The University of Texas at Austin
Center for Identity

Self-Sovereign Identity and User Control for Privacy-Preserving Contact Tracing

Wenting Song
Razieh Nokhbeh Zaeem
David Liau
Kai Chih Chang
Michael R. Lamison
Manah M. Khalil
K. Suzanne Barber

UTCID Report #20-14

November 2020

Self-Sovereign Identity and User Control for Privacy-Preserving Contact Tracing

Abstract

Contact tracing apps use mobile devices to keep track of and promptly identify those who come in contact with an individual who tests positive for COVID-19. However, privacy is a major obstacle to the wide-spread use of such apps since users are concerned about sharing their contact and diagnosis data. This research overcomes multiple challenges facing contact tracing apps: (1) As researchers have pointed out, there is a need to balance contact tracing effectiveness with the amount of user identity and diagnosis information shared. (2) No matter what information the user chooses to share, the app should safeguard the privacy of user information. (3) On the other hand, some essential test result information must be shared for the contact tracing app to work. While contact tracing apps have done a good job maintaining contact information on the user's device, most such apps publish positive COVID-19 test results to a central server which have some risks for compromise. (4) Finally, following the spirit of privacy and in the absence of significant collection of user information, the app must innovate new methods to identify deliberate false reports of COVID-19. We address these challenges by (1) giving the user the right to choose how much information to share about their diagnosis and their identity, (2) building our novel contact tracing app on top of Self Sovereign Identity (SSI) to assure privacy preserving user authentication with verifiable credentials, (3) decentralizing the storage of COVID-19 test results, and (4) incorporating innovate fraud detection methods with limited user information. We, in collaboration with a top multi-national telecommunications corporation, have implemented our Privacy-preserving Contact Tracing (PpCT) app, leveraging Self-Sovereign Identity advances based on the blockchain for their 5G network.

Key Takeaways

- Privacy is a major obstacle to the wide-spread use of contact tracing apps (applications that use mobile devices to keep track of and promptly identify those who come in contact with an individual who tests positive for COVID-19). This research overcomes multiple challenges facing contact tracing apps.
- Our contract tracing app gives users the right to choose how much information to share about their diagnosis and their identity.
- We built our novel contact tracing app using a Self-Sovereign Identity (SSI) platform, to assure privacy-preserving user authentication with verifiable credentials.
- Storage of COVID-19 test results are decentralized in our app, which also incorporates innovate fraud detection methods using relatively little user information.
- In collaboration with Verizon, The Center for Identity has implemented our Privacy-preserving Contact Tracing (PpCT) app, leveraging Self-Sovereign Identity advances based on the blockchain for Verizon's 5G network.

Self Sovereign Identity and User Control for Privacy-Preserving Contact Tracing

Contact tracing apps use mobile devices to keep track of and promptly identify those who come in contact with an individual who tests positive for COVID-19. However, privacy is a major obstacle to the wide-spread use of such apps since users are concerned about sharing their contact and diagnosis data. This research overcomes multiple challenges facing contact tracing apps: (1) As researchers have pointed out, there is a need to balance contact tracing effectiveness with the amount of user identity and diagnosis information shared. (2) No matter what information the user chooses to share, the app should safeguard the privacy of user information. (3) On the other hand, some essential test result information must be shared for the contact tracing app to work. While contact tracing apps have done a good job maintaining contact information on the user's device, most such apps publish positive COVID-19 test results to a central server which have some risks for compromise. (4) Finally, following the spirit of privacy and in the absence of significant collection of user information, the app must innovate new methods to identify deliberate false reports of COVID-19. We address these challenges by (1) giving the user the right to choose how much information to share about their diagnosis and their identity, (2) building our novel contact tracing app on top of Self Sovereign Identity (SSI) to assure privacy preserving user authentication with verifiable credentials, (3) decentralizing the storage of COVID-19 test results, and (4) incorporating innovate fraud detection methods with limited user information. We, in collaboration with a top multi-national telecommunications corporation, have implemented our Privacy-preserving Contact Tracing (PpCT) app, leveraging Self-Sovereign Identity advances based on the blockchain for their 5G network.

CCS Concepts: • **Security and privacy** → **Software and application security**; • **Human and societal aspects of security and privacy** → *Privacy protections*; • **Software and application security** → Social network security and privacy; • **Security services** → Privacy-preserving protocols.

Additional Key Words and Phrases: contact tracing, privacy, self sovereign identity, fraud detection, application development, blockchain.

1 INTRODUCTION

The unprecedented COVID-19 pandemic has wreaked havoc on literally every aspect of life. As of November 2020, the coronavirus has infected over fifty seven million and killed over one million three hundred thousand people globally[1]. To curb the spread of this virus (named SARS-CoV-2), many governments have enforced stay-at-home orders and social (i.e., physical) distancing measures. These measures, however, drastically disrupt financial and social activities. The economic impact of COVID-19 is still unfolding, very much like its social impacts. In March 2020, the U.S. National Bureau of Economic Research estimated a year-on-year contraction in U.S. real GDP of 11% as of the fourth financial quarter of 2020 [8].

As societies seek to reopen after the stay-at-home orders eventually lift, effective contact-tracing of infected individuals is paramount. Contact-tracing involves identifying those who have come in close contact with an infected individual during the time he/she is potentially infectious with COVID-19, and notifying contacted individuals to take further actions such as getting tested, monitoring for symptoms, and self-quarantining. Mathematical models have demonstrated how "highly effective contact-tracing and case isolation is enough to control a new outbreak of COVID-19 within 3 months." [20].

A mobile app that automatically detects close contact with other individuals in real-time, keeps track of the contact lists, and proactively notifies contacts is shown to be considerably more effective than traditional reactive contact-tracing [17]. A major concern with such mobile apps, however,

[1] https://coronavirus.jhu.edu/map.html

Author's address:

is the potential privacy breach of user sensitive information. At a minimum, these apps need to record contacts of individuals and positive COVID-19 test results, but some solutions go as far as collecting exact [29] or approximate [31] geographical locations. Even with the minimal contact information, many have raised privacy concerns [11, 32] with respect to the government access to the data [7], potential for snooping, and lack of privacy from contacts [13].

A number of simultaneous attempts are being made to produce privacy-preserving contact tracing apps. Most of these apps rely on the Bluetooth/Bluetooth Low Energy (BLE) [4] signals to collect the contact list. Each device constantly broadcasts hashed identifiers over Bluetooth/BLE. When a user's device comes in contact with another, the app collects the hashed identifier of the other user, and keeps this contact information exclusively on the user's device. When a user tests positive for COVID-19, he/she voluntarily publishes their diagnosis and his/her identifier to a centralized server (typically government server) which in turn propagates the diagnosis to the other users. The other users match the hashed identifier of the infected user against their contact lists to determine if they have been in contact with the infected individual. The matching usually happens on the user's device. Researchers have proposed a range of privacy-preserving components into this high-level solution, including a public server to collect and propagate positive COVID-19 diagnosis (in lieu of a trusted third-party server) [12], a decentralized peer-to-peer system that eliminates the server altogether [10, 37], or a zero-knowledge proof approach [24]. Amid these solutions, some researchers have aimed to replace the server with blockchain technology to enable privacy-preserving contact-tracing [7, 36].

Yet, prominent privacy researchers have argued that [14] "digital contact tracing may protect privacy, but [without proper balance between privacy and contact tracing effusiveness] it is unlikely to stop the pandemic". While some researchers feel confident that privacy and contact tracing can go together [2], others view such apps as major risk to privacy [32]. To address this concern, we offer our **first contribution**: we give users the right to choose how much information to share with other users with no sharing to a centralized public authority. The user can share as much or as little identity and diagnosis data he/she wishes to share. We are unaware of any other contact tracing app that gives the user such wide range of options about what to share without decreasing contact tracing efficacy.

No matter how much the user decides to share, user information should be safeguarded. To provide the ultimate protection of user privacy, we make our **second contribution**. We utilize Self Sovereign Identity (SSI) built on top of blockchain, and available to 5G network users through our industrial partner in this research. Blockchain was first introduced with Bitcoin [28], a crypto-currency (i.e., electronic cash) technology that allows online transactions to take place without going through a trusted financial third party. Digital signatures and a peer-to-peer network form the backbone of the blockchain technology. The two parties of the transaction communicate through digital signatures (i.e., public and private keys). The peer-to-peer network timestamps transactions by hashing them into a chain of blocks, forming a record of transactions. This ledger (also known as blockchain) cannot be altered without the consensus of the network majority. **Self-sovereign identity (SSI)** is the concept of individuals or organizations having sole ownership of their digital and analog identities. Blockchain exhibits several properties that make it a suitable candidate for self-sovereign identity applications [16], including but not limited to distributed consensus, immutability and irreversibility of ledger state, distributed data control, accountability and transparency. Because of its support for self-sovereign identity, blockchain platforms have already been exploited to develop self-sovereign identity applications, such as uPort [25], Jolocom [15], Sovrin [30] and Blockcerts[33]. These applications are deployed at the top (application) layer where a blockchain platform resides underneath. Our industrial partner has a similar implementation of SSI on top of the Hyperledger blockchain.

While our use of SSI strengthens the privacy protection of user private information, there is an essential part of COVID-19 positive test results that must be available to other PpCT users for the app to function. Individuals on the contact list of a user who has tested positive for COVID-19 must be notified. Even this limited notification is under the user's control in our app. To the best of our knowledge, however, the publication of positive test results to a central server seems almost ubiquitous in contact tracing apps. Such central server is a single point of failure and may be compromised to flood the PpCT users with deliberately fabricated COVID-19 positive results or other formulations of inaccurate results. To resolve this issue, we take advantage of the blockchain network (in particular Hyperledger Fabric) as a distributed repository of positive test results, disassociated from user identities. As our **third contribution**, blockchain replaces the central repository of COVID-19 positive test results (hashed identifiers) and adds decentralization, immutability, transparency, and security. Note that the *public* Fabric ledger is used only for the part of test results that must be shared with others.

Finally, we introduce novel trust measures to serve as fraud detectors. In the absence of significant user information, recognizing trustworthy users from attackers that seek to report false diagnosis seems particularly daunting. Our **final contribution** is the definition of trust for PpCT users with limited information. It is notable that we keep even that limited information on user devices protected by SSI (i.e., in their SSI wallet).

We have implemented our PpCT for Android devices in collaboration with our industrial telecommunication corporation that funds this research [name redacted]. We are currently testing PpCT with college students at [name redacted]. It is also noteworthy that our Privacy-preserving Contact Tracing app is transferable for tracing any other infectious disease.

2 PRELIMINARY

2.1 Privacy Concerns in Traditional Contact Tracing Applications

There are lots of important issues waiting to be addressed for a traditional contact tracing app. Among these are privacy and security concerns of users' information: Contact tracing apps involve the storage of users' contact data, associated with users' interaction history log for a pre-set duration (usually set as two weeks). The collection of this information imposes possible threats to user privacy as the log data may disclose users' private information such as identities, locations, trajectories, and lifestyle-related information. A contact tracing app aggravates the privacy problem even further because it may disclose users' health and diagnosis information. Such threats to privacy would discourage people from participating in contact tracing through apps.

2.2 Our Design for Privacy Preserving Contact Tracing

To ensure the maximum possible privacy of users, and at the same time assure the proper functioning of the contact tracing app, we make the following attempts in our privacy-preserving contact tracing application:

(1) We make contact tracing happen strictly through Bluetooth Low Energy (BLE) beaconing. User location information is never recorded or stored.
(2) BLE Identifiers change every fifteen minutes, which reduces the risk of privacy exposure during the process of broadcasting identifiers and collecting contacts.
(3) The participation in each functionality of the app requires users' explicit consent. Users decide to opt into registration, contact collection, self-reporting, and notification. In line with this privacy preserving design decision, we allow users to self-report instead of having third party COVID testing labs report on their behalf.

(4) Anonymization. The basic idea is to remove identification information from all interactions between users, and of course between the user and application. One way to make users anonymous is to replace their identifiable information with pseudonyms or suppressing users' identities.

(5) The user selects the extent to which they wish to share their identity through self-reports and/or add identity verification from an organization who knows them. Users may have third party organizations attest their identities.

(6) All user authentication information is secured with SSI and saved in their wallet on the device. User private key for SSI is generated on the device and never leaves it.

(7) User trustworthiness measurement calculations are processed exclusively on the device. The original intention of the measurement is fraud detection in order to stop malicious users' false reporting attacks that may cause a huge panic among the users.

(8) The trustworthiness score is protected and secured as one of the identity owner's personal identifiable information. The score is treated as another form of identity information and saved in the SSI wallet for privacy. The use of the measurement score should satisfy all the guiding principles of self-sovereign identity. User reporting trustworthiness and user privacy preservation are two conflicting objectives in a privacy-preserving contact tracing application. A strict mechanism to preserve privacy may influence user reporting trustworthiness. Nevertheless, The method to ensure the user's trustworthiness imply collecting users' personal data for verification or authentication, which is against the preserving privacy purpose. Therefore, a trade-off between user information collection (albeit securely saved on user device with their SSI wallet) and detecting fraud is necessary.

(9) Persons in a user's contact list are notified the user's test results have been verified if the user is willing to provide their positive test report, but report contents are never shared outside the PpCT app.

(10) Contact information never leaves user devices and contact matching takes place locally on their devices.

3 SELF-SOVEREIGN IDENTITY (SSI)

Self-sovereign identity, is the concept of individuals or organizations having sole ownership of their digital and analog identities. SSI adds a layer of security and flexibility allowing the identity holder to only reveal the necessary data for any given transaction or interaction. Allen proposed his ten guiding principles of SSI [3], which are summarized in Table. 1.

3.1 Background: Flow of Typical SSI

Hyperledger Indy combined with Aries is an example of practical implementation of the SSI concept. While Indy itself is a form of SSI implementation already, Aries provides the flexibility and additional functionality in practical software development across different platforms.

Figure 1 displays the high-level workflow of SSI. In addition, a typical Indy SSI example adds the concept of derived credentials in addition to root credentials as follows: At the beginning, an individual user has root credentials like one's names or driver's licence that are issued and documented by trusted government agencies. An entity like a local bank or a university can create SSI for the user to use in future applications by creating a derived credential with the root credentials. For example, a university can create a diploma (derived credential) with one's name (root credential) and the information is stored in its database. In this case, the university offers and the user accepts the diploma. By accepting this offer, the user utilizes root credentials to create a derived credential. Once the university gets that information as a whole, it fills in all the other information it had for the diploma and sends the completed SSI back to the user. The user now stores this completed SSI

Table 1. SSI can be defined as ten guiding principles, which was categorized into three sections: Security, Controllability and Portability[35], [16]. Transparency and Access are solely from the identity owner's perspective and ensure that identity owners can utilize their own identities anywhere and anytime.

Security	Controllability	Portability
Keep identity information secure.	Users must take control of their data, e.g who can see or access.	Users can utilize their identities wherever they want, without being tied to any provider.
Protection	Existence	Interoperability
Persistence	Persistence	Transparency
Minimisation	Control	Access
	Consent	

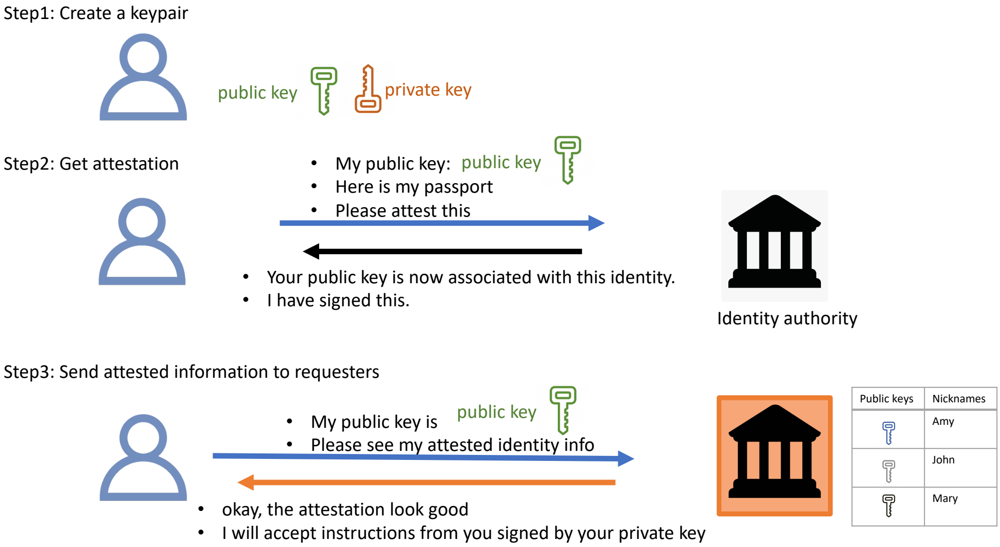

Step1: Create a keypair
public key
private key
Step2: Get attestation
• My public key: public key
• Here is my passport
• Please attest this
• Your public key is now associated with this identity.
• I have signed this.
Identity authority
Step3: Send attested information to requesters
• My public key is public key
• Please see my attested identity info
• okay, the attestation look good
• I will accept instructions from you signed by your private key
Public keys
Nicknames
Amy
John
Mary

Fig. 1. Flow of SSI [5].

in their wallet on their device and use it in the future without a typical authentication process that has to involve the university. This process would greatly increase user controllability of the identity because the technology enables an authentication process that minimizes the involvement of administrative authorities.

4 HIGH-LEVEL USE CASE OF PPCT

In this section, we explain the high-level use case of our PpCT app. Inspired by SSI, we provide four distinct levels of authentication. The user may choose any level to authenticate himself/herself without exposing the identity document used outside the secure SSI wallet. In the remainder of

Table 2. Four SSI layers.

	Technology	Purpose	Examples	Thoughts
Level 0	Zero Knowledge Proof Scene	Achieve proof of asset or identification without exchange of the identity itself	Zero-Knowledge Password Proof (ZKPP) (standardized as part of IEEE IEEE 1363.2)	Needs sophisticated design depending on the application scenario
Level 1	General Blockchain Technology without Know Your Customer (KYC)	A virtual identity which is not related to a limit of real world identity is established within some systems	Hyperledger Fabric enabled blockchain, Ethereum	Usually requires additional resource to maintain the system
Level 2	Private Blockchains, Hyperledger Indy	Achieve self- sovereign identity in this level of Trust	College Certificate, Loan Application, Work Pre-screening	Self-sovereign Identity can be achieved but privacy preserving needs further discussion
Level 3	Traditional authentication/authorization			

this section, we select one sample level of assurance from these four levels and explain our PpCT functions in more details.

4.1 Four Privacy Layers of SSI

Four privacy layers of SSI allow users to share as much as they desire about their identities.

- Level 0: No sharing
- Level 1: I own an identity defined by a public/private key pair
- Level 2: I own an identity known to a third party/organization I trust
- Level 3: Individual Personal Identifiable Information sharing

Table 2 elaborates on these four levels of assurance.

For the sake of brevity, we do not extensively explain all the four layers. Rather, we select layer two and dive into details of its implementation.

4.2 Overarching Flow

We display the swim-lane diagram of the entire PpCT at level two of authentication in Figure 2. The overarching flow of the swim-lane diagram can be divided into six parts: Registration, Contact Collection Setting, Contact Collection, Contact Notification Setting, COVID Test Reporting, and Contact Notification.

Procedure 1: Registration.

John and Mary, two users of PpCT, both have PpCT installed on their phones. In the registration phase, they each provide identity information in accordance with Section 3.1 to have their identities attested and verified. Similar to a typical SSI application, their identity information and their private keys are stored in their SSI wallets, solely on their devices, and never leave the device. The figure

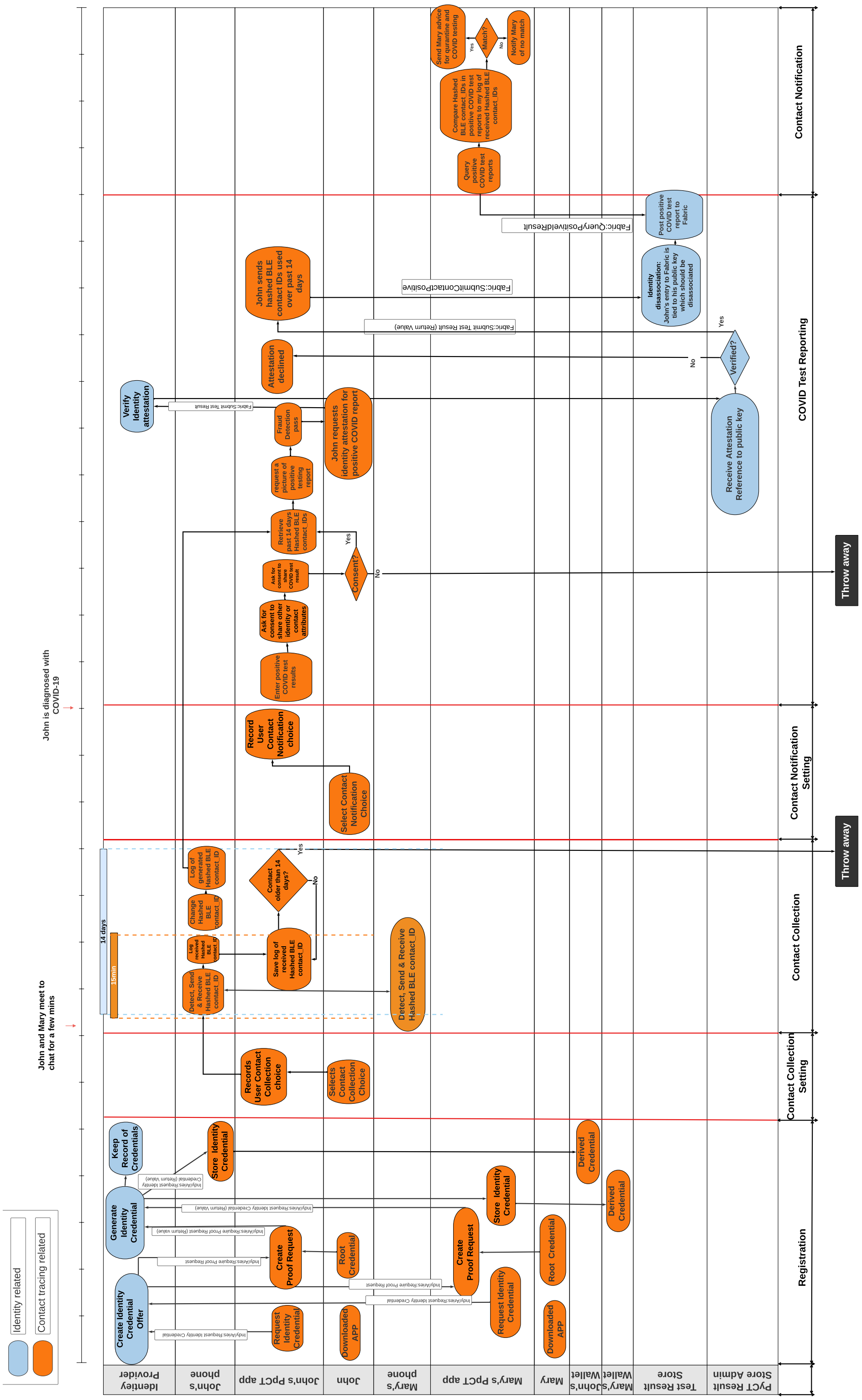
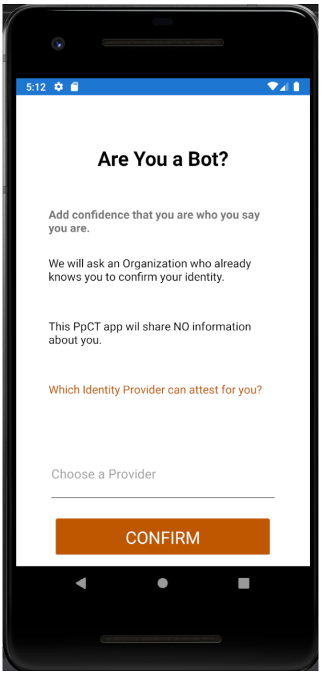

Identity related
Contact tracing related
Identity Provider
John's phone
John's PpCT app
John
Mary's phone
Mary's PpCT app
Mary
John's Wallet
Mary's Wallet
Test Result Store
PpCT Result Store Admin
Registration
Contact Collection Setting
Contact Collection
Contact Notification Setting
COVID Test Reporting
Contact Notification
John and Mary meet to chat for a few mins
John is diagnosed with COVID-19
Create Identity Credential Offer
Generate Identity Credential
Keep Record of Credentials
Request Identity Credential
Downloaded APP
Create Proof Request
Root Credential
Store Identity Credential
Derived Credential
Indy/Aries:Request Identity Credential
Indy/Aries:Require Proof Request
Indy/Aries:Require Proof Request (Return Value)
Indy/Aries:Request Identity Credential (Return Value)
Selects Contact Collection Choice
Records User Contact Collection choice
Detect, Send & Receive Hashed BLE contact_ID
Log received Hashed BLE contact_ID
Save log of received Hashed BLE contact_ID
Change Hashed BLE contact_ID
Log of generated Hashed BLE contact_ID
Contact older than 14 days?
Yes
No
15min
14 days
Throw away
Select Contact Notification Choice
Record User Contact Notification choice
Enter positive COVID test results
Ask for consent to share other identity or contact attributes
Ask for consent to share COVID test result
Consent?
Retrieve past 14 days Hashed BLE contact_IDs
request a picture of positive testing report
Fraud Detection pass
John requests identity attestation for positive COVID report
Verify Identity attestation
Fabric:Submit Test Result
Receive Attestation Reference to public key
Verified?
Attestation declined
Fabric:Submit Test Result (Return Value)
John sends hashed BLE contact IDs used over past 14 days
Fabric:SubmitContactPositive
Identity disassociation: John's entry to Fabric is tied to his public key which should be disassociated
Post positive COVID test report to Fabric
Fabric:QueryPositiveResult
Query positive COVID test reports
Compare Hashed BLE contact_IDs in positive COVID test reports to my log of received Hashed BLE contact_IDs
Match?
Send Mary advice for qurantine and COVID testing
Notify Mary of no match

Fig. 2. Swim lane diagram.



Are You a Bot?

Add confidence that you are who you say you are.

We will ask an Organization who already knows you to confirm your identity.

This PpCT app wil share NO information about you.

Which Identity Provider can attest for you?

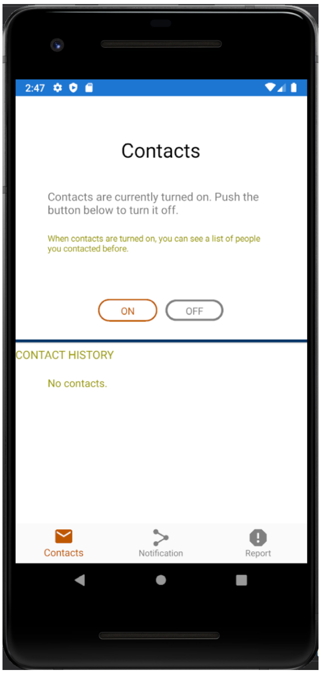
Fig. 3. The screenshot of the identity provider page.

includes methods provided by the SSI API that PpCT calls in white boxes. First, the PpCT app sends Request Identity Credential to the identity provider. The identity provider follows up by Creating Identity Credential Offer and sending back to the user to accept. Once the user accepts this offer, the identity provider generates the verified Identity Credential, sends it back to the user for future use and also stores the Identity Credential at the provider side. John's PpCT app (also Mary's) receives Unique Identity Credential and the keys are stored in the wallet.

Figure 3 shows the screenshot of PpCT that implements registration. Users are allowed to pick an identity provider in the page and then press the confirm button to submit their choice. If the user selects no identity provider, they remain at level 0. If they choose an identity provider, the user's identity is verified at level 1. If the user further has a third party organization attest their identities through single sign on (SSO), they will be verified at level 2. If the user provides the full identity, they are verified at level 3.

Procedure 2: Contact Collection Setting.
In PpCT, the user's explicit consent is asked every step of the way. Users can decide whether or not to give the PpCT app permission to broadcast BLE signals (Contact Collection Setting of the swim lane) on the PpCT Contacts page. Only when contacts are turned on, they could see the list of people they have been in contact before, and at the same time, they themselves will show up in other users' contacts correspondingly.

Procedure 3: Contact Collection.
As shown in Figure 2, when John and Mary meet to chat for a few minutes, their smartphones exchange hashed (anonymous) identifiers over Bluetooth to register that they have been in contact.



Contacts

Contacts are currently turned on. Push the button below to turn it off.

When contacts are turned on, you can see a list of people you contacted before.

CONTACT HISTORY

No contacts.

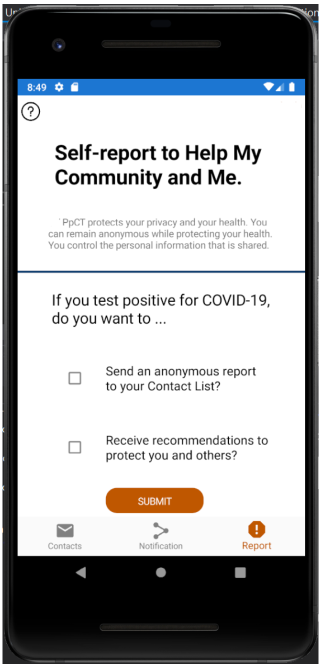
Fig. 4. The Contacts tab of PpCT app with the switch turned on.

To avoid the possibility of associating an identifier with a device, these identifiers change every 15 minutes and remain resident on user phones. Each contact is saved for a preset number of days for which the viral disease is believed to be infectious—in the case of COVID-19 for 14 days. Figure 4 is the screenshot of the Contacts page. The past contacts will show in the contact history with hashed IDs and time stamps.

Procedure 4: Contact Notification Setting.
Again, the user's consent is requested in the settings of contact notification. We show a screenshot of this page after reviewing the reporting process.

Procedure 5: COVID Test Reporting.
When a user (John) is diagnosed with COVID-19, he voluntarily enters his positive COVID-19 test results into the PpCT app. Then, PpCT asks for his informed consent on the PpCT app Report tab. If John grants consent, our app will bring John to the next two steps to finish the notification procedure. Users can absolutely decide not to share anything in our app. In that case, the user receives links to information on COVID-19. However, the main purpose of PpCT is achieved when the user decides to share parts of his diagnosis (Figure 5).

With John's approval, the app will lead him to the other page of the Report Tab (Figure 6) to select the information to attest—the test result from the test agency or trust scores from PpCT, and the personal information to share (Name, Contact duration or Contact time). For example, if John selects to share contact duration and contact time (only month) and append his Trust Score as attestation, anonymous and time-stamped notifications stating the following will be sent out to users on his contact log list, including Mary: "You came into contact with a COVID-19 positive individual in June. The contact lasted for one hour and the trust score of the individual



8:49

Self-report to Help My Community and Me.

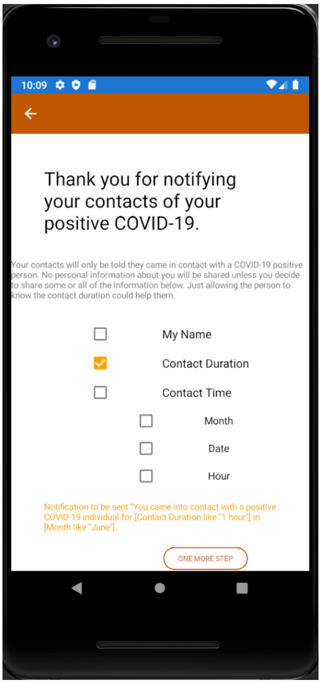
PpCT protects your privacy and your health. You can remain anonymous while protecting your health. You control the personal information that is shared.

If you test positive for COVID-19, do you want to ...

□ Send an anonymous report to your Contact List?

□ Receive recommendations to protect you and others?

Fig. 5. The Report tab of PpCT app.



10:09

Thank you for notifying your contacts of your positive COVID-19.

Your contacts will only be told they came in contact with a COVID-19 positive person. No personal information about you will be shared unless you decide to share some or all of the information below. Just allowing the person to know the contact duration could help them.

□ My Name

☑ Contact Duration

□ Contact Time

□ Month

□ Date

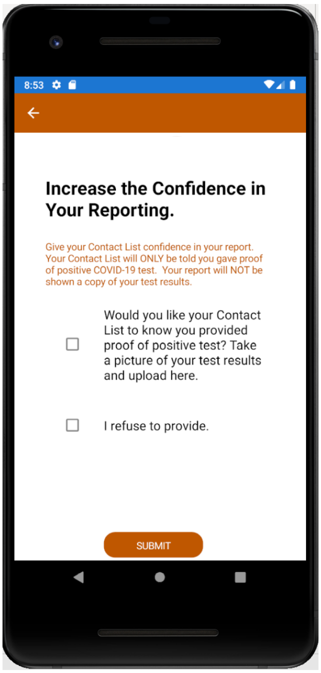
□ Hour

Notification to be sent "You came into contact with a positive COVID-19 individual for [Contact Duration like "1 hour"] in [Month like "June"].

Fig. 6. The information selection page of PpCT app.

(the likelihood that they truthfully reported a positive test result) is Y."

In figure 7, users can choose to pick a picture of positive COVID-19 test results in their album and then provide it to PpCT.



Increase the Confidence in Your Reporting.

Give your Contact List confidence in your report. Your Contact List will ONLY be told you gave proof of positive COVID-19 test. Your report will NOT be shown a copy of your test results.

☐ Would you like your Contact List to know you provided proof of positive test? Take a picture of your test results and upload here.

☐ I refuse to provide.

Fig. 7. The information selection page of PpCT app.

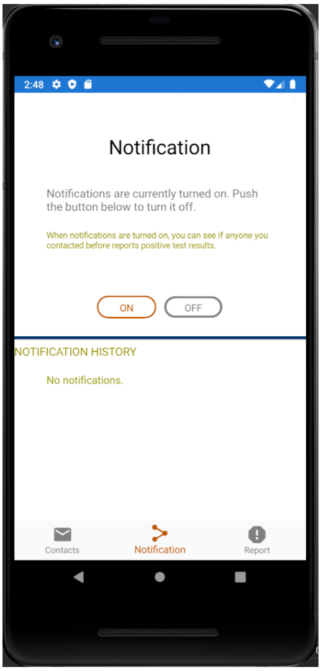
Note the nuances of test reporting in Figure 2. Once the user grants permission, the hashed BLE IDs from his past 14 days are retrieved from his device. If the user is still considered contagious according to the timing of test results, his future hashed BLE IDs will also be periodically retrieved. PpCT then asks for a picture of positive test results (if the user is willing to share one) and calculates trust scores for fraud detection on the user device. Then, John may decide to have his identity attested with the identity provider, without actually sharing his identity. Depending on his chosen level of assurance (Section 4.1), his report to the distributed PpCT Result Store is attested and automatically verified by the PpCT Result Store Admin. Then John's hashed BLE IDs are saved to Hyperledger Fabric's ledger—publicly available to other PpCT users.

Procedure 6: Contact Notification.

Mary's PpCT app periodically queries positive COVID test reports, and compares Hashed BLE contact IDs in positive COVID test reports to her log of received Hashed BLE contact IDs. If there is a match, the app will send Mary advice for self-quarantine and COVID testing. Hence, Mary receives notifications sent by John if she has turned on the notifications in PpCT. Figure 8 shows the screenshot of the notification tab. Past notifications will show under the notification history.

5 THE TECHNOLOGICAL STACK

In this section, we cover various technologies we use for the development of the PpCT mobile app. Fig 9. gives a top level description of how different technologies interact with one another.



Notification

Notifications are currently turned on. Push the button below to turn it off.

When notifications are turned on, you can see if anyone you contacted before reports positive test results.

NOTIFICATION HISTORY

No notifications.

Fig. 8. The Notification tab of PpCT app with the switch turned on.

Although Google/Apple have released a contact tracing library [1] that utilizes the same technology as this work, the Google/Apple API is available to health authorities only. As a result, we build our own tracking libraries from scratch.

5.1 Bluetooth Low Energy

We leverage Bluetooth Low Energy (BLE), a short-range radio communication standard that uses less transmission power than normal Bluetooth to minimize its impact on battery consumption. The technology utilized the same ISM band of 2.4GHz as many of the wireless communication protocols sit in, while giving 40 physical channel with separation of 2 Mhz. Of the 40 physical channels, 3 are dedicated for BLE advertisement and 37 are for data. In our application, we do not establish bi-directional data link with the BLE protocol but rather advertise one-sided the device identifier generated in our app periodically. It is worth noting that the contact tracing functionality provided by the Google/Apple's Exposure Notification system also utilize this technology to achieve the goal. A typical detection of contact would looks like the following: John's phone is periodically advertising its own device identifier. Mary's phone would be scanning for such identifiers with a certain power measurement and records the time of receiving. We consider contact duration and tune for contact distance so that PpCT can evaluate different risk levels of contact with the information provided (e.g., 15 minute contact within 6 feet distance). After a pre-set period of time, John's phone will generate a new identifier for it to be detected. Note that even though Mary's phone received the first and second identifier from John's phone, the system is designed so that it cannot determine whether these two identifiers are from the same phone. This feature was enabled by the Media Access Control (MAC) address randomisation introduced in the BLE standard. In order to lower the risk of being tracked by other devices, the MAC address and the device identifier shall be changed at the same time. Another feature of BLE protocols is that it can be configured to

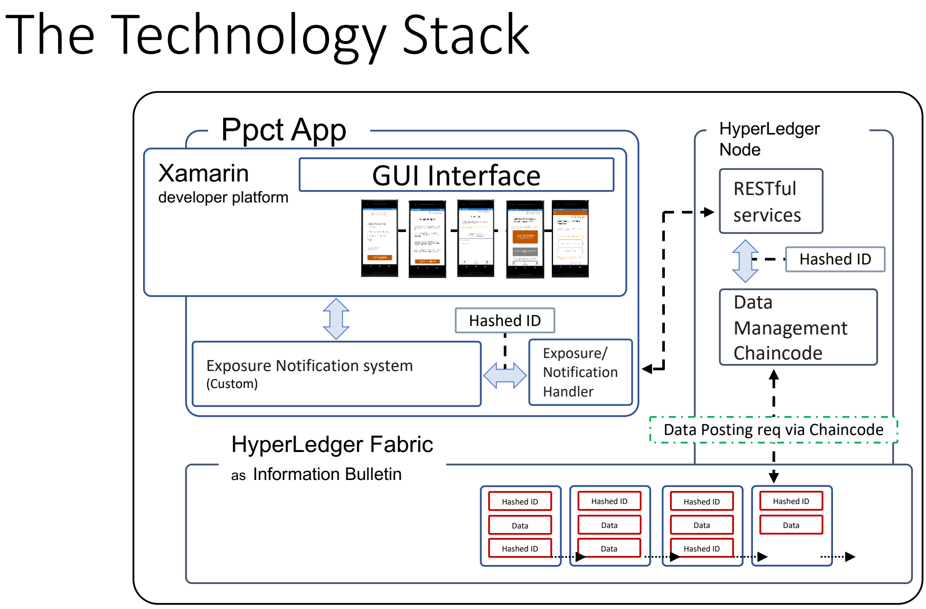


Fig. 9. The Technology Stack of PpCT.

be extra power saving than many other communication protocols. Therefore, even if our PpCT app requires users to turn on Bluetooth the entire time, their batteries still last long.

5.2 App Development

We use Xamarin for the implementation of the mobile app and XMPP as our extensible messaging and presence protocol used for streaming XML elements between device and the cloud. Signal Protocol is applied for cryptography and Amazon Web Services (AWS) for deployment. To allow our PpCT app to run in the background even if the app exits or the phone restarts, we use WorkManager which is also a battery-friendly API like BLE. To perform responsive UI design, we use ViewModel, DataBinding, and LiveData so that our PpCT app can adapt to different sizes of screen and these APIs also make it easier to manage UI when data in the background has changed.

5.3 Implementation of Self Sovereign Identity

We choose the HyperLedger Indy combined with Aries as the tool to implement the SSI identity management system. The app starts with establishing derived credentials that are being used in the app from root credentials. Root credentials are usually credentials that are strongly trusted by the app designer. For instance, since we have a strong understanding of how the Single Signed On (SSO) system works at our organization, the membership of the the organization can be considered a valid root credential to create a derived credential that will be used in future authentication processes. The advantage of this method is that once derived, the SSI is fully controllable by the user. Only the identity owner has the ability to initiate an authentication or authorization process.

5.4 Public Ledger as Information Bulletin

Even though a main important design concept of our PpCT app is privacy, the hashed BLE IDs voluntarily shared by PpCT users have to be accessible to other users for the app to function properly. In our design, we emphasize the need for designing an entity for storing the proper amount of information for exposure notification. In our case, this entity should not store information that can be directly linked to the identity of app users but should still enable users to know if they have been exposed to COVID-19. We choose to implement the entity utilizing the HyperLedger technology, which enables us to build a decentralized ledger with extra power of smart contracts to perform crucial privacy preserving operations in a decentralized way which can be modified after deployment.

HyperLedger is one of the blockchain projects that has gained popularity over the past few years since launching in 2015. Within the maturing ecosystem of HyperLedger, HyperLedger Indy is chosen for identity management and credential distributing. Indy project is a ledger project that specialized in identity management by providing digital identities rooted on blockchain. Once a credential is issued with Indy, users have full control over that credential comparing to other common digital identities. For development flexibility, we also adopt Hyperledger Aries for cross system identity support. Last but not the least, a distributed ledger supported by Hyperledger Fabric is also adopted for the purpose of securely storing the information needed within the design scope. Since distributed ledger empowered by Hyperledger Fabric also supports smart contracts, more higher level functionalities can be developed and added.

We want to emphasize that our app can operate the notification functionality without storing information that relates to the real world identity of the user, but information that may be traced back to the user can still be processed with the app user's consent to support higher level functionalities such as the calculation of trust scores. Note that even in the scenario that the user is posting data with derived credentials, the user is the one initiating the authentication process. Once the credential is used for establishing some level of trust, the information will not be stored at any entity. In this case, a one-time posting key is generated for the user to post the data he/she would like to share. By design, the user's privacy is more protected and enhanced compared to transitional peer-to-peer systems.

We reiterate that even though the blockchain ledger that replaces the central server of COVID-19 positive test results is public, what is saved is hashed anonymous identifiers that would not be possible to map back to identify devices. However, blockchain protects PpCT from a single point of failure and adds decentralization and immutability.

6 FRAUD DETECTION PROBLEMS

One issue of a privacy-preserving contact-tracing app is the reliability of users' COVID test result reports and provided image proofs. As data are reported by users out of voluntary, they could possibly be falsified. The worst cases are malicious users, who may provide numerous false reports. If these false reports are left undetected, they could cause panic among the users during the COVID-19 pandemic.

Our PpCT opted to not directly receive COVID-19 test results from COVID-19 testing labs because we would like to empower users by allowing them to share as much or as little as they wish. As a result, PpCT requires innovative methods to detect false and misleading COVID-19 test reporting by untrustworthy users. SSI plays an essential role to detect false reporting by authenticating users, if the user consents to such authentication. Nonetheless, authentication is voluntary and we enhance our fraud detection further with methods explained in this section.

The trustworthiness issue of user-provided data inherently conflicts with the user privacy preserving issue. The conflicts lie in that those falsified or even fabricated reports could not be detected or eliminated, if users' identities or personal information are not disclosed at all. In other words, if full anonymity is provided to application users, guaranteeing the trustworthiness of reported data is almost impossible. Therefore, a trade-off between guaranteeing user privacy and detecting fraud users is necessary. Our solution to this problem is to evaluate the trustworthiness of users' report. By evaluating their trustworthiness and detecting those fraud cases, the app can know the limit of reports provided by users. We ask for user consent and use limited information solely kept on the user device (i.e., SSI wallet) for trust score calculation.

6.1 Trust Score Kept in the SSI Wallet for Privacy Protection

To dispel users' privacy concerns, we make the trustworthiness score another form of identity information, which is saved in the SSI wallet for privacy, because user trustworthiness calculation may disclose users' private information. All the fraud detection related measurement is calculated on the user device, kept at the user side, and is protected and secured as one of the identity owner's personal identifiable information. Personal identifiable information are those personal information usually used to identify a particular person, such as a full name, email address, social security number, driver's license, bank account number, or passport number. The measurement result will only serve as a proof provided to the users as an option requiring the user's explicit concept. The use of the trustworthiness score should satisfy all the guiding principles of self-sovereign identity.

6.2 Fraud Detection Attempt - User Editing Trustworthiness

6.2.1 ***Self-reporting preference settings.*** To ensure user privacy, we build user trustworthiness with minimal personal identity information. One option that works both for user personal identifiable information privacy and reporting data trustworthiness is to track users' editing process during self-reporting. Here are the choices users would be asked to make under the report section, according to their personal preferences:

(1) Do you consent to report yourself as COVID-19 positive?
(2) Are you willing to share your anonymous contact list?
(3) Do you need more advice on what to do next?
(4) What personal information you would like to share in the notifications sent to your contacts?
   - My Name
   - Contact Duration
   - Contact Time (Month only)
   - Contact Time (Month and Date)
   - Contact Time (Month, Date and Hour)

We add more questions for Editing Trustworthiness calculation:

(1) How many tests have you taken? how many of them are negative and how many are positive?
(2) Any COVID-19 symptoms, like a cough or fever?
(3) Have you taken any self-quarantining actions?
(4) Do you need our help with hospital information or advice from physicians?

6.2.2 ***Atomic editing types.*** Similarly to the work illustrated in [21], we build up our editing trustworthiness measurements based on a sequence of basic editing operations which is expressible as the following four atomic editing types:

Initialization. If a user just starts to use the app and then he decides to self-report as tested positive when finding the report tab for the first time, he will be treated as trustworthy for his initial editing. In other words, normal users are considered trustworthy on self-reporting by default.

For example, John who volunteers to report by walking through the whole process step by step, implicitly confirms the trustworthiness of his report.

Originality. If a user tends to keep the originality of his report since clicking "Finish the report", the moment when he completes all the report contents, all his edits during the report process will be given a high trustworthiness score. Thence, his trustworthiness score will be quite high.

Modifications. Because changes to the simple preference choices may reveal his intention to forge some false facts, for those actions which intend to modify the answer to the same question too frequently, it will be assigned a lower trustworthiness value. The more frequent the changes are made to those simple questions, the lower the trustworthiness score will be.

Rollbacks. Modifications should intuitively decrease a user's reputation if he reverts a preference setting to a state before. There are special cases remain to be taken into consideration as users may make some edits inadvertently. There is no doubt that the user's trustworthiness score should not decrease for the edits to fix his unintended mistakes.

6.2.3 *User Editing Trustworthiness* $T_e(u,t)$.

We denote user editing trustworthiness by $T_e(u,t)$, where the trustworthiness score $T_e(u,t)$ is a function of both user u and time t, that is the trustworthiness score of user u at time t . The trustworthiness of each atomic editing is denoted as A_i, where i is the i^{th} edits that were made by a given user.

Following the way in [26], we bound the values of both user editing trustworthiness $T_e(u,t)$ and single editing trustworthiness A_i between and 0 and 1, that is $0 \leq T_e(u,t) \leq 1$ and $0 \leq A_i \leq 1$.

User editing trustworthiness $T_e(u,t)$ depends on the trustworthiness of all editing he has produced and is defined as the average of such values:

$$T_e(u,t) = \frac{\sum_{A_i \in F(u,t)} A_i}{|F(u,t)|} \tag{1}$$

where $F(u,t)$is the set of all the edits made by user u until time t.

6.3 Fraud Detection Attempt - User App Liveness Trustworthiness based on Anomaly Detection

Anomaly detection [22] has been widely studied and applied in a variety of domains, one of which is fraud detection. Anomaly detection detects the occurrence of anything unusual with statistical models, considering unexpected behaviors could be the sign of an attack happening. It is useful as it can detect a threat without any explicitly defined signature or specific standards. This allows us to identify the so-called "zero day attacks".

Inspired by [34], we try to distinguish from the mode of user utilizing PpCT application if there are any unusual performance from the application side, which could be a sign of potential attacks. To be more specific, we measure users' trustworthiness level in self-reporting from the normality of users' application liveness level.

6.3.1 *Application Liveness Normality*.

The basic idea is to detect the abnormal changes of users' daily activity through their **application liveness**—i.e., the period of time that the PpCT app is live and in use. Any normal changes in the activity scope are totally acceptable and the threshold will be definitely different for different users, considering different people have different lifestyle and application habits.

We try to detect the anomaly of users' liveness mode change by calculating the similarity score of users' application liveness level between different time duration dur_p, dur_q of same length k. The time duration k can be an hour, a day, or a week, and thus $dur_p = t_p - t_{p-k}$, $dur_q = t_q - t_{q-k}$, where t_p, t_{p-k}, t_q, t_{q-k} are the time point. The parameters are designed to be adjustable based on the user types and application scenarios we will be facing.

When calculating the liveness similarity, we aim to obtain the liveness of users in the contact tracing application. Liveness of a user can be measured by many aspects, like the number of tweets a user posted if it is for Twitter in time duration dur_p. Likewise in the contact tracing app case, we choose the following several criteria $\eta_n(u, dur_p)$ as measurements whether users are active. η_n denotes the n^{th} measurement number for user u during the time duration dur_p.

- Number of repeated checks of his/her own contacts history.
- Time duration the contact tracing app is kept open.
- Number of close contacts.
- Number of checks on the COVID-19 information and self-quarantine suggestions.
- Time duration through which Bluetooth is turned on.

Similarly as the editing trust measurement in section 6.2, the values of single trustworthiness $\eta_n(u, dur_p)$ are bounded between 0 and 1, that is $0 \leq \eta_n(u, dur_p) \leq 1$.

When the liveness difference is smaller than a pre-set threshold ϵ, trustworthiness is set as $1 - \frac{|\eta_n(u,dur_p)-\eta_n(u,dur_q)|}{\epsilon}$ and trustworthiness is set as 0, if not:

$$L_n(dur_p, dur_q) = \begin{cases} 1 - \frac{|\eta_n(u,dur_p)-\eta_n(u,dur_q)|}{\epsilon} & \left|\eta_n(u, dur_p) - \eta_n(u, dur_q)\right| \geq \epsilon \\ 0 & \left|\eta_n(u, dur_p) - \eta_n(u, dur_q)\right| < \epsilon \end{cases} \tag{2}$$

where $L_n(dur_p, dur_q)$ represents the user application liveness similarity for the n^{th} measure η_n between time duration dur_p and dur_q. One point to clarify, the similarity score is not a measurement for trustworthiness, while the dataset including all the similarity scores will serve as a basis for judgment in anomaly detection.

Again, all the calculation and the whole dataset are kept at the user application side, and will never be uploaded from device. Final assessment outcomes will be kept, protected and secured as one of the identity owner's personal identifiable information in users' SSI wallets.

6.3.2 ***User Activity Anomaly Detection Trustworthiness*** $T_a(u, t)$. To measure users' activity trustworthiness from the perspective of application use, we take advantage of anomaly detection to detect any abnormal behaviors happen, on the premise that any unexpected user behavior could be the sign of an attack or a malicious user fraud. The methodology behind anomaly detection is to test a new data sample against the data samples history, which is always useful to detect anything abnormal happening. To apply in our scenario, the detection pertains to investigating the user's current behavior and measuring how much the difference is between the current behavior and the accumulating past behaviors. One of the most common algorithms to apply is the Gaussian distribution algorithm [22] which implies that a normal sample x is distributed with a mean μ and variance σ^2.

6.4 Image Proofs Provided Verification Trustworthiness

In the procedure of COVID Test Reporting, there is a step in which we recommend users who volunteer to self-report provide image proofs for credibility attestation. Considering user trustworthiness measurement and user privacy preservation are two conflicting objectives in a privacy-preserving contact tracing application. One way to measure the trustworthiness without revealing users' private information is to measure the trustworthiness of user-generated content instead of users themselves. In other words, the measure target shall be moved from user trustworthiness to user-generated content trustworthiness, through tracking content, not people.

The general idea is to decouple the contents generated by users and users' personal information, especially those private identity-related information, while as a substitute, to couple the content with a spatial timestamp representing a system-verified time or location. This approach preserves the privacy of users by not exposing their identity information to any potential attackers. A wide range of research [23], [19] has been conducted on how to establish some trust in the authenticity of user-generated content, assuming the information were reported or provided by unknown individuals or even un-trusted users. Paper [23] proposed a localization certification method where users could tag their contents with a spatial timestamp to increase credibility.

Inspired by this idea, we apply the model [19] as our prototype and add more features to verify the authenticity and credibility of the image proofs provided by users. To self-report and send notifications, users are recommended to provide an image proof for testing positive. With these user provided contents, we verify the trustworthiness of users' self-reporting by verifying image timestamps, considering user generated content is more valuable and trustworthy when its spatial and temporal properties can be verified. **If the user provides an image that is digitally altered, the image proof is considered untrustworthy.**

Again, all the provided images are kept at the user application side and the verification is conducted on the device. The verification result will be kept at the user side, protected and secured as one of the identity owner's personal identifiable information in identity owner's SSI wallet. Only the final trust score is provided to other users, with the provider's explicit consent.

6.5 Fraud Detection Attempt - Binomial Reputation (To Combine Different Measurements Above)

Calculating the above trustworthiness measurements, (1) user editing trustworthiness $T_e(u, t)$ (section 6.2), (2) user activity normality trustworthiness $T_a(u, t)$ (section 6.3)and (3) image proofs verification trustworthiness (section 6.4), helps the application distinguish normal users from malicious ones, and consequently make decisions whether to trust the COVID-19 positive test report provided by users and spread in the form of sending notifications.

It is worth emphasizing that if PpCT detects one user with any abnormal behavior from any trustworthiness perspective, it does not mean that this user would be evaluated as a malicious user or his report would be discarded. For any specific measurements, low trustworthiness score does not directly lead to a judgment for users. The final conclusion will be made after a thorough evaluation, taking into account all the measurements, which therefrom enhance the robustness of the trustworthiness evaluation system.

In this part we demonstrate our method to identify the malicious users based on the measurements proposed above. In paper [18] , the authors proposed a reputation management mechanism called binomial reputation. Similarly in our application scenario, each user will only be classified as a normal user or a malicious user (which can be trusted). The binomial distribution helps compute the probability of a user being credible:

$$f = (p|n_g, n_b) = \frac{\tau(n_g, n_b)}{\tau(n_g) + \tau(n_b)} p^{n_g - 1}(1 - p)^{n_b - 1} \tag{3}$$

where τ denotes the beta distribution, applied for its flexibility and simplicity. n_g and n_b are the number of good and bad behaviors captured, respectively, and p is the probability of n_g. The expectation of binomial distribution probability is:

$$E(p) = \frac{n_g}{n_g + n_b}, 0 \leq E(p) \leq 1 \tag{4}$$

This equation denotes the degree or probability of a user being normal and thus we assumed normal users as credible and reliable. We can pre-set a threshold value ξ as a tolerable bottom line, $0 \leq \xi \leq 1$. For example, If $E(p) = \xi$, it means the user's behaviors captured are half and half (if

$\xi = 0.5$), and thence hard to classify. If $E(p) > \xi$, there is a great chance that the evaluated user is normal, according to the predetermined standard and his report will be spread to all close contacts as a reminder to self-quarantine or schedule a COVID-19 test. (The trust score may be attached to the final report with the user's explicit consent.) Otherwise, the user is more likely to be malicious under the expected standard and his positive test report should be discarded.

7 RELATED WORK

In this section, we review the work most close to ours on leveraging blockchain for contact-tracing. To the best of our knowledge, there are two manuscripts that cover blockchain-based contact-tracing with the goal of preserving user privacy: BeepTrace [36] and Pronto-C2 [7]. BeepTrace [36] utilizes blockchain to propose the architectural design of a privacy-preserving contact-tracing app. It further numerically analyzes network storage and computing capacity requirements of such a design. In comparison with our work, BeepTrace is an abstract open initiative without concrete prototype implementation. Pronto-C2 [7] presents a decentralized BLE-based peer-to-peer contact-tracing system, which can optionally be implemented with blockchain. Pronto-C2 seeks to protect users against mass surveillance by governments and authorities through decentralization. Like BeepTrace, Pronto-C2 is a high-level design proposal lacking a concrete implementation or prototype with blockchain. While there are many different concrete implementations of contact-tracing apps [6, 9, 27], none are based on blockchain. The use of self sovereign identity is the contribution of our work and we are unaware of any contact tracing app based on SSI.

Finally, to the best of our knowledge, there is no other work that takes advantage of the concept of trust in contact-tracing to compensate for the possible lack of attestations from health authorities. In fact, all the solutions we have found take such attestations for granted.

8 CONCLUSION

We presented the prototype of our privacy-preserving contact-tracing (PpCT) mobile app, which leverages Self-Sovereign Identity (SSI) on top of blockchain (Hyperledger Indy/Aries). We put the control of identity and diagnosis information back in the hands of users by enabling them to choose how much to share and keeping their identity/diagnosis information in their SSI wallets except for a part of diagnosis data that is absolutely necessary to be shared for the app to function. Even for this shared public information, we eliminate the use of a central server by employing peer to peer networks (Hyperledger Fabric). We envision two methods for reporting positive diagnosis results: (1) through direct app connection with testing labs and (2) voluntarily by users. We opt for the latter, following the same rationale to make the users the sole owners of their diagnosis data. Our choice of letting the users directly report diagnosis data, however, introduces the potential of flooding the network with false reports by malicious users. With very limited user information that is always kept on the user device, we introduce multiple fraud detection techniques inspired by previous work to detect and eliminate such false reports. We covered technical details of the implementation of PpCT in collaboration with a multinational telecommunications industrial partner. Our PpCT app is now fully developed and is in the beta testing phase by college students. We plan to deploy PpCT to actual customers of our industrial partner after beta testing. Our PpCT can help stop the spread of COVID-19, while at the same time protect user information privacy by giving the user the right to choose how much to share.

REFERENCES

[1] [n.d.]. Privacy-Preserving Contact Tracing - Apple and Google. https://covid19.apple.com/contacttracing

[2] Johannes Abeler, Matthias Bäcker, Ulf Buermeyer, and Hannah Zillessen. 2020. COVID-19 contact tracing and data protection can go together. *JMIR mHealth and uHealth* 8, 4 (2020), e19359.

[3] Christopher Allen. 2016. The path to self-sovereign identity. *Life with Alacrity* (2016).

[4] Thamer Altuwaiyan, Mohammad Hadian, and Xiaohui Liang. 2018. EPIC: efficient privacy-preserving contact tracing for infection detection. In *2018 IEEE International Conference on Communications (ICC)*. IEEE, 1–6.

[5] antonylewis2015. [n.d.]. A gentle introduction to self-sovereign identity. https://bitsonblocks.net/2017/05/17/gentle-introduction-self-sovereign-identity/ Accessed: Nov 23, 2020.

[6] Apple and Google. [n.d.]. Exposure Notifications Android. https://github.com/google/exposure-notifications-android Accessed: July 5, 2020.

[7] Gennaro Avitabile, Vincenzo Botta, Vincenzo Iovino, and Ivan Visconti. 2020. Towards Defeating Mass Surveillance and SARS-CoV-2: The Pronto-C2 Fully Decentralized Automatic Contact Tracing System. *IACR Cryptol. ePrint Arch.* 2020 (2020), 493.

[8] Scott R Baker, Nicholas Bloom, Steven J Davis, and Stephen J Terry. 2020. *Covid-induced economic uncertainty*. Technical Report. National Bureau of Economic Research.

[9] Jason Bay, Joel Kek, Alvin Tan, Chai Sheng Hau, Lai Yongquan, Janice Tan, and Tang Anh Quy. 2020. BlueTrace: A privacy-preserving protocol for community-driven contact tracing across borders. *Government Technology Agency-Singapore, Tech. Rep* (2020).

[10] Samuel Brack, Leonie Reichert, and Björn Scheuermann. 2020. Decentralized Contact Tracing Using a DHT and Blind Signatures. *IACR Cryptol. ePrint Arch.* 2020 (2020), 398.

[11] Vinay Chamola, Vikas Hassija, Vatsal Gupta, and Mohsen Guizani. 2020. A Comprehensive Review of the COVID-19 Pandemic and the Role of IoT, Drones, AI, Blockchain, and 5G in Managing its Impact. *IEEE Access* 8 (2020), 90225–90265.

[12] Justin Chan, Shyam Gollakota, Eric Horvitz, Joseph Jaeger, Sham Kakade, Tadayoshi Kohno, John Langford, Jonathan Larson, Sudheesh Singanamalla, Jacob Sunshine, et al. 2020. Pact: Privacy sensitive protocols and mechanisms for mobile contact tracing. *arXiv preprint arXiv:2004.03544* (2020).

[13] Hyunghoon Cho, Daphne Ippolito, and Yun William Yu. 2020. Contact tracing mobile apps for COVID-19: Privacy considerations and related trade-offs. *arXiv preprint arXiv:2003.11511* (2020).

[14] Lorrie Faith Cranor. 2020. Digital contact tracing may protect privacy, but it is unlikely to stop the pandemic. *Commun. ACM* 63, 11 (2020), 22–24.

[15] Ch Fei, J Lohkamp, E Rusu, K Szawan, K Wagner, and N Wittenberg. 2018. Jolocom: Self-sovereign and decentralised identity by design. *White paper* (2018).

[16] Md Sadek Ferdous, Farida Chowdhury, and Madini O Alassafi. 2019. In search of self-sovereign identity leveraging blockchain technology. *IEEE Access* 7 (2019), 103059–103079.

[17] Luca Ferretti, Chris Wymant, Michelle Kendall, Lele Zhao, Anel Nurtay, Lucie Abeler-Dörner, Michael Parker, David Bonsall, and Christophe Fraser. 2020. Quantifying SARS-CoV-2 transmission suggests epidemic control with digital contact tracing. *Science* 368, 6491 (2020).

[18] Saurabh Ganeriwal, Laura K Balzano, and Mani B Srivastava. 2008. Reputation-based framework for high integrity sensor networks. *ACM Transactions on Sensor Networks (TOSN)* 4, 3 (2008), 1–37.

[19] Bela Gipp, Norman Meuschke, and Andre Gernandt. 2015. Decentralized Trusted Timestamping using the Crypto Currency Bitcoin. In *Proceedings of the iConference 2015 (to appear)*. Newport Beach, CA, USA. http://ischools.org/the-iconference/

[20] Joel Hellewell, Sam Abbott, Amy Gimma, Nikos I Bosse, Christopher I Jarvis, Timothy W Russell, James D Munday, Adam J Kucharski, W John Edmunds, Fiona Sun, et al. 2020. Feasibility of controlling COVID-19 outbreaks by isolation of cases and contacts. *The Lancet Global Health* (2020).

[21] Carsten Keßler, Johannes Trame, and Tomi Kauppinen. 2011. Tracking editing processes in volunteered geographic information: The case of OpenStreetMap. In *Identifying objects, processes and events in spatio-temporally distributed data (IOPE), workshop at conference on spatial information theory*, Vol. 12. 6–8.

[22] Eric Knapp. 2011. Chapter 7 - Establishing Secure Enclaves. In *Industrial Network Security*, Eric Knapp (Ed.). Syngress, Boston, 147 – 187. https://doi.org/10.1016/B978-1-59749-645-2.00007-0

[23] Vincent Lenders, Emmanouil Koukoumidis, Pei Zhang, and Margaret Martonosi. 2008. Location-based trust for mobile user-generated content: applications, challenges and implementations. In *Proceedings of the 9th workshop on Mobile computing systems and applications*. 60–64.

[24] Joseph K Liu, Man Ho Au, Tsz Hon Yuen, Cong Zuo, Jiawei Wang, Amin Sakzad, Xiapu Luo, and Li Li. 2020. Privacy-Preserving COVID-19 Contact Tracing App: A Zero-Knowledge Proof Approach. *IACR Cryptol. ePrint Arch.* 2020 (2020), 528.

[25] Christian Lundkvist, Rouven Heck, Joel Torstensson, Zac Mitton, and Michael Sena. 2017. Uport: A platform for self-sovereign identity. *URL: https://whitepaper. uport. me/uPort_ whitepaper_DRAFT20170221. pdf* (2017).

[26] Nicola Mezzetti. 2004. A socially inspired reputation model. In *European Public Key Infrastructure Workshop*. Springer, 191–204.

[27] Paul Mozur, Raymond Zhong, and Aaron Krolik. 2020. In coronavirus fight, China gives citizens a color code, with red flags. *New York Times* (2020). https://www.nytimes.com/2020/03/01/business/china-coronavirus-surveillance.html Accessed: July 5, 2020.

[28] Satoshi Nakamoto. 2019. *Bitcoin: A peer-to-peer electronic cash system (2008).* Technical Report. Manubot.

[29] Sangchul Park, Gina Jeehyun Choi, and Haksoo Ko. 2020. Information technology–based tracing strategy in response to COVID-19 in South KoreaâĂŤprivacy controversies. *Jama* (2020).

[30] Drummond Reed, Jason Law, and Daniel Hardman. 2016. The technical foundations of sovrin. *The Technical Foundations of Sovrin* (2016).

[31] Leonie Reichert, Samuel Brack, and Björn Scheuermann. 2020. Privacy-Preserving Contact Tracing of COVID-19 Patients. *IACR Cryptol. ePrint Arch.* 2020 (2020), 375.

[32] Frantz Rowe. 2020. Contact tracing apps and values dilemmas: A privacy paradox in a neo-liberal world. *International Journal of Information Management* 55 (2020), 102178.

[33] Philipp Schmidt. 2016. BlockcertsâĂŤAn open infrastructure for academic credentials on the blockchain. *MLLearning (24/10/2016)* (2016).

[34] Jeffry A Simpson and William Steven Ed Rholes. 1998. *Attachment theory and close relationships.* Guilford Press.

[35] Andrew Tobin and Drummond Reed. 2016. The inevitable rise of self-sovereign identity. *The Sovrin Foundation* 29, 2016 (2016).

[36] Hao Xu, Lei Zhang, Oluwakayode Onireti, Yang Fang, William Bill Buchanan, and Muhammad Ali Imran. 2020. BeepTrace: Blockchain-enabled Privacy-preserving Contact Tracing for COVID-19 Pandemic and Beyond. *arXiv preprint arXiv:2005.10103* (2020).

[37] Tyler M Yasaka, Brandon M Lehrich, and Ronald Sahyouni. 2020. Peer-to-Peer contact tracing: development of a privacy-preserving smartphone app. *JMIR mHealth and uHealth* 8, 4 (2020), e18936.

WWW.IDENTITY.UTEXAS.EDU