Towards Exchanging Connection Information by Using the Blockchain Technology for Decentralized Social Networking Services

Yi ZHOU [1] Yasushi SHINJO [1] Kazuki TAKARADA [1]
Zhiyuan LIN [1]

1. Introduction

With the development of society and technology, we are enjoying the convenience brought by the Internet but it also highlights some drawbacks. For example, many popular SNSs (Social Networking Services) such as Facebook and Twitter and many cooperative services such as Dropbox are centralized systems which require users to exchange their private information through centralized servers. These centralized architectures are always controlled by specific organizations that have unlimited power over users' information and they can manage the systems as they wish. In addition, the breakdown of central servers may lead to series of problems such as information loss or data breach.

To solve these problems, we are building a decentralized SNS. In this SNS, users can communicate with the following channels.

- Web Realtime Communication (WebRTC) for web browsers.
- Virtual Private Networks (VPNs) for desktop PCs [1].
- Session Initiation Protocol (SIP) for mobile devices tools [1].

In these social channels, before two nodes are able to communicate, they must do the following things.

- Exchange connection information. A node must know the IP address or other location information of the other node. In addition, WebRTC needs complex exchanging of messages based on session description protocol (SDP).
- Perform user authentication each other.

We are implementing these things by using the blockchain technology. Before two nodes communicates, they exchange connection information with smart contracts of the blockchain technology. At this time, they perform user authentication using blockchain wallets.

2. Target communication channels

2.1 WebRTC

WebRTC provides peer-to-peer (P2P) communication channels among web browsers. The design of WebRTC focuses on communication between browsers via network address translation (NAT) routers. It appears to be a promising mechanism for addressing the problems of centralized servers. However, establishing WebRTC channels usually requires web servers [2]. Before two peers communicate with each other through WebRTC, they must exchange short messages. This process is called signaling, which usually requires a centralized server.

The signaling process of WebRTC includes the following four protocols: Session Description Protocol (SDP), Interactive Connectivity Establishment (ICE), Session Traversal Utilities for NAT (STUN) and Traversal Using Relays around NAT (TURN). STUN and TURN are options for enhancing the reachability for peers behind NAT routers.

To achieve our goals, we must implement signaling without centralized servers. In our decentralized SNS, we implement a smart contract that realizes signaling of WebRTC.

2.2 VPNs and SIP

VPNs enables users to send and receive data across public networks as they are directly connected with a private networks. In our decentralized SNS, establishing a VPN channel requires obtaining the IP addresses of peers [1].

Session Initiation Protocol (SIP) is a signaling protocol used for initiating, maintaining and terminating real-time sessions that include voice, video and messaging applications. In our decentralized SNS, users run SIP servers at home and they connect with friends' servers. Before two SIP servers establish a SIP connection, they need the IP addresses of peers [1].

In our decentralized SNS, we implement a smart contract that realizes the exchange of IP addresses. Exchanging IP addresses is much easier than exchanging WebRTC signaling messages. We implement the former by simplifying the latter. In the following sections, we focus on the former.

3. WebRTC signaling with a smart contract

In our decentralized SNS, web browsers are connected with WebRTC in a P2P manner. We implement an Ethereum smart contract that performs WebRTC signaling. When two peers are not behind symmetric NAT routers and one of the peers has a public IP address, the signaling can be done by exchanging two messages of SDP via the smart contract. An SDP message includes basic parameters to establish connections using ICE, namely IP addresses, port numbers and session public keys of peers which will be used for end-to-end encryption of the WebRTC connections.

In the Ethereum blockchain, when a node sends a transaction message to a smart contract, the smart contract knows the wallet address of the sender. A smart contract can emit events, which are much cheaper than contract storage. A node can watch events that include its wallet address.

Figure 1 shows a signaling process using the smart contract. In our decentralized SNS, two friends exchange their wallet addresses of Ethereum in advance. This smart contract has two functions: offer() and answer(). When two Peers A and B establish a WebRTC connection, Peer A creates an SDP offer message by calling the WebRTC API RTCPeerConnection.createOffer() of the web

1 University of Tsukuba

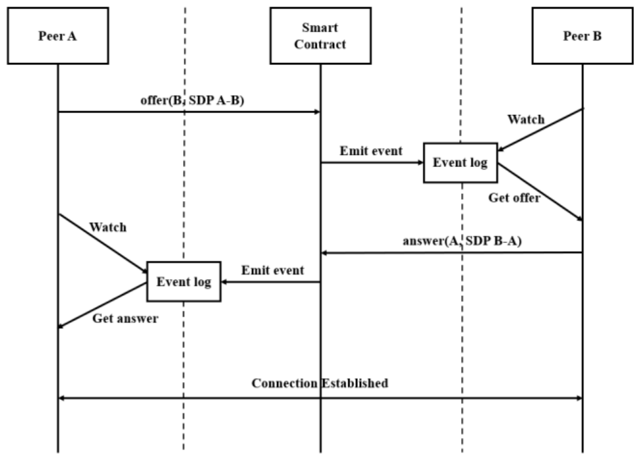

Figure 1: WebRTC signaling with a smart contract.

browser. Next, Peer A calls the function offer() of the smart contract with two arguments: the wallet address of Peer B and the SDP offer message. The function offer() emits an event, and a node that is watching the event log receives the event. When Peer B receives a new Offer event, Peer B extracts the address of Peer A and the SDP offer message in the event. After that Peer B creates an answer message of SDP by calling the WebRTC API RTCPeerConnection.createAnswer() of the web browser, using the received SDP offer message. Then Peer B sends this SDP answer message to Peer A in the reverse direction using the same smart contract. Finally, Peer A and Peer B establish a WebRTC connection.

We have implemented this signaling mechanism, and performed several experiments in the Ethereum test network Ropsten by using Ethereum client geth version 1.9.23. The execution times of signaling with the smart contract were 20 to 60 seconds. This time is affected by the block interval of the Ethereum network.

This signaling with the smart contract is slow and using it directly is impractical for applications including text chat and video calling. We solve this problem by exchanging signaling messages through WebRTC data channels. When a user boots a web browser, it establishes the WebRTC data channels with those of the user's friends by using the slow smart contract. This can take several minutes. When the user executes an application, video calling for example, the web browser exchanges the signaling messages through an existing WebRTC data channel to a friend.

4. Privacy enhanced signaling with smart contract

A blockchain is a shared database in essence. The data or information stored in it has the characteristics of unforgeable, traceable, open and transparent. Every transaction we conduct on the blockchain can be traced. The information of a wallet address can also be traced. If we continue to use a fixed wallet address, the information we exchange is easily traced, and this causes a privacy issue.

We solve this problem with warpwallets [3]. A warpwallet is a deterministic address generator in a blockchain. A warpwallet can generate multiple deterministic wallet addresses through a passphrase and a salt string entered by a user.

In our decentralized SNS, we are implementing WebRTC signaling using warpwallets. The wallet address of each user changes according to the passphrase and salt sequence. For each user, we give a random user ID and use it as a salt string. We include a date string to a random passphrase. For example, a user can use "Picard Delta 5 2020-12-01" as a passphrase. When two users establish a social relationship in our decentralized SNS, they exchange the random user IDs and prefixes (without the date string) of passphrases. When a user performs WebRTC signaling, the user uses the address of the warpwallet with the date string of the day instead of the fixed wallet address.

5. Related work

Twister [4] is a distributed microblogging social network modeled after Twitter. It utilizes a blockchain for user registration and a Distributed Hash Table (DHT) for routing and indexing of peers and contents. SAND [5] utilizes Social VPNs to establish peer-to-peer connections between trusted friends and distribute social media contents reliably through the VPN connections. In research, we establish peer-to-peer channels of WebRTC, VPN and SIP by the blockchain technology.

6. Conclusion

This article proposes a method of exchanging connection information using a smart contract in a blockchain network for building a decentralized SNS. We have implemented the smart contract that exchanges SDP messages and IP addresses with fixed normal addresses. While the signaling using this smart contract is slow, it is done once at the boot time. We are solving the privacy problem of fixed addresses by using deterministically changing wallet addresses of warpwallets.

Reference

[1] Takarada, K., Shinjo, Y., Zhou, Y. and Lin, Z.: "Towards the implementation of social communication channels powered by distributed ledger", IPSJ Computer Symposium Post Session (2020).

[2] Sredojev, B., Samardzija, D. and Posarac, D.: "WebRTC technology overview and signaling solution design and implementation," 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, , pp. 1006-1009, doi: 10.1109/MIPRO.2015.7160422 (2015).

[3] Krohn, M. and Coyne, C.: warpwallet, GitHub repository, https://github.com/kebase/warpwallet (2017).

[4] Freitas, M.: "Twister: the development of a peer-to-peer microblogging platform", International Journal of Parallel, Emergent and Distributed Systems, Vol.31, No.1, pp.20–33 (2016).

[5] Ding, D., Conti, M. and Figueiredo, R.: "Sand: Social-aware, network-failure resilient, and decentralized microblogging system", Future Generation Computer Systems, Vol.93, pp.637–650 (2019).