

Original Research Article

BLOCK CHAIN AND CRYPTOGRAPHY BASED SECURE COMMUNICATION SYSTEM

Abstract- The block chain is a cutting-edge technology that reduces these risks by allowing vital tasks to be decentralized while retaining a high level of security. It eliminates the need for dependable intermediaries. The block chain, which records all past transactions, is accessible to all network nodes. The goal of our project is to create a block chain-based secure communication system. We explain why block chain would improve communication security and propose a model design for block chain-based messaging that focuses on training the performance and security of data recorded on the block chain, as well as using a smart contract to verify identities and their associated public keys, as well as validating the users certificate.

Keywords: Block Chain, Security, Validation, Public Keys

1. INTRODUCTION

Block chain is a decentralized, traceable, non-tamper able, secure, and trustworthy distributed database. This project integrates the P2P protocol, digital encryption technology, a consensus method, a smart contract, and other technologies. Instead of the conventional central node maintenance method, a strategy of mutual maintenance by diverse users is used to establish information supervision among numerous parties, ensuring the data's credibility and integrity. Block chain systems are classified into three types: public chains, private chains, and alliance chains. The public chain allows any node to join or leave at any moment, while the private chain has tight requirements for participating nodes and the alliance chain is administered collaboratively by multiple partnering institutions.

As a representation of distributed databases, all user transaction information is recorded on the block chain, which has strict security standards. Block chain is a peer-to-peer network that is decentralized. There is no need for nodes to trust one another, and there is no central node. As a result, transactions on the block chain must maintain the confidentiality of transaction

information while preserving transaction integrity over insecure connections. As can be seen, cryptography is the most important block chain technology.

As a representation of distributed databases, all user transaction information is recorded on the block chain, which has strict security standards. Block chain is a peer-to-peer network that is decentralized. There is no need for nodes to trust one another, and there is no central node. As a result, transactions on the block chain must maintain the confidentiality of transaction information while preserving transaction integrity over insecure connections. As can be seen, cryptography is the most important block chain technology. Cryptography is widely utilized in block chain to safeguard user privacy and transaction information, as well as to assure data integrity.

The hash algorithm is a function that turns a series of messages of variable lengths into a single shorter value. Some of its properties are susceptibility, unidirectionality, collision resistance, and high sensitivity. Hash is typically used to check data integrity or to confirm that data has not been tampered with incorrectly. When the data being tested changes, so does the hash value. As a result, even if the data is in a dangerous environment, the hash value may be utilized to determine the data's integrity.

Wu et al. proposed a method for ensuring the authenticity and non-repudiation of digital material while maintaining privacy for traceable encryption in block chain. The authors address the issue of the user's private key, which, when shared with other entities, does not contain the user's specific information. If the shared key is damaged or misused, it is difficult to determine where the secret key came from. Furthermore, access control leakage of sensitive information is a barrier to current solutions. The authors devised a privacy protection approach, such as attribute-based encryption, to safeguard the private keys (ABE). However, the decryption approach does not appear to be more efficient.

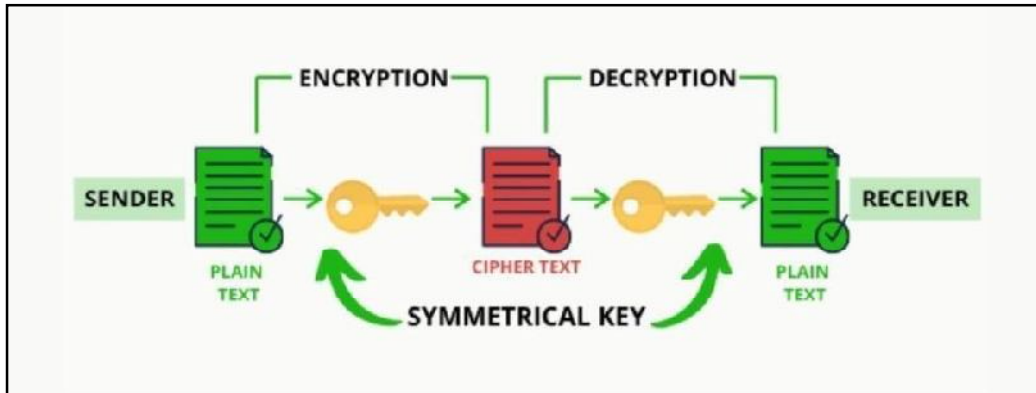


Figure 1: B and C Example

2. METHODOLOGY

Python language is used to write the code. Python provides a wide variety of libraries for scientific and computational usage. Libraries such as hashlib, rsa.

Consider the probability distribution D . $V \sim D$ denotes a random variable V that is distributed according to D . We indicate $u \sim D$ if an element u is sampled from a probability distribution of D . U denotes the uniform probability distribution on a set X . We use standard nomenclature for probabilistic algorithms. When an algorithm A is run with the input x and produces the result y , we call it $y = A(x)$. Oracles can be made available to algorithms. Oracles are "black boxes" that can compute things like functions and other methods. A^O denotes an algorithm A with oracle access to an oracle O . The time complexity of calling an oracle and obtaining an answer is assumed to be constant time $O(1)$.

Algorithm 1: Pseudorandom Cipher text Experiment

```

Algorithm 1 Pseudorandom Ciphertext Experiment
1: procedure  $\text{PRC\_EXP}_A^{\text{SE}}(1^s)$ 
2:    $k \leftarrow \text{Gen}(1^s)$ 
3:    $(m, S) \leftarrow A_1^{\text{Enc}_k}(1^s)$  ▷  $S$  is internal state information of  $A$ 
4:    $b \leftarrow U(\{0, 1\})$ 
5:   if  $b = 1$  then
6:      $c \leftarrow \text{Enc}(k, m)$  ▷ Use the actual encryption
7:   else
8:      $c \leftarrow U(\{0, 1\}^{|\text{Enc}(k, m)|})$  ▷ Use a random string
9:   end if
10:   $b' \leftarrow A_2^{\text{Enc}_c}(c, S)$ 
11:  if  $b = b'$  then
12:    return 1 ▷  $A$  guessed correctly
13:  else
14:    return 0 ▷  $A$  did not guess correctly
15:  end if
16: end procedure

```

Algorithm 2: Embedding Algorithm

```

Algorithm 2 Embedding Algorithm
1: procedure Embed( $(k, \lambda), m, \mathcal{B}$ )
2:    $c \leftarrow \text{Enc}(k, m)$ 
3:   Concatenate  $c' = \lambda || c$ 
4:   Set  $N = |c'|$ 
5:   Interpret  $c'$  as a bit representation  $c'_1 c'_2 \dots c'_N \in \{0, 1\}^N$ 
6:    $i = 1$ 
7:   while  $i \leq N$  do
8:     Generate unseal  $(s_k, p_k) \leftarrow \text{Gen}_\Sigma(1^s)$ 
9:      $a \leftarrow H(p_k^{(i)})$ 
10:    Interpret  $a$  as a bit representation  $a_1 a_2 \dots a_n \in \{0, 1\}^n$ 
11:    if  $a_n = c'_i$  then
12:       $\mu \leftarrow \mathcal{M}_\mathcal{H}$ 
13:      Generate a unique identifier  $t$  for the payment
14:       $\sigma \leftarrow \text{Sign}(s_k^{(A)}, (p_k^{(A)}, a, \mu, t))$ 
15:      Submit  $(p_k^{(A)}, a, \mu, t, \sigma)$ 
16:      Wait for the blockchain to publish a new block
17:      Update  $\mathcal{H}$ 
18:       $i \leftarrow i + 1$ 
19:    end if
20:  end while
21: end procedure

```

Algorithm 3: Extraction Algorithm

```

Algorithm 3 Extraction Algorithm
1: procedure Extract( $(\lambda, k), \mathcal{B}$ )
2:    $i = 1$ 
3:    $j = 1$ 
4:   while have not found  $\lambda$  yet do ▷ Scan for  $\lambda$ 
5:      $C = \text{Read}(j)$ 
6:     if  $C = \perp$  then
7:       Wait until a block appears and read it:  $C = \text{Read}(j)$ 
8:     end if
9:     for any payment  $P \in C$  do
10:      if  $P$  is from  $p_k^{(A)}$  then
11:        Extract address  $a$  from  $P$  and get the LSB  $a_n$ 
12:        Scan if we have found the entire  $\lambda \in \{0, 1\}^{\sigma_\lambda}$ 
13:      end if
14:    end for
15:     $j \leftarrow j + 1$ 
16:  end while
17:   $i = 1$  ▷ Now reading the encrypted hidden message
18:  while  $i \leq N - \sigma_\lambda$  do
19:     $C = \text{Read}(j)$ 
20:    if  $C = \perp$  then
21:      Wait until a block appears and read it:  $C = \text{Read}(j)$ 
22:    end if
23:    for any payment  $P \in C$  do
24:      if  $P$  is from  $p_k^{(A)}$  then
25:        Extract address  $a$  from  $P$  and get the LSB  $a_n$ 
26:         $c_i \leftarrow a_n$ 
27:         $i \leftarrow i + 1$ 
28:      end if
29:    end for
30:     $j \leftarrow j + 1$ 
31:  end while
32:  Compile  $c = c_1 c_2 \dots c_{N - \sigma_\lambda}$ 
33:   $m \leftarrow \text{Dec}(k, c)$ 
34:  output  $m$ 
35: end procedure

```

Algorithm 4: Payment Extinguishing Algorithm

Algorithm 4 Payment Distinguishing Experiment

```

1: procedure PAY_DIST_EXP $\Pi, \mathcal{B}$  $_A^{1^s}$ ( $1^s$ )
2:    $(s_k, p_k) \leftarrow \text{Gen}_\Sigma(1^s)$ 
3:    $(m, S) \leftarrow A_1^{\text{Read, Submit}}(p_k)$   $\triangleright S$  is internal state information of the adversary that can be passed
   to the second stage
4:    $(\lambda, k) \leftarrow \text{Gen}_\Pi(1^s)$ 
5:    $b \leftarrow U(\{0, 1\})$ 
6:   if  $b = 1$  then  $\triangleright$  Actual message is sent to the blockchain
7:      $\text{Embed}((\lambda, k), m, \mathcal{B})$ 
8:   else  $\triangleright$  Random payments are sent to the blockchain
9:      $n_\lambda \leftarrow |\lambda|$ 
10:     $N \leftarrow |\text{Enc}(k, m)| + n_\lambda$   $\triangleright$  Enc is the encryption scheme used by  $\Pi$ 
11:    Generate  $N$  random addresses  $a_i$  for  $i \in \{1, 2, \dots, N\}$ 
12:    Simulate  $\text{Embed}$  to generate payments to  $a_i$ 
13:    Submit payments to blockchain one-by-one as  $\text{Embed}$  does
14:  end if
15:   $b' \leftarrow A_2^{\text{Read, Submit}}(p_k, S)$ 
16:  if  $b = b'$  then
17:    return 1  $\triangleright$  A guessed correctly
18:  else
19:    return 0  $\triangleright$  A did not guess correctly
20:  end if
21: end procedure

```

Algorithm 5 First Stage of Adversary A'

Algorithm 5 First Stage of the Adversary A'

```

1: procedure  $A'_1{}^{\text{Enc}_i}(1^s)$ 
2:   Initialize a blockchain  $\mathcal{B}$ 
3:    $(s_k, p_k) \leftarrow \text{Gen}_\Sigma(1^s)$ 
4:    $(m, S) \leftarrow A_1^{\text{Read, Submit}}(p_k)$   $\triangleright$  Answers the queries according to the specification of  $\mathcal{B}$ 
5:    $S' \leftarrow$  state and internal information of  $\mathcal{B}$ 
6:   output  $(m, (S, S', p_k, s_k))$ 
7: end procedure

```

Algorithm 6 Second Stage of Adversary A'

Algorithm 6 Second Stage of the Adversary A'

```
1: procedure  $A_2^{Enc_t}(c, (S, S', p_k, s_k))$ 
2:   Initialize a blockchain  $\mathcal{B}$  according to the state  $S'$ 
3:    $(\lambda, k) \leftarrow \text{Gen}_{\text{BLOCCE}}(I^s)$ 
4:   Embed  $\lambda || c$  into  $\mathcal{B}$  by simulating Embed
5:    $b' \leftarrow A_2^{\text{Read, Submit}}(p_k, S)$ 
6:   output  $b'$ 
7: end procedure
```

3. RESULTS AND ANALYSIS

As a representation of distributed databases, all user transaction information is recorded on the block chain, which has strict security standards. Block chain is a peer-to-peer network that is decentralized. There is no need for nodes to trust one another, and there is no central node. As a result, transactions on the block chain must maintain the confidentiality of transaction information while preserving transaction integrity over insecure connections. As can be seen, each message while entered allocates and display's a block number, Data and Hash value for the block.

```
Block Hash: b843dd2f3cfa492d4fb4260a4c28968d7615492f9906b0ad6e2d7bbc5c69d7dc
BlockNo: 1
Block Data: Block 1
Hashes: 1013952
-----
Block Hash: 6156c9f7df237e11aab86dcd1e605fb69150c724a7e6da452939dd964d0d2973
BlockNo: 0
Block Data: Genesis
Hashes: 0
-----
Block Hash: b843dd2f3cfa492d4fb4260a4c28968d7615492f9906b0ad6e2d7bbc5c69d7dc
BlockNo: 1
Block Data: Block 1
Hashes: 1013952
```

The two types of cryptographic algorithms utilized in block chains are asymmetric-key algorithms and hash functions. Hash functions are used to provide each participant with a unified picture of the block chain. In block chains, the SHA-256 hashing algorithm is often employed as

the hash function. As seen below, while we enter a message, it asks for Encryption or Decryption and converts respectively.

```
Enter your message: attack china
Enter you key [1 - 26]: 19
Encrypt or Decrypt? [E/D]: E
tmmtvdvabgt
```

The SHA-256 hashing method is frequently used as the hash function in block chains. While entering a message, it asks for Encryption or Decryption and transforms accordingly, as shown below.

```
Enter your message: tmmtvdvabgt
Enter you key [1 - 26]: 19
Encrypt or Decrypt? [E/D]: D
attackchina
```

4. Conclusion and Future Scope

Block chain technology has emerged as a critical focus area for all multinational organizations' development in recent years, with a great number of startups emerging in this industry. This study covers current block chain challenges and presents cryptography's major uses. To begin, the block chain technology, starting with the block chain infrastructure, is simplified. Second, to better understand the block chain, cryptography technology is offered. Finally, the block chain's current security weaknesses are evaluated. It demonstrates that digital encryption is employed throughout the block chain system and is a necessary component. For maximum security, the communication system's message will be transmitted via encryption and the block chain protocol. Future plans include developing and deploying social media tools, particularly for communication. Encryption techniques are now used in every system. As a result, in the future, block chain and encryption will combine for more privacy.

References

[1] Nakamoto, S.(2008) Bitcoin:Apeer-to- peer electronic cash system. Consulted. 165:55-61.

- [2] Zhu, Y., Gan, G.H., Deng, D. (2016) Security Research in Key Technologies of Blockchain. Information Security Research., 12:1090-1097.
- [3] Liu, X.F. (2017) Research on block chain performance improvement of Byzantine fault-tolerant consensus algorithm based on dynamic authorization. Zhejiang University.
- [4] Wang, X., Lai, X., Feng, D. (2005) Cryptanalysis of the Hash Functions MD and RIPEMD. Advances in Eurocrypt., 3494:1-18.
- [5] Shen, Y., Wang, G. (2017) Improved preimage attack on RIPEMD-160 and SHA-160. KSII Transactions on Internet & Information Systems., 12:727-746.
- [6] Wang, H.Q., Wu, T. (2017) Cryptography in Blockchain. Journal of Nanjing University of Posts

UNDER PEER REVIEW