RAMAIAH
Institute of Technology

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Cryptography and Network Security – CS62

Non CIE Component Report

5th Semester 'B' Section

Submitted by

Mihika Dhariwal	**1MS21CS075**
Prachi Patil	**1MS21CS094**
Chethana Peramana	**1MS21CS089**
Moulya R Gowda	**1MS21CS145**

Under the guidance of

Nandini S. B.
Assistant Professor, Dept. of CSE

RAMAIAH INSTITUTE OF TECHNOLOGY
(Autonomous Institute, Affiliated to VTU)
BANGALORE-560054
www.msrit.edu, December 2022

RAMAIAH
Institute of Technology

CERTIFICATE

This is to certify that the Project work carried out by **Mihika Dhariwal (1MS21CS075), Prachi Patil (1MS21CS094), Chethana Peramana (1MS21CS0--) and Moulya R Gowda (1MS21CS145)** as a 20-mark component for the course Cryptography and Network Security (CS62), VI semester B.E, CSE during the academic year January 2023 - May 2023 satisfies the academic requirements for awarding the marks.

Signature of the Faculty

Content

CRYPTOGRAPHIC ALGORITHMS IN BLOCKCHAIN

ABSTRACT

With the growing importance of secure communication and data integrity in the digital landscape, cryptographic techniques play a pivotal role in safeguarding sensitive information. This project explores the integration of the RSA and SHA 256 algorithm into a simple blockchain. The SHA 256 algorithm is used to hash transaction data within the blockchain whereas RSA algorithm's key pair generation, encryption, and decryption processes are leveraged to enhance the security and authenticity of transactions within the blockchain. Each block in the blockchain contains an encrypted amount, digital signatures, and hash values, providing a secure and tamper-evident environment for data transfer. This project serves as an educational and practical exploration of RSA's application in the context of blockchain technology. It underscores the practical application of RSA in real-world scenarios, emphasizing its role in ensuring the confidentiality, integrity, and authenticity of data within a decentralized and distributed ledger. As the digital landscape continues to evolve, understanding the synergy between cryptographic algorithms and blockchain technology becomes imperative for building resilient and secure systems that can withstand the challenges of an increasingly interconnected world. Through this exploration, the project contributes to the broader discourse on the integration of cryptographic techniques within emerging technologies, fostering a deeper understanding of their practical implications for secure data transfer and communication.

Keywords: *Blockchain, RSA, SHA 256, cryptography, decentralized technology, encryption*

INTRODUCTION

Blockchain technology is a decentralized and distributed ledger system that enables secure and transparent record-keeping of digital transactions. It is the underlying technology behind cryptocurrencies like Bitcoin, but its applications extend far beyond digital currencies. Blockchain has gained significant attention for its potential to revolutionize various industries by providing a secure and tamper-resistant way to record, store, and verify data.

Key features of blockchain technology include:

1. Decentralization: Unlike traditional centralized systems, where a single entity or authority controls the data, blockchain operates on a decentralized network of computers (nodes). Each node in the network has a copy of the entire blockchain, promoting transparency and resilience.

2. Immutable Ledger: Once data is added to the blockchain, it becomes extremely difficult to alter or delete. Each block contains a cryptographic hash of the previous block, creating a chain of blocks that are linked together. This feature ensures the integrity and immutability of the transaction history.

3. Consensus Mechanism: To validate and add new transactions to the blockchain, nodes in the network must agree on the state of the ledger. Consensus mechanisms, such as Proof of Work (used in Bitcoin) or Proof of Stake, ensure that a majority of nodes reach an agreement before a new block is added.

4. Smart Contracts: Smart contracts are self-executing contracts with the terms of the agreement directly written into code. They automatically enforce and execute the terms when predefined conditions are met. Smart contracts run on the blockchain, reducing the need for intermediaries and increasing efficiency.

5. Security: The decentralized and cryptographic nature of blockchain makes it highly secure. Transactions are secured through cryptographic algorithms, and the distributed nature of the ledger makes it resistant to hacking or fraud attempts.

USE OF CRYPTOGRAPHY IN BLOCKCHAIN

Blockchain relies on cryptography to secure data, ensuring trust and privacy. Cryptographic tools guarantee immutability, data integrity, and user privacy within the decentralized network. Let's explore the practical role of cryptography in the blockchain ecosystem.

1. Hashing:

At the heart of blockchain lies the concept of a "block," a record containing data along with a unique identifier called a "hash." Cryptographic hash functions play a vital role here, acting as complex mathematical formulas that "squeeze" any amount of data into a fixed-size string of alphanumeric characters known as a hash. These functions are deterministic and unique, meaning they generate the same output for the same input and produce entirely different hashes for even slight changes in the input. This chaining mechanism forms the backbone of blockchain, with each block including the hash of the previous block, creating an unbreakable chain. Any attempt to tamper with data in a block would alter its hash, triggering inconsistencies in subsequent blocks, making manipulation practically impossible.

2. Digital Signatures:

Digital signatures in blockchain use asymmetric cryptography, involving a pair of mathematically linked keys: a public key (widely shared) and a private key (kept secret). When signing a transaction, the user uses their private key to encrypt a unique string representing the transaction data. Verification of the signature can be done by anyone using the corresponding public key, ensuring the authenticity of the signer's identity and preventing repudiation.

3. Public Key Infrastructure (PKI):

PKI provides the framework for managing and distributing public and private keys across the blockchain network. Trusted entities, known as "certificate authorities," verify user identities and issue signed certificates containing their public keys. This infrastructure enables secure communication and identity verification within the decentralized ecosystem.

4. Secure Communication:

Communication between peers (nodes) in the blockchain network requires secure channels to protect sensitive data from eavesdropping or tampering. Cryptographic protocols like Transport Layer Security (TLS) encrypt communication between nodes, safeguarding data and ensuring the network's integrity.

5. Privacy Enhancements:

While pseudonymity is inherent in blockchain systems, users may seek enhanced privacy protection for their transactions. Cryptographic tools like ring signatures and zero-knowledge proofs are explored to allow users to prove their involvement in transactions without revealing their specific identity or the exact amount transacted.

Beyond Security:

Cryptography's role in blockchain extends beyond securing data. It facilitates trust and transparency by enabling decentralization, immutability, and auditability. Advanced encryption techniques are employed to scramble sensitive data, making it unreadable to unauthorized parties. The sender's use of their private key to sign transactions ensures only authorized senders can initiate transactions and that transactions remain unaltered once recorded on the blockchain. This process guarantees the historical accuracy of transactions and promotes transparency and accountability in the decentralized network.

RSA ALGORITHM

The RSA algorithm, named after its inventors Ron Rivest, Adi Shamir, and Leonard Adleman, is a widely used public-key cryptosystem that revolutionized secure communication in the digital age. Developed in 1977, RSA remains a cornerstone of modern cryptography, providing a secure method for encrypting sensitive information over untrusted networks.

Characteristics:

1. Public-Key Cryptosystem

RSA is a public-key cryptosystem, which means it uses a pair of keys: a public key for encryption and a private key for decryption. The security of the algorithm relies on the difficulty of factoring the product of two large prime numbers.

2. Key Generation

The key generation process involves selecting two large prime numbers, multiplying them to obtain a modulus, and then deriving the public and private keys. The security of RSA is closely tied to the difficulty of factoring the modulus back into its prime components.

3. Asymmetric Encryption

RSA employs asymmetric encryption, where the public key is used for encryption, and only the corresponding private key can decrypt the data. This ensures a secure communication channel even if the public key is known to potential attackers.

How It Works:

1. Key Generation

The RSA algorithm begins with key generation, where two large prime numbers, p and q, are selected. The product of these primes, n = pq, is used as the modulus for both the public and private keys.

2. Public Key Infrastructure

The public key, consisting of the modulus n and an exponent e, is widely distributed and made available to anyone who wants to send encrypted messages to the owner of the corresponding private key.

3. Encryption

When someone wants to send a secure message, they use the recipient's public key to encrypt the message. This encrypted message can only be decrypted by the recipient's private key.

4. Decryption

The recipient uses their private key to decrypt the message and retrieve the original content. The security of RSA relies on the difficulty of factoring the product of two large primes, ensuring that unauthorized parties cannot easily decrypt the message.

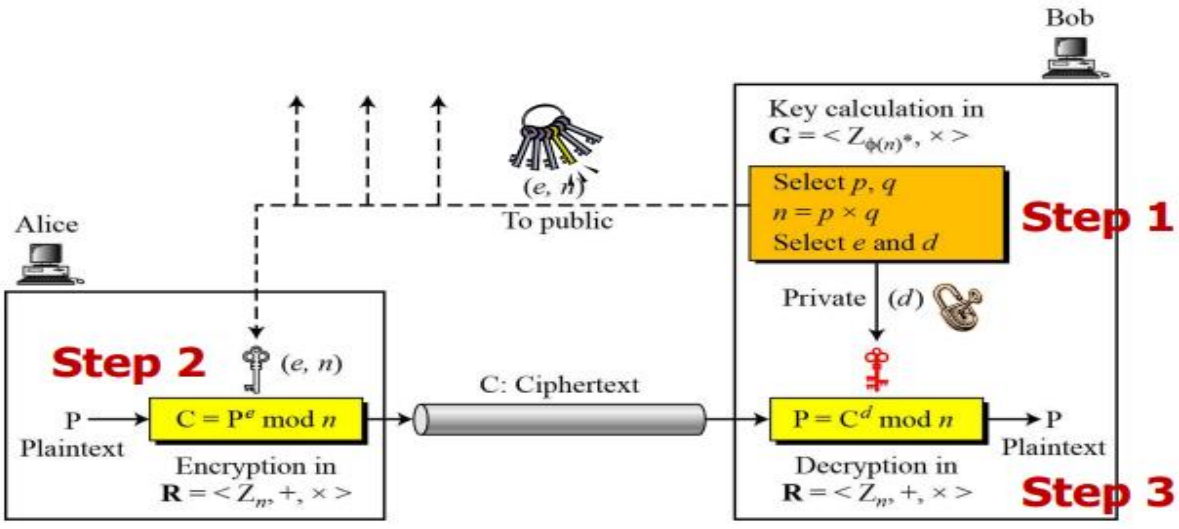


Fig: How RSA Algorithm Works

Key Generation:

1.Choose two large prime numbers (p and q)

2.Calculate n = p*q and $\Phi(n) = (p-1)(q-1)$

3.Choose a number e , where $1 < e < \Phi(n)$, e is coprime to $\Phi(n)$, $\gcd(e, \Phi(n)) = 1$

4.Calculate $d = e^{-1} \bmod n$, or $de \bmod \Phi(n) = 1$

5.Public key pair as (e,n)

6.Private key is d

Encryption/Decryption Function:

1.If the plaintext is P, ciphertext

$C = P^e \bmod n$

2.If the ciphertext is C, plaintext

$P = C^d \bmod n$

Advantages:

1. Security

RSA offers a high level of security due to the complexity of factoring large numbers, making it resistant to brute-force attacks. The security of the algorithm is further enhanced by regularly updating key pairs.

2. Authentication

RSA is not only used for encryption but also for digital signatures and authentication. Digital signatures generated using RSA provide a way to verify the authenticity and integrity of digital messages.

Applications:

1. Secure Communication

The primary application of RSA is in securing communication over untrusted networks. It is widely used in protocols such as HTTPS for securing online transactions, email encryption, and virtual private networks (VPNs).

2. Digital Signatures

RSA's capability for creating digital signatures ensures the authenticity and integrity of digital documents. This is crucial in applications such as electronic commerce, legal documents, and software distribution.

3. Key Exchange

RSA is employed in key exchange protocols, allowing parties to securely exchange secret keys for symmetric encryption. This is commonly used in secure communication systems and online banking.

SHA ALGORITHM

The SHA-256 (Secure Hash Algorithm 256-bit) is a cryptographic hash function that belongs to the SHA-2 family. It's widely used for generating fixed-size hash values (256 bits or 64 characters in hexadecimal) from variable-sized input data.

Algorithm:

1. **Padding:**

If the input message length is not a multiple of 512 bits, padding is applied to make it so. Padding includes appending a single '1' bit, followed by a series of '0' bits, and finally, the length of the original message in bits.

2. **Initial Hash Values (H):**

SHA-256 uses eight 32-bit words as initial hash values (H0 to H7). These values are derived from the first 32 bits of the fractional parts of the square roots of the first eight prime numbers.

3. **Breaking Message into Blocks:**

The padded message is broken into blocks of 512 bits each.

4. **Processing Blocks:**

Each 512-bit block is processed through a series of logical and arithmetic operations to update the hash values.

5. **Message Schedule (W):**

Each 512-bit block is divided into sixteen 32-bit words, forming the initial message schedule (W[0] to W[15]).

6. **Expansion of Message Schedule:**

The message schedule is expanded to create a total of 64 words using a recurrence relation.

7. Main Compression Loop (64 Rounds):

The main processing loop involves 64 rounds of operations that update the hash values based on the message schedule and constant values. Each round uses different functions such as Ch (choose), Maj (majority), Sigma functions, and bitwise operations like bitwise rotation, bitwise shifting, and modular addition.

8. Update Hash Values:

The hash values (H0 to H7) are updated after each 512-bit block is processed.

9. Final Hash Value:

The final hash value is a concatenation of the updated hash values H0 to H7.

USE OF RSA AND SHA 256 TO ENCRYPT AND DECRYPT DATA

RSA and SHA-256 are utilized for different purposes within a blockchain:

USE OF SHA-256

In a blockchain transaction, relevant information includes details like the sender's address, receiver's address, and the amount being transferred. This data is collectively referred to as the transaction data.

SHA-256 is a cryptographic hash function that takes an input and produces a fixed-size output of 256 bits, commonly represented as a 64-character hexadecimal number. The goal of hashing the transaction data with SHA-256 is to create a unique fingerprint or hash value for that specific set of transaction details.

The hash is deterministic, meaning the same input will always produce the same hash value. Any change in the transaction data, no matter how small, will result in a completely different hash value. This property ensures the integrity of the transaction. If someone attempts to modify the sender's address, receiver's address, or the amount being transferred, the resulting hash will be entirely different.

The hash generated by SHA-256 serves as a critical component in the creation of a block. In a blockchain, each block contains a hash of the previous block (making the blockchain a linked list of blocks). This creates a chain of blocks where each block is linked to its predecessor through the hash. The immutability of the blockchain is enforced by the fact that altering any block's content would require changing its hash. Since each block's hash is used as input for the next block, any change in a previous block would cascade through the entire chain. This makes it computationally infeasible to alter historical transactions without detection.

USE OF RSA

The hash value generated is then encrypted using RSA with the sender’s private key. The private key, known only to the sender, plays a crucial role in proving the origin of the data. The result of this encryption is a digital signature, representing a secure and unique encapsulation of the transaction's hash.

During the verification process, the recipient, or any validating entity, utilizes the sender's public key to decrypt the digital signature. The decryption reveals the original hash of the transaction data. Simultaneously, the recipient independently computes the hash of the received transaction data (or its hash if not provided) using the same hash function. The comparison of the decrypted hash from the signature with the newly generated hash determines the validity of the signature. If the two hash values match, the signature is considered valid, assuring that the data has not been tampered with.

The significance of this cryptographic process in blockchain is multifaceted. Firstly, it ensures data integrity within a block by detecting any unauthorized alterations through hash value mismatches. Secondly, the public key used for signature verification serves as a means of authentication, confirming the identity of the sender. Additionally, non-repudiation is established as the private key holder cannot deny involvement in creating the transaction, given that only their private key could produce the correct signature. Lastly, the use of asymmetric cryptography, with public and private keys, adds a robust layer of security. It makes it computationally infeasible for others to forge a valid signature without possessing the private key, thus safeguarding the integrity and authenticity of transactions in the blockchain.

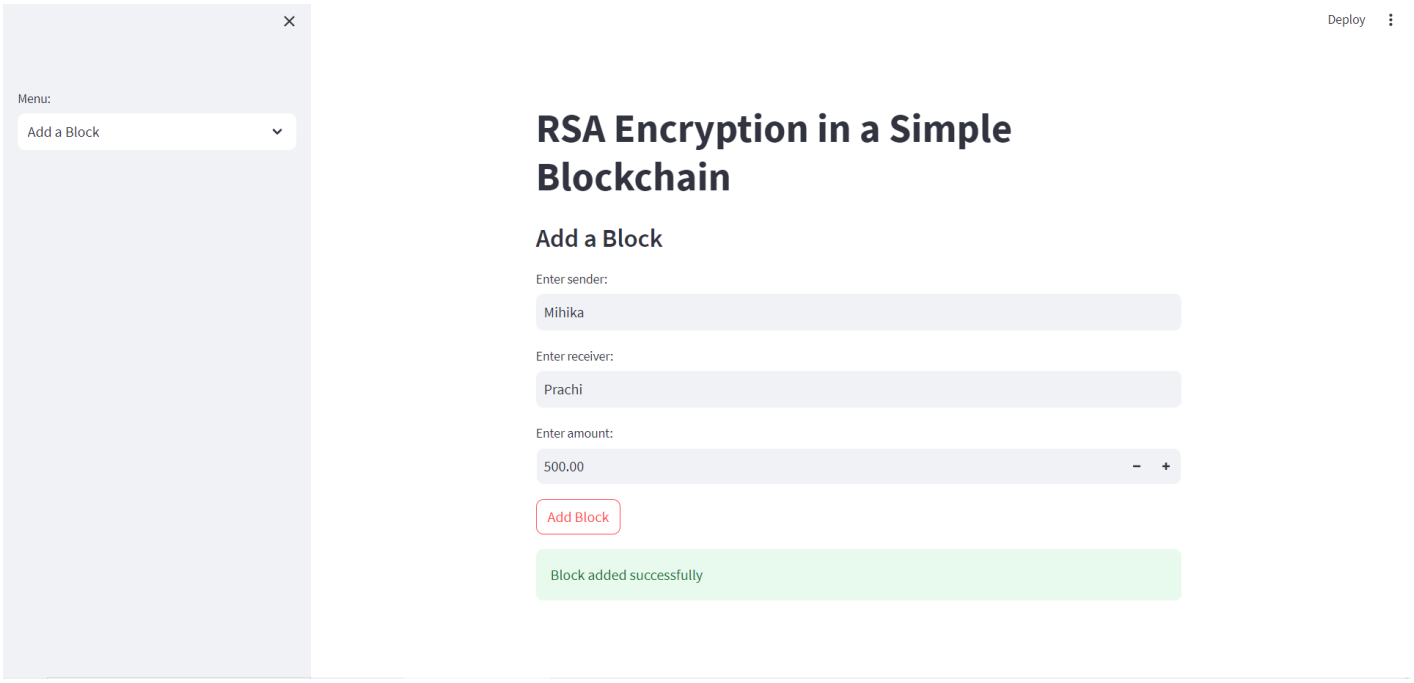
SOURCE CODE

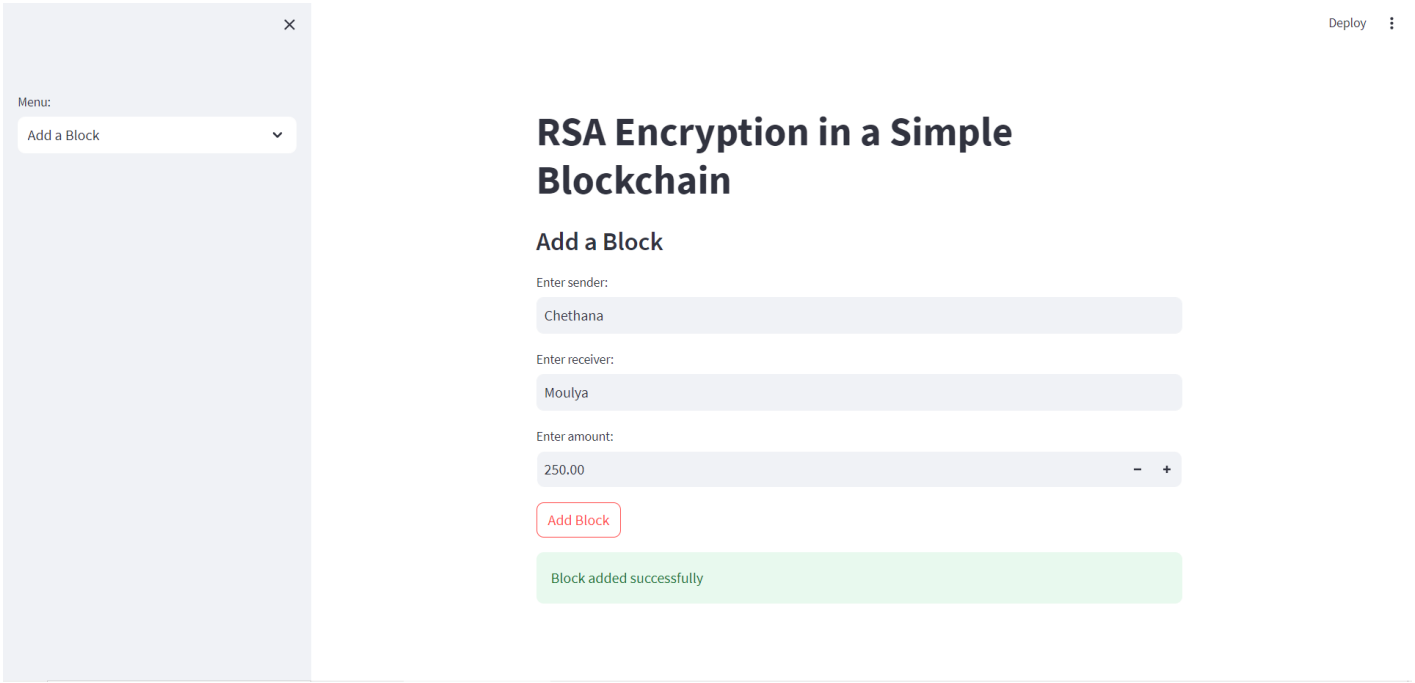
```
import streamlit as st
import hashlib

# Functions for RSA encryption and decryption, and key generation
def gcd(a, b):
    while b != 0:
        a, b = b, a % b
    return a

def mod_inverse(a, m):
    m0, x0, x1 = m, 0, 1
    while a > 1:
        q = a // m
        m, a = a % m, m
        x0, x1 = x1 - q * x0, x0
    return x1 + m0 if x1 < 0 else x1

def generate_keypair(p, q):
    n = p * q
    phi = (p - 1) * (q - 1)
    e = 65537  # Commonly used public exponent
    d = mod_inverse(e, phi)
    return ((e, n), (d, n))

def encrypt(message, private_key):
    d, n = private_key
    cipher = [pow(ord(char), d, n) for char in message]
    return cipher

def decrypt(cipher, public_key):
    e, n = public_key
    plain = [chr(pow(char, e, n)) for char in cipher]
    return ''.join(plain)

class Block:
    def __init__(self, sender, receiver, amount):
        self.sender = sender
        self.receiver = receiver
        self.amount = amount
        self.public_key, self.private_key = generate_keypair(61, 53)

        # Hashing and signing using SHA-256
        transaction_data = f"{sender}{receiver}{amount}"
        self.hash = hashlib.sha256(transaction_data.encode()).hexdigest()
        self.signature = encrypt(self.hash, self.private_key)

    def verify_signature(self):
        decrypted_hash = decrypt(self.signature, self.public_key)
```

```
        return self.hash == decrypted_hash

class Blockchain:
    def __init__(self):
        self.chain = []

    def add_block(self, sender, receiver, amount):
        block = Block(sender, receiver, amount)
        self.chain.append(block)

    def display_chain(self):
        return self.chain

def main():
    st.title("RSA Encryption in a Simple Blockchain")

    if 'blockchain' not in st.session_state:
        st.session_state.blockchain = Blockchain()
    menu_choice = st.sidebar.selectbox("Menu:", ["Add a Block", "Display Blockchain"])

    if menu_choice == "Add a Block":
        st.subheader("Add a Block")
        sender = st.text_input("Enter sender:")
        receiver = st.text_input("Enter receiver:")
        amount = st.number_input("Enter amount:")

        if st.button("Add Block"):
            st.session_state.blockchain.add_block(sender, receiver, amount)
            st.success("Block added successfully")

    elif menu_choice == "Display Blockchain":
        st.subheader("Blockchain")
        chain = st.session_state.blockchain.display_chain()

        if len(chain) == 0:
            st.write("Blockchain is empty.")
        else:
            for index, block in enumerate(chain, start=1):
                st.write(f"Block {index}:")
                st.write(f"Sender: {block.sender}")
                st.write(f"Receiver: {block.receiver}")
                st.write(f"Amount: {block.amount}")
                st.write(f"Block Hash: {block.hash}")
                st.write(f"Signature: {block.signature}")
                st.write(f"Signature Verified: {block.verify_signature()}")
                st.write("")
if __name__ == "__main__":
    main()
```

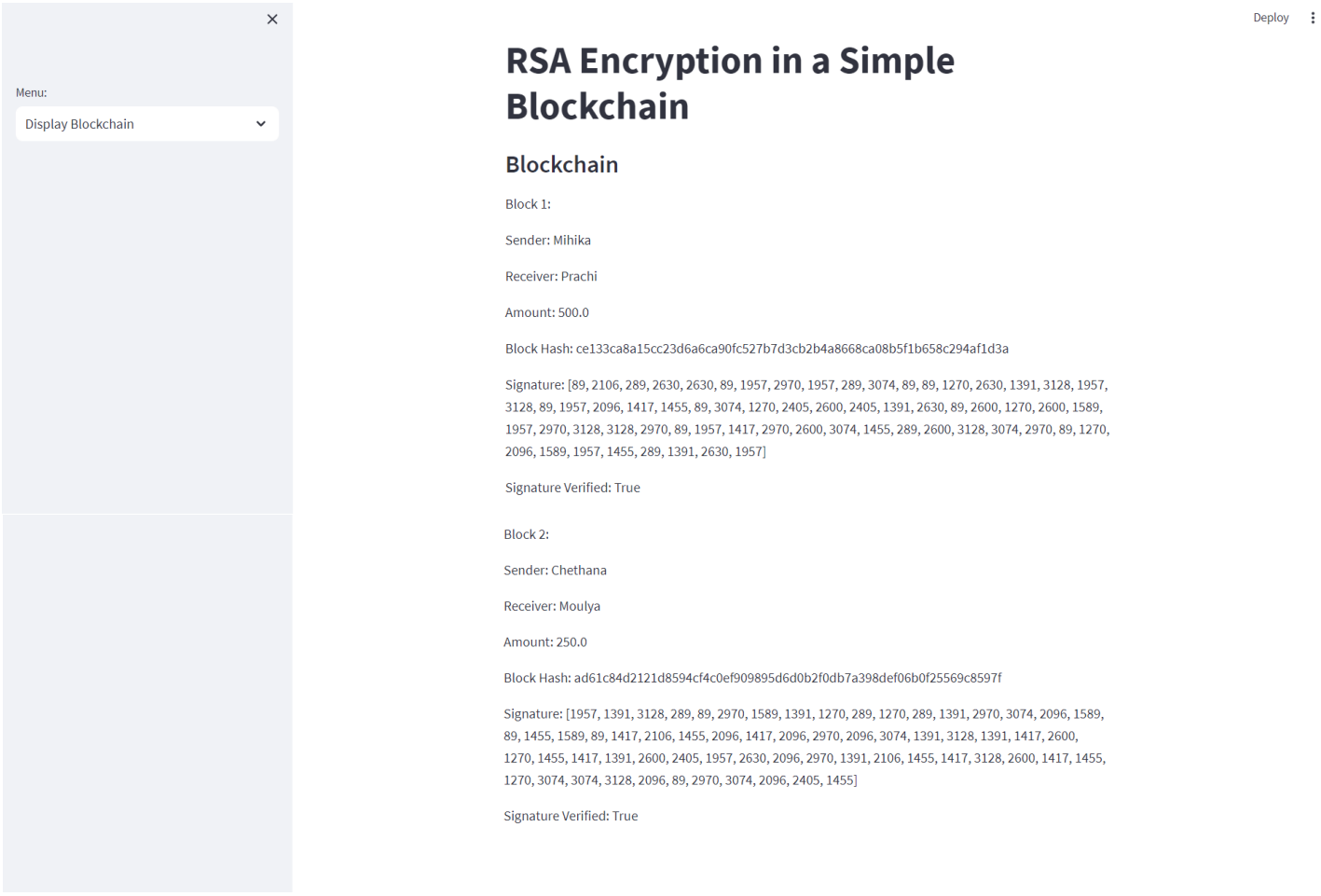
GRAPHICAL USER INTERFACE

2 Blocks are added to the blockchain





Displaying the 2 added blocks



CONCLUSION

In conclusion, the integration of the RSA and SHA 256 algorithm into a simple blockchain showcases the versatility of cryptographic techniques in ensuring secure and trustworthy transactions. The SHA 256 algorithm’s hashing and the RSA algorithm's asymmetric key pair generation, encryption, and decryption capabilities contribute to the robustness of the blockchain's security infrastructure. The project not only demonstrates the encryption and decryption of transaction amounts but also incorporates digital signatures and hash values for data integrity verification. The blockchain's transparency and immutability, combined with RSA's cryptographic strength, create a reliable framework for secure communication and financial transactions. As we navigate an era where cybersecurity is paramount, understanding and implementing such cryptographic solutions become crucial for developing resilient systems in the digital realm. This project stands as a testament to the symbiotic relationship between blockchain and RSA encryption in addressing the evolving challenges of secure data exchange.

REFERENCES

https://www.geeksforgeeks.org/rsa-algorithm-cryptography/

https://www.simplilearn.com/tutorials/cyber-security-tutorial/sha-256-algorithm

https://www.ibm.com/topics/blockchain

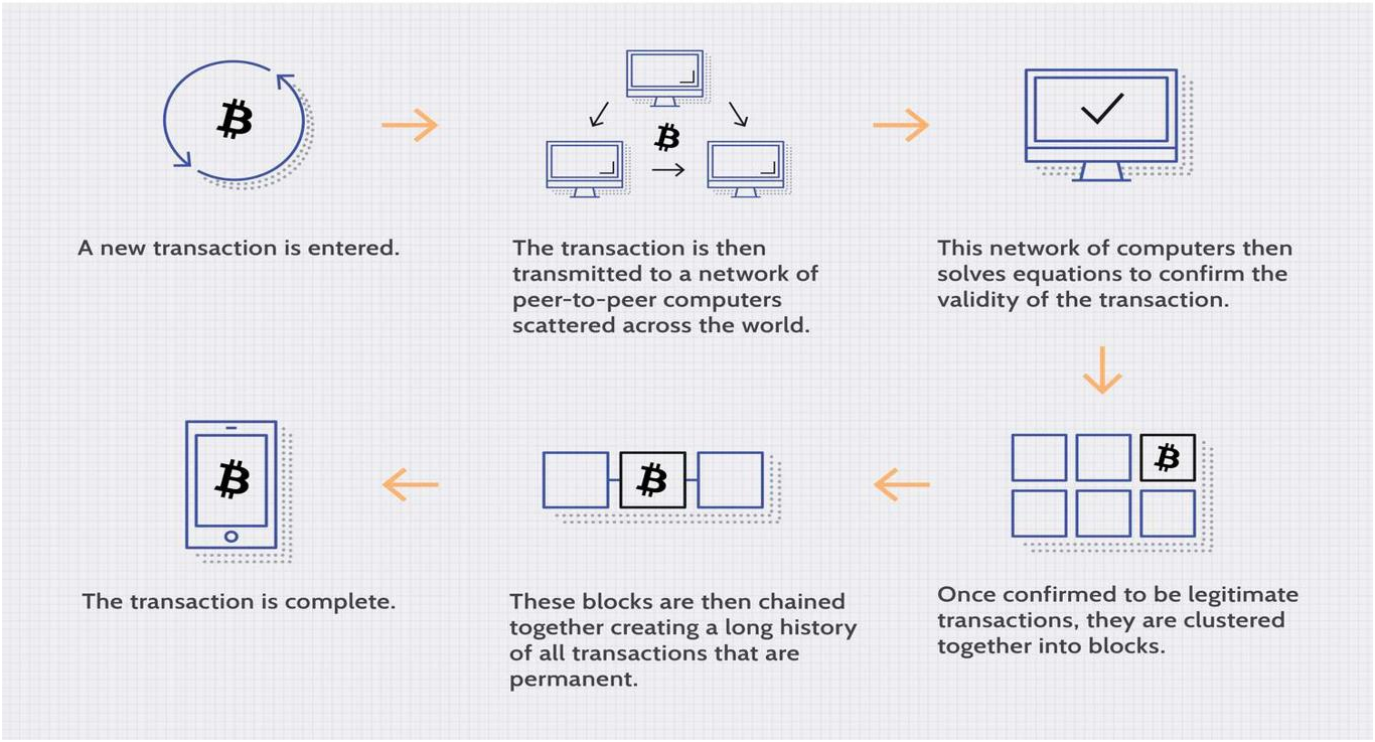
https://www.geeksforgeeks.org/cryptography-in-blockchain/

https://101blockchains.com/public-key-cryptography-in-blockchain/

PROJECT PRESENTATION SLIDES

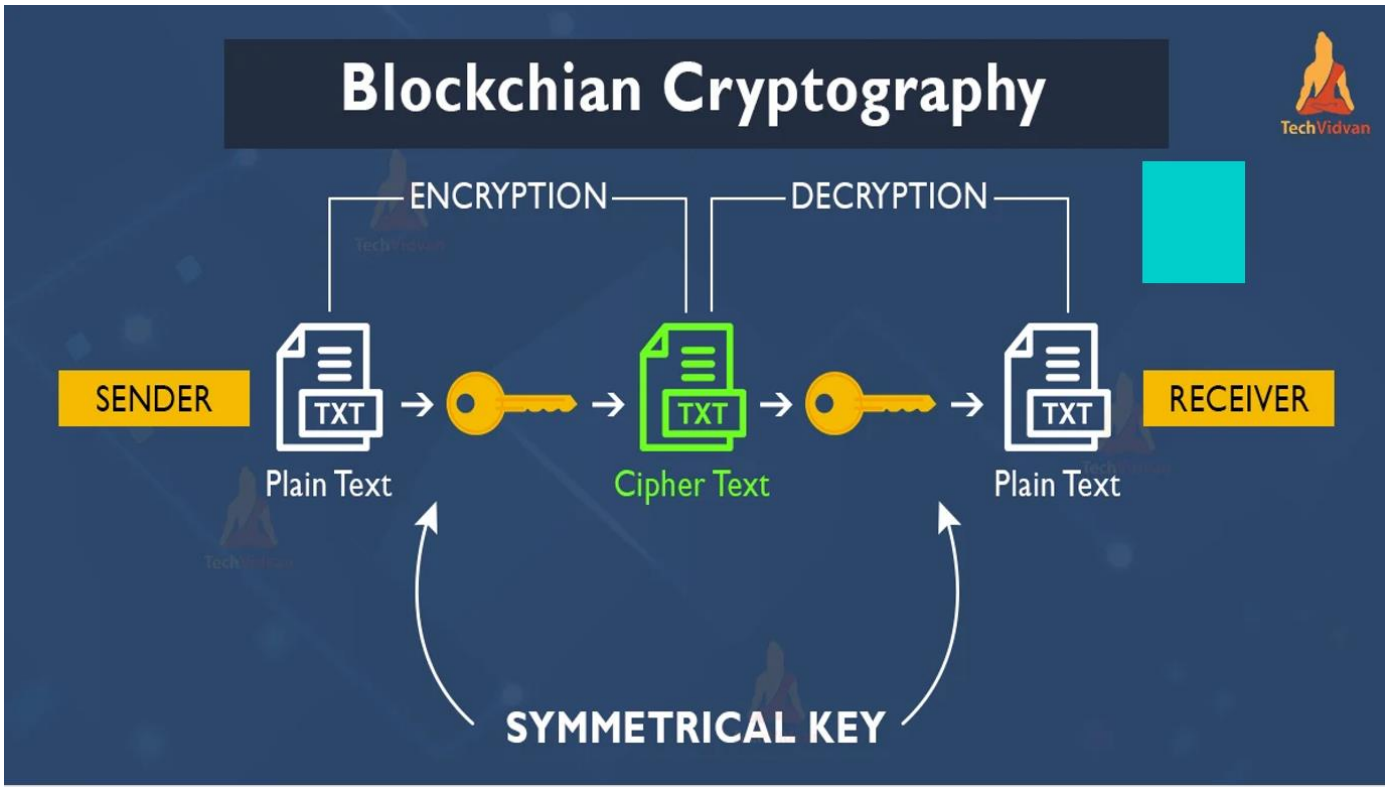
Introduction

- Blockchain technology is an advanced database mechanism that allows transparent information sharing within a business network. A blockchain database stores data in blocks that are linked together in a chain.
- The data is chronologically consistent because you cannot delete or modify the chain without consensus from the network.
- As a result, you can use blockchain technology to create an unalterable or immutable ledger for tracking orders, payments, accounts, and other transactions. The system has built-in mechanisms that prevent unauthorized transaction entries and create consistency in the shared view of these transactions.



How is cryptography used in blockchain

- Blockchain is a distributed ledger technology that allows for secure and transparent transactions. It is a decentralized database that is stored across multiple computers, making it resistant to tampering and fraud.
- Cryptography employs advanced encryption techniques to scramble sensitive data like user identities, transaction details, and confidential information stored on the blockchain, making it unreadable to unauthorized parties.

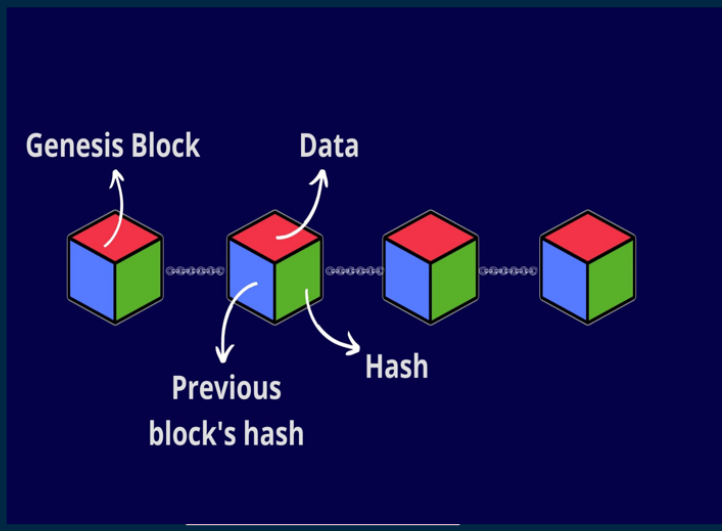


Transaction Authenticity

- Digital signatures act as digital fingerprints, they protect against double-spending, unauthorized transactions, and maintain the immutability.
- The sender uses their private key to "sign" the transaction, creating a unique digital signature. This signature is then verified by the blockchain network using the sender's public key.
- If the digital signature matches the sender's public key, the network confirms the transaction's authenticity and adds it to the blockchain.
- This process ensures that only authorized senders can initiate transactions and that transactions remain unaltered once recorded on the blockchain.

Hash functions

Hash functions are used to verify the integrity of data stored on a blockchain. A hash function produces a unique and fixed-size output for any given input.



RSA Algorithm

- RSA encryption is used to protect transaction amounts, while SHA-256 hashing ensures data integrity and immutability. SHA-256 creates a unique fingerprint for each transaction block and detects even minor data changes.
- This hashed value is then encrypted with RSA to form a digital signature, verifying the transaction's authenticity.

RSA AND SHA IN BLOCKCHAIN