**Open Access Library Journal**
2024, Volume 11, e11179
ISSN Online: 2333-9721
ISSN Print: 2333-9705

# Replay Attack Prevention in Decentralised Contact Tracing: A Blockchain-Based Approach

**Lavanya-Nehan Degambur**

Faculty of Information Communication and Digital Technologies, University of Mauritius, Reduit, Mauritius
Email: nehan24@live.com

**How to cite this paper:** Degambur, L.-N. (2024) Replay Attack Prevention in Decentralised Contact Tracing: A Blockchain-Based Approach. *Open Access Library Journal*, **11**: e11179.
http://doi.org/10.4236/oalib.1111179

**Received:** January 6, 2024
**Accepted:** February 18, 2024
**Published:** February 21, 2024

## Abstract

Digital contact tracing solutions have aided humanity in the first line of defense against the COVID-19 pandemic, but not without major technical drawbacks such as attacks against digital contact tracing technology and loss of privacy and security. The most popularly used digital contact tracing system is the decentralised DP-3T protocol and it suffers from the replay attack. A replay attack involves taking contact tracing data from one location and retransmitting it to another location and creating multiple issues such as false positive cases and inhibiting the COVID-19 pandemic fight. This project's aim was to try to prevent replay attacks in digital contact tracing systems using blockchain. The research methodology used was an empirical study using both qualitative and quantitative techniques. A literature review was performed by systematically reviewing and analyzing digital contact tracing concepts, theories, and research work. The DP-3T protocol was critically analysed to discover the threat surface that is vulnerable to replay attacks. A remodeled version of the DP-3T protocol was proposed by applying blockchain technology to store different keys and broadcast data, using hash values of location coordinates to ensure privacy, redefining the roles of participating entities, and enabling the authentication and validation of data using the blockchain when received by a user. The proposed solution was implemented and tested in a Python simulation. The simulation was input with real-life data which was saved on the blockchain, and broadcasts were simulated between senders and receivers before simulating replay attacks. Hence, all replay attacks are prevented during the Normal Operation phase of the protocol owing to the four layers of conditions verifications and validations that must be performed on a received broadcast. As compared with the DP-3T protocol, Vaudenay's Interactive Protocol and Pietrzak's Delayed Authentication scheme, our proposed solution prevents 100% of replay attacks and protects user privacy.

## Subject Areas

Digital Contact Tracing, Blockchain

## Keywords

COVID-19, Decentralised Digital Contact Tracing, DP-3T, Blockchain, Privacy

## 1. Introduction

Contact tracing in the COVID-19 context is the process of identifying and providing supported quarantine to people who have been in contact with individuals infected with SARS-Cov-2. It is used to discover the source of an infection by identifying events, settings, and circumstances where infection could have happened, hence paving the way for targeted sanitary measures. Data obtained from contact tracing contributes to the understanding of the worldwide epidemiology of COVID-19 [1] [2] [3].

Contact tracing can be done in two ways, namely, the manual classical method of recruiting contact tracers who will interview and trace contacts following infection and the digital way of using digital tools which minimise human intervention and allow for faster detection and alerting [4].

Smartphone-based digital contact tracing can be done in two models, namely, the centralised model and the decentralised model. When the centralised model is used, digital contact tracing users' applications report and upload locally saved contact tracing information to a central server, while also revealing the identity of infectious patients. The centralised server in turn informs the contacts of the infected user from the latter's contact list. This model is promoted by the Pan-European Privacy-Preserving Proximity Tracing (PEPP-PT) group that encourages the use of standardised methodologies for European data privacy in accessing and using user data. The advantage of the centralised model is that the authority in charge of the server gets an effective overview or picture of the state of the epidemic while also allowing infected users to remain anonymous among each while being visible to the server and to another single user who later tests positive [4].

Using the decentralised model, positively tested users only have to report their positively tested reports to the server while revealing their identity only to a limited level. The server then needs to update the anonymised public list of infected persons, which will be regularly queried and used by the digital contact tracing participants to check for their own respective statuses. The decentralised model is promoted by Google and Apple through the GAEN framework and by the DP-3T initiative. It is beneficial as the infected users do not need to expose their whole contact list to the server so they always keep control over their own data [4].

Most existing digital contact tracing solutions use one or more smartphone-based technologies such as Bluetooth Low Energy (BLE), Camera, Global Positioning System (GPS), Magnetometer, and WiFi. BLE utilises BLE scanning to discover nearby devices and received signal strength (RSSI) for relative distance estimation between devices. Machine learning techniques like computer vision and facial recognition can be used with smartphone cameras to track people and trace their contacts. GPS technology can be used to discover contacts by cross-verifying registered users' location details and trails. Smartphone magnetometers record the magnetic fingerprints of users that they leave as they travel, because of the magnetic anomalies caused by metals in buildings and other infrastructures. Public WiFi access points can be used to detect co-located smartphones that are connected to them at a certain moment while also using RSSI to estimate distance [4].

Currently, available solutions that have been evaluated are EPIC, Tracetogether, Reichert's MPC-based solution, CAUDHT, Berke's location-based contact tracing system, DP-3T, PEPP-PT, and ROBERT. Following a methodical evaluation strategy, three types of attacks can affect contact tracing solutions, namely generic attacks, Bluetooth-based attacks, and GPS-based attacks since most contact tracing solutions either use Bluetooth or GPS as sensing technologies. Generic attacks can be in the form of resource drain attacks, trolling attacks, replay attacks, proximity app attacks, tracking, and deanonymisation attacks, ransomware attacks, backend impersonation attacks, and false injection or false report attacks. Bluetooth-based attacks include bluejacking, bluesnarfing, and blue-bugging attacks. Finally, GPS-based attacks are GPS jamming attacks and GPS spoofing attacks [5] [6].

Bluetooth-based and GPS-based attacks exploit flaws in the hardware and firmware implementation of those technologies and hence they are beyond the scope of this research.

Additional digital contact tracing systems are COVID Trace, HowWeFeel, NOVID, COVID Shield, ShareTrace, Safe2, Aarogya Setu App, and Exposure Notification [6].

The DP-3T protocol is state-of-the-art, transparent, open-source, and decentralised and hence provides a trustworthy and visible digital contact tracing process and solution. The functioning decentralised nature of DP-3T makes it vulnerable to the dreaded replay attack among the thirteen other attacks it is vulnerable to.

Vulnerabilities that threaten DP3T can be categorised into two main categories namely False Alert Injection Attacks and People Tracking Tool. False alert injection attacks involve an attacker making the application of a victim raise false alerts either by sending the smartphone some ephemeral IDs of infected people or making the smartphone receive a daily key which can possibly derive one of the already received ephemeral IDs. False alert injection attacks include backend impersonation attacks, false report attacks, replay of released cases attacks, replay attacks, and replay attacks [7].

Since DP3T uses a wide variety of information such as dated daily keys of infected people, the reported date of those keys, smartphone contents, data shared user data, and data leaked through side channels, it can become a tool to track people if an adversary can access that information. A tool for tracking people could be done by using the beacons, deanonymising known reported users, disclosing private encounters, and making coercion threats [7].

Given that a bluetooth device is continuously broadcasting messages while turned on and also has a fixed MAC address, it can be recognised while also allowing the device's user to be recognised. Hence the smartphone that is broadcasting ephemeral IDs is also acting as a beacon, enabling a user to be deanonymised through a mapping process. Deanonymising known reported users can be done in four ways namely, occasional disclosure, paparazzi attack, nerd attack, and militia attack [7].

To combat replay attacks on DP-3T systems, interactive schemes that add a bit more complexity to the protocol are proposed without affecting the privacy aspects of the protocol, and delayed authentication has been proposed [7] [8].

Blockchain-leveraging digital contact tracing solutions include Hasan *et al.*'s DApp [9], Liu *et al.*'s Solution [10], BeepTrace [11], DIMY [12], Bari *et al.*'s DApp [13], Bychain [14], Connect [15].

A blockchain is a series of chained blocks in which each block contains a group of transactions that have been cryptographically signed by the entity that verified it and then stored in a distributed network of nodes to allow all authorised and authenticated stakeholders to access and verify them. Each block is cryptographically connected to the preceding one following a verification process and going through a consensus decision [16].

The more blocks are added to the chain, the more difficult older blocks become to modify and as new blocks are replicated across the network, the chain becomes increasingly tamper-proof and conflict resolution becomes automated within established rules [16].

The findings of the literature review are summarized as follows:

1) Replay attack is an interesting scenario to tackle since it is enabled by the basic functioning of the processes taking place in the decentralised methodology employed by protocols such as DP-3T and GAEN.

2) Moreover, preventing replay attacks can be done only using attack detection techniques if the protocol itself is not modified.

3) Nevertheless, modifying the protocol's algorithms can be done in two ways to fight replay attacks, namely, by adding interactive schemes to make the protocol more complex or by using delayed authentication in a simpler and non-interactive way; all while protecting privacy.

4) Hence, the DP-3T protocol can be modified while keeping its functionalities and continuing to protect privacy. Interactions and authentications can also not be exclusive to senders and receivers but can be extended to the server as well, while also not increasing the server's exposure and roles with respect to the user data it receives. The server's role can hence be limited only to querying, matching, and authenticating data. The distributed nature of blockchain makes

it an ideal solution to complement decentralised contact tracing systems such as DP-3T.

5) Blockchain data are tamper-proof, immutable, and privacy-preserving so that medical data or personally identifiable information can be secured in blockchains.

6) Since blockchains allow multiple parties to communicate without additional intervention, the autonomous nature of DP-3T remains intact while also allowing the authority's server to perform its data-matching tasks.

7) As all participating blockchain entities have an identical duplicate of the distributed ledger, queries to the blockchain do not use up network resources which can instead be used to sync the ledger, which provides an advantage to the phased contact tracing process used by DP-3T.

8) Since secure hash code usage is the basis of blockchain technology, the privacy of users can be preserved by strong hashes of personally identifiable data instead of the raw data itself.

9) Since replay attacks involve sending duplicates of collected messages, the basic functioning of blockchain allows the detection of duplicate data being stored and also prevents such storage of such data, if applied to specific scenarios with the contact tracing protocol's functioning.

### 1.1. Problem Statement

However with digital contact tracing, came new technology-borne threats to human lives, namely, attacks against the digital contact tracing systems and the loss of privacy and security [5] [6].

One of those attacks is the replay attack which takes data from one location or from a set of people and replays it to another location or another set of people so that false positive cases are detected, frenzies are created, health systems are overwhelmed, human adoption of digital contact tracing is decreased and the fight against the COVID-19 pandemic falters and resources are wasted [5] [6].

If someone has been falsely identified as COVID-19 positive by digital contact tracing, that person must stay at home to self-isolate, perform a test and wait for the results before deciding the way forward. Staying at home for extended periods of time in the torment of potentially being ill with COVID-19 can economically inhibit the person and negatively affect their physical and mental health although the COVID-19 test later reveals itself to be negative. Moreover, many resources are wasted in the process of performing COVID-19 tests for persons later revealed to be negative so the economic and logistical weight of the pandemic is enhanced on health systems worldwide [5] [6].

The only currently available practical way to prevent replay attacks in digital contact tracing is to detect them using attack detection techniques using antivirus or attack detection software or solutions inherent to the smartphone and infrastructure being used. However, the digital contact tracing protocols themselves remain vulnerable to replay attacks, since their mechanisms themselves allow it and do not provide any mechanism against it [5] [6]. Hence this is a real threat against the COVID-19 pandemic fight that needs solving.

### 1.2. Potential Solution

In light of the issues and challenges posed by digital contact tracing, blockchain technology can be a significant tool in resolving those issues and challenges. This is because blockchain provides a distributed peer-to-peer network connection between nodes and diminishes the gap between end-users and application managers. The decentralised network provided by blockchain ensures that data management becomes user-centric and returns data ownership to the users. Data security is enhanced because blockchains only store encrypted data which can be decrypted by users only. Since information stored in a blockchain is time-stamped upon entry by the digital signature of the source, the provenance and legitimacy of data are proven [6].

As data is distributed among the blockchain network participating nodes, data is always available as long as enough nodes are running. Data immutability, reliability, and transparency are guaranteed because once data is input into a blockchain, it can never be modified. Data discrepancies are also eliminated because of the timestamping of all data in a blockchain network [6].

**Aim**

The aim of this research was to determine if replay attacks can be prevented in digital contact tracing systems using novel technologies like Blockchain.

**Objectives**

1) Systematic reviews were used to understand all aspects of digital contact tracing.

2) Several digital contact tracing solutions were analysed to unearth the surface of the solution that is vulnerable to replay attacks.

3) Novel technologies such as blockchain were studied to understand their applicability to digital contact tracing.

4) A new approach or protocol for digital contact tracing was proposed, designed and simulated to demonstrate how replay attacks can be prevented.

5) The proposed solution's results were compared to the current state-of-the-art.

## 2. Methodology

The research methodology utilised is an empirical study that consists of both qualitative and quantitative techniques.

The research process used consists of a series of steps, namely Background Study, Research Problem Definition, Literature Review, Proposing a Hypothetical Solution, Research Solution Design, Data Collection and Execution, Data Analysis and Hypothesis Testing, and Results Evaluation and Reporting. Feedback and Forward loops interconnect certain steps and hence create a cycle.

A background study was performed by reading and critically analysing technical documents concerning digital contact tracing and identifying research problems that need solving. A research problem was defined by identifying that digital contact tracing solutions such as the decentralised DP-3T protocol are vulnerable to Replay attacks.

A deeper literature review was done by systematically reviewing and technically analysing theories, concepts, and previous research results found in selected and collected research papers, scientific journals, conferences, reviews, white papers, websites, and technical and scientific guides about the various aspects of digital contact tracing in relation to the problem statement previously defined. The main findings of the review were tabulated and several observations were noted.

A hypothetical solution was hence proposed using information and observations gathered in the literature review. Consequently, the DP-3T protocol was further analysed to understand the threat surface that is vulnerable to replay attack. The results of that analysis were then used to design the research solution.

Data used to test the hypothetical solution was collected based on the attributes of the solution design. The hypothetical solution was tested and the new results data was analysed using the solution design data fed forward from the research solution design step and the collected data. The results of that analysis are fed back to the data collection step if fresh data are needed for further analysis.

The results of the hypothesis testing and data analysis were interpreted and reported, using information fed forward from the research problem definition step to see if the problem is solved, and can even be fed back to the hypothesis testing and data analysis step if further investigations are needed. The results can also be fed back to define another research problem if the need arises.

## 2.1. Analysis of the DP-3T Protocol for Replay Attack Surface

This section describes the DP-3T protocol's functioning and shows how replay attacks can occur.

The DP-3T system works in 4 phases [17]:

1) Installation: When the application that uses DP-3T is installed, it generates a secret seed (SK), and derives Ephemeral Bluetooth IDs (EphIDs or EBIDs) from it. A new SK is generated every day using the first SK.

2) Normal Operation: All applications broadcast EphIDs using bluetooth, and locally save EphIDs, the time, and the exposure measurement that are broadcasted by other applications in the covered zone.

3) Handling Infected Patients: When patients are diagnosed, and only with their consent and with authorization from a health authority, they can upload data such as the SK of that day and the time data itself from their phone to the backend server.

4) Decentralized Contact Tracing: Each application can use the SK and time data of the infected patient from the backend to locally compute all possible EphIDs of the patient and calculate whether the application's user was in physical proximity of an infected patient using the locally stored EphID, Time and Exposure Measurement and devise the potential infection risk. The application can notify the user if there is any infection risk.

A replay attack can be performed in two stages as per Figure 1 below:

Firstly by collecting EphID broadcasts alongside time and exposure measurements from other smartphones to create a pool, and secondly by re-advertising EphIDs that belong to positively tested people, to the replay attack victims [5].

Research has suggested using attack detection techniques to prevent such replay attacks, however, since the protocol in itself does not provide the solution, replay attacks are a threat to DP-3T [5].

First of all, an attacker creates a collector computer that listens to and collects BLE beacons containing the broadcasted EphIDs and uploads them to a command-and-control server. The collector does not necessarily need to be in proximity to an honest DP-3T running smartphone to collect one single instance of a beacon; since a collector with a powerful antenna can also catch beacons. The collectors can also be smartphones infected with malware that transmit the collected EphIDs to the server. The attacker also creates a spreader that downloads collected broadcasts from the command-and-control server and retransmits them as fast as possible and for as long as possible so the broadcasts can be considered "valid". This scenario is even more amplified if some DP-3T implementations do not check the timeliness of published EphIDs and only perform EphID matching. The attacker could also retransmit collected EphIDs from a farther range using a powerful transmitter [18].

When a person tests positive and uploads their secret seeds for the last 14 days that can generate a replayed EphID, then everyone who has been in proximity to the spreader during retransmission will be deemed a false positive. Even

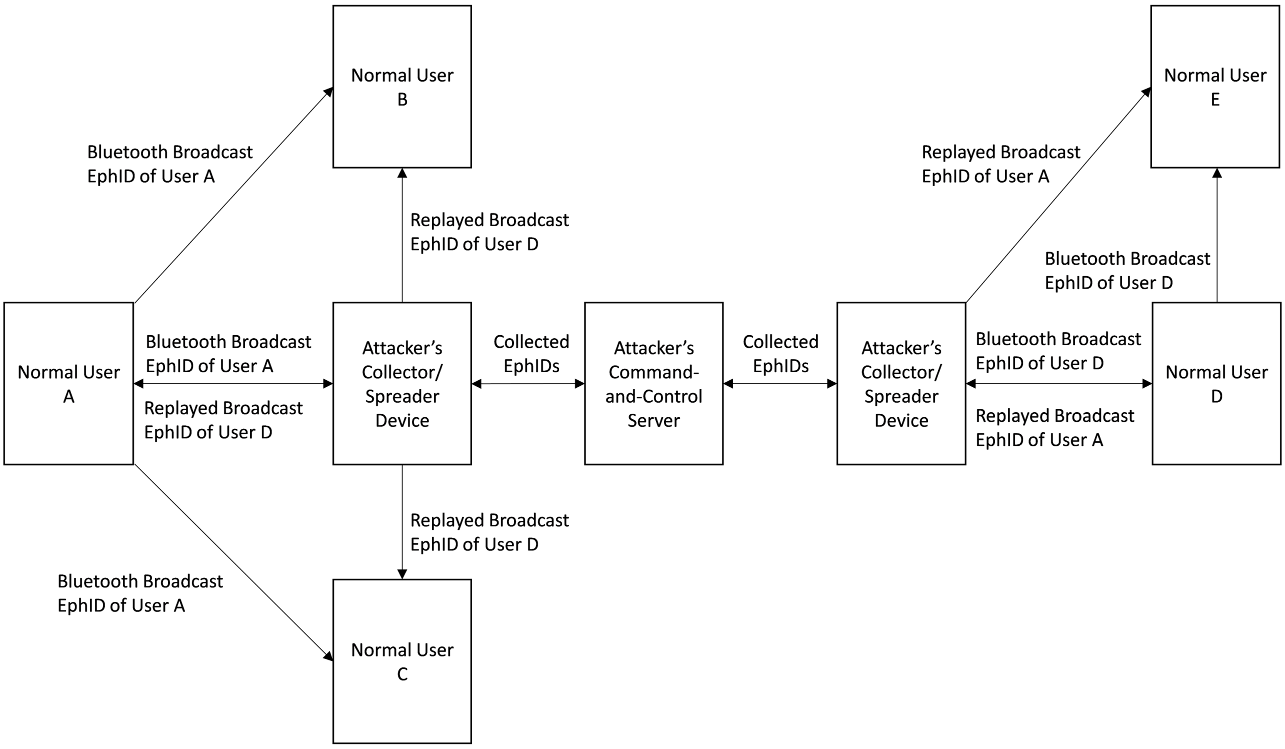
Normal User B
Bluetooth Broadcast EphID of User A
Replayed Broadcast EphID of User D
Normal User A
Bluetooth Broadcast EphID of User A
Replayed Broadcast EphID of User D
Attacker's Collector/ Spreader Device
Collected EphIDs
Attacker's Command-and-Control Server
Collected EphIDs
Attacker's Collector/ Spreader Device
Replayed Broadcast EphID of User D
Bluetooth Broadcast EphID of User A
Normal User C
Normal User E
Replayed Broadcast EphID of User A
Bluetooth Broadcast EphID of User D
Bluetooth Broadcast EphID of User D
Replayed Broadcast EphID of User A
Normal User D

Figure 1. Replay attack process.

worse, many spreaders can be set up in strategic places to further amplify the reach and create an avalanche of false positives and a collective frenzy. Another scenario might be to set up collectors at places with a likely risk of infected people passing by and broadcasting their EphIDs, such as in a COVID-19 testing station, and then placing spreaders at sites where people tend to linger around without being COVID-19 positive such as a hospital's emergency department [18].

## 2.2. Proposed Solution

The proposed revamped protocol adds a variety of different interactions between users and the backend server and also with the blockchain but maintains several aspects of the original DP-3T protocol such as the four phases of Installation, Normal Operation, Handling Infected Patients, and Decentralised Contact Tracing, the use of an original secret key, the generation of daily secret keys and ephemeral IDs, the broadcasting of ephemeral IDs and the use of exposure measurement.

The blockchain is used by the users to store secret keys, ephemeral IDs, and their corresponding generation times and location hashes, and broadcasted ephemeral IDs and their corresponding broadcast locations' hashes and broadcast times. The data stored remains anonymous and entities that are not data owners, such as the central server and the other users, can only query the blockchain to perform verifications and validations to prevent replay attacks.

To explain the protocol, users can be Senders or Receivers. A Sender's main roles involve generating daily secret keys, using them to generate daily ephemeral IDs, and broadcasting ephemeral IDs and their corresponding data via Bluetooth while also saving the data needed on the blockchain. The Receiver's role is to receive the broadcasts and validate their authenticity by querying the blockchain so that the receiver is the first line of replay attack prevention. The receivers also query the blockchain to perform the matching of daily secret keys and corresponding ephemeral IDs of patients, which they receive from the central server as part of the decentralised contact tracing phase of the protocol.

The server's roles consist of receiving diagnostics, daily secret keys, and time data from patients, querying the blockchain and the patient to verify and validate the received data and ensuring that ephemeral IDs corresponding to the daily secret keys are found on the blockchain, and finally forwarding the patients' daily secret keys to application users.

The proposed revamped protocol works as follows:

1) Installation:

a) When the application that uses DP-3T is installed, it generates a secret seed ($SK_0$).

2) Normal Operation:

a) A new $SK_i$ is generated every day using the first $SK_0$.

b) A pool of $EphID_1$ - $EphID_n$ is derived from $SK_i$.

c) When the EphIDs are generated, the GenerationTime and Location$_{SK}$ of the smartphone generating the EPhIDs are logged.

d) The pool of EphIDs, SK$_{i,}$ GenerationTime, and the hash of Location$_{SK}$ are stored on the blockchain.

e) Sender will broadcast EphID$_1$ - EphID$_n$ at time T$_1$ - T$_n$ respectively.

f) When Sender broadcasts EphID$_i$ at time T$_i$, the hash of the Location$_{EphID}$ of the smartphone at time T$_i$ is logged and broadcasted alongside EphID$_i$.

g) Sender saves EphID$_i$ at time T$_i$ and the hash of Location$_{EphID}$ is saved on the Blockchain.

h) Receiver gets EphID$_i$ at T$_i$, the hash of Location$_{EphID}$, and calculates Exposure Measurement.

i) Receiver queries the blockchain to check if EphID$_i$, T$_i$, and the hash of Location$_{EphID}$ match and if the Receiver's Location Hash matches Location$_{EphID}$ Hash**,** if Ti matches the time at which the Receiver received the broadcast and T$_i$ is within the EphID validity time period.

j) If Yes: Locally store exposure measurement, the exposure time, the broadcast time, the received time, the broadcast location, and the recipient's own location at receive time.

k) If No: Discard.

3) Handling Infected Patients:

a) When patients are diagnosed, and only with their consent and with authorization from a health authority, they can upload data such as the SKs of the last 14 days and the time data itself from their phone to the backend server.

b) The Server queries the positively tested user for the Location Hashes and Time of SK generation and also checks the blockchain using SK$_{i,}$ GenerationTime, and the hash of Location$_{SK}$ and EphIDs as search criteria to verify if the user who has uploaded his SKs as a positively tested user, is the user who generated the SKs in the first place.

c) The server also queries the blockchain to verify that the EphIDs generated from SK$_i$ have been previously saved by the user on the blockchain at the broadcast time T$_i$ and Location$_{EphID}$.

d) The server sends positively tested user's fully verified SKs to application users.

4) Decentralized Contact Tracing:

a) The recipient application can use the positively tested user's SK from the server to query the blockchain and see which EphIDs have been generated from the SK. After matching the EphIDs, the application then checks the exposure measurement, the exposure time, the broadcast time, the received time, the broadcast location, and the recipient's own location at receive time to perform contact tracing.

### 2.3. Simulation Method for the Proposed Solution

To study the behaviour of the proposed system and evaluate its viability to prevent replay attacks in digital contact tracing, a computer model was produced programmatically to test various roles, functionalities, and scenarios. It is also advantageous to programmatically simulate the solution before actually implementing it in real time on actual hardware or equipment so that resources are not wasted in the implementation of an untested concept and also to obtain a clear pathway to develop a working product.

First of all, the roles of each of the participating entities in the system were defined and the possible actions of each role were clearly identified so that they can be programmed. The four identified roles are the Sender, Receiver, Central Server, and the Blockchain. Next, since the solution involves data transfer between each of the entities via a communication medium such as Bluetooth, the simulation also includes the ability to programmatically transfer data between each entity.

A program was written using the Python programming language to simulate the proposed solution, mostly using the object-oriented programming paradigm. Each entity was programmed in the form of a Python class and the functionalities of each entity were implemented in the form of methods of the Python class.

Data transfer between each entity was simulated as arguments passed to methods that process the data to perform various calculations, comparisons, and make decisions.

To program the simulation, an integrated development environment (IDE) that supports Python development was needed as it provides tools for programmers to edit source code, automate builds and debug issues. The hardware platform on which the programming was done has the following specifications: an Intel Core i7 2.80 GHz CPU, 16 GB of RAM, and 64-bit Windows 10 Home operating system.

Since the proposed solution was simulated instead of being developed fully fledged, there were many differences between the simulation and the real-life implementation to ensure the simulation's practicability as enumerated below:

1) Bluetooth broadcasts were not programmed since the simulation was self-contained into demonstrating the use of blockchain to prevent replay attacks, instead of programming the hardware behaviour of the Bluetooth technology.

2) Since Bluetooth's hardware technology was not programmed, the signal attenuation calculation used to devise exposure measurement was not simulated. The proposed solution does not modify the way exposure measurement is done and is not directly related to the prevention of replay attacks in digital contact tracing so its implementation by the DP-3T consortium will be trusted and not included in our simulation.

3) The DP-3T protocol utilises Cuckoo filters during the various stages of secret keys generation, however, Cuckoo filters were not included in the simulation because they do not directly influence the prevention of replay attacks. Pseudorandom number generators offered by Python were used in key generation instead for simplicity and practicability.

4) Since the proposed solution includes the smartphone's location, a smartphone application named Sensor Logger was used to collect location coordinates at various locations and used in the simulation.

5) The implementation hence focussed on key generation, broadcast data handling, and decision-making between the sender, receiver, server and blockchain and did not include the physical hardware layer implementations.

Python has been chosen as the programming language for the simulation because it is safe, and advanced enough to be reliably stable, secure, scalable, and performant while also being easily learned. Most of all, python provides free fully-fledged, and tested packages and libraries such as JSON, Hashlib, Secrets, and Datetime that are essential in the development of blockchain programs as they need cryptographic modules, mathematical calculations, timestamping, and special notations for data storage [19] [20].

Spyder is a powerful integrated development environment written in Python to develop Python programs. Spyder integrates scientific packages such as IPython, SciPy, QTConsole, SymPy, NumPy, Matplotlib, Pandas, Secrets, Hashlib, and JSON which can also be further extended with third-party packages. When installed with Anaconda, Spyder is lighter but has a more expansive library than other similar IDEs [21].

Anaconda is an open-source, community-supported, free distribution of Python programming language that eases deployment and package management processes. Anaconda was used to install, use and administrate the Spyder IDE while also providing all the necessary Python libraries for the simulation's development [22].

Sensor Logger is an Android smartphone application that allows data collection from Android smartphones using numerous available sensors such as accelerometer, gravity sensor, gyroscope, magnetometer, barometer, proximity sensor, microphone and GPS. Collected data can be exported in CSV and JSON formats. Sensor Logger was used to get the various GPS coordinates needed for the simulation [23].

There were two types of tests to be performed on the simulation program, namely, unit tests and integration tests. Unit tests consisted of testing the simulation's individual modules' methods granularly to see if they behave as expected. The Input data was entered on the programming interface provided by the Spyder IDE and the results were the output from Spyder's console. Integration tests consisted of verifying that the simulation's different modules functioned properly together to perform the needed overall objective. The integration test for the simulation program consisted of running the Main module and the simulation to evaluate if they prevent replay attacks and preserve privacy in decentralised digital contact tracing.

### 2.4. Evaluation

The proposed solution's simulation was evaluated using tabled sample data, providing the observations consequently obtained, and interpreting the observations. Data was input one by one into the simulation and the data was in terms of the locations coordinates, their respective coarser versions, and their respective SHA256 hash values and also in terms of date-time strings.

Data has been input for six scenarios, Normal Broadcast, Initial Broadcast, Normal Broadcast with Coarse Location, Replay Attack with Different Locations, Replay Attack with Different Times, and Total Replay Attack. The results of this experiment were compared with those of the previously proposed ones from Vaudenay in 2020 [7] and Peitrzak in 2020 [8]. According to the literature reviewed, neither Vaudenay, 2020's paper nor Pietrzak, 2020's one provided quantitative results of their research, hence this section will showcase their qualitative results as compared to both the qualitative and quantitative results of our proposed solution.

## 3. Results and Discussion

Ten observations have been made following the evaluation process and compared to the original DP-3T protocol and two other previously proposed solutions. The observations obtained during the evaluation process of the solution's simulation are as follows:

1) The main replay attack prevention process is done in the Normal Operation phase among the four phases of the proposed solution. The receiver validates a received broadcast's EphID, Broadcast Location Hash, and Broadcast Date Time by matching them with data from the Broadcast Blockchain, then checking if the Broadcast Location Hash and Broadcast Date Time match the Receiver's Location Hash and the Receiver's Date Time when the broadcast was received and finally verifying if the broadcast was already received and if it has passed the EphID validity period.

2) Since the Broadcast Location Hash matching is done using the exact hash codes of the different Location Hashes, any granular change in location coordinates of the receiver changes the latter's location hash and the received broadcast is rejected as a replayed broadcast either correctly or as a false positive replay attack.

3) The granularity of the location coordinates directly affects the broadcast location hash matching process so that the coarser or rounded-off a set of coordinates become, more data points have the same coordinates as that of the broadcast, and fewer broadcasts are flagged as replayed broadcasts. However, it should be noted that false positives replay attacks are diminished by half in a set of 10 incremental coordinates with the rounding-off process as compared to those using the more granular coordinates.

4) Since the broadcast validation also involves matching the broadcast date time with the receiver's and that of the blockchain, only a broadcast with matching date times is accepted. When the receiver checks a received broadcast's date time with what is registered on the blockchain, any discrepancy between the two date times results in the broadcast's rejection.

5) Likewise, even if a broadcast's date time has been validated with the block-

chain if the receiver's date time at the time of broadcast receipt does not match the broadcast date time, the broadcast is always rejected.

6) However, it should be noted that EphIDs have been configured to have a 15-minute validity period so that only date times falling within the validity period are accepted while the others are systematically rejected.

7) A total replay attack includes several types of modified and replayed broadcast data such as modified broadcast date time, expired EphIDs, and replaying broadcasts of one location to receivers at another location and at other date times.

8) In a total replay attack, any broadcast is that received twice is systematically rejected as a replay attack, even if all the attributes used by a receiver for the broadcast validation, positively match.

9) In a total replay attack whereby the attacker has also modified the broadcast's date time, the blockchain validation process performed by the receiver enables the standardised rejection of the broadcast.

10) All scenarios of total replay attacks result in the rejection of the replayed broadcasts owing to the four layers of verifications, namely blockchain validation, receiver location and date time validation, EphID validity period checking, and locally stored EphID checking, done before a receiver saves a broadcast.

The following **Table 1** highlights the side-by-side comparison between the DP-3T protocol, our proposed solution and the proposals from Vaudenay's proposal [7] and Pietrzak's [8] based on two criteria namely, security and privacy.

## 4. Conclusions

Digital contact tracing solutions such as the DP-3T protocol are prone to several attacks that exploit the sensing, transmission, and storage technologies being used and one of those attacks is the Replay Attack. After a thorough literature review, the DP-3T protocol was analysed to find the attack surface using which attackers can take broadcasted data from one location and broadcast it to groups of people at another location. Following the analysis, a proposed solution that consists of applying blockchain technology and revamping the DP-3T protocol's functioning was proposed to find if replay attacks can be prevented.

The reworked protocol keeps the four phases of DP-3T protocol namely Installation, Normal Operation, Handling Infected Patients, and Decentralised Contact Tracing. The main replay attack prevention mechanism acts in the Normal Operation phase whereby a Sender saves the keys generated and each different broadcast data on the blockchain. Broadcast data include ephemeral keys, broadcast location hash values, and the date time at which data is broadcasted.

The receiver of a broadcast will validate the data using blockchain, then verify if the receiver's location and date time data match that of the broadcast, check if the broadcast is within its validity period, and finally ensure that no duplicate broadcast is saved locally.

Table 1. Comparison between proposals.

| Solutions/Criteria | Security | Privacy |
|---|---|---|
| DP-3T | Vulnerable against Replay Attacks. | Encounters information can be revealed. |
| Interactive Protocol [7] | Prevent Replay Attacks if the same course times are used. | Digital evidence is protected only to a certain degree if small tags are used. |
| Delayed Authentication Scheme [8] | Probably prevents replay attacks. | Digital evidence about encounters can be revealed in parties deflecting from the protocol. |
| Our Proposed Solution | 1. Prevents 100% of Replay attacks whereby Receiver's location hash differs from Broadcast's Location Hash.<br>2. Prevents 100% of Replay attacks whereby Receiver's date time differs from Broadcast's date time.<br>3. Prevents 100% of Total Replay attacks whereby broadcasts have different locations and different date times as compared to the receiver, ephemeral keys that have passed their validity period, and/or broadcasts that have already been received by the receiver.<br>4. Any broadcast that does not fulfil all four criteria needed to be accepted is automatically flagged as a replayed broadcast, resulting in an acceptance rate of only 9.10% in a batch of 11 receivers with variable locations and date times.<br>5. The acceptance rate can be increased by making date times or location coordinates coarser. | Location Hashes prevent encounters from being disclosed. |

In the Handling Infected Patients phase, a positively tested patient uploads the last 14 daily secret keys to the authority server, which then uses the blockchain to validate other user-supplied data before sending the daily secret keys to other users.

In the Decentralised Contact Tracing phase, a user queries the blockchain to obtain the ephemeral keys that belong to the received daily secret keys and perform matching with locally stored ephemeral keys.

Following the simulation, the evaluation process has concluded that a broadcast will be accepted only if the following conditions are met:

1) The broadcasted Ephemeral Key, Location Hash, and Date Time should have an entry saved in the blockchain that matches the broadcasted Ephemeral Key, Location Hash, and Date Time.

2) The broadcasted Location Hash and Date Time should match the Receiver's Location Hash and Date Time at which the broadcast data was received.

3) The broadcasted Ephemeral Key's Date Time of broadcast should be within the Ephemeral Key's validity period of 15 minutes as saved on the blockchain.

4) The broadcast data should not be already present in the Receiver's local storage.

The granularity of the location coordinates can also be modulated to make them coarser so that more identical coordinates and hash values are obtained to allow more users in the same vicinity to accept broadcasts and hence generate fewer false-positive replay attack flags.

Thus the proposed solution effectively prevents replay attacks while also protecting privacy and providing better security in digital contact tracing. Finally, as compared to the DP-3T protocol, Vaudenay's Interactive Protocol and Pietrzak's Delayed Authentication scheme, our proposed solution prevents 100% of replay attacks and protects user encounters and hence privacy.

## Acknowledgements

I wish to wholehearted thank everyone who helped and supported me during the course of this research, including my supervisor, my family and my friends.

## Conflicts of Interest

The author declares no conflicts of interest.

## References

[1] Centers for Disease Control and Prevention. (2021) Contact Tracing for COVID-19. https://www.cdc.gov/museum/pdf/cdcm-pha-stem-lesson-contact-tracing-lesson.pdf

[2] European Centre for Disease Prevention and Control (2021) Contact Tracing in the European Union: Public Health Management of Persons, Including Healthcare Workers, Who Have Had Contact with COVID-19 Cases, Stockholm.

[3] World Health Organisation (2021) Contact Tracing in the Context of COVID-19. Interim Guidance 1 February 2021. World Health Organisation, Geneva. https://doi.org/10.15557/PiMR.2020.0005

[4] Nguyen, K.A., Luo, Z. and Watkins, C. (2020) Epidemic Contact Tracing with Smartphone Sensors. *Journal of Location Based Services*, **14**, 92-128. https://doi.org/10.1080/17489725.2020.1805521

[5] Dar, A.B., Lone, A.H., Zahoor, S., Khan, A.A. and Naaz, R. (2020) Applicability of Mobile Contact Tracing in Fighting Pandemic (COVID-19): Issues, Challenges and Solutions. *Computer Science Review,* **38**, Article 100307. Https://Doi.Org/20/2026/J.Cosrev.2020.100307

[6] Idrees, S.M., Nowostawski, M. and Jameel, R. (2021) Blockchain-Based Digital Contact Tracing Apps For COVID-19 Pandemic Management: Issues, Challenges, Solutions, and Future Directions. *JMIR Medical Informatics*, **9**, e25245. https://doi.org/10.2196/25245

[7] Vaudenay, S. (2020) Analysis of DP3T. Between Scylla and Charybdis. https://infoscience.epfl.ch/record/277808/files/dp3t-ana.pdf

[8] Pietrzak, K. (2020) Delayed Authentication: Preventing Replay and Relay Attacks in Private Contact Tracing. In: Bhargavan, K., Oswald, E. and Prabhakaran, M., Eds., *Progress in Cryptology—INDOCRYPT* 2020, Vol. 12578, Springer, Cham. Https://Eprint.Iacr.Org/2020/418 https://doi.org/10.1007/978-3-030-65277-7_1

[9] Hasan, H.R., Salah, K., Jayaraman, R., Yaqoob, I., Omar, M. and Ellahham, S. (2021) COVID-19 Contact Tracing Using Blockchain. *IEEE Access*, **9**, 62956-62971. https://doi.org/10.1109/ACCESS.2021.3074753

[10] Liu, M., Zhang, Z., Chai, W. and Wang, B. (2022) Privacy-Preserving COVID-19

Contact Tracing Solution Based on Blockchain. *Computer Standards & Interfaces*, **83**, Article 103643. https://doi.org/10.1016/j.csi.2022.103643

[11] Xu, H., Zhang, L., Onireti, O., Fang, Y., Buchanan, W.J. and Imran, M.A. (2021) Beeptrace: Blockchain-Enabled Privacy-Preserving Contact Tracing for COVID-19 Pandemic and Beyond. *IEEE Internet of Things Journal*, **8**, 3915-3929. https://doi.org/10.1109/JIOT.2020.3025953

[12] Ahmed, N., Michelin, R.A., Xue, W., Putra, G.D., Ruj, S., Kanhere, S.S. and Jha, S. (2022) Dimy: Enabling Privacy-Preserving Contact Tracing. *Journal of Network And Computer Applications*, **202**, Article 103356. https://doi.org/10.1016/j.jnca.2022.103356

[13] Bari, N., Qamar, U. and Khalid, A. (2021) Efficient Contact Tracing For Pandemics Using Blockchain. *Informatics in Medicine Unlocked*, **26**, Article 100742. https://doi.org/10.1016/j.imu.2021.100742

[14] Lv, W., Wu, S., Jiang, C., Cui, Y., Qui, X. and Zhang, Y. (2020) Decentralized Blockchain for Privacy-Preserving Large-Scale Contact Tracing. Https://Arxiv.Org/Abs/2007.00894

[15] Bandara, E., Liang, X., Foytik, P., Shetty, S., Hall, C., Bowden, D., Ranasinghe, N. and De Zoysa, K. (2021) A Blockchain Empowered and Privacy-Preserving Digital Contact Tracing Platform. *Information Processing and Management*, **58**, Article 102572. https://doi.org/10.1016/j.ipm.2021.102572

[16] Shrimali, B. and Patel, H.B. (2021) Blockchain State-of-the-Art: Architecture, Use Cases, Consensus, Challenges and Opportunities. *Journal of Kind Saud University—Computer and Information Sciences*, 34, 6793-6807. https://doi.org/10.1016/J.Jksuci.2021.08.005

[17] Troncoso, C., Payer, M., Hubaux, J-P. and Salath, M. (2020) Decentralized Privacy-Preserving Proximity Tracing. Https://Github.Com/Dp-3t/Documents

[18] Farrell, S. and Leith, D.J. (2020) A Coronavirus Contact Tracing App Replay Attack with Estimated Amplification Factors. Https://Down.Dsg.Cs.Tcd.Ie/Tact/Replay.Pdf

[19] Protasiewicz, J. (2018) 5 Reasons Why Python Is Good for Blockchain. Https://Www.Netguru.Com/Blog/Python-Blockchain

[20] Python Org. (2022) Python. Https://Www.Python.Org

[21] Anaconda Org. (2022) Anaconda/Packages/Spyder. The Scientific Python Development Environment. Https://Anaconda.Org/Anaconda/Spyder

[22] Anaconda Documentation (2022) Getting Started with Anaconda. Https://Docs.Anaconda.Com/Anaconda/User-Guide/Getting-Started

[23] Choi, K.T.H.C. (2022) Sensor Logger. Https://Www.Tszheichoi.Com/Sensorloggerhelp