SECURE ARTWORK TRANSFER

Siddhartha Nath B 21CSB0A08
Sachin A.D 21CSB0A54

The Secure Artwork Transfer focuses on securing artwork and its metadata during transmission between multiple parties with a server in between them. The problem addressed in this is the sensitivity of metadata during transmission, which can be intercepted and manipulated by unauthorized parties. This problem is crucial because metadata contains valuable information about the artwork, such as an image of the art, its origin, author and ownership, which, if tampered with, can lead to authenticity and integrity issues.

Existing solutions typically involve encrypting the metadata using symmetric or asymmetric encryption techniques during the transmission.

The key contributions of our project include:

Implementation of a secure communication between multiple systems through a server to transmit encrypted metadata. Utilization of symmetric key encryption for securing metadata during transmission by leveraging RSA encryption for securely transmitting the symmetric key between the system and the server.

Objectives of Secure Art Transfer:

- To develop a strong encryption mechanism to protect artwork(image) and its metadata during transmission between multiple parties(systems) using a single server.

- To develop a server which can efficiently store the artwork received from different systems and securely transfer artwork and its ownership if requested by any system.

The terminology used in the implementation of Secure Artwork Transmission:

-PC1: System which shared its Artwork with the server

-PC2: Server which stores, receives and transfers the artwork and metadata securely.

-PC3: System which requests artwork from the server.

Implementation of Secure Artwork Transfer:

1. Key Generation: PC1 generates its data consisting of wallet address, and pair of cryptographic keys (PRK1, PCK1) using RSA Cryptosystem. The server generates its own pair of cryptographic keys(PRKS, PCKS) using the same RSA algorithm.

2. Connection Establishment: PC1 establishes its connection with the PC2(server) using socket programming. PC2 shares its public key(PCKS) with PC1.

3. Metadata Generation: PC1 generates the metadata for the image it is going to share with PC2.

4. Encryption and Transfer: PC1 encrypts the image and its metadata using AES symmetric encryption and shares it along with the Symmetric key to the PC2 server using the RSA public cryptosystem.

5. Decryption and Storage: The server receives the encrypted image along with the metadata and decrypts it using the RSA public cryptosystem. The server generates the unique identification number corresponding to the decrypted image and metadata received and stores them with that unique ID.

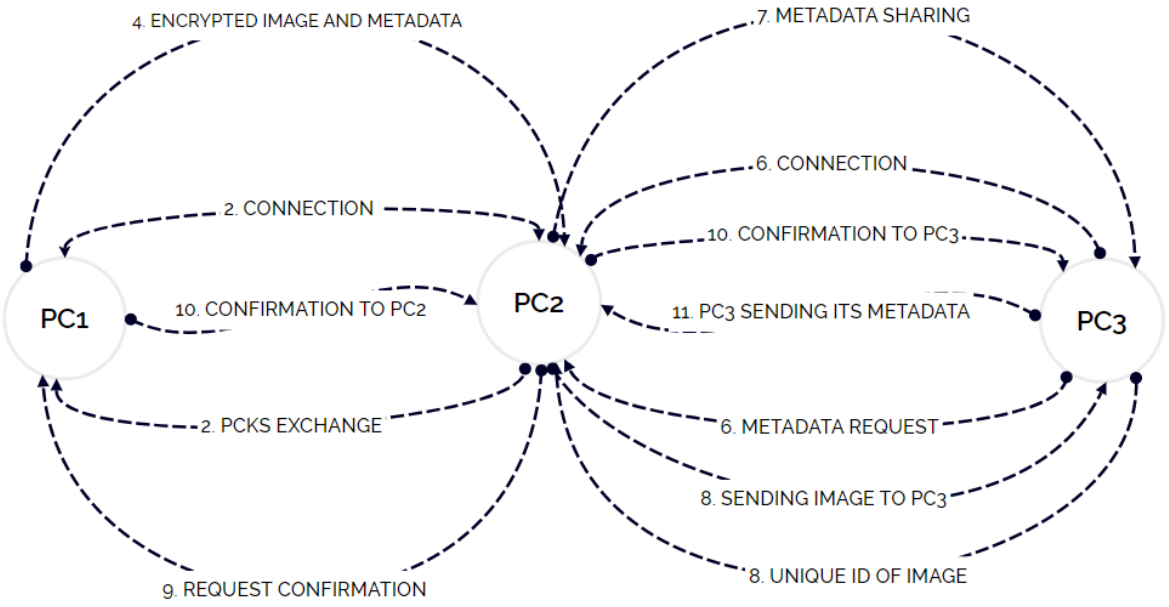
6. Metadata Request: PC3 establishes the connection with PC2 using sockets and requests for the metadata of artwork that the server has stored by sending its public key(PCK3).

7. Metadata Transfer: PC2 shares the metadata of artwork it has stored to PC3 using symmetric encryption and RSA key exchange process.

8. Metadata Analysis and Image Request: PC3 analyzes all the metadata received by decrypting it using PRK3 and sends the unique ID of the image which he is interested in to the server. The server sends the encrypted form of the image that PC3 has requested. If PC3 needs the image, he sends the confirmation to PC2 with the unique ID.

9. Ownership Update Request: PC2(server) upon receiving the unique ID confirmation from PC3, sends a request to the actual owner(PC1) of the art using the metadata it stored along with the image to inform the owner about the update of ownership to PC3.

10. Confirmation and Metadata Request: If PC1 confirms about updation of ownership to server, server passes this confirmation to PC3 and asks for its metadata.



Conceptual Diagram

11. Metadata Update: PC3 sends its metadata to the server using symmetric encryption with the RSA key exchange process. The server, on the other side, decrypts the received metadata and updates the ownership of the image in its database.

Outcomes of Implementation:

1. Image encryption and metadata generation by PC1:

```
PS C:\Users\Siddhartha Nath\OneDrive\Documents\CRYPTO_IMAGE_RSA_NFT> python -u "c
A_NFT\image_encry.py"
Image encrypted and stored successfully.
PS C:\Users\Siddhartha Nath\OneDrive\Documents\CRYPTO_IMAGE_RSA_NFT>
```

```
PS C:\Users\Siddhartha Nath\OneDrive\Documents\CRYPTO_IMAGE_RSA_NFT> python -u "c:\Users\Siddhartha Nath\OneDrive\Documents\CRYPTO_IMAGE_RSA_NFT\m
eta_data_gen.py"
{'title': 'Mountain', 'description': 'Beautiful Mountain', 'artist': 'Sid', 'creation_date': '2024-04-21', 'file_path': 'encrypted_image.enc', 'wa
llet_address': '64a778398b08bcad181713f2e610da34d73d73e6aec53e45d5b8e065bb4b7954', 'sender_ip': '192.168.137.56', 'sender_port': 8080}
NFT metadata saved successfully.
PS C:\Users\Siddhartha Nath\OneDrive\Documents\CRYPTO_IMAGE_RSA_NFT>
```

2. Send the encrypted metadata and image to PC2.

```
PS C:\Users\Siddhartha Nath\OneDrive\Documents\CRYPTO_IMAGE_RSA_NFT> python
ending_metadata_encrypt.py"
PC1 waiting for PC2 to connect...
--------------------------------
PC1 connected with ('192.168.137.185', 51851)
Encrypted metadata sent to PC2 successfully.
PS C:\Users\Siddhartha Nath\OneDrive\Documents\CRYPTO_IMAGE_RSA_NFT>
```

```
PS C:\Users\Siddhartha Nath\OneDrive\Documents\CRYPTO_IMAGE_RSA_NFT> python
ending_encrypt_img.py"
PC1 waiting for PC2 to connect...
--------------------------------
PC1 connected with ('192.168.137.185', 51970)
Encrypted IMAGE data sent to PC2 successfully.
PS C:\Users\Siddhartha Nath\OneDrive\Documents\CRYPTO_IMAGE_RSA_NFT>
```

3. PC2 receives the encrypted data and stores them in a duplicate_store with a unique identification number.

```
PS C:\Users\sachi\OneDrive\Desktop\crypto2> & C:/Python312/python.exe c:/Users/sachi/OneDrive/Desk
top/crypto2/receive_decrypt_store1.py
pc2(server) connected to pc1.
Receiving encrypted data...
Encrypted data received successfully.
Image decrypted and stored successfully.
Metadata stored successfully with unique ID: ldLjO1apo5
Decrypted image moved to 'image_store' directory with unique ID name: ldLjO1apo5
Duplicate copy of the decrypted image created in 'duplicate_store' directory with unique ID name:
ldLjO1apo5
Duplicate copy of metadata created in 'duplicate_metadata_store' directory with unique ID name: ld
LjO1apo5
PS C:\Users\sachi\OneDrive\Desktop\crypto2>
```

4. PC3 requests metadata from the server and the server sends encrypted metadata to PC3.

```
PS C:\Users\sachi\OneDrive\Desktop\crypto2> & C:/Python312/python.exe c:/Users/sachi/OneDrive/Desk
top/crypto2/send_to_pc3.py
PC2 (server) is listening for connections.
Connected to PC3 at ('192.168.137.56', 52254).
Sending metadata from pc2(server) to pc3
Sent encrypted metadata for ldLjO1apo5.json to PC3.
Sent encrypted metadata for QT4bM8fcqq.json to PC3.
```

```
PS C:\Users\Siddhartha Nath\OneDrive\Documents\CRYPTO_IMAGE_RSA_NFT> python -u "c:\Users\Siddhartha Nath\OneDrive\Documents\CRYPTO_IMAGE_RSA_NFT\p
c3_receiver_encrypt.py"
PC3 Connected to PC2.
PC3 Received metadata from PC2:
{"title": "Mountain", "description": "Beautiful Mountain", "artist": "Sid", "creation_date": "2024-04-21", "file_path": "encrypted_image.enc", "wa
llet_address": "64a778398b08bcad181713f2e610da34d73d73e6aec53e45d5b8e065bb4b7954", "sender_ip": "192.168.137.56", "sender_port": 8080, "unique_id"
: "ldLjO1apo5"}
{"title": "Doreamon", "description": "A cute white crying Doreamon", "artist": "Sid", "creation_date": "2024-04-15", "file_path": "encrypted_image
.enc", "wallet_address": "64a778398b08bcad181713f2e610da34d73d73e6aec53e45d5b8e065bb4b7954", "sender_ip": "192.168.137.56", "sender_port": 62201,
"unique_id": "QT4bM8fcqq"}
PS C:\Users\Siddhartha Nath\OneDrive\Documents\CRYPTO_IMAGE_RSA_NFT>
```

5. PC3 sends the unique identification of the image of his interest. PC2 encrypts the image with the received unique ID and sends it back to the PC3. PC3 decrypts the image and stores it.

```
PS C:\Users\Siddhartha Nath\OneDrive\Documents\CRYPTO_IMAGE_RSA_NFT> python
Waiting for PC2 to accept
Connected to PC2.
Sent request to see the image with unique ID: ldLjO1apo5
Request sent to PC2 to see the image.
PS C:\Users\Siddhartha Nath\OneDrive\Documents\CRYPTO_IMAGE_RSA_NFT>
```

```
PS C:\Users\sachi\OneDrive\Desktop\crypto2> & C:/Python312/python.exe c:/Users/sachi/OneDrive/Desk
top/crypto2/receive_id_from_pc3_image_encry_store.py
PC2 (server) is listening for connections from PC3.
Connected to PC3 at ('192.168.137.56', 52388).
Received unique ID (ldLjO1apo5) from PC3.
Image encrypted and stored successfully.
Process completed.
PS C:\Users\sachi\OneDrive\Desktop\crypto2>
```

```
PS C:\Users\sachi\OneDrive\Desktop\crypto2> & C:/Python312/python.exe c:/Users/sachi/OneDrive/Desk
top/crypto2/sending_encrypt_img_to_pc3.py
pc2(server) waiting for receiver to connect...
------------------------------
Sender connected with ('192.168.137.56', 52410)
Encrypted data sent to PC3 successfully.
PS C:\Users\sachi\OneDrive\Desktop\crypto2>
```

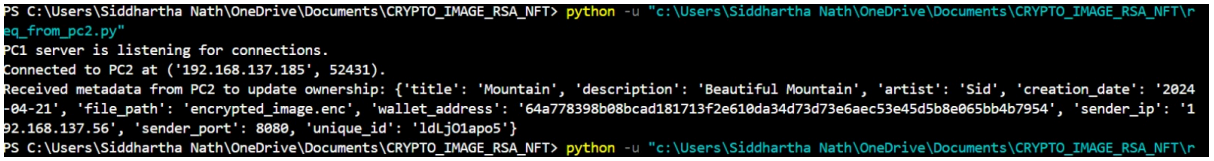
```
PS C:\Users\Siddhartha Nath\OneDrive\Documents\CRYPTO_IMAGE_RSA_NFT> python
m_pc2.py"
Waiting for PC2 to send the image...
PC3 connected to PC2.
Receiving encrypted data...
Encrypted data received successfully.
Image decrypted and stored successfully.
PS C:\Users\Siddhartha Nath\OneDrive\Documents\CRYPTO_IMAGE_RSA_NFT>
```

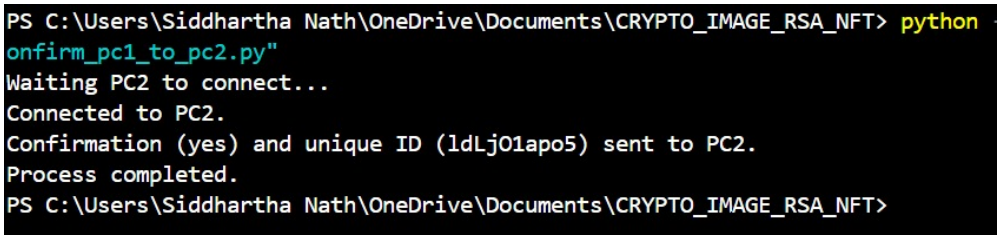
6. PC3 sends the unique ID of the image to PC2 to update ownership. PC2 forwards the metadata of the requested image to the actual owner.

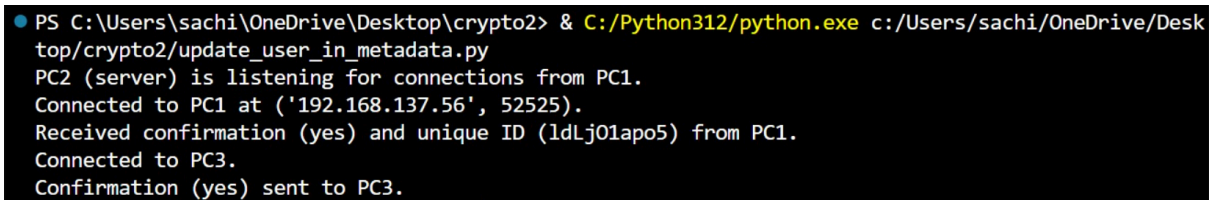
```
PS C:\Users\Siddhartha Nath\OneDrive\Documents\CRYPTO_IMAGE_RSA_NFT> python
eq_pc3_img.py"
Waiting for PC2 to accept
Connected to PC2.
Sent request to update the ownership on image with unique id ldLjO1apo5
PS C:\Users\Siddhartha Nath\OneDrive\Documents\CRYPTO_IMAGE_RSA_NFT>
```

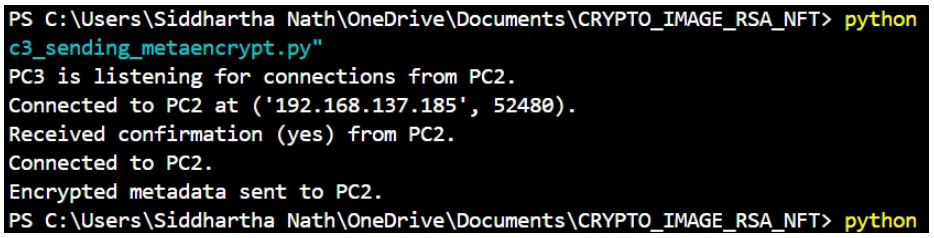
```
PS C:\Users\sachi\OneDrive\Desktop\crypto2> & C:/Python312/python.exe c:/Users/sachi/OneDrive/Desk
top/crypto2/receive_uid_from_pc3_send_to_pc1.py
PC2 (server) is listening for connections.
Connected to PC3 at ('192.168.137.56', 52467).
Received unique ID from PC3: ldLjO1apo5
Metadata sent to PC1.
Connected to PC3 at ('192.168.137.56', 52470).
Received unique ID from PC3: ldLjO1apo5
```

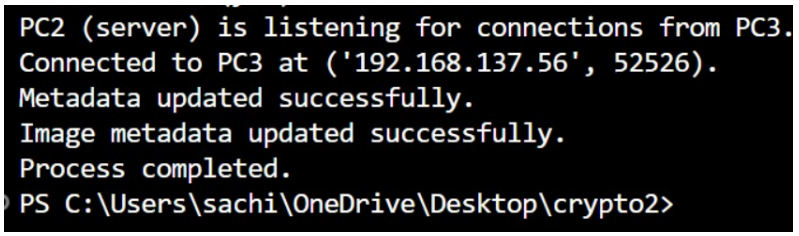
7. PC1(actual owner) receives the metadata and sends its confirmation to the PC2. PC2 forwards the confirmation to PC3. If the confirmation is ‘yes’, PC3 sends its data by encrypting it back to PC2 to update the ownership of the image. PC3 decrypts the metadata and updates it in the database.











Result Analysis:

The implemented secure artwork transfer effectively employs RSA and AES encryption techniques to ensure both image and metadata confidentiality and integrity throughout transmission and storage. RSA key exchange establishes secure communication channels, while symmetric encryption with AES adds an extra layer of protection for both artwork and metadata. Authentication is achieved through the exchange of symmetric key encryption, facilitating mutual authentication and integrity, preventing unauthorized access.

Potential improvements include implementing digital signatures or hash functions for enhanced authenticity and integrating error-handling mechanisms for improved robustness. Overall, the implementation provides a strong foundation for secure art transfer, emphasizing

the importance of regularly updating and improving security measures to keep up with new threats and technology changes.

Learning Outcomes:

- Understanding of symmetric and asymmetric encryption techniques for images and files.
- Understanding of symmetric key exchange using RSA Public Cryptosystem.
- Proficiency in implementing secure communication using sockets in Python.

Conclusion:

In conclusion, Secure Artwork Transfer successfully addresses the vulnerability of artwork and its metadata during transmission by implementing a robust encryption mechanism. The secure communication ensures that metadata is transmitted and stored securely, minimizing the risk of unauthorized access or tampering.