Blockchain-Driven Privacy-Preserving Contact-Tracing Framework in Pandemics

Xiao Li, *Graduate Student Member, IEEE*, Weili Wu, *Senior Member, IEEE*, and Tiantian Chen

***Abstract*—Blockchain technology, recognized for its decentralized and privacy-preserving capabilities, holds potential for enhancing privacy in contact tracing applications. Existing blockchain-based contact tracing frameworks often overlook one or more critical design details, such as the blockchain data structure, a decentralized and lightweight consensus mechanism with integrated tracing data verification, and an incentive mechanism to encourage voluntary participation in bearing blockchain costs. Moreover, the absence of framework simulations raises questions about the efficacy of these existing models. To solve above issues, this article introduces a fully third-party independent blockchain-driven contact tracing (BDCT) framework, detailed in its design. The BDCT framework features an Rivest-Shamir-Adleman (RSA) encryption-based transaction verification method (RSA-TVM), achieving over 96% accuracy in contact case recording, even with a 60% probability of individuals failing to verify contact information. Furthermore, we propose a lightweight reputation corrected delegated proof of stake (RC-DPoS) consensus mechanism, coupled with an incentive model, to ensure timely reporting of contact cases while maintaining blockchain decentralization. Additionally, a novel simulation environment for contact tracing is developed, accounting for three distinct contact scenarios with varied population density. Our results and discussions validate the effectiveness, robustness of the RSA-TVM and RC-DPoS, and the low storage demand of the BDCT framework.**

***Index Terms*—Blockchain, contact tracing, COVID-19 pandemic, delegated proof of stake (DPoS), RSA.**

I. Introduction

SINCE the first case of the novel corona-virus COVID-19 discovered in December 2019, there have been over 510 million globally confirmed cases, including 6 million deaths by April 2022.[1] Contact tracing is known to be an effective way for controlling the virus spread [1].

Formally, the contact tracing is the process of identifying history contact cases of people who may have come into contact with infected patients. Many countries or companies have developed contact tracing methods, such as TraceTogether in Singapore [2], the QR code system in China [3], and the Exposure Notification system developed by Google and Apple [4]. These apps mostly use Bluetooth to recognize nearby devices or GPS signal to get the accurate location coordinates to determine contact cases. Most of these tracing systems rely on central servers controlled by governments or healthcare authorities, which may collect the users' identities and other privacy data through an application installed on smart phones.

Systems based on centralized servers suffer single-point failure and are weak to attacks. Decentralized contact tracing methods are then promoted. As an emerging decentralized data generating, sharing, and storing technique, blockchain systems are introduced to solve contact tracing tasks to promote the security and privacy. Blockchain stores data into blocks that are connected to each other as a chain. The data stored in blocks are not able to be tempered. Smart contract deployed on blockchain can perform various functionalities. Furthermore, encryption and anonymization technologies can be applied in blockchain system to protect user's identity. The consensus mechanism in Blockchain allows blockchain systems keep working stably without a central server.

A few state-of-art contact tracing systems using blockchain technologies have been proposed. Hasan et al. [5] propose proof of location (PoL) and develop smart contracts to ensure the privacy of contact list. However, no simulation is provided. In addition, there is no incentive mechanism to motivate users to join the system. Authors assume there are plenty of users in the system behaving honestly, while the situation is hard to achieve in practice. Xu et al. [6] propose BeepTrace blockchain-based contact tracing solution, where a blockchain system plays the neutral role in bridging data transmission between different parties, such as patients, doctors, and government authorities. The users' geodata are securely preserved in specially designed blockchain. However, the efficiency of this system is not demonstrated, and no clear consensus mechanism and incentive mechanism are specified in the article. Lv et al. [7] propose Bychain, a three-layer contact tracing framework without reliance on trusted third parties. PoL is proposed to verify the contact record and incentive mechanism is design for maximizing contact tracing range. However, unlike this article, Bychain is not able to produce person-to-person accurate contact cases.

We conclude four main challenges to develop a third-party free blockchain-based contact tracing method. The four challenges are overlapped with each other. 1) Instead of simply treating blockchain as a separated storage method, how to leverage powerful consensus mechanism in blockchain system to

Manuscript received 17 November 2022; revised 22 November 2023; accepted 4 January 2024. Date of publication 25 January 2024; date of current version 31 May 2024. This work was supported by the U.S. National Science Foundation (NSF) under Grant 1822985. *(Corresponding author: Xiao Li.)*

The authors are with the Computer Science Department, The University of Texas at Dallas, Richardson, TX 75080 USA (e-mail: xiao.li@utdallas.edu).

Digital Object Identifier 10.1109/TCSS.2024.3351191

[1]https://covid19.who.int/

promote data security. 2) How to design an effective consensus mechanism to organize data storage and meanwhile achieve low latency of recording contact information. People should afford the computation cost and be able to access latest contact records timely to prevent further possible virus spread. 3) How to design the incentive mechanism so that people are motivated to join the contact tracing system and behavior honestly. 4) Due to lack of real-world contact data, as well as high cost of testing whole system in practice, it is hard to evaluate the effectiveness and efficiency of whole systems. The difficulty of collecting real-world contact information is not only from privacy concerns but also diversity of people contact scenarios. Contact cases happened in crowded cities and those happened in rural areas are totally different scenarios with different frequencies and amounts. This diversity brings challenge to design incentive mechanism fair to every one.

There is few work that clearly addresses all above mentioned four challenges. In this article, we aim to tackle the four challenges with the users' privacy ensured by proposing a fully third-party free contact tracing framework with blockchain technology that can produce accurate direct contact cases. A RSA-based transaction verification algorithm is proposed to ensure the correctness of recorded contact information and improve system robustness. To efficiently store contact information into blocks, we propose reputation corrected delegated proof of stake (RC-DPoS) consensus mechanism, which can control the right of appending new blocks. An incentive mechanism is then designed to work with RC-DPoS motivating people to work honestly and maintaining system decentrality. Finally, we design a contact tracing simulation method that simulates different real-world people contact scenarios to evaluate the effectiveness of proposed framework.

The reminder of this article is organized as follows. In Section II, we discuss existing related work on contract tracing. Section III is dedicated to presenting the overview of proposed BDCT framework. Next, we elaborate transaction verification method, RC-DPoS, and incentive mechanism in Sections IV, V, and VI, respectively. Experimental simulation and discussion are conducted in Section VII. Finally, Section VIII concludes the article.

II. Related Work

Traditional contact tracing tools are usually developed on central server using location technologies such as GPS, Wifi, cell tower, or Bluetooth [8], [9], [10]. Most of existing works are centralized where third-party servers are used collecting user's personal data and contact history to match contact records [11]. Centralized models are exposed to risks of single point failure, privacy data leaking, and security compromising. Though some contact tracing methods are proposed to be decentralized [12], [13], these methods still require a server to process data computing functions and are vulnerable to dishonest behaviors from malicious users.

Blockchain technology, first known as distributed ledger [14], can make a system work stably without any trust built among parties. Blockchain technology has demonstrated significant feasibility in IoT applications, which have similar requirements as contact tracing systems [15], [16], [17], [18], [19], [20].

Blockchain technology has been extensively explored in the domain of healthcare, such as healthcare sensor data protection [21], [22], drug traceability control [23], and medical supply chain management [24]. Blockchain technology shows great potential for developing privacy-preserving and efficient contact tracing applications [25].

Besides the above-mentioned work [5], [6], [7] in Introduction, there exists many other blockchain-based contract tracing frameworks or systems. Arifeen et al. [26] propose a high-level blockchain-based contract tracing framework where blockchain is used for patients to publish contact list. Zhang et al. [27] propose privacy-preserving contact tracing scheme in 5G-integrated and blockchain-based medical applications (PTBM) leveraging both permissionless and permissioned blockchain to manage users' location data, and 5G technique provides support for low latency communication. Liu et al. [28] include multiple involved parties in contact tracing, e.g., patients, doctors, and governments. Though the communication schema within blockchain among these parties are clearly defined, neither the blockchain consensus and incentive mechanism are not defined, nor blockchain simulation is provided. Hee and Salam [29] identify the challenge of communication delay if only Bluetooth signal is considered in distance approximation, and propose to use additional sound system to promote the contact tracing robustness. Blockchain nodes are clearly defined, however the proof of work (PoW) consensus mechanism adopted in the framework brings extra computation cost to devices.

Peng et al. [30] propose P^2B, where users can upload contact information to blockchain storage to be further verified and cross-checked by clients and authorities. P^2B is demonstrated with higher data transmission efficiency than BeepTrace. Vangipuram et al. [31] propose a three-tier architecture for storing numerous data collected by Internet-of-MedicalThings (IoMT) for contact tracing. In the architecture, blockchain is employed to securely transfer the data from the infected person to the hospital system using the edge infrastructure.

Zuhair et al. [32] consider a sixth-generation (6G)-assisted unmanned aerial vehicles (UAVs) empowered mass surveillance system in dense areas, which can monitor body temperature of persons. Salimibeni et al. [33] consider in-door contact tracing scenarios and propose trustworthy blockchain-enabled system for an indoor CT (TB-ICT) contact tracing framework. TB-ICT can motivate people to behave honestly since better credit can decrease mining difficulty. However, PoW-based consensus mechanism may bring high computation overhead. The comparison between this work and the recent start of the arts is presented in Table I.

The emerging probabilistic blockchain [34] that incorporates distributed nonbinary decisions offers unique functionality for contact tracing. For instance, probabilistic blockchains can be useful for predicting "high risk region," by allowing visitors to collectively determine whether the region carries a high contact risk and addressing dishonest user behavior by reaching consensus among historical contacts with probabilistic evaluation.

TABLE I
COMPARISON OF THIS WORK WITH RECENT STATE OF THE ARTS

		BDCT	[5] (2021)	[6] (2021)	[7] (2022)	[26] (2020)	[27] (2021)	[28] (2023)	[29] (2022)	[30] (2021)	[33] (2022)
Person2Person Tracing		✓	✓	✓	✗	✓	✗	✓	✓	✓	✓
Blockchain	Data Verification	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓
	Consensus Mechanism	✓	✓	✗	✓	✗	✓	✗	✗	✗	✓
	Incentive Mechanism	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓
Framework Simulation		✓	✗	✗	✓	✗	✓	✗	✓	✓	✓

III. CONTRACT TRACING FRAMEWORK OVERVIEW

A. Problem Definition and Preliminary Settings

In this article, we study the contact tracing problem as: given Bluetooth signals on smart devices, with the constraints of preserving privacy, we aim to output pairwise users' contact lists by discovering nearby Bluetooth devices. The goals to achieve for the contact tracing problem are the completeness and correctness of contact list, the contact tracing robustness, and attack resistance. We consider the attacks in this article as malicious users trying to disrupt the contact tracing or to take control of the blockchain ledgering system.

We assume our blockchain-driven contact tracing (BDCT) framework is implemented and deployed through clients on smart devices. People can join the contact tracing system by installing the client on their smart devices. It is assumed each user carries one device with the client installed. The client will generate private-public key pair and a unique device ID for each device. We use Bluetooth to evaluate the distance between two devices within a certain range based on the strength of Bluetooth signal. The furthest contact distance considered in this article is 5 m where Bluetooth can produce strong enough signal to support accurate computation [35].

B. Contact Tracing Procedure

At a given frequency, the client on a smart device will scan and record all the device IDs of nearby devices within a range. This process is fast and secure since the client only scans surrounding devices without having to pair two devices, which also avoids cyberattack through Bluetooth channel. If there is a device detected within 2 m,[2] the client will identify this as a contact case. The client will then store the device IDs of contacted devices into contact list locally in a special format which will be specified in the next section.

Most of previous works ignore the fact that mobile devices are not as robust as computers in terms of internet connectivity, system robustness, and security level. The device may fail to collect the contacted device information or be attacked to record false contact list. To improve the data integrity, a special role *witness* is proposed in this article. All the devices that are 2 m away but still within 5 m from the current device are considered witnesses of the contact case. Witnesses play important roles in BDCT, which help verify the reported contact list, speed up the verification process, and recover the missed contacts. The client will also store the device IDs of witnessed devices into witness list locally in the similar format as contact list. Users can check locally about whom they have contacted with and who have witnessed their contact cases at any time without knowing the real identity of the device owner.

[2]The distance can be adjusted according to particular scenarios.

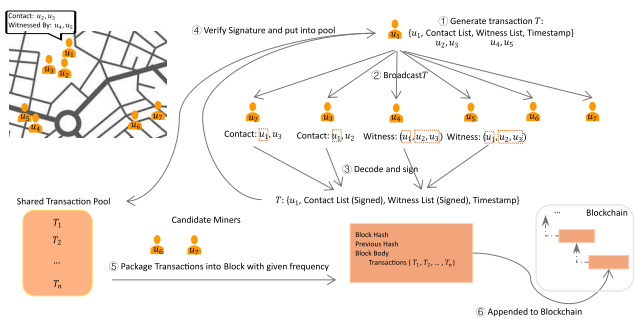

Fig. 1. Overview of contact tracing framework (user u_1 generates a transaction).

Based on the above setting, we now illustrate the overall BDCT framework in Fig. 1 with an example.

In Fig. 1, user u_1 tries to report his current contact case at *Timestamp*. Let's assume users u_2 and u_3 are within 2 m from u_1 and hence considered a contact case with user u_1. Users u_4 and u_5 are 2 m away but still within 5 m from u_1, and they witness that u_1 is with u_2 and u_3 at *Timestamp*. u_6 and u_7 are considered irrelevant to this contact case. There are six steps to record this contact case into blockchain.

Step 1: User u_1 initiates a blockchain transaction T_{con}, which is used for recording the contact case of u_1 at *Timestamp*. One transaction represents one contact case of users at some timestamp. Formal definition of T_{con} will be presented in Section IV.

Step 2: User u_1 then broadcasts the transaction T_{con} through internet to every user who have the client installed. Since no one knows others' identities, u_1 is not able to directly send message to u_2, u_3, u_4, and u_5.

Step 3: When other users receive the transaction T_{con}, it will check its local record if it contacted with u_1 at *Timestamp* or if it witnessed the reported contact case. Then, the contacted users in this example, u_2 and u_3, and the witnessed users, u_4 and u_5, will sign the transaction T_{con} and broadcast this signed transaction. The transaction generator u_1 will receive the signed transaction.

Step 4: After receiving the signed transaction T_{con}, u_1 will verify the signature. u_1 will wait for the signatures within a specific delay d, such as 60 min. Only the records in *ContactList*

verified valid will be finally preserved in T_{con}. u_1 will put transaction T_{con} into a shared transaction pool which is synchronized on every device along with blockchain.

Step 5: At given block generation frequency, one of the candidate miners will be selected to package all the transactions in the transaction pool into a block.

Step 6: The block is finally appended to the blockchain by the miner and broadcasts to all users in the network for synchronizing.

Steps 1–4 will be elaborated in Section IV by proposing RSA-based transaction verification method (RSA-TVM). In Section V, RC-DPoS and corresponding incentive mechanism are presented to complete steps 5 and 6.

IV. Transaction Verification Method

There are two major goals on contact tracing system: 1) data integrity—the collected contact cases should be as complete, untampered, and correct as possible; and 2) privacy—the whole system should never initiatively disclose any location or identity information of users.

In this article, we propose RSA-TVM to make sure the contact records in the transaction are valid meanwhile ensuring the anonymity, so that both above two goals can be achieved. We employee RSA algorithm as encryption module [36]. RSA algorithm is an asymmetric encryption algorithm and is able to generate a key pair (public key, private key) for a user. Public key is known by public, while the private key is only known by the owner. The secret message encrypted by public key can only be decoded by private key owner. A message can be signed by private key and then be verified only by corresponding public key.

Next, we will first describe how to initialize credentials for each user and then present RSA-TVM.

A. Generate User Credential

When a person u installs the tracing client on a smart device and become an user of the tracing system, the client will first name the device with an unique device ID, denoted as $u_{D_{\text{ID}}}$, and then generate a RSA key pair (public key, private key), denoted as $(u_{\text{Pub\_key}}, u_{\text{Pri\_key}})$. The length of each key is set to 1024 bits. The private key will be stored locally in the smart device. The public key and the device information will be included in a transaction through the client, then be stored into blockchain. This transaction is called "Registration Transaction," which is defined as $T_{\text{reg}} = \{T_{\text{id}}, (u_{D_{\text{ID}}}, u_{\text{Pub\_key}}, t)\}$. T_{id} is the unique id for each transaction and is generated by SHA256 algorithm [37] based on timestamp t as well as the transaction content $\{u_{D_{\text{ID}}}, u_{\text{Pub\_key}}, t\}$, so that any change made on the content will cause a different T_{id}.

After the registration transaction is stored in the blockchain, since every user in the system have a synchronized copy of the whole blockchain, every user will hold the public keys for every others.

Next, if the user wants to report contact cases and store the contact information into Blockchain, "Contact Transaction" will be initialized.

B. RSA-Based Transaction Verification Method (RSA-TVM)

Users will scan the nearby devices (through Bluetooth) at a given transaction generation frequency to get the nearby devices' IDs and record them locally. If there are nearby devices within 2 m detected by user u's device, u can generate "Contact Transaction," denoted as $T_{\text{con}} = \{T_{\text{id}}, (u_{D_{\text{ID}}}, C, W, t)\}$, where T_{id} is the unique id for each transaction and is generated based on timestamp t and the transaction content $\{u_{D_{\text{ID}}}, C, W, t\}$. The $u_{D_{\text{ID}}}$ is the device ID of u, and t is the timestamp for this contact case. C and W are Contact List and Witness List, which contain the information of the contacted people (devices) and the witness of this contact case, respectively. To generate C and W, user u first needs to decide a original secret message D, and then encrypt it with the public key of the contacted people (e.g., u_i) and the witnesses (e.g., u_j) of this contact case. For each contacted person u_i, $u_{i\text{Pub\_key}}$ encrypted text, denoted as $D_{u_{i\text{Pub\_key}}}$ is generated. Similarly, for each witness u_j, $D_{u_{j\text{ Pub\_key}}}$ is generated. Formally, the Contact List C is defined as a set of tuples: $C = \{(u_{i\text{Pub\_key}}, D_{u_{i\text{Pub\_key}}}) | \forall u_i\}$. Similarly, the Witness List is defined as: $W = \{(u_{j\text{Pub\_key}}, D_{u_{j\text{ Pub\_key}}}) | \forall u_j\}$.

In practice, we set each secret message D containing ten hex characters $(0-9, a-f)$, which is able to represent about 1.1×10^{12} different messages.

Witness list W can be very helpful to avoid contact case loss and improve robustness against dishonest user behaviors or system failure. We will show this later in Section VII.

The transaction T_{con} will then be broadcast to all users in order to protect privacy. Each user will check if C or W in the received T_{con} contains his/her public key. If so, the related tuples require his/her verification. Since the messages are all encrypted, therefore only the user who holds the public key can decrypt the encrypted secret message by his/her public key.

When u_i identifies the tuple $(u_{i\text{Pub\_key}}, D_{u_{i\text{Pub\_key}}})$ in C, u_i will decrypt the encrypted text $D_{u_{i\text{Pub\_key}}}$ with the private key $u_{i\text{Pri\_key}}$ to get the secret message D. Then, u_i will check local contact history. If u_i has the record that u_i contacted with u at timestamp $t \pm 3$ min, then u_i can confirm the tuple $(u_{i\text{Pub\_key}}, D_{u_{i\text{Pub\_key}}})$ valid in T_{con}. Then, u_i needs to send a message back to u to indicate that the contact record about u_i in T_{con} is confirmed. Specifically, u_i signs the secret message D with his private key. The signed text is denoted as $SD_{u_{i\text{Pri\_key}}}$. Then, u_i replaces $(u_{i\text{Pub\_key}}, D_{u_{i\text{Pub\_key}}})$ with $(u_{i\text{Pub\_key}}, SD_{u_{i\text{Pri\_key}}})$ in T_{con}, and broadcast to all users.

u_j will conduct similar verification on the related tuple in witness list W. If u_j has the record that the transaction generator u contacted with all users in C at $t \pm 3$ min, then u_j will consider all tuples in C valid by signing secret message in related tuple in W.

If u_i cannot find any local record showing u_i contacted with u at timestamp $t \pm 3$ min, then u_i believes this is a wrong record. u_i will sign a predefined warning message $Z =$ "Wrong Record" instead of signing the secret message D. The tuple $(u_{i\text{Pub\_key}}, D_{u_{i\text{Pub\_key}}})$ in T_{con} will then be $(u_{i\text{Pub\_key}}, Z_{u_{i\text{Pri\_key}}})$. Once the transaction generator u receives the updated T_{con} from user u_i, u will verify the signature with the public key of u_i.

A tuple in contact list C in T_{con} is considered valid if: 1) there is no signed warning message in the tuple; and 2) the

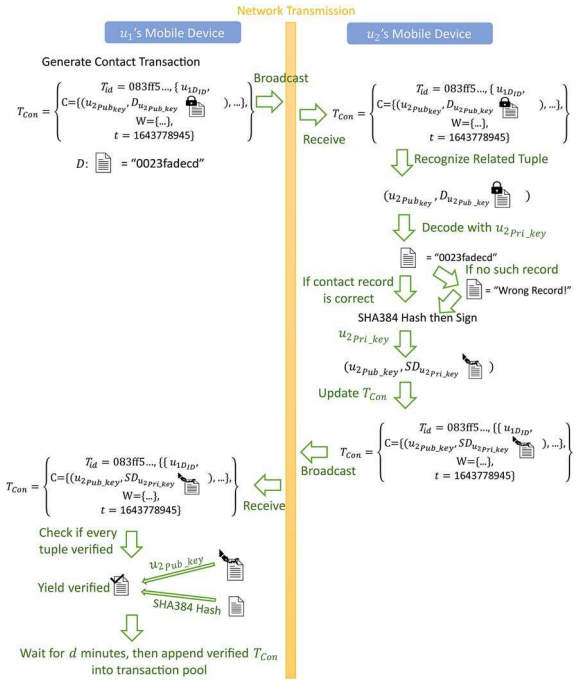

Fig. 2. Illustration of RSA-TVM.

secret message in tuple is correctly signed by the contacted person or at least one tuple in witness list is correctly signed by the witness. Due to network or system failure of users' smart devices, users may have no response to the related tuple in C or W within given delay d. In this case, the tuple will still be considered valid as long as one witness has verified that this contact case is correct. Only the valid tuples in C will be finally preserved in T_{con}. The transaction T_{con} will be put into the shared transaction pool waiting to be packaged into block. Any user u_i or u_j who signed D or Z will get reward for helping verify the contact case. We will discuss reward policies in Section VI. Fig. 2 shows the process of RSA-TVM that u_2 verifies u_1's contact case.

V. Reputation-Corrected DPoS Mechanism (RC-DPoS)

Most existing work considers it is straightforward to let the transaction generator directly package the verified transactions into blocks and then broadcast to all users instead of choosing a miner to do the job. However, the above strategy will cause *unfair incentive reward problem* due to the nature of diverse contact scenarios.

Unfair incentive reward problem: Users are rewarded for reporting contact cases. However, people naturally have different chances to have contact cases people who live or work in human-dense areas, such as cashiers in markets, will obviously have much more contact cases than those who stay or work at home, thus gain much more reward. This actually encourages people to make contacts in order to earn reward, which is against the social distancing policy during pandemics.

The simulation in Section VII demonstrates the existence of the problem. In addition, if miners can be the one not in the *Contactlist* or *WitnessList*, it helps avoid group cheating that small verify fake contact transactions for each other and append new blocks in order to gain great amount of reward rapidly. Therefore, it is imperative to carefully design consensus mechanism and incentive mechanism to balance the reward. The consensus mechanism is required computational lightweight and has high transaction throughput to satisfy the huge data storage demand on smart devices.

Delegated Proof of Stake (DPoS) consensus mechanism [38] is a popular light-weight consensus mechanism. The DPoS can produce high throughput without compromising decentrality of blockchain system if everyone is honest and the voting is random. However, it cannot be directly applied in our proposed BDCT. In order to motivate people to share their contact information, reward must be given for generating contact transactions. In DPoS the reward is stake, people living or working in human-dense areas will gather stakes quickly. Thus, their votes will gradually become highly weighted due to high stakes, hence their votes will easily determine the selection of candidate miners. In other words, the whole blockchain system will be dominated and become centralized.

We propose RC-DPoS consensus mechanism to solve the issue. In RC-DPoS, we assign reputation to each user, which is represented by credit c. Users will gain reputation reward instead of stake reward for honestly reporting their contact cases, while only gain stake reward for working as a miner. Specifically, the RC-DPoS mechanism works as follows.

Step 1: All new users in the contact tracing framework will be given initial stake s_0 and credit c_0.

Step 2: Initially, the candidate miner set is empty, the candidate selection process will start. Each user votes for another one trusted user and users cannot vote for themselves. Similar to DPoS, the vote is weighted according to the voter's stake. But the total votes received by a user will be corrected by receiver's credit. Formally, let N denotes the total number of users in the system. For user $u_i, i \in Z^N$, the corrected total weighted received by u_i is calculated according to the following equation:

$$G_i = \frac{\text{RF}(u_i) + 1}{2} \sum_{u_k} \frac{s_k}{\sum_{j \in Z^N} s_j} \tag{1}$$

where the sum taken over user u_k who votes u_i is the total weighted vote received by u_i without correction, and s_k is the current stake amount of u_k. c_i is the current credit amount of u_i. $\text{RF}(u_i)$ is the *reputation correction factor* of user u_i, which is defined as

$$\text{RF}(u_i) = \frac{c_i - \min_{j \in Z^N}(c_j)}{\max_{j \in Z^N}(c_j) - \min_{j \in Z^N}(c_j)}. \tag{2}$$

$\text{RF}(u_i) \in [0, 1]$, and $[\text{RF}(u_i) + 1/2] \in [0.5, 1]$. The intuition behind this equation is that users with good reputation should have higher chance to be a candidate miner, meanwhile, we need avoid applying too much punishment on other users with lower reputation (maximum 50% off on received votes).

Step 3: Rank all users in descending order according to their vote scores. The top $\lceil N/5 \rceil$ users are selected into candidate miners set. The size of candidate miners set can be adjusted based on specific applications.

Step 4: At a given block generation frequency (3 min, 5 min, or so on), one arbitrary miner selected from the candidate

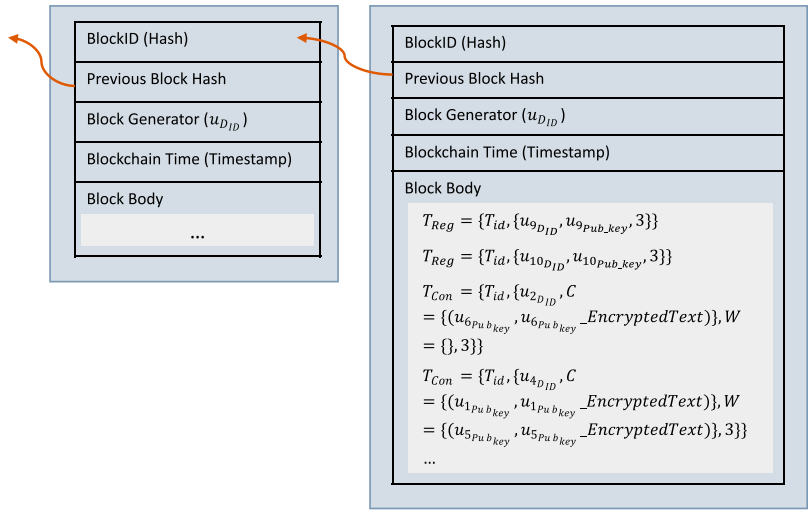

Fig. 3. Blockchain storage structure in proposed framework.

miner set will package all the transactions in the shared transaction pool into a block and append it into the miner's local blockchain. Then, shared transaction pool is empty and waits for new verified transactions. The structure of the blockchain storage is illustrated in Fig. 3.

Step 5: The miner then broadcasts this blockchain update to all users. Users in the system will update their local blockchain and the local transaction pool. The miner will be given stake reward and reputation reward. Reward detail will be elaborated in Section VI. Then, the miner will be removed from the candidate miner set.

Step 6: When a miner fails to do this job within a excusable delay (e.g., 10 min) due to network disconnection or system failure, a penalty will be applied on the miner by taking away some credits and no stake reward will be given. The miner will be removed from the candidate miner list and another miner will be delegated to do the job. Whenever the candidate miner set is empty, back to *Step 2*.

VI. Incentive Mechanism

The proposed BDCT contact tracing framework is third-party free and fully relies on people to generate transactions, store contact cases into blocks and maintain system decentrality. Therefore, it is crucial to design an incentive mechanism to motivate people to generate contact transactions and append blocks into blockchain honestly. It is also important to ensure the incentive mechanism does not specially benefit a particular group of people to avoid the system becoming centralized and dominated. In this article, we design the incentive mechanism as a composition of following four incentive policies.

1) Users will be rewarded with one unit[3] credit for generating transactions. The users cannot get the reward until the transaction is accepted by the shared transaction pool. This will motivate users to honestly report their contact list.
2) Users will be rewarded with one unit credit after successfully verifying related tuple in contact transactions. This will motivate users to participate in generating transactions and improve the speed of verifying contact cases.
3) Users will be rewarded with R_i unit stake reward and one unit credit reward for mining a block, e.g., append a new block into existing blockchain. R_i is corresponding to the total amount of transactions that u_i generated, which is formally defined as

$$R_i = w * \frac{\mathrm{TF}(u_i) + 1}{2} \tag{3}$$

$$\mathrm{TF}(u_i) = 1 - \frac{t_i - \min_{j \in Z^N}(t_j)}{\max_{j \in Z^N}(t_j) - \min_{j \in Z^N}(t_j)}. \tag{4}$$

w is a predefined reward amount (e.g., five units) and t_i is the total number of transaction generated by u_i. $\mathrm{TF}(u_i) \in [0, 1]$ is called the *transaction correction factor*. From the above definition, we could find that the more transaction u_i has generated, the lower stake reward will be given to u_i. The intuition behind R_i is that we do not want people who generate much transactions also gain much stake reward for completing every mining job since they naturally have more chance to become a miner according to incentive policy 1) and (1). On the other hand, people who generate less transactions will get more stake reward per mining job they complete. $[\mathrm{TF}(u_i) + 1]/2 \in [0.5, 1]$ will make a maximum 50% off on the stake reward. Therefore, R_i can help balance the stake reward among users in different contact scenarios, hence help maintain the vote power distributed.
4) A user will lose five units credits and earn no stake reward if the user fails to complete a mining job.

VII. Simulation and Discussion

A. Simulation Method

There is no real-world trajectory dataset that can produce real-word people contact cases in terms of frequency and

[3]Numbers in the incentive mechanism only for indicating the relative amount; they can be of any unit.

amount. Since it is hard to collect real-word trajectory in a wide range due to privacy concerns and diversity of contact scenarios, we conduct experiments on synthetic datasets that simulates different people contact scenarios.

We propose to consider three general contact scenarios decided based on population density: *Low density (Sparse)*, *Medium density (Medium)*, and *High density (Crowded)*. Each scenario can intuitively represent one kind of real-world contacting cases. "Sparse" can represent the contacting cases in rural area or residential area. "Medium" can represent the contact cases in schools, parks, or other common public areas. "Crowded" represents or contacting cases happening in some very crowded places, such as shopping malls and sports events.

People in three scenarios have different frequencies of having contact cases, different numbers of contacted people and witnesses. To achieve this goal, we specify the settings for the three scenarios as follows.

Low density (sparse) case: In each transaction, the length of contact list and witness list follow normal distribution $\mathcal{N}(\mu = 0, \sigma = 2)$ and $\mathcal{N}(\mu = 0, \sigma = 1)$, respectively. The frequency of generating transaction is 1 case/h.

Medium density (medium) case: In each transaction, the length of contact list and witness list follow the normal distribution $\mathcal{N}(\mu = 2, \sigma = 4)$ and $\mathcal{N}(\mu = 2, \sigma = 2)$, respectively. The frequency of generating transaction is 3 cases/h.

High density (crowded) case: In each transaction, the length of contact list and witness list follow the normal distribution $\mathcal{N}(\mu = 5, \sigma = 2)$ and $\mathcal{N}(\mu = 7, \sigma = 2)$, respectively. The frequency of generating transaction is 12 cases/h.

We implement the framework with Python 3.7, and all simulations are conducted on a machine with Intel Core i7-10875h 8 cores and 32GB memories. Each user is implemented as a thread of python, and all threads run paralleled to simulate real time contacting. We randomly generate the contact cases for each user without considering reasonable trajectories for them, since the trajectories do not affect the evaluation of the effectiveness. For the length of contact list or witness list, we only adopt the nonnegative numbers sampled from corresponding normal distributions.

B. Decentrality Evaluation

It is crucial to maintain the decentrality of BDCT, so that the voting power is distributed and uses can be motivated to keep contributing contact cases honestly.

We simulate 200 users for each contact scenario, hence totally 600 users are in the whole simulation environment. To measure the dencentrality of the system, we draw Lorenz curve and calculate the Gini coefficient/index of three factors of the 600 users: user *balance* (cumulative stake reward), user *credits* (cumulative reputation reward), and the *total number of mined blocks*.

Lorenz curve is originally proposed for drawing the cumulative income from different units when they are in the ascending order [39]. The closer the income distribution is to uniform distribution, the closer the corresponding Lorenz curve is to line $y = x$. We extend Lorenz curve in this article to illustrate the decentrality of the proposed RC-DPoS.

The Gini coefficient *Gini* is a metric for quantitatively measure inequality of a distribution, which can derived from Lorenz curve [40]. It is defined as a ratio with values between 0 and 1. Specifically, the numerator is the area between the Lorenz curve of the distribution and the uniform distribution line; the denominator is the area under the uniform distribution line. Hence, $Gini = 0$ indicates perfect equality of a distribution, and $Gini = 1$ indicates the distribution is total skew to one unit.

We adopt the DPoS mechanism as the baseline. In the baseline, no credit reward is given to users, and users will get 1 unit stake reward for generating or verifying transactions and 5 units stake reward for mining a block. Other settings are kept the same as proposed BDCT. The initial stake and credit of users are set to 100 units. Random voting strategy is adopted for voting the candidate miners.

We run the simulation for ten times and evaluate statistical significance of Gini coefficient of user balance between baseline and proposed BDCT. We conduct a two-sided T-test for the null hypothesis that two frameworks' stake reward distribution has identical average (expected) values. The $p = 1.22 \times 10^{-32}$ indicates that BDCT achieves definitely better stake reward dencentrality.

Fig. 4 shows the results of Gini coefficient and Lorenz curve of the three factors. Fig. 4(a) and 4(b) shows how the Gini coefficient changes with more and more data stored in the blockchain. Fig. 4(c) shows that the Gini coefficient of balance of baseline DPoS remains as high as 0.56 when blockchain height is 10k, indicating the stake rewards are mostly given to people in Dense scenario. Since the baseline DPoS mechanism does not consider any credit reward, the Gini coefficient of credit stays 0 in Fig. 4(a)–(c). In addition, the Gini coefficient of mined blocks count of baseline is close to 0, this is because under the random vote strategy in DPoS, users have the same expectation to be selected as a miner.

In Fig. 4(b), our BDCT framework makes the Gini coefficient decrease with the height of blockchain, which means BDCT is achieving balanced stake reward when we continue recording more data. In Fig. 4(d), the Gini coefficient of credit is 0.57 when blockchain height is 10 000, showing users in dense area can indeed earn more credit than other users as expected. The Gini coefficient of mined blocks count is 0.27 which is higher than 0.12 in baseline, indicating people in dense areas indeed have higher chance to be a miner. Gini coefficient of credit is 0.19 which is significantly lower than 0.56 in baseline and demonstrates our RC-DPoS and proposed incentive mechanism can successfully balance the stake reward among different groups of users.

We further investigate the stake reward distribution. In DPoS baseline, the 200 users (1/3 of total users) in the Crowded scenario together holds more than 85% stake rewards giving them more than 85% vote power. This demonstrate the necessity of dealing with Unfair incentive reward problem as described in Section V to avoid people in Crowded areas can naturally earn more reward and take control of the whole contact tracing system. Compared with the baseline, the three groups of users in three different contact scenarios in our proposed BDCT hold stake reward 23%, 42%, and 34% respectively.

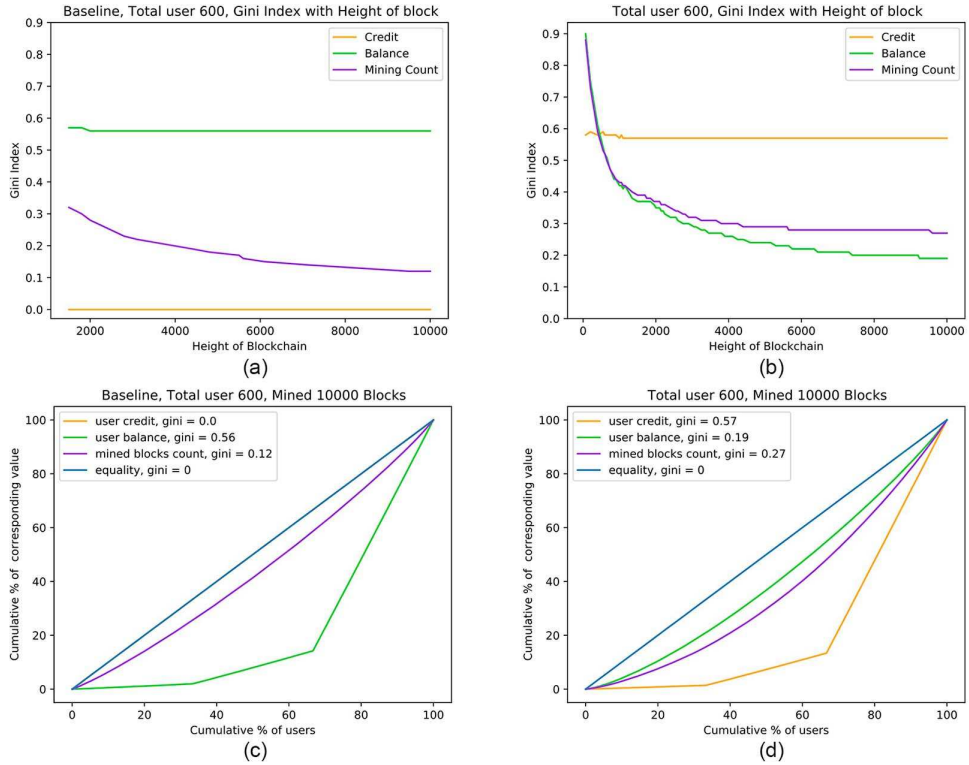

Fig. 4. Lorenz curve and Gini coefficient of proposed BDCT framework and baseline DPoS. (a) Gini coefficient, baseline DPoS. (b) Gini coefficient, our BDCT. (c) Lorenz curve, baseline DPoS. (d) Lorenz curve, our BDCT.

TABLE II
AVERAGE CONTACT CASE LOSS PERCENTAGE AT DIFFERENT FAILURE RATE p

User Failure Rate	p = 0.1	p = 0.2	p = 0.3	p = 0.4	p = 0.5	p = 0.6
BDCT	0.18±0.02%	0.45±0.03%	0.68±0.01%	1.19±0.02%	1.96±0.12%	3.69±0.03%
BDCT-w/o-Witness	2.25±0.03%	6.12±0.03%	11.12±0.04%	18.70±0.09%	27.63±0.07%	38.74±0.14%

This also demonstrates the resistance to the attack that even though malicious group can fake contact cases and generate a number of transactions, they will not gain significant increase on rewards.

C. Tracing Robustness Evaluation

Mobile devices are usually with low computational power and low security level, and sometimes may suffer from system failure or network delay and disconnection. All those factors can cause failure of detecting contact case, verifying contact list or receiving transactions. In this article, we proposed witness for every contact case in BDCT framework, which can improve the robustness of recording correct contact cases. As mentioned in Section IV-B, if a tuple in contact case is not verified due to the failure mentioned above, as long as there is one witness in W verified the contact list C, the tuple will be considered valid.

To evaluate the robustness of recording contact information of the proposed system, we set a failure rate p of each user, representing that the user has a probability p of failing to verify the corresponding transaction. Then, we compute how many contact cases, that u_i contacts with u_j at timestamp t, e.g., (u_i, u_j, t), will lose comparing with the given 300k contact cases. We use a baseline $BDCT - w/o - Witness$ that is BDCT without witness role, therefore (u_i, u_j, t) will be lost when both u_i and u_j fail to verify the corresponding tuples in C. This is also a common design in exiting works [12], [41]. We simulate this experiment for ten times and report the average results.

Table II shows the simulation results. It can be seen that our framework lost significantly less contact cases than baseline at any failure rate p. BDCT can correctly record nearly 96.31% (1%–3.69%) total contact cases even every node has 0.6 failure probability, which is 35% more than the baseline that can only preserve about 61.26% (1%–38.74%) contact cases.

D. Time Complexity Analysis

We analyze the computation cost of a single contact case from being generated as contact transaction to being stored in blockchain. The contact case will be first verified by RSA-TVA, then be packaged by miner into a block. The major computation is for RSA-TVA as the packaging is constant time for a miner to produce block hash and put all required field together in one block. Assume the contact case has n_c contacts, n_w witnesses and every contact and witness will participate in RSA-TVA.

We adopt RSA-1024, hence the public key $u_{i_{\text{Pub\_key}}} = (E, N)$ where $E < 2^{1024}$ and $N < 2^{1024}$. Encrypting a message or verifying a signature using $u_{i_{\text{Pub\_key}}}$ requires $O(\log^2(n) * \log(e)) = O(\log^2(n))$ where e and n are the number of bits of E and N, respectively, and $e < n \leq 1024$. Decrypting a encoded message or sign a message using corresponding private key requires $O(\log^3(n))$ [42].

The computation cost for verifying the contact case is composed by three parts: first $(n_c + c_w) * O(\log^2(n))$ for generating encrypted message in contact list and witness list, then $2 * (n_c + c_w) * O(\log^3(n))$ for these message will be decrypted and signed by correct users and finally $(n_c + c_w) * O(\log^2(n))$ for verifying the signature. As one contact case involves n_c participants, and at most each participant can generate a contact transaction then conduct RSA-TVA, therefore the upbound of computation cost is $n_c * (2 * (n_c + c_w) * O(\log^2(n)) + 2 * (n_c + c_w) * O(\log^3(n))) = n_c * (O((n_c + c_w) * \log^3(n)) = O(n_c * (n_c + n_w) * \log^3(n))$, and $\log^3(n)$ will be approximately constant for a fixed RSA key length.

E. Storage Cost Analysis

In BDCT framework, every user is holding the whole copy of blockchain. We evaluate the expected blockchain storage cost of the proposed BDCT framework by calculating the expected number of transactions and blocks generated per hour with respect to our experiment setting.

We denote the expected size of total blockchain segment generated per hour as $E(S_{\text{TB}/h})$, expected size of all block heads per hour as $E(S_{\text{BH}/h})$, and expected size of all block bodies per hour as $E(S_{\text{BB}/h})$. The size of single block head is denoted as s_{BH}, the expected amount of blocks generated per hour is $N_{B/h}$. Hence, $E(S_{\text{BH}/h}) = E(N_{B/h}) * s_{\text{BH}}$. The block bodies contain transactions, therefore $E(S_{\text{BB}/h}) = E(S_{T/h})$, where $E(S_{T/h})$ is the expected size of total transactions generated per hour. Consequently, $E(S_{\text{TB}/h})$ is calculated by the following equation:

$$\begin{aligned} E(S_{\text{TB}/h}) &= E(S_{\text{BH}/h}) + E(S_{\text{BB}/h}) \\ &= E(N_{B/h}) * s_{\text{BH}} + E(S_{T/h}). \end{aligned} \tag{5}$$

The speed of generating a block is predefined in the system, e.g., every 5 min. Therefore, $E(N_{B/h}) = 12$. Based on the block structure illustrated in Fig. 3, s_{BH} can be calculated as follows:

$$\begin{aligned} s_{\text{BH}} &= s_{\text{BlockHash}} + s_{\text{PreviousBlockHash}} \\ &\quad + s_{u_{D_{\text{ID}}}} + s_{\text{Timestamp}} \\ &= 2 * s_{\text{BlockHash}} + s_{u_{D_{\text{ID}}}} + s_{\text{Timestamp}} \\ &= 2 * 256 \text{ bits} + 10 \text{ bytes} + 32 \text{ bits} = 78 \text{ bytes}. \end{aligned} \tag{6}$$

$s_{\text{BlockHash}}$ is the size of a unique block ID, which is a SHA256 hash value, therefore $s_{\text{BlockHash}} = 256$ bits. $s_{u_{D_{\text{ID}}}}$ is the size of block generator's device ID. In our framework, the set device ID a string contains ten hex characters, which can represent $2^{10*4} \approx 1.1 \times 10^{12}$ unique devices. Since each char type hex character take 1 byte, $s_{u_{D_{\text{ID}}}} = 10$ bytes. $s_{\text{Timestamp}}$ is size of an Unix timestamp of 32 bit integer type, hence $s_{\text{Timestamp}} = 32$ bits.

In our experiment settings, three different contact scenarios are considered with different contact case generating frequency, number of contacted people, and number of witnesses. Though there may also registration transactions in block bodies, registration transactions cost minor storage. In this discussion, we consider the general case that block bodies contain only contact transactions. Then, $E(S_{T/h})$ is the sum of expected all transactions generated by three contact scenarios as follows:

$$E(S_{T/h}) = E(S_{\text{TS}/h}) + E(S_{\text{TM}/h}) + E(S_{\text{TC}/h}). \tag{7}$$

$E(S_{\text{TS}/h})$, $E(S_{\text{TM}/h})$, and $E(S_{\text{TC}/h})$ are the expected numbers of transactions generated per hour in Sparse scenario, Medium scenario, and Crowded scenario, respectively.

We denote $E(N_{\text{TS}/h})$ as the expected number of transactions (contact cases) per hour in Sparse scenario and $E(s_{\text{TS}})$ as the expected size of a transaction generated in Sparse scenario. Given the contact transaction structure described in Section IV-B, $E(s_{\text{TS}})$ is composed by size of transaction ID $s_{T_{\text{id}}}$, size of transaction generator's device ID $s_{u_{D_{\text{ID}}}}$, size of transaction timestamp $s_{\text{Timestamp}}$, expected size of contact list $E(s_C)$, and expected size of witness list $E(s_W)$. In each contact list C or witness list W, there are signed tuples $(u_{i_{\text{Pub\_key}}}, SD_{u_{i_{\text{Pri\_key}}}})$. The size of signed tuples is denoted as s_{st}. In our experiments, we generate 1024 bits RSA Keys with the Python package Crypto.[4] With the secret message as ten hex characters, $s_{\text{st}} =$ 56 bytes $+ 161$ bytes $= 217$ bytes in our simulation.

Though the length of contact list and witness list are sampled from normal distribution described in Section VII-A, we set the length to 0 if the sampled length is less than 0. The expected length of contact list $E(N_C)$ based such sample strategy satisfies the following equation:

$$\begin{aligned} E(N_C) &= \int_0^{\infty} x \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) dx \\ &= \frac{\sqrt{2}\sigma}{\sigma\sqrt{2\pi}} \int_{-\frac{\mu}{\sqrt{2}\sigma}}^{\infty} (\sqrt{2}\sigma t + \mu) \exp^{-t^2} dt \\ &\quad \times \left(\text{with } t = \frac{x-\mu}{\sqrt{2}\sigma}\right) \\ &= \frac{1}{\sqrt{\pi}} \left(\sqrt{2}\sigma \int_{-\frac{\mu}{\sqrt{2}\sigma}}^{\infty} t \exp^{-t^2} dt + \mu \int_{-\frac{\mu}{\sqrt{2}\sigma}}^{\infty} \exp^{-t^2} dt\right) \\ &= \frac{1}{\sqrt{\pi}} \left(\frac{\sigma}{\sqrt{2}} \exp^{-\frac{\mu^2}{2\sigma^2}} + \mu Z(t)\right) \end{aligned} \tag{8}$$

$$\begin{aligned} Z(t) &= \int_{-\frac{\mu}{\sqrt{2}\sigma}}^{\infty} \exp^{-t^2} dt \\ &= \int_{-\frac{\mu}{\sqrt{2}\sigma}}^{0} \exp^{-t^2} dt + \int_0^{\infty} \exp^{-t^2} dt \\ &\leq \int_0^{\frac{\mu}{\sqrt{2}\sigma}} \frac{1}{t^2+1} dt + \frac{\sqrt{\pi}}{2} \\ &= \arctan\left(\frac{\mu}{\sqrt{2}\sigma}\right) + \frac{\sqrt{\pi}}{2}. \end{aligned} \tag{9}$$

[4] https://pycryptodome.readthedocs.io

Since $Z(t)$ is limited by an upper bond, we can calculate upper bond of the expected number of transactions per hour in Sparse scenario by the following equation. Similarly, $E(S_{\mathrm{TM}/h}) \leq 3374$ bytes and $E(S_{\mathrm{TC}/h}) \leq 36\,227$ bytes

$$\begin{aligned} E(S_{\mathrm{TS}/h}) &= E(N_{\mathrm{TS}/h}) * E(s_{\mathrm{TS}}) \\ &= 1 * (s_{T_{\mathrm{id}}} + s_{u_{D_{\mathrm{ID}}}} + s_{\mathrm{Timestamp}} + E(s_C) + E(s_W)) \\ &= 256 \text{ bits} + 10 \text{ bytes} + 32 \text{ bits} \\ &\quad + E(s_C) + E(s_W) \\ &= 46 \text{ bytes} + E(N_C) * s_{\mathrm{st}} + E(N_W) * s_{st} \\ &\leq 46 \text{ bytes} + \left(\frac{\sqrt{2}}{\sqrt{\pi}} + \frac{\sqrt{2}}{2\sqrt{\pi}}\right) * 217 \text{ bytes} \\ &\approx 306 \text{ bytes}. \end{aligned} \tag{10}$$

Hence, $E(S_{T/h}) \leq 306 + 3374 + 36227 = 39\,907$ bytes. Then, $E(S_{\mathrm{TB}/h}) = E(N_{B/h}) * s_{\mathrm{BH}} + E(S_{T/h}) = 12 * 78 + 39\,907$ bytes $= 40\,843$ bytes ≈ 39.89 KB. Therefore the expected storage cost for the blockchain generated per day is $24 * E(S_{T/h}) = 24 * 39.89\,\mathrm{KB} = 957.36\,\mathrm{KB} < 1\,\mathrm{MB}$, which is totally affordable for most smart devices.

VIII. Conclusion

In this article, we propose a BDCT framework, which is a fully decentralized framework without any third-party required. We propose the role "witness" in the framework to promote contact tracing data integrity, and the RSA-TVM to verify the correctness of the reported contact cases. RC-DPoS consensus mechanism is applied to select miners based on both users' reputation and users' stake. An incentive mechanism is further developed to motivate people to keep reporting contact cases honestly and work with RC-DPoS achieving balanced stake reward distribution to maintain the whole framework decentralized. In the simulation, we propose a simulation environment, which mixes three contact scenarios based on different population density. The simulation results demonstrate our proposed framework can achieve significantly decentrality than the baseline framework, and RSA-TVM incorporated with "witness" role in the framework can hugely improve the system robustness.

References

[1] World Health Organization and others, "Contact tracing during an outbreak of Ebola virus disease," 2014. Accessed: Oct. 30, 2020. [Online]. Available: https://www.who.int/publications/i/item/9789290232575

[2] J. Bay, J. Kek, A. Tan, and C. S. Hau, "BlueTrace: A privacy-preserving protocol for community-driven contact tracing across borders," 2020. Accessed: Oct. 30, 2020. [Online]. Available: https://bluetrace.io/static/bluetrace_whitepaper-938063656596c104632def383eb33b3c.pdf

[3] F. Liang, "COVID-19 and health code: How digital platforms tackle the pandemic in China," *Social Media + Soc.*, vol. 6, no. 3, 2020, Art. no. 2056305120947657.

[4] Apple and Google, "Apple and Google partner on COVID-19 contact tracing technology," Apple and Google, Dec. 2021. Accessed: Sep. 30, 2010. [Online]. Available: https://www.apple.com/newsroom/2020/04/apple-and-google-partner-on-covid-19-contact-tracing-technology/

[5] H. R. Hasan, K. Salah, R. Jayaraman, I. Yaqoob, M. Omar, and S. Ellahham, "COVID-19 contact tracing using blockchain," *IEEE Access*, vol. 9, pp. 62956–62971, 2021.

[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 Things J.*, vol. 8, no. 5, pp. 3915–3929, Mar. 2021.

[7] W. Lv, S. Wu, C. Jiang, Y. Cui, X. Qiu, and Y. Zhang, "Towards large-scale and privacy-preserving contact tracing in COVID-19 pandemic: A blockchain perspective," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 1, pp. 282–298, Jan./Feb. 2022.

[8] A. Trivedi, C. Zakaria, R. K. Balan, and P. J. Shenoy, "WiFiTrace: Network-based contact tracing for infectious diseases using passive WiFi sensing," *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 5, no. 1, pp. 37:1–37:26, 2021.

[9] A. Hekmati, G. S. Ramachandran, and B. Krishnamachari, "CONTAIN: Privacy-oriented contact tracing protocols for epidemics," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM)*, 2021, pp. 872–877.

[10] P. C. Ng, P. Spachos, and K. N. Plataniotis, "COVID-19 and your smartphone: BLE-based smart contact tracing," *IEEE Syst. J.*, vol. 15, no. 4, pp. 5367–5378, Dec. 2021.

[11] M. A. Azad et al., "A first look at privacy analysis of COVID-19 contact-tracing mobile applications," *IEEE Internet Things J.*, vol. 8, no. 21, pp. 15796–15806, Nov. 2021.

[12] N. Ahmed et al., "A survey of COVID-19 contact tracing apps," *IEEE Access*, vol. 8, pp. 134577–134601, 2020.

[13] J. Chan et al., "PACT: Privacy sensitive protocols and mechanisms for mobile contact tracing," 2020, *arXiv:2004.03544*.

[14] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2009. Accessed: Oct. 30, 2020. [Online]. Available: https://assets.pubpub.org/d8wct41f/31611263538139.pdf

[15] J. Guo, X. Ding, and W. Wu, "A blockchain-enabled ecosystem for distributed electricity trading in smart city," *IEEE Internet Things J.*, vol. 8, no. 3, pp. 2040–2050, Feb. 2021.

[16] Y. Fan, L. Wang, W. Wu, and D. Du, "Cloud/edge computing resource allocation and pricing for mobile blockchain: An iterative greedy and search approach," *IEEE Trans. Comput. Soc. Syst.*, vol. 8, no. 2, pp. 451–463, Apr. 2021.

[17] X. Ding, J. Guo, D. Li, and W. Wu, "An incentive mechanism for building a secure blockchain-based Internet of Things," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 1, pp. 477–487, Jan.–Mar. 2021.

[18] L. Dong, W. Wu, Q. Guo, M. N. Satpute, T. Znati, and D. Du, "Reliability-aware offloading and allocation in multilevel edge computing system," *IEEE Trans. Reliab.*, vol. 70, no. 1, pp. 200–211, Mar. 2021.

[19] C. Luo, L. Xu, D. Li, and W. Wu, "Edge computing integrated with blockchain technologies," in *Complexity Approximation: In Memory of Ker-I Ko*, vol. 12000 (Lecture Notes in Computer Science), D. Du and J. Wang, Eds., New York, NY, USA: Springer-Verlag, 2020, pp. 268–288.

[20] P. Kumar et al., "PPSF: A privacy-preserving and secure framework using blockchain-based machine-learning for IoT-driven smart cities," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 3, pp. 2326–2341, Jul.–Sep. 2021.

[21] M. F. Younis, W. Lalouani, N. Lasla, L. Emokpae, and M. Abdallah, "Blockchain-enabled and data-driven smart healthcare solution for secure and privacy-preserving data access," *IEEE Syst. J.*, vol. 16, no. 3, pp. 3746–3757, Sep. 2022.

[22] R. Kumar et al., "An integration of blockchain and AI for secure data sharing and detection of CT images for the hospitals," *Comput. Med. Imag. Graph.*, vol. 87, 2021, Art. no. 101812.

[23] G. Kumar, "Blockchain in enterprise application for pharmaceutical drug traceability," *Int. J. Sci. Res.*, vol. 12, no. 8, pp. 130–134, 2023.

[24] S. K. Nanda, S. K. Panda, and M. Dash, "Medical supply chain integrated with blockchain and IoT to track the logistics of medical products," *Multimedia Tools Appl.*, vol. 82, no. 21, pp. 32917–32939, 2023.

[25] S. M. Idrees, M. Nowostawski, and R. Jameel, "Blockchain-based digital contact tracing apps for COVID-19 pandemic management: Issues, challenges, solutions, and future directions," *JMIR Med. Inform.*, vol. 9, no. 2, 2021, Art. no. e25245.

[26] M. M. Arifeen, A. Al Mamun, M. S. Kaiser, and M. Mahmud, "Blockchain-enable contact tracing for preserving user privacy during COVID-19 outbreak," *Preprints*. Jul. 2020, 2020070502.

[27] C. Zhang, C. Xu, K. Sharif, and L. Zhu, "Privacy-preserving contact tracing in 5G-integrated and blockchain-based medical applications," *Comput. Standard Interfaces*, vol. 77, 2021, Art. no. 103520.

[28] M. Liu, Z. Zhang, W. Chai, and B. Wang, "Privacy-preserving COVID-19 contact tracing solution based on blockchain," *Comput. Standards Interfaces*, vol. 83, 2023, Art. no. 103643.

[29] Z. Hee and I. Salam, "Blockchain based contact tracing: A solution using Bluetooth and sound waves for proximity detection," *IACR Cryptol. ePrint Arch.*, vol. 2022, 2022, Art. no. 209.
[30] Z. Peng, C. Xu, H. Wang, J. Huang, J. Xu, and X. Chu, "P^{2}B-Trace: Privacy-preserving blockchain-based contact tracing to combat pandemics," in *Proc. Int. Conf. Manage. Data, Virtual Event (SIGMOD)*, China, G. Li, Z. Li, S. Idreos, and D. Srivastava, Eds., New York, NY, USA: ACM, Jun. 20–25, 2021, pp. 2389–2393.
[31] S. L. T. Vangipuram, S. P. Mohanty, and E. Kougianos, "CoviChain: A blockchain based framework for nonrepudiable contact tracing in healthcare cyber-physical systems during pandemic outbreaks," *SN Comput. Sci.*, vol. 2, no. 5, 2021, Art. no. 346.
[32] M. Zuhair, F. Patel, D. Navapara, P. Bhattacharya, and D. Saraswat, "BloCoV6: A blockchain-based 6G-assisted UAV contact tracing scheme for COVID-19 pandemic," in *Proc. 2nd Int. Conf. Intell. Eng. Manage. (ICIEM)*, Piscataway, NJ, USA: IEEE Press, 2021, pp. 271–276.
[33] M. Salimibeni, Z. Hajiakhondi-Meybodi, A. Mohammadi, and Y. Wang, "TB-ICT: A trustworthy blockchain-enabled system for indoor contact tracing in epidemic control," *IEEE Internet Things J.*, vol. 10, no. 7, pp. 5992–6017, Apr. 2023.
[34] T. Salman, R. Jain, and L. Gupta, "Probabilistic blockchains: A blockchain paradigm for collaborative decision-making," in *Proc. 9th IEEE Annu. Ubiquitous Comput., Electron. Mobile Commun. Conf. (UEMCON)*, Piscataway, NJ, USA: IEEE Press, 2018, pp. 457–465.
[35] A. Kotanen, M. Hannikainen, H. Leppakoski, and T. D. Hamalainen, "Experiments on local positioning with bluetooth," in *Proc. Int. Conf. Inf. Technol., Coding Comput. (ITCC)*, Piscataway, NJ, USA: IEEE Press, 2003, pp. 297–303.
[36] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, 1978.
[37] H. Gilbert and H. Handschuh, "Security analysis of SHA-256 and sisters," in *Proc. 10th Annu. Int. Workshop Sel. Areas Cryptogr. (SAC)*, Ottawa, Canada, Revised Papers, vol. 3006 (Lecture Notes in Computer Science), M. Matsui and R. J. Zuccherato, Eds., New York, NY, USA: Springer-Verlag, Aug. 14–15, 2003, pp. 175–193.
[38] D. Larimer, "Delegated proof-of-stake (DPoS)," Bitshare whitepaper, vol. 81, p. 85, 2014.
[39] N. C. Kakwani, "Applications of Lorenz curves in economic analysis," *Econometrica*, vol. 45, no. 3, pp. 719–727, 1977.
[40] R. Dorfman, "A formula for the Gini coefficient," *Rev. Econ. Statist.*, vol. 61, no.1, pp. 146–149, 1979.
[41] L. Reichert, S. Brack, and B. Scheuermann, "A survey of automatic contact tracing approaches using Bluetooth low energy," *ACM Trans. Comput. Healthcare*, vol. 2, no. 2, pp. 18:1–18:33, 2021.
[42] R. P. Brent and P. Zimmermann, *Modern Computer Arithmetic*, vol. 18. Cambridge, U.K.: Cambridge Univ. Press, 2010.

Xiao Li (Graduate Student Member, IEEE) received the B.S. and M.S. degrees in software engineering from Dalian University of Technology, Dalian, China, in 2016 and 2019, respectively. He is currently pursuing the Ph.D. degree in computer science with the Department of Computer Science, The University of Texas at Dallas, Richardson, TX, USA.

His current research interests include blockchain technology, distributed systems, data mining, distributed machine learning, and reinforcement learning.

Weili Wu (Senior Member, IEEE) received the M.S. and Ph.D. degrees both in computer science and engineering from the Department of Computer Science, University of Minnesota, Saint Paul, MN, USA, in 1998 and 2002, respectively.

She is currently a Full Professor with the Department of Computer Science, The University of Texas at Dallas, USA. Her research focuses on the design and analysis of algorithms for optimization problems that occur in wireless networking environments and various database systems.

Tiantian Chen received the B.S. degree in mathematics and applied mathematics and the M.S. degree in operational research and cybernetics from the Ocean University of China, Qingdao, China, in 2016 and 2019, respectively, and the Ph.D. degree in computer science from The University of Texas at Dallas, USA, in 2023.

Her research focuses on reinforcement learning, deep learning, social networks, blockchain, and design and analysis of approximation algorithms.