint Video Team (JVT) of the International Telecommunication Union (ITU) and the ISO Moving Picture Experts Group (MPEG), H.264 offers significant improvements in compression efficiency over its predecessors, enabling the delivery of high-quality video content over bandwidth-constrained networks. **H.264 algorithm** presents both opportunities and challenges. On one hand, its sophisticated compression techniques provide a robust framework for embedding steganography payloads within the video data stream. By exploiting the spatial and temporal redundancies present in video sequences, H.264 facilitates the seamless integration of encrypted information while minimizing the impact on visual quality.

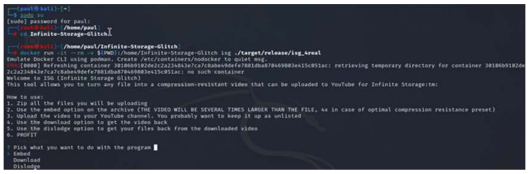
II. OBJECTIVES:

- To provide an another form of encryption method.
- To prove files (or) any other data that can be turned into a video which hides the data.
- To ensure security of the data which is converted into a video.
- Error correction coding techniques may be employed to enhance the resilience of the glitched video stream to data loss or corruption during transmission or storage.

III. CONTRIBUTION OF THE PAPER:

This study contributes the development of new methodology of the plain text file encryption. To achieve this, the plain text file is converted into a binary codes and then the binary codes are converted to (Black & White) Pixels and the data are hidden behind this glitch image each binary codes are converted into pixels and 0's are represented as White and 1's are converted into Black.

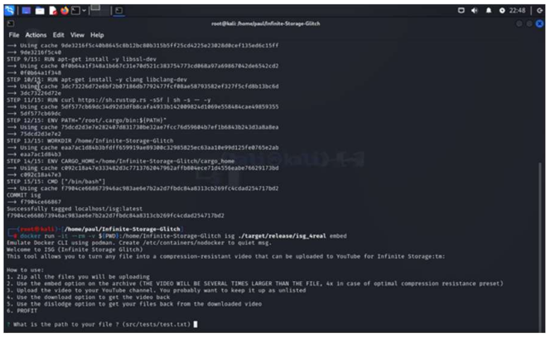
IV. IMPLEMENTATION



```
┌──(paul㉿kali)-[~]
└─$ sudo su
[sudo] password for paul:
┌──(root㉿kali)-[/home/paul]
└─# cd Infinite-Storage-Glitch

┌──(root㉿kali)-[/home/paul/Infinite-Storage-Glitch]
└─# docker run -it --rm -v $(PWD):/home/Infinite-Storage-Glitch isg ./target/release/isg_4real
Emulate Docker CLI using podman. Create /etc/containers/nodocker to quiet msg.
ERRO[0000] Refreshing container 30106b9102de2c2a234843e7ca7c8abe49defe7881dba870469803e415c051ac: retrieving temporary directory for container 30106b9102de2c2a234843e7ca7c8abe49defe7881dba870469803e415c051ac: no such container
Welcome to ISG (Infinite Storage Glitch)
This tool allows you to turn any file into a compression-resistant video that can be uploaded to YouTube for Infinite Storage:tm:

How to use:
1. Zip all the files you will be uploading
2. Use the embed option on the archive (THE VIDEO WILL BE SEVERAL TIMES LARGER THAN THE FILE, 4x in case of optimal compression resistance preset)
3. Upload the video to your YouTube channel. You probably want to keep it up as unlisted
4. Use the download option to get the video back
5. Use the dislodge option to get your files back from the downloaded video
6. PROFIT

? Pick what you want to do with the program
> Embed
  Download
  Dislodge
```

Figure 1.1: Interface of the project

In this process, we enabled the ISG tool for encrypting the file and in this step it gives the option for encryption and decryption to the user and the user selects the option accordingly. The user is allowed to select what to do with the program. The embed option allows the user to insert the plain text file and encrypt the plain text file, the download option allows the user to download the video from the social media and the dislodge option allows the user to decrypt the glitch video.

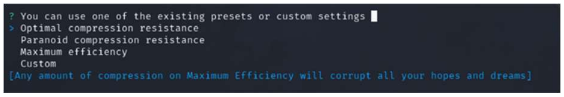


22:48
root@kali: /home/paul/Infinite-Storage-Glitch
File Actions Edit View Help

```
→ Using cache 9de3216f5c40b8645c8b12bc80b315b5ff25cd4225e23028d0cef135ed6c15ff
→ 9de3216f5c40
STEP 9/15: RUN apt-get install -y libssl-dev
→ Using cache 0f0b64a1f348a1b667c31e70d521c383754773cd068a97a69867042de6542cd2
→ 0f0b64a1f348
STEP 10/15: RUN apt-get install -y clang libclang-dev
→ Using cache 3dc73226d72e6bf2b07186db7792477fcf08ae58793582ef327f5cfd8b13bc6d
→ 3dc73226d72e
STEP 11/15: RUN curl https://sh.rustup.rs -sSf | sh -s -- -y
→ Using cache 5df577cb69dc34d92d3dfb8cafa4933b142009824d1069e558484cae49859355
→ 5df577cb69dc
STEP 12/15: ENV PATH="/root/.cargo/bin:${PATH}"
→ Using cache 75dcd2d3e7e282407d831730be32ae7fcc76d59604b7ef1b6843b243d3a8a8ea
→ 75dcd2d3e7e2
STEP 13/15: WORKDIR /home/Infinite-Storage-Glitch
→ Using cache eaa7ac1d84b3bfdff659919ae89300c32985825ec63aa10e99d125fe0765e2ab
→ eaa7ac1d84b3
STEP 14/15: ENV CARGO_HOME=/home/Infinite-Storage-Glitch/cargo_home
→ Using cache c092c18a47e333482d3c7713762047962affb804ece71d4556eabe76629173bd
→ c092c18a47e3
STEP 15/15: CMD ["/bin/bash"]
→ Using cache f7904ce668673946ac983ae6e7b2a2d7fbdc84a8313cb269fc4cdad254717bd2
COMMIT isg
→ f7904ce66867
Successfully tagged localhost/isg:latest
f7904ce668673946ac983ae6e7b2a2d7fbdc84a8313cb269fc4cdad254717bd2

┌──(root㉿kali)-[/home/paul/Infinite-Storage-Glitch]
└─# docker run -it --rm -v $(PWD):/home/Infinite-Storage-Glitch isg ./target/release/isg_4real embed
Emulate Docker CLI using podman. Create /etc/containers/nodocker to quiet msg.
Welcome to ISG (Infinite Storage Glitch)
This tool allows you to turn any file into a compression-resistant video that can be uploaded to YouTube for Infinite Storage:tm:

How to use:
1. Zip all the files you will be uploading
2. Use the embed option on the archive (THE VIDEO WILL BE SEVERAL TIMES LARGER THAN THE FILE, 4x in case of optimal compression resistance preset)
3. Upload the video to your YouTube channel. You probably want to keep it up as unlisted
4. Use the download option to get the video back
5. Use the dislodge option to get your files back from the downloaded video
6. PROFIT

? What is the path to your file ? (src/tests/test.txt)
```

Figure 1.2: File Selection Interface

Here, the user needs the provide the file path of the file that need to be encrypted and the tool extract the file using the given file path.



```
? You can use one of the existing presets or custom settings
> Optimal compression resistance
  Paranoid compression resistance
  Maximum efficiency
  Custom
[Any amount of compression on Maximum Efficiency will corrupt all your hopes and dreams]
```

Figure 1.3: glitch video intensity selection

The above four options are used for different glitch video encryption and that uses different RGB colors for the encryption according to the binary codes and also it is provided with custom settings option where the user can able to use their own custom settings for the encryption process.

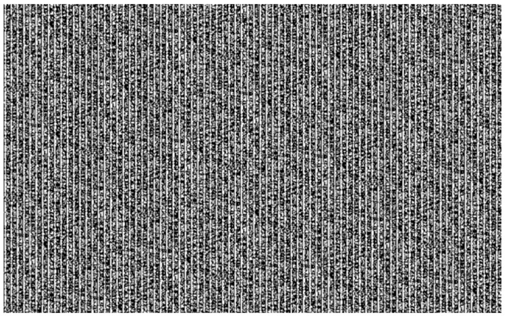


Figure 1.4: Glitch Video

After these steps, the final output of the plain text file is converted to a glitch video

ARCHITECTURE OF THE PROJECT :

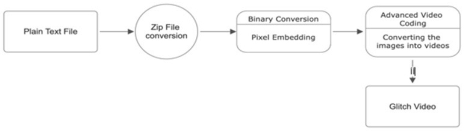
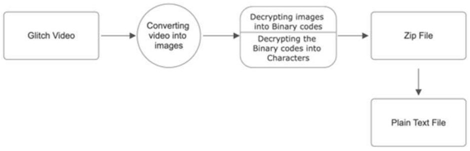


Figure 2.1: The above diagram represents the work flow of the encryption process



Glitch Video
Converting video into images
Decrypting images into Binary codes
Decrypting the Binary codes into Characters
Zip File
Plain Text File

Figure 2.2: The above diagram represents the work flow of the decryption process

V. GLITCHED VIDEO ENCRYPTION:

Glitched video encryption is a unique approach to securing video content by intentionally introducing controlled glitches or distortions into the video data while maintaining cryptographic integrity. This method combines elements of glitch art, which intentionally distorts digital media for aesthetic purposes, with encryption techniques, which encode data to protect it from unauthorized access. Which ensures that only authorized users can access the original content. Glitched video encryption intentionally introduces imperfections or distortions into the encrypted video stream. These glitches can take various forms, such as pixelated, noise, color shifts, or geometric distortions. By this the synergy between glitch art and cryptography, glitched video encryption provides a unique blend of security, creativity, and resilience, opening up new possibilities for safeguarding sensitive information in the digital environment.

VI. ALGORITHM & TECHNIQUES:

1.1 STEGANOGRAPHY:

Steganography is the practice of concealing messages or information within other innocuous data in such a way that the existence of the hidden information is not readily apparent to observers. Unlike cryptography, which focuses on encrypting messages to make them unreadable without the proper decryption key, steganography aims to hide the very existence of the communication. Detecting and extracting hidden messages in steganography typically requires knowledge of the steganography method used and access to the original cover data. Specialized software tools and algorithms can be employed to analyze digital media files for signs of hidden information.

Such as statistical anomalies or irregular patterns, and extract the embedded messages. This is enhance security, steganography is often combined with encryption and compression techniques. The secret message may first be encrypted to protect its contents, and then embedded within the cover data using steganography methods. Compression techniques can also be applied to minimize the size of the hidden message, making it more difficult to detect. In text steganography, Concealing messages within whitespace, punctuation marks, or formatting elements. Employing homoglyphy or alternate character representations to encode information without altering the appearance of the text. Utilizing invisible characters or non-printable Unicode symbols to hide information within the text.

1.2 H264 ALGORITHM:

The H.264 algorithm, also known as AVC (Advanced Video Coding), is a highly efficient video compression standard that was jointly developed by the International Telecommunication Union (ITU-T) Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG). It is one of the most widely used video compression formats and serves as the foundation for various applications, including streaming video, broadcast television, Blu-ray discs, and video conferencing. The H.264 algorithm employs sophisticated compression techniques to reduce the size of video data while preserving visual quality. It achieves this by exploiting spatial and temporal redundancies present in video sequences, thereby minimizing the amount of information needed to represent the video content accurately. Each frame is divided into smaller blocks called macroblocks (is a fundamental unit used for encoding and decoding video frames. It represents a rectangular region of pixels within a video frame and serves as the basic building block for motion estimation, prediction, and compression). These macroblocks are further partitioned into smaller units, such as prediction blocks and transform blocks, which are used for encoding and compression. The H.264 algorithm achieves a remarkable balance between compression efficiency and visual quality, making it an indispensable tool for video encoding and transmission in a wide range of applications. Its versatility, scalability, and widespread adoption have solidified its position as one of the most important video compression standards in the digital multimedia landscape.

1.3 Pixel embedding:

Pixel embedding is a technique used to conceal information within the pixels of digital images. It's a method of hiding data in an image by subtly altering the color values of individual pixels in a way that's imperceptible to the human eye but detectable by specialized software. The process begins with selecting a cover image, which serves as the carrier for the hidden message. The cover image is typically chosen to be a high-resolution image with a large number of pixels, providing ample space for embedding data without significantly degrading visual quality. The information to be hidden, known as the secret message or payload, is encoded into a format suitable for embedding within the cover image. This could be text, binary data, or other forms of information. The embedding process involves altering the color values of selected pixels in the cover image to encode the hidden message. This can be done using various techniques, such as least significant bit (LSB) substitution, where the least significant bits of the pixel values are replaced with the bits of the hidden message. Pixel embedding typically involves both an encoding algorithm, used to embed the hidden message into the cover image, and a decoding algorithm, used to extract the hidden message from the modified image. These algorithms must be carefully designed to ensure that the embedded data can be reliably recovered without introducing artifacts or perceptible distortions in the cover image. While pixel embedding can provide a covert means of communication, it's important to consider security implications. Depending on the sensitivity of the hidden message and the threat model, additional encryption or cryptographic techniques may be employed to protect the confidentiality and integrity of the embedded data. In LSB substitution, the color values of each pixel in the cover image are modified to encode the bits of the hidden message. Since the human visual system is less sensitive to changes in the least significant bits of color values, these alterations are generally imperceptible to the naked eye. By replacing the LSBs of the pixel values with the bits of the hidden message, the cover image is subtly modified to carry the encoded information. Pixel embedding is a versatile technique that can be used for various purposes, including digital watermarking, copyright protection, and covert communication. By exploiting the imperceptible changes in pixel values, it allows for the concealment of information within digital images without visibly altering their appearance.

1.4 Binary Converting:

Binary conversion is the process of representing numbers, characters, or data using the binary number system, which consists of only two digits which is 0 and 1. This system is fundamental in computing and digital electronics, where data is represented in the form of binary digits (bits). In computing, characters and data are often represented using binary codes. For example, ASCII (American Standard Code for Information Interchange) is a widely used character encoding standard that assigns a unique 7-bit binary code to each character in the English alphabet, digits, punctuation marks, and control characters. Binary conversion is a fundamental concept in computer science and digital technology, forming the basis for data representation, arithmetic, and communication in digital systems. Converting text to binary involves encoding each character of the text into its corresponding binary representation. This process typically follows a standardized character encoding scheme such as ASCII or Unicode.

1.5 Error Correction Coding:

Error correction coding techniques may be employed to enhance the resilience of the glitched video stream to data loss or corruption during transmission or storage. Methods such as Reed-Solomon codes or convolutional codes can add redundancy to the data, allowing for the detection and correction of errors. The original data is processed using an error correction code algorithm to generate redundant bits. These redundant bits are calculated based on the original data using mathematical operations such as parity checks, checksums, or more sophisticated encoding schemes like Reed-Solomon codes or convolutional codes. The encoded data consisting of both the original message and the added redundant bits, is transmitted over a communication channel or stored in a medium such as a disk drive or memory device. During transmission or storage, the data may be subject to noise, interference, or other forms of distortion that can introduce errors. The received data is decoded using the same error correction code algorithm used for encoding. The decoder analyzes the received bits and uses the redundant information to detect and correct errors that may have occurred during transmission or storage. Error correction coding introduces additional overhead in terms of the number of redundant bits added to the original data. The amount of redundancy required depends on factors such as the error rate of the communication channel or storage medium and the desired level of error correction capability. Balancing the trade-off between error correction capability and the efficiency of data transmission or storage is a key consideration in designing error correction coding schemes.

Error correction coding is essential in ensuring the reliability and integrity of digital communication and data storage systems, particularly in applications where data integrity is critical, such as telecommunications, satellite communications, wireless networks, and digital storage media. It allows for robust and resilient transmission and storage of data in the presence of noise and other sources of errors.

Error correction is a crucial aspect of data communication and storage systems, aimed at detecting and correcting errors that occur during transmission or retrieval. There are various techniques and algorithms used for error correction, each with its own advantages and applications

VII. CONCLUSION:

The conclusion of the paper on glitched video encryption would summarize the key findings, contributions, and implications of the research in the experimental results or theoretical analyses demonstrate the effectiveness of glitched video encryption in securely concealing data within video streams while maintaining visual quality. Reflect on the advancements made in cryptographic techniques, steganography methods, and video processing algorithms to enable glitched video encryption. The significance of glitched video encryption as a novel approach to combining artistry and security in digital media. Emphasize the potential of this technique to inspire new forms of creative expression, foster interdisciplinary collaborations, and advance the state-of-the-art in multimedia security.The integration of supervised models further enhances the capabilities of encryption methods, The glitched video encryption method is more effective and secured compared than the other encryption methods.

VIII. REFERENCES:

[1] Zhongyun Hua, Yifeng Zheng, Ziyi Wang, Yuanman Li, Yongyong Chen, “Enabling Large Capacity Reversible Data Hiding Over Encrypted Jpeg Bitstream”,Vol. 33, 2023.

[2] yijing chen, Hongxia wang, Wanjie li, Jie luo, “ Cost Reassignment For Improving Security of Adaptive Steganography Using an Artificial Immune System”Vol 23,2020

[3] Wanli teng, Tao wang, Zhenxing Qian, Sheng li, Xinpeng zhang, “Cross-Model Text Steganography Against Synonym Substiution-based text attack”Vol 30,2023

[4] Mingzhi hu, Hongxia wang, “Image Steganalysis against adversarial Steganography by Combining Confendences and Pixel artifacts” Vol 30,2023

[5] Ruiyi yan, yating yang, Tian song “ A Secure and Disambiguating approach for Gerative linguistic Steganography” Vol 30,2023

[6] Rong wang, Ling yua xiang, Yangfan liu , Chunfang yang “Png-stega:Progressive Non-autoregressive Generative linguistic Steganography” Vol 30,2023

[7] Shiv Prasad, Arup kuram pal, Soumya mukhergee “An Rebcolour Image Steganography Scheme by binary lower Triangular Matrix” Vol 24,2023

[8] Chenwei huang, Zhongliang ying, Zhiwen hu, Jinshuai wang, Haochenqy, Jian zhang, Lei zheng “DNA Synthetic Steganography based on Conditional Probiliaty Adaptive Coding” Vol 18,2023

[9] Chenwei huang, Zhongliang ying, Zhiwen hu, Jinshuai wang, Haochenqy, Jian zhang, Lei zheng, Jian zhang, Lei zheng “ Ameliorating Lsb using Piecewisc Linear Chaotic Map and one-time pad for Superlative Capaticy, Imperceptibility and secure audio Steganography” Vol 11,2023

[10] A Suresh, R Kishorekumar, MS Kumar, K Elaiyaraja ," Assessing transmission excellence and flow detection based on Machine Learning"- Optical and Quantum Electronics, 2022.