BeepTrace for COVID-19 Pandemic: A Demo

Hong Kang[*]
James Watt School of Engineering
University of Glasgow
Glasgow, United Kingdom
2357544k@student.gla.ac.uk

Zaixin Zhang[*]
James Watt School of Engineering
University of Glasgow
Glasgow, United Kingdom
2357620z@student.gla.ac.uk

Junyi Dong
James Watt School of Engineering
University of Glasgow
Glasgow, United Kingdom
2357626d@student.gla.ac.uk

Yinghao Ji
James Watt School of Engineering
University of Glasgow
Glasgow, United Kingdom
damienji@icloud.com

Hao Xu
James Watt School of Engineering
University of Glasgow
Glasgow, United Kingdom
h.xu.2@research.gla.ac.uk

Lei Zhang
James Watt School of Engineering
University of Glasgow
Glasgow, United Kingdom
Lei.Zhang@glasgow.ac.uk

***Abstract*—Given the heavy economic losses and threats to human health caused by COVID-19 worldwide, as well as the privacy concerns from existing contact tracing technologies, we developed a novel contact tracing mobile application BeepTrace based on blockchain, targeted at mitigating the pandemic and easing the privacy concerns of contact tracing. The app features dual-mode, where the passive mode uses GPS, and the active mode with Bluetooth Low Energy (BLE) technology to solve contact information respectively. Thanks to blockchain technology, it solves third-party trust concerns and opens up anonymous data sharing for COVID-19 protection and beyond by adopting BeepTrace approach as an open platform of health/disease tracing data. At present, we have released the alpha version of the app with the preliminary realization of relevant functions. We believe this application will have significant implications for the global response to the pandemic of and beyond COVID-19.**

I. Introduction

COVID-19, caused by SARS-CoV-2 , has become a global pandemic with thousands of death, tremendous social and economic upheavals. As the number of infected cases raise and fall in many waves, governments have been implementing measures to reduce the impact of disease, by maintaining social distancing, wearing masks in crowded spaces, and tracing contacts between people, in order to contain the disease [1].

Traditional contact tracing is laborious and time-consuming as it relies on the memories of patients to work [2], thus Digital Contact Tracing (DCT) solutions are being developed across the world [3], [4]. Other similar protocols and solutions are emerging but they are challenged by the concerns of data centralization. They may suffer from malicious attacks, unauthorized access of personal identity and contacts, or even to spread fake information or false alarms by impersonating health authorities. Therefore, we introduced a novel integration of DCT technologies, including Bluetooth Low Energy (BLE), Geographic Information System (GIS), and Blockchain technology (Ethereum), named Blockchain-enabled Privacy-preserving Contact Tracing (BeepTrace) [5], [6]. This App has two modes (active and passive) and we intend to demo the active mode using BLE and Ethereum in this paper. Next, we describe the overall framework and implementation details of the application.

II. Methodology and the application

For the purpose of demo, we developed the BeepTrace-active in the first release. The passive mode has also been planned and will be covered in future work. In the following section, we will focus on the framework of BeepTrace-active. Note that, BeepTrace common functions, such as data upload and download on the blockchain, and account management, are shared between two modes.

Application of BeepTrace-Active uses Bluetooth Low Energy (BLE) as the sensing technology. It has great advantages in an effective distance, reactive delay and power consumption in particular. Besides, in the initial verification and testing stage of our BeepTrace application, we use Ethereum [7] as the blockchain platform with smart contract to store and manage the tracing data on the blockchain.

1) Framework: For BeepTrace-active, the overall framework is shown in Figure 1. Temporary pseudo-IDs are generated by users' smartphone at first. In step 2, the application broadcasts the generated pseudo-IDs to devices within the Bluetooth coverage, triggering a contact event. In step 3, a local list of all contact events is created, consisting of all received pseudo-IDs history from other BLE devices. Once a person is tested positive by a medical institution, either the health authorities or the user should be responsible for uploading its local contact list of the last 14 days to the blockchain, as shown in steps 4 and 5. In step 6, the application in active mode periodically downloads and updates local lists of positive cases from the blockchain. After that, the risky pseudo-IDs from the blockchain are matched with the users' local contact list in step 7. Lastly, the user is alerted of the risk level of COVID-19 if there is a match between the local contact list and risky pseudo-IDs from the blockchain.

2) Implementation details:

- **Generation of pseudo-ID** is required to initiate the service. The pseudo-IDs are based on the BLE UUID, which is generated by using a cryptographically strong pseudo random number generator. UUID is a 128-bit value and the probability of UUID crash is negligible, so

[*]Hong Kang and Zaixin Zhang contribute equally to this demo. BeepTrace repo:https://github.com/GlasgowCSI/BeepTrace_release

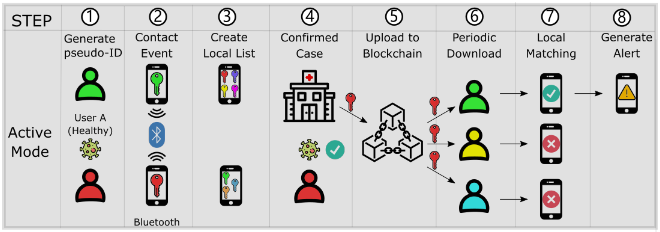

Figure 1: Framework of BeepTrace-active

it can be utilized as an encrypted anonymous identifier. To avoid storing UUID advertised by unrelated services, part of the UUID is modified and being recognized as a BeepTrace service identifier. When a new scan result is received by the users' device, it will be converted to string type and the BeepTrace service identifier will be checked. For each user, personal pseudo-ID will be updated periodically and get stored every time it changes. Personal pseudo-ID is advertised continuously via BLE. Meanwhile, Beeptrace will continuously listen to BLE devices nearby and register pseudo-ID with correct BeepTrace service identifier.

- **Users' interaction with Ethereum** is designed to upload and distribute contact tracing information. The users' requests are transferred to the Ethereum nodes by interfaces provided by Web3j based on the JSON-RPC framework. Web3j is a lightweight Java and Android class library that provides rich APIs for integration with clients (nodes) on the Ethereum network. It fits nicely with the Android apps of BeepTrace. The generation of public and private key pairs in the BeepTrace application is also referred to as Ethereum and uses the API provided by Web3j for generating and managing accounts. In addition, we replaced the password needed to generate the Ethereum account with a random number of 6 bits generated by the java.security.SecureRandom class.
- **Ether distribution** is important, due to the way Ethereum works. There is a fee (gas) needed when making a transaction or changing the status of a smart contract, such as uploading data. This poses some challenges to the implementation of our BeepTrace application. Our solution in the BeepTrace demo distributes the genesis address and the private key to individual clients, in which case, clients may pay any amount to the consensus network using the genesis account balances. In order to make sure there is enough fund in the genesis address, an initial deposit of a huge amount of ethers will be put into the genesis address by setting the genesis block. More specifically, we define accounts with a lot of ether (maximum value of uint256) into the genesis block of the Ethereum private chain. The genesis account's private keys and a string of random numbers used to generate the private keys are packaged into the application's APK. BeepTrace estimates the gas required before each user initiates a transaction to an Ethereum node and uses the genesis account to transfer the minimum cost into the user's account to ensure that the transaction can be executed by a miner, with the following balance calculation: $Balance_{minimum} = gas\ price \times gas\ limit + gas\ used \times gas\ price$
- **Upload and download data** are implemented using both smart contract and logs (description of events in a smart contract). Considering the storage overhead, logs are easier and cheaper to store. In BeepTrace-active, all pseudo-IDs and the corresponding timestamps are uploaded to the blockchain, sending transactions with data to a smart contract account. The proposed ether distribution solution works out the concern of mining cost. The way of downloading contact data from the blockchain is an event listener service supported by Web3j. This approach is consistent with the fact that we only use logs to store data, and there are two advantages. First, users can customize the Settings of the listening service, such as the starting block of the listening. Second, contact information uploaded on the blockchain can be downloaded in real-time if the user does not actively turn off the listener.
- **Data validation within 14 days** is recommended by health authorities for COVID-19, and varies for each diseases. In order to save the user storage, the life-cycle of stored information is 14 days. We record block ID of each pseudo-ID on blockchain. Every time the user downloads the data, a function embedded in the smart contract is used to get the minimum block ID within 14 days by calculating the timestamps of the pseudo-IDs on the blockchain. Then, the application will compare the latest local block ID to decide the block from which the tracing data is downloaded.

III. Conclusion

In this demo paper, we show the feasibility of BeepTrace in detail with implementations. Although we focus on the active mode, the blockchain functions are commonly shared within BeepTrace, such as blockchain data handling, ether distribution, etc. BeepTrace-active with BLE and GIS can efficiently and accurately obtain users' contact information with privacy by design. Besides, the well-designed encryption mechanism and blockchain technology comprehensively protect users' privacy with an open future. We cannot wait to launch the open-source beta version of BeepTrace and contribute to the public safety of humankind.

References

[1] A. Hekmati, G. Ramachandran, and B. Krishnamachari, "Contain: Privacy-oriented contact tracing protocols for epidemics," *arXiv preprint arXiv:2004.05251*, 2020.

[2] C. Watson, A. Cicero, J. Blumenstock, and M. Fraser, "A national plan to enable comprehensive COVID-19 case finding and contact tracing in the US," 2020.

[3] "NHS COVID-19 app," https://www.nhsx.nhs.uk/covid-19-response/nhs-covid-19-app/, accessed: 2020-05-24.

[4] India Goverment, "Aarogya Setu Mobile App," 2020. [Online]. Available: https://www.mygov.in/aarogya-setu-app/

[5] P. V. Klaine, L. Zhang, B. Zhou, Y. Sun, H. Xu, and M. Imran, "Privacy-preserving contact tracing and public risk assessment using blockchain for covid-19 pandemic," *IEEE Internet of Things Magazine*, vol. 3, no. 3, pp. 58–63, 2020.

[6] H. Xu, L. Zhang, O. Onireti, Y. Fang, W. J. Buchanan, and M. A. Imran, "Beeptrace: blockchain-enabled privacy-preserving contact tracing for covid-19 pandemic and beyond," *IEEE Internet of Things Journal*, 2020.

[7] V. Buterin *et al.*, "A next-generation smart contract and decentralized application platform," *white paper*, vol. 3, no. 37, 2014.