Secure Peer-to-Peer Communication Based on Blockchain

Kahina Khacef(✉) and Guy Pujolle(✉)

Sorbonne University, 4 Place Jussieu, 75005 Paris, France
{kahina.khacef,guy.pujolle}@lip6.fr

Abstract. Nowadays, electronic messaging is the most used network application, and the authentication between users is a vital property. The most commonly employed approaches to ensure this property are PKI and S/MIME email encryption protocols, but indeed they are facing multiples security threats, such as the MITM attack and EFAIL attack. The blockchain is an innovative technology that overcomes these threats and allows to decentralize sensitive operations while preserving a high level of security. It eliminates the need for trusted intermediaries. The blockchain is accessible to all network nodes and keeps track of all transactions already made. The goal of our work is to propose a secure messaging solution based on the blockchain technology. In this paper, we explain why blockchain would make communications more secure, and we propose a model design for blockchain-based messaging maintaining the performance and security of data recorded on the blockchain, using a smart contract to verify the identities and their associated public keys, and validate the user's certificate. The system is entirely decentralized and allows users to exchange messages securely.

1 Introduction

Public Key Infrastructure (PKI) is a significant component to resolve authentication in networks and provides guarantees to trust a certificate signed by a certification authority (CA). A new concept named web of trust for his Pretty Good Privacy (PGP) encryption program uses PKI to provide confidentiality with encryption, authentication via the signature and web of trust via identity validation from peers. The certificates authenticate the public keys and allow you to perform cryptographic operations, such as encryption and digital signing. As authentication and identity validation is centralized in the PKI, that creates single points of failure.

The blockchain technology was designed to make the transaction more reliable. Blockchain technology is a potential solution to achieve data integrity, relying on cryptography to provide tamper-resistance. Blockchain can be utilized in such cases to achieve the secure communication and integrity of data. The blockchain is decentralized, and no centralized authority can approve transactions. All network participants must reach a consensus to validate transactions in a secure way, and previous records cannot be changed. A very high cost must be spent if someone

L. Barolli et al. (Eds.): WAINA 2019, AISC 927, pp. 662–672, 2019.
https://doi.org/10.1007/978-3-030-15035-8_64

wants to alter previous records. External attackers would have to gain access to every computer in the network that hosts the blockchain database at the same time to manipulate it, which is as practically impossible [1].

The cryptocurrency Bitcoin was created by an unknown person using the alias Satoshi Nakamoto in 2008. Bitcoin generates and broadcast transactions to the Blockchain. Once validated, the transactions are propagated through the network and added into a block. Once a block is full, the block is appended to the Blockchain by performing a mining process. Miners try to solve a hard cryptographic puzzle named Proof of Work (PoW), and the node that solves the puzzle first adds the new block to the Blockchain. Bitcoin is based on decentralized trust. The trust is achieved as an emergent property from the interactions of different participants in the bitcoin system [2]. The data stored on the blockchain cannot be hacked, modified or deleted. The immutability of data in this blockchain is strongest when the chain is long [3].

1.1 Motivations

Blockchain has been of great interest to engineers and investors because of its immense commercial potential and its use in applications as diverse as a cryptocurrency.

Several layering schemes of new features on bitcoin are proposed. A proposal to create a decentralized domain name service over Bitcoin appeared as BitDNS, eventually evolved into Namecoin, the first altcoin [4]. Bitcoin has the most substantial amount of computational power securing the blockchain data. However, it's hard to introduce new functionality to bitcoin because that requires consensus breaking changes. Transactions in bitcoin did not include a standard method for carrying a data payload, and the blockchain grows at an exponential rate [5] that extends pressures upward on storage space and importantly network bandwidth. Due to the security concerns, only a handful of functions are permitted in standard transactions [6].

Peer to peer (P2P) systems help overcome many problems that go beyond traditional client-server approaches, but these features also introduce new issues such as how to establish the trust relationship within P2P networks [7].

The Ethereum blockchain is a platform for decentralized applications called smart contracts. Because Ethereum addresses are unique identifiers whose ownership does not change, their activity can be tracked and analyzed [8]. A smart contract is an executable code that runs on top of the blockchain, allows running an agreement between two parties automatically, without one of the parties being able to obstruct its execution [9].

At the time of the digital economy, data is brought to transit more and more between companies, from a client to a supplier, moving from one cloud to another with the advent of virtualization and containers. Our work describes the potential for applying blockchain to assure the traceability of data, certificate individual and secure messaging based on the blockchain technology.

1.2 Contribution and Organisation of Paper

In this contribution, we are interested in the security of the blockchain for messaging, which can be an email, a website or some other form of message. We propose a protocol that performs encrypted messaging on the public blockchain. The goal of our contribution is to achieve the three functionalities achieved by PKI using the blockchain as the database to store public keys, digital signature, and peer information, allowing each entity of the network to validate information about every other node in the network.

The main contribution of this paper is: Sect. 2 a presentation of the benefits of blockchain in cybersecurity, Sect. 3 briefly description of the current state of the art for Blockchain based PKI, and in Sect. 4 we present our proposed solution with a smart contract to use the legitimate secure certificate.

2 Blockchain

A blockchain is essentially a fault-tolerant distributed database of records of a public ledger of all transactions that have been executed and shared among participating parties [10]. Each transaction in the public ledger is verified by a consensus of a majority of participants in the system. Blockchain provides a tool for increasing data integrity using a combination of cryptography and consensus [11]. As such, there is no central server point of control. The information is stored in blocks, Each block contains transactions, a timestamp and a link to the previous block, and is cryptographically protected. What makes it secure is:

- **Public key infrastructure:** It uses public/private key encryption and data hashing to store and exchange data safely.
- **Distributed ledgers:** There is no central authority to hold and store the data, it removes the single point of failure.
- **Peer to peer Network:** The communication is based on the P2P network architecture and inherits the decentralized characteristics.
- **Cryptography:** Blockchain uses a variety of cryptographic techniques, hash functions, Merkle trees, public and private key [12]. It is difficult to alter the blockchain, to make a modification, it is necessary to succeed in a simultaneous attack on more than 51% of the participants.
- **Consensus Algorithm:** The rules which the nodes in the network follow to verify the distributed ledger. Consensus algorithms are designed to achieve reliability in a system involving multiple unreliable nodes. A consensus of the nodes validates the transactions. Choice of consensus algorithm has significant implications for performance, scalability, latency and other Blockchain parameters. The consensus algorithms must be fault-tolerant [13].
- **Transparency:** Each transaction is visible to anyone with access to the system. If an entry can not be verified, it is automatically rejected. The data is therefore wholly transparent. Each node of a blockchain has a unique address that identifies it. A user may choose to provide proof of identity to others.

2.1 How Public Blockchain Solves PKI Problems?

In blockchain, the shared data is the history of every transaction ever made. The ledger is stored in multiple copies on a network of computers, called "nodes." Each time someone submits a transaction to the ledger, the nodes check to make sure the transaction is valid. The transaction is being created in the chained data structure of blockchain, a new timestamp will be recorded at the same time, and any modification of data created before will not be allowed anymore.

A block is a data structure which consists of a header and a list of transactions. Each transaction is generally formed as trx := (M, Signature) where M := (PKSender, receiver, data); PKSender denotes the public key of a sender (address of the sender is derived from the PKSender), and receiver is the address of a receiver, Signature := Sign SKSender (H(M)) where SKSender is the private key of the sender and H() is a secure hash function. A subset of them competes to package valid transactions into "blocks" and add them to a chain. The owners of these nodes are called miners and the miners who successfully add new blocks to the chain earn a reward.

A cryptographic hash unique to each block and a consensus protocol makes it tamper proof [13]. There are different types of the blockchain, and there are several consensus mechanisms used in the blockchain, we present the two most used consensus proof of work (PoW) and proof of stake (PoS).

2.1.1 Types of Blockchains

1. **Public blockchain:** A blockchain that anyone in the world can read, can send transactions to and expect to see them included if they are valid. This means anyone can become part of the network and participate in the consensus process making them permissionless. There is no way to censor transactions on the network nor change transactions retrospectively [14]. The content of the blockchain can be trusted to be correct. Public blockchains are, however, very inefficient. The more computing power is required to support trust. So, an attacker would need to acquire 51% of the network's computing power to change an entry in the blockchain. (e.g., Bitcoin, Ethereum, ZCash).
2. **Consortium blockchain:** It is a blockchain where a pre-selected set of nodes control the consensus process.
3. **Private blockchain:** A blockchain where access permissions are more lightly controlled, where rights to modify or even read the blockchain state restricted to a few users, where only known nodes are allowed to participate in the network. Ideally, it is internal for an organization. The writes permissions are kept centralized to one organization. Private blockchain reduces counterparty risk by enabling the exchange of data without the intermediation of third parties.
4. **Permissioned Blockchain:** It is a blockchain where we can allow specific actions to be performed only by specific addresses. The participants in the network can restrict who can participate in the consensus mechanism and who can create a smart contract and give the authority for some participants to provide the validation of blocks of transactions. A control access layer into

the blockchain nodes is used. However, raise their questions, Who has the authority to grant permission? A permission blockchain may make its owners feel more secure, giving the database rigorous security and privacy capabilities but can see as violating the idea of blockchain because only some participants have more control, which means they can make changes whether or not other network participants agree.

2.1.2 Proof of Work

PoW is the consensus algorithm used in bitcoin. The protocol use of elliptic curve cryptography for transactions signatures, and the elliptic curve digital signature algorithm (EDCSA) [15] is used. The user needs to provide her public key and her signature to prove ownership. A hash takes a lot of computing time and energy to generate initially. It thus serves as a proof that the miner who added the bloc to the blockchain did the computational work to earn. The hashes also serve as the links in the blockchain, each block includes the previous block's unique hash.

If an attacker wants to tamper with the blockchain, he needs to control more than 50% of the hashing power to become the first one to generate the new bloc.

2.1.3 Proof of Stake

There are no particularly heavy calculations in a PoS. PoS does not require powerful and expensive hardware for maintenance of the blockchain and do not consume a lot of electricity. PoS require significantly smaller resources than PoW. The digital currency has the concept of coin age. Coin age of a coin is its value multiplied by the period after it was created [16]. The probability that an account will succeed in confirming the next block of transactions to add to the blockchain is proportional to the amount of money that is on that account.

Hash(hash(Bprev), A, t) <= Sold(A) M/D, with Sold(A), is the balance of address A, and t is the timestamp. Unlike the PoW, the only variable that the user can change is the timestamp.

3 Related Work

In this section, we present the works made use Blockchain for secure transactions between users. Since BitDNS [4], numerous approaches are proposed.

Namecoin it is the first system to build a decentralized naming system using blockchain, the first altcoin from bitcoin with its blockchain [14]. Satoshi believed that BitDNS should use its independent blockchain and event offered the first proposal for merged mining as a way to secure blockchain. Namecoin was also the first solution to zooko's triangle producing a naming system that is simultaneously secure, decentralized, and human-meaningful. Namecoin suffers from 51% attacks [17], because of its insufficient computing power.

Blockstack uses the existing internet transport layer (TCP or UDP) and underlying communication protocols and focuses on removing points of centralization that exist at the application layer. The underlying blockchain of Bitcoin limits the performance of Blockstack [18].

Certcoin [19] removes central authorities and uses the blockchain Namecoin as a distributed ledger of domains and their associated public keys. Every certcoin user stores the entires blockchain, and this causes two problems, the latency for the controller and the security problems of merged mining used by Namecoin.

Emercoin [20]: Blockchain-based PKI that doesn't remove central authorities but uses Blockchains to store hashes of issued and revoked certificates. Emercoin has the side benefit of optimizing network access by performing key and signature verification on local copies of the blockchain.

However, all of these systems faced the same barrier. This consensus involves significant energy, delay, and computation overhead because of high resource demand for solving the PoW.

4 Contribution

4.1 The Proposed Model

The primary goal of our approach is to secure communications between entities of the network. The proposed model is to use the blockchain to validate the user's identity and to ensure trust between users for exchanging messages with a high level of security. Each user must communicate only with the user's identity validated by the smart contract, and consider every other interaction as malicious. Each user how want uses the system to communicate with others must register their identity and their public key an store it on blockchain. To achieve such a system, we can use Ethereum public blockchain that implements smart contracts. The Ethereum network has up to 200,000 developers currently working on the protocol and derived projects, ensuring that the protocol remains

Table 1. Notations

Notation	Definition
Alice, Bob	Entities
K_{pub}^{userA}	Public key of Alice
K_{pub}^{userB}	Public key of Bob
K_{pri}^{userA}	Private key of Alice
K_{pri}^{userB}	Private key of Bob
ID^{A}	Id of Alice
ID^{B}	Id of Bob
S_{pri}^{userA}	Signature with private key of Alice
E_{pub}^{userA}	Encryption with public key of Alice
t	Validity of the Public Key
T	Time stamps

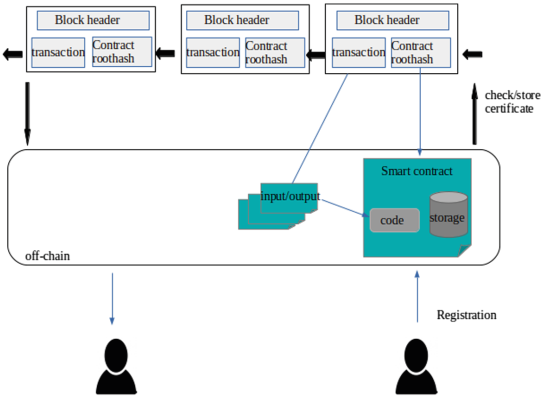

Fig. 1. System architecture

cutting edge, it has a level of security. The most frequently used notations in our scheme are mentioned in Table 1.

Once the code conditions are validated in smart contract, it can run as scheduled. This execution is carried out in practice by sending a specific transaction to the blockchain that causes the execution of the smart contract (Fig. 1).

4.1.1 Smart Contract

The smart contract is the code that is stored and executed on a blockchain. Once a smart contract is deployed, users cannot modify it since the contract was sented and validated. A user can send a transaction to the contact's address, and the transaction will be executed.

Each interaction with the smart contract is recorded as a "transaction" on the blockchain. These transactions are grouped in a Merkle tree and stored in a block on the blockchain.

4.1.2 Blockchain Registration

1. The user Alice generate a pair key using ECDSA algorithm, a public key K_{pub}^{userA} and private key K_{pri}^{userA}, then derives the identity ID^A, as the hash of the public key as shown in Fig. 2. Alice keeps the private key safely and requests to register its identity with the corresponding public key into the blockchain after to be verified and validate by the network.
   Each public key have an associated timestamp.
2. Alice signs transactions with the corresponding private key and transfers $(ID^A, S_{pri}^{userA}(K_{pub}^{userA}, t, T))$ to the blockchain.
3. The miner checks:
   a. $K_{pub}^{userA}(ID^A, S_{pri}^{userA}(K_{pub}^{userA}, t, T))$.
   b. That the ID^A has never previously been registered.

4. If verified, The certificate's Alice $(ID^A, S_{pri}^{userA}(K_{pub}^{userA}), t, T)$, is then stored on the blockchain, with the following pieces of information: (Id Alice, Public key of Alice, the validity of the public key and timestamp).
5. This is the registration process for all network entities.

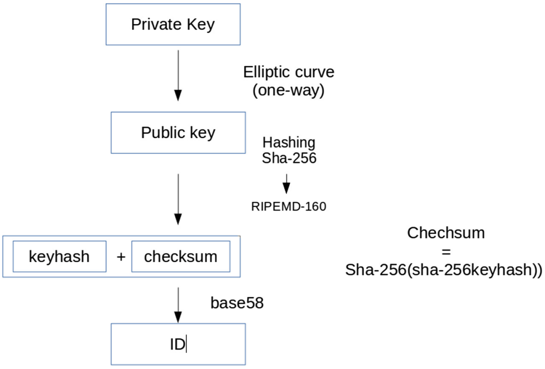

Fig. 2. ID generation

4.1.3 Smart Contract-Based Verification

When user Bob wants to send a message to Alice, Bob only presents ID^B previously recorded on the blockchain and ID^A of Alice with a time-stamp T to the blockchain. Each message must include the time.

1. Bob sents a transaction: $T_B = [ID^B$, ID^A, T, $S_{pri}^{userB}(ID^B, ID^A, T)]$.
   The Smart contract receives the request from the user Bob:
2. The smart contract checks if the presented ID^B and ID^A exist on the blockchain. The Smart contract reads and parses the two Alice and Bob recordings.
   a. It performs a Lookup of (K_{pub}, ID), If existed the output return true.
   b. Once the validity of the public keys verified, the Smart contract checks the validity of the timestamp and the signature of the transaction.
   c. Finally, The smart contract validates the request and return true.
3. Bob sends a transaction $T_B^1 = [ID^B$, ID^A, T, $E_{pub}^{userA}(ID^B, ID^A, T, K_{pub}^{userB})]$ to Alice's address.
4. Next, Alice checks the transaction with his private key and then sends to Bob's address $T_A^2 = [ID^A$, ID^B, T+1, $E_{pub}^{userB}(ID^A, ID^B, T+1, K_{pub}^{userA})]$.
5. After receipt of the transaction by Bob, the same verification will be performed, and mutual authentication is established between the two entities.

4.1.4 Send/Receive Message

Once mutual authentication is established, Alice uses Bob's public key and Alice's private key to generate a shared secret. The shared secret can be generating using the Elliptic-curve-Diffie-Hellman (ECDH) to encrypt Message. ECDH is a variant of the Diffie-Hellman algorithm for elliptic curves. Once the shared key is generated, it is used to encrypt messages using a symmetric key algorithm [21].

4.2 Evaluation

4.2.1 Advantages

Our approach removes central authorities (CA) and uses the blockchain public as a distributed ledger of identity and their associated public keys. We use Blockchain to store public keys, digital signature, and peer information.

Once published, the smart contract code works precisely as programmed. This is one of the main advantages of the platform, the code always interacts as promised it cannot be falsified, and it never has a downtime. The system is trust, transparent and traceable.

1. **Confidentiality:** Once the communication channel between users is secured, peer to peer encryption between endpoints can be set and only authorized users to have access to the messages exchanged.
2. **Message integrity and Authentication:** The blockchain checks the validity of the signature, before being stored. Another person can not change/modify the signed agreement or alter exchanged message during the network transit. Each user has a certificate stored on the blockchain. The smart contract checks the certificate and proves the identity of users. All exchanged messages are signed with private keys associated with the public key on certificates using the ECDSA algorithm.
3. **Reliability:** It is impossible to shut down all computers participating in the blockchain simultaneously. As a result, this database is always online, and its operation never stops.

4.2.2 Limitations

In blockchain, the smart contract is executed sequentially that affect the performance of blockchain negatively. With the growing number of smart contract, the blockchain will not be able to scale.

In practice, it is impossible to modify an existing contract in the blockchain after it has been registered in it. The design phase of these contracts will, therefore, require special attention to avoid any future disappointment.

4.3 Implementation

The smart contract plays a vital role in executing and performing the agreement among various users in the system. For implementing the proposed system, a smart contract can be created by developing the codes that run on the blockchain to execute the agreement signed by the nodes to validate the certificates of individuals. Smart contract promise low transaction fees compared to traditional systems that require a trusted third party to execute an agreement. The contract's state is stored on the blockchain, and it is updated each time the contract is invoked.

Before sending/receiving the message, the smart contract first checks the identity and its associated public key, previously stored in the blockchain. Then, it checks the validity of the timestamp. Finally, the smart contract verifies the signature.

A smart contract can be developed with Solidity.

5 Conclusion

Our contribution benefits from an entirely decentralized architecture offering inherent fault tolerance, redundancy, and transparency. We have firstly proposed an approach that secure communication, it benefits of security properties of blockchain public. It shows how to use to the immutability of the blockchain to provide a solution to high problematics in the field of centralized PKI.

We plan to implement our proposal to base our results empirically and demonstrate its viability.

Our next proposal is to set up an architecture with smart contracts to validate, store and revoke the certificate on a public blockchain. The certificate of the individual will contain, his address and public key, the address of smart contract that issued it, and stored it in Off-chain.

References

1. Schuermann, F.: Bitcoin and beyond: a technical survey on decentralized digital currencies. IEEE Commun. Surv. Tutor. **18**, 2084–2092 (2016)
2. Bano, S., Sonnino, A., Bassam, M., Azouvi, S., McCorry, P., Meiklejohn, S., Danezis, G.: SoK: consensus in the age of blockchains, arxiv (2016)
3. Cachin, C., Vukoli, M.: Blockchain Consensus Protocols in the Wild (2016)
4. Loibi, A.: Namecoin, Seminar innovative Internettechnologien und mobilkommunikation SS2014. IEEE (2014)
5. Hung, T., Svetinovic, D.: Data analysis of digital currency network: namecoin case study. In: International Conference on Engineering of Complex Computer Systems (2016)
6. Deepak, K., Shetty, S., Liang, X., Kamhoua, C., Njilla, L.: Consensus Protocols for Blockchain-based Data Provenance: Challenges and Opportunities. IEEE (2017)
7. Mudliar, K., Parekh, H., Bhavathankar, P.: A comprehensive integration of national identity with blockchain technology. In: 2018 International Conference on Communication, Information and Computing Technology (ICCICT), 2–3 February, Mumbai, India (2018)
8. Aung, Y.N., Tantidham, T.: Review of Ethereum: smart home case study. In: 2017 2nd International Conference on Information Technology (INCIT) (2017)
9. Ward, M.: Untangling Blockchain: A Data Processing View of Blockchain Systems, Elsevier Information Security Technical report, pp. 89–92 (2006)

10. Tewari, H., Nuallain, E.: Netcoin: A Traceable P2P Electronic Cash System. IEEE (2016)
11. Guy, Z., Oz, N., Alex, P.: Decentralized Privacy: Using Blockchain to Protect Personal Data (2018)
12. Conti, M., Kumar, S., Lal, C., Ruj, S.: A Survey on Security and Privacy Issues of Bitcoin. IEEE (2017)
13. Tosh, D.K., Shetty, S., Liang, X., Kamhoua, C., Njilla, L.: Consensus Protocols for Blockchain based Data Provenance: Challenges and Opportunities. IEEE (2017)
14. Kalodner, H., Carlsten, M., Ellenbogen, P., Bonneau, J., Narayanan, A.: An empirical study of Namecoin and lessons for decentralized namespace design. Blockchain (2014)
15. Sankar, L.S., Sindhu, M., Sethumadhavan, M.: Survey of consensus protocols on blockchain applications. In: International Conference on Advanced Computing and Communication Systems (ICACCS 2017)
16. Mingxia, D., Xiaofeng, M., Zhe, Z., Xiangwei, W., Qijin, C.: A Review on Consensus Algorithm of Blockchain. IEEE (2017)
17. Dai, F., Shi, Y., Meng, N., Wei, L., Ye, Z.: From bitcoin to cybersecurity: a comparative study of blockchain application and security issues/ICSAI, pp. 975–979 (2017)
18. Ali, M., Nelson, J., Shea, R., Freedman, M.J.: Blockstack: a global naming and storage system secured by blockchain. In: USENIX Annual Technical Conference (2016)
19. Fronknecht, C., Velicannu, G., Sophia, Y.: CertCoin: A Namecoin Based Decentralized Authentication System (2014)
20. Khovayko, O., Shumilov, E.: EMCSSL Decentralized identity management, passwordless logins, and client SSL certificates using Emercoin NVS (2016). http://emercoin.com
21. Andrea, C.: ECDH and ECDSA (2015). https://andrea.corbellini.name/2015/05/30/elliptic-curve-cryptography-ecdh-and-ecdsa