International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

Series X

Supplement 38
(09/2022)

SERIES X: DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

ITU-T X.1152 – Supplement on use cases for contact tracing technologies to prevent spread of infectious diseases

ITU-T X-series Recommendations – Supplement 38

ITU-T X-SERIES RECOMMENDATIONS

DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

PUBLIC DATA NETWORKS	X.1–X.199
OPEN SYSTEMS INTERCONNECTION	X.200–X.299
INTERWORKING BETWEEN NETWORKS	X.300–X.399
MESSAGE HANDLING SYSTEMS	X.400–X.499
DIRECTORY	X.500–X.599
OSI NETWORKING AND SYSTEM ASPECTS	X.600–X.699
OSI MANAGEMENT	X.700–X.799
SECURITY	X.800–X.849
OSI APPLICATIONS	X.850–X.899
OPEN DISTRIBUTED PROCESSING	X.900–X.999
INFORMATION AND NETWORK SECURITY	
General security aspects	X.1000–X.1029
Network security	X.1030–X.1049
Security management	X.1050–X.1069
Telebiometrics	X.1080–X.1099
SECURE APPLICATIONS AND SERVICES (1)	
Multicast security	X.1100–X.1109
Home network security	X.1110–X.1119
Mobile security	X.1120–X.1139
Web security (1)	X.1140–X.1149
Application Security (1)	X.1150–X.1159
Peer-to-peer security	X.1160–X.1169
Networked ID security	X.1170–X.1179
IPTV security	X.1180–X.1199
CYBERSPACE SECURITY	
Cybersecurity	X.1200–X.1229
Countering spam	X.1230–X.1249
Identity management	X.1250–X.1279
SECURE APPLICATIONS AND SERVICES (2)	
Emergency communications	X.1300–X.1309
Ubiquitous sensor network security	X.1310–X.1319
Smart grid security	X.1330–X.1339
Certified mail	X.1340–X.1349
Internet of things (IoT) security	X.1350–X.1369
Intelligent transportation system (ITS) security	X.1370–X.1399
Distributed ledger technology (DLT) security	X.1400–X.1429
Application Security (2)	X.1450–X.1459
Web security (2)	X.1470–X.1489
CYBERSECURITY INFORMATION EXCHANGE	
Overview of cybersecurity	X.1500–X.1519
Vulnerability/state exchange	X.1520–X.1539
Event/incident/heuristics exchange	X.1540–X.1549
Exchange of policies	X.1550–X.1559
Heuristics and information request	X.1560–X.1569
Identification and discovery	X.1570–X.1579
Assured exchange	X.1580–X.1589
Cyber Defence	X.1590–X.1599
CLOUD COMPUTING SECURITY	
Overview of cloud computing security	X.1600–X.1601
Cloud computing security design	X.1602–X.1639
Cloud computing security best practices and guidelines	X.1640–X.1659
Cloud computing security implementation	X.1660–X.1679
Other cloud computing security	X.1680–X.1699
QUANTUM COMMUNICATION	
Terminologies	X.1700–X.1701
Quantum random number generator	X.1702–X.1709
Framework of QKDN security	X.1710–X.1711
Security design for QKDN	X.1712–X.1719
Security techniques for QKDN	X.1720–X.1729
DATA SECURITY	
Big Data Security	X.1750–X.1759
Data protection	X.1770–X.1789
IMT-2020 SECURITY	X.1800–X.1819

For further details, please refer to the list of ITU-T Recommendations.

Supplement 38 to ITU-T X-series Recommendations

ITU-T X.1152 – Supplement on use cases for contact tracing technologies to prevent spread of infectious diseases

Summary

Supplement 38 to ITU-T X-series Recommendations defines a contact tracing application as a tool that enables the identification, assessment, and management of people who have been in contact with individuals that may have been infected with a contagious disease to prevent onward transmission. These applications help prevent the spread of infectious diseases by proactively finding people at higher risk than others due to potential exposure, notifying them if possible and determining whether a quarantine is necessary.

This Supplement guides the development of interoperable systems to automatically trace and inform potentially infected users, in addition to manual notification methods, with consideration for reducing potential security risks associated with data processed in contact tracing applications. It also describes various use cases for contact tracing applications, provides data processing models including their data processing flow and identifies threats and risks from a security and personally identifiable information (PII) protection perspective.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T X Suppl. 38	2022-09-02	17	11.1002/1000/15165

Keywords

Centralized contact tracing system, contact tracing, decentralized contact tracing system, smartphone application.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

This is an informative ITU-T publication. Mandatory provisions, such as those found in ITU-T Recommendations, are outside the scope of this publication. This publication should only be referenced bibliographically in ITU-T Recommendations.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this publication may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the publication development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents/software copyrights, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the appropriate ITU-T databases available via the ITU-T website at http://www.itu.int/ITU-T/ipr/.

Table of Contents

Supplement 38 to ITU-T X-series Recommendations

ITU-T X.1152 – Supplement on use cases for contact tracing technologies to prevent spread of infectious diseases

1 Scope

This Supplement describes various use cases for contact tracing technologies. It also provides data processing models including their procedures, data processing flow and security considerations. In addition, practical use cases are described in Appendix I.

2 References

None.

3 Definitions

3.1 Terms defined elsewhere

This Supplement uses the following terms defined elsewhere:

3.1.1 data processing [b-ISO/IEC 20944-1]: Systematic performance of operations on data.

NOTE – The term data processing is not a synonym for information processing. Information processing includes data communication (e.g., computer networks) and office automation (e.g., satisfying the business needs of an entity), whereas data processing does not include data communication and office automation.

3.1.2 mobile device [b-ISO 12812-1]: Personal device with mobile communication.

3.1.3 objective [b-ISO/IEC 27000]: Result to be achieved.

3.1.4 personally identifiable information [b-ISO/IEC 29100]: Any information that (a) can be used to identify the PII principal to whom such information relates, or (b) is or might be directly or indirectly linked to a PII principal.

NOTE – To determine whether a PII principal is identifiable, account should be taken of all the means which can reasonably be used by the privacy stakeholder holding the data, or by any other party, to identify that natural person.

3.1.5 risk [b-ISO/IEC 27000]: Effect of uncertainty on objectives (clause 3.1.3).

3.1.6 threat [b-ISO 7498-2]: Potential violation of security.

3.2 Terms defined in this Supplement

This Supplement defines the following terms:

3.2.1 contact tracing: Process of identifying, assessing, and managing people who were in contact with individuals that may have been infected with a contagious disease to prevent onward transmission.

3.2.2 manual contact tracing: Conventional contact tracing means that requires interviewing the infected patient(s) regarding their lifestyle and sustained contacts using a long list of questions to trigger memories and elicit names/addresses.

4 Abbreviations and acronyms

This Supplement uses the following abbreviations and acronyms:

AES Advanced Encryption Standard

BLE	Bluetooth Low Energy
DoS	Denial of Service
ECC	Encrypted Country Code
EphID	Ephemeral Identifier
GPS	Global Positioning System
ID	IDentifier
PII	Personally Identifiable Information
PoW	Proof of Work
PUID	Pseudonymous persistent User ID
RSSI	Received Signal Strength Indicator
TAN	Transaction Authentication Number

5 Conventions

None.

6 Overview of contact tracing

6.1 General

Contact tracing is the process of identifying, assessing, and managing people who have been in contact with individuals that may have been infected with a contagious disease to prevent onward transmission.

Contact tracing applications are designed to limit the spread of infectious diseases by tracking a person who has been in close contact with an infected patient. These applications typically use a Bluetooth-based system [b-Apple & Google] that stores data on people's phones. When someone has officially tested positive for an infectious disease, the system can send a notification to anyone else who has recently been near that infected person, requesting them to contact their local health authority for medical advice and for testing.

Bluetooth low energy (BLE) power-conserving variant has emerged as the most promising short range wireless network technology to implement the contact tracing service. GPS signals could also be used to implement the contact tracing service.

The data gathered by contact tracing applications can be used to alert people if they pose a risk of spreading a contagion and need to isolate. However, a split has emerged between two different types of contact tracing applications, the centralized and decentralized models.

Both models normally use Bluetooth signals to log when smartphone owners are close to each other – so if someone has tested positive for a contagious disease, an alert can be sent to other users they may have infected. Some other models use GPS signals to provide a contact tracing service.

There are two types of models of contact tracing applications [b-Criddle]: the centralized contact tracing model and decentralized contact tracing model, as shown in Figure 1. Under the centralized contact tracing model, the anonymized data gathered is uploaded to a remote server where matches are made with other contacts in case a person starts to present with symptoms of a contagious disease.

In contrast, the decentralized contact tracing model [b-Apple], [b-EU] gives users more control over their data by storing it on their mobile device. The mobile device makes matches with people who may have contracted the virus. This model may comply with data protection principles described in [b-ISO/IEC 29100] and controls described in [b-ITU-T X.1058].

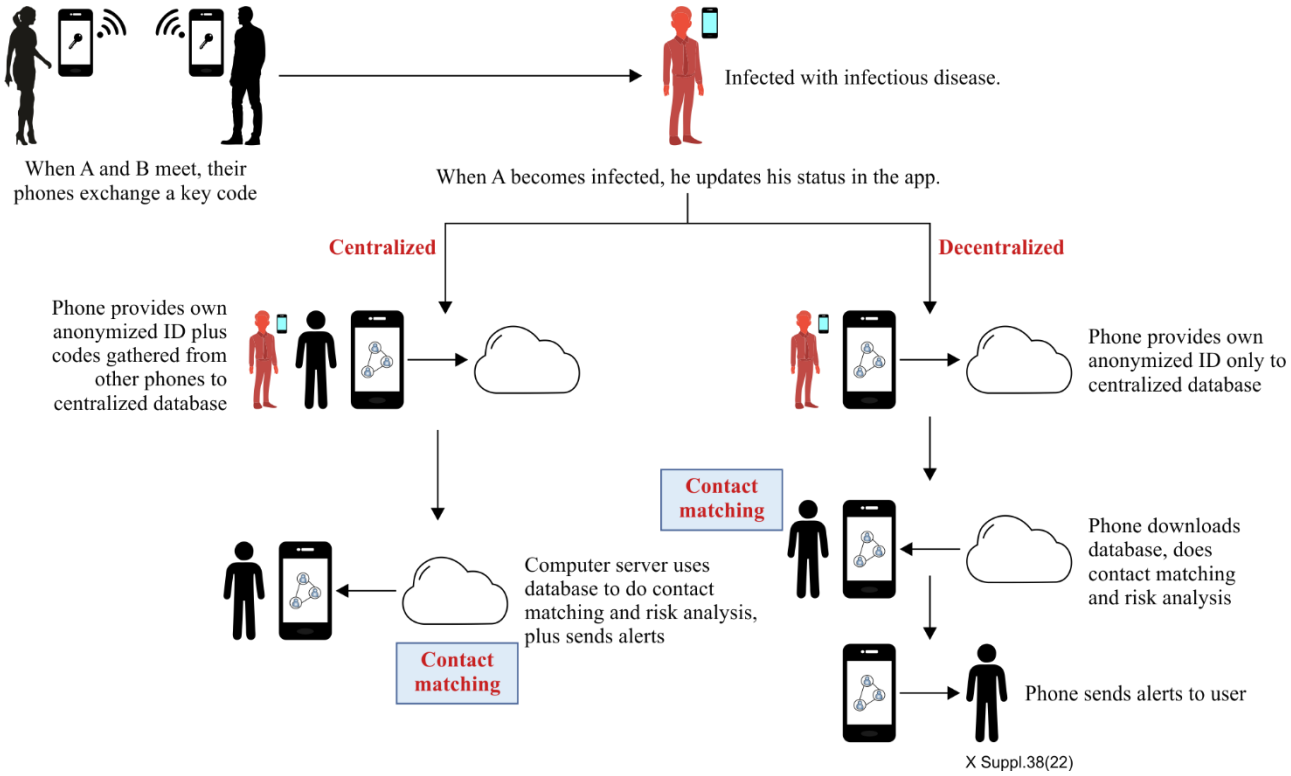

Figure 1 – Centralized contact tracing model and decentralized contact tracing model

A centralized contact tracing model can give authorities more insight into the spread of the virus and how well the application is performing. However, the decentralized contact tracing model offers users a higher degree of privacy, protecting them from hackers or the State itself revealing their social contacts.

6.2 Entities, methods and protocols for a digital contact tracing system

A digital contact tracing system informs its users that they have been in contact with individuals that may have been infected with a contagious disease. It can be a tool that supports manual contact tracing, making it more efficient [b-ETSI GR E4P 002].

The digital contact tracing system is composed of the following entities, as shown in Figure 2:

– **User**: A person that uses the digital contact tracing system through a mobile application on a mobile device.

– **Mobile device**: An electronic device that utilizes communication networks while in motion, responsible for providing the proximity information that is obtained via the proximity detection method, by communicating with other mobile devices and with an infrastructure through a mobile application.

– **Mobile application**: A piece of software running on the mobile device, responsible for registering and managing proximity information, communicating with the infrastructure, informing the end user that he or she may be infected and notifying the central infrastructure if the user tests positive.

– **Infrastructure (i.e., backend system/server)**: A set of technology elements (computers, databases, networks, etc.) that provides authoritative, trusted information to the mobile device. The main role of the infrastructure is to support information sharing between users through their mobile devices and apps. Should there be multiple infrastructures potentially using different contact tracing protocols, they might exchange information through a federation protocol to provide interoperability between the different digital contact tracing systems.

– **Health authority**: The public authority overseeing the whole digital contact tracing system and process and responsible for certifying the infection of a user.

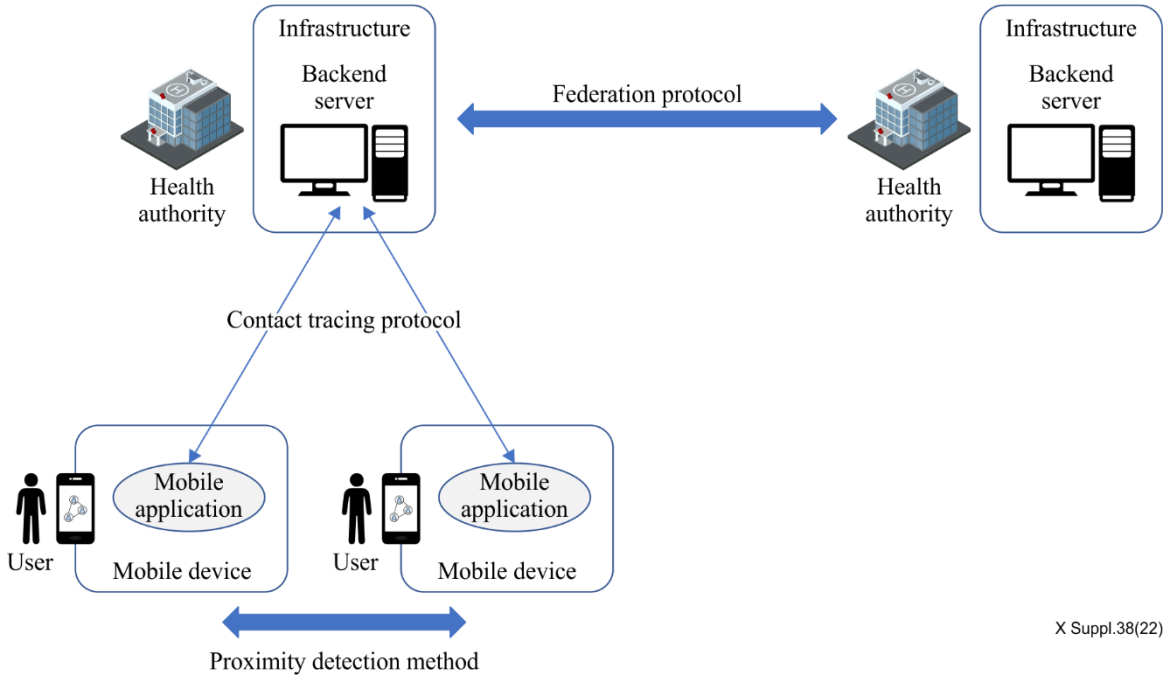

Figure 2 – Entities, methods and protocols for a digital contact tracing system

The following methods and protocols are needed to implement the digital contact tracing system.

– **Proximity detection method**: The method used by mobile devices for detecting their proximity (based on Bluetooth signals sent between devices) with potential sources of infection.

– **Contact tracing protocol**: The protocol between mobile devices and the infrastructure used by the mobile application.

– **Federation protocol**: A protocol used to exchange information between different infrastructures.

7 Contact tracing models

7.1 Centralized contact tracing model

7.1.1 Procedure

The centralized contact tracing model helps authorities responsible for collecting, processing and investigating information to identify exposure to infectious diseases. Records collected through the application are sent to a centralized server, then authorities can determine devices of people exposed to a confirmed infected patient from the records. After identifying the IDs of devices of exposed people, authorities can send alerts to inform them that they may have been exposed to an infectious disease.

This model has been confirmed to give the authorities more insight into the spread of infection, although a reluctance to confirm its superiority exists since it is inherently weak in data protection.

In centralized systems, ephemeral identifiers are derived from a pseudonym of the user, who has been previously registered in the backend system. The central backend server has a trapdoor allowing the retrieval of the pseudonym of the user from the ephemeral identifier. Hence, the backend server can determine if a user, with a certain pseudonym, is at risk as soon as it recognizes the pseudonym from the received ephemeral identifier(s). Therefore, when the user's application connects to the server and authenticates under his or her pseudonym, the server can directly tell whether the user is at risk.

The procedure for a centralized system is as follows:

– **User registration**: Each user's application registers to the server. During user registration, a pseudonymous user ID is generated by the server and sent to the application. Therefore, the backend server sets a pseudonym (pseudonymous user ID) for the application. The backend server and the application may determine a way to authenticate the application under pseudonym.

– **Setup of identifiers**: The application frequently connects and authenticates to the backend server to get new identifiers. The backend server or application creates a list of ephemeral identifiers which can be mapped to application's pseudonym by using the backend server's trapdoor. The ephemeral identifiers are given to the application which stores them in a list (of to send ephemeral identifiers).

– **Broadcast of ephemeral identifier**: During a given internal, the application constantly broadcasts an ephemeral identifier. Every application from other users collects the broadcasted ephemeral identifier and stores it in a list of received ones together with coarse-time information.

– **Reporting**: Upon testing positive, a diagnosed user provides his or her own application with the appropriate (anonymous) credential to upload (part of) its list of received ephemeral identifiers to the backend server. The first user’s application does not authenticate to the server. Elements to report are sent separately to prevent the server from linking them. The server associates each reported ephemeral identifier with a pseudonym of the original (first) user in its database (using the trapdoor) and remembers that said pseudonym has to be notified.

– **Status verification**: Regularly, the first user’s application connects and authenticates to the server to check the status of its user on the server. The server answers whether the user is at risk. If at risk, data about the pseudonym of that first user are erased and the user (his or her app) should register again.

7.1.2 Data processing

When a user installs this application, the backend server is always active. During user registration, a pseudonymous user ID is generated by the server and sent to the app.

For every period *t*, for example 1 h, the backend server generates a single secret key BK_t shared with all users. The backend server generates enough BK_t keys to cover a larger period in the future, for example, 2 days. Then, the application generates an ephemeral BLE ID (*EBID*) for every *t*, by encrypting their pseudonymous persistent user ID (PUID) with BK_t.

Each application broadcasts its current valid $EBID_t$ via BLE advertisements using the BLE privacy feature to prevent the tracking of users who send out continuous BLE advertisements.

When a user tested positive as infected, the collected data are sent to the backend server for assessing which other users are at risk and notifying them. The backend holds these data for up to 3 weeks. To upload the data to the backend, a health care professional provides a transaction authentication number (TAN) to the infected user by out-of-band means.

The backend server associates each *EBID* received with its corresponding PUID and calculates the risk for the PUID holder.

7.1.3 Security consideration

In decentralized use cases, the following are considered for security:

– The application should compute the solution to the PoW challenge, if necessary, to impede mass creation of user accounts.

– The application should provide an authentication to the server and vice versa.

– The backend server should generate a single secret key BK_t.

– The backend server should share a single secret key BK_t with all users in a secure manner.

– Each application should generate an *EBID* for every *t*, by encrypting their PUID with BK_t.

– Each application should broadcast its current valid $EBID_t$ via BLE advertisements using the BLE privacy feature to prevent tracking of users who send out continuous BLE advertisements.

– It should not store PII data or location information other than proximity information.

– To detect proximity, it should use an anonymous ID, $EBID_t$, which is cryptographically derived from the anonymous ID, PUID.

– An ephemeral BLE ID should be renewed for the short term to make it less unlinkable.

– The random Bluetooth ID should be cryptographically derived from the anonymous ID so that it cannot be inferred from the random Bluetooth ID.

– Users may decide on the application or transmission of their information.

– If a user is diagnosed with the infection or is notified as having been exposed to an infector, he or she may determine whether to send his or her information to the server or health authorities.

– When a user tests positive for infection, the collected data should be sent to the backend server for assessing which other users are at risk and notifying them.

– Data should be stored and processed only within a device except for a propagation of the random Bluetooth ID, or unless a user has consented to transmit his or her information to the server.

– Information generated in the application should strictly be limited to be used for the prevention of infectious diseases.

– Information in the application should be deleted after a period that has been proven to be safe epidemically.

7.2 Decentralized contact tracing model

7.2.1 Procedure

The decentralized contact tracing model uses Bluetooth technology on mobile applications for contact tracing. To allow contact tracing, the application is designed to create a random Bluetooth identifier, broadcast and receive it to measure nearby devices. When the devices are nearby, they receive beacons that contain the temporary Bluetooth ID of the other device and metadata that indicates the distance from the device. The device lists these IDs and metadata for further use.

After an updated positive diagnosis list is received from the server, the device compares the list of received IDs to the positive diagnosis list. If there is a match, a user is notified on when they were in contact with an infected person based on an approximate distance between the users.

When a user is diagnosed as positive with an infectious disease or notified that he or she has been in contact with someone who has been diagnosed as positive, a user can determine whether to send his or her information to the server with explicit consent.

The procedure for a decentralized system is as follows:

– **Setup of identifiers**: A user's application regularly prepares a list of random ephemeral identifiers to be used (broadcast) and stores them in a list.

– **Broadcasting of ephemeral identifier**: During a certain interval of time, the user's application constantly broadcasts an ephemeral identifier. A few weeks after that ephemeral identifier is broadcasted for the last time, it is erased from the list of ephemeral identifiers of the user. Every other user application collects the ephemeral identifier broadcast by the first user and stores it in the other users' list of received ephemeral identifiers with some time information.

– **Reporting**: Upon testing positive, a diagnosed user provides his or her own application with the appropriate credential to upload (part of) his or her list of ephemeral identifiers to send (broadcast) to the backend server. The latter publishes it.

– **Status verification**: Regularly, the application of the first user checks the newly uploaded ephemeral identifiers on the server and checks whether they are an element of his or her list of received ephemeral identifiers. This way, the user's application determines if someone is at risk.

7.2.2 Data processing

The application in a decentralized use case creates a temporary anonymous key within the device to derive Bluetooth IDs for broadcast. Bluetooth IDs and metadata to estimate proximity are derived from the temporary anonymous key, broadcast and renewed for very short intervals.

Devices in close proximity receive each other's ID and data and store them in the device. These records are not sent to a server nor to health authorities unless a user consents to send his or her information after he or she receives a positive diagnosis of infectious disease or is notifies as having been exposed to infection. The list of ID and metadata received in the device will be deleted after a period that has been epidemically proven to be safe.

The application could regularly receive list of temporary anonymous key of users who have been diagnosed with an infection. Then the application compares matches between the received list and records of recent proximity stored in the devices. If there is a match, it notifies to the user when he or she was exposed and how far the distance between devices was.

Once a user is diagnosed with the infection, the user can determine whether to send his or her information to the server or health authorities.

7.2.3 Security consideration

In decentralized use cases, the following are considered for security:

– The application should not store PII data nor location information other than proximity information.

– To detect proximity, it should use anonymous ID and random Bluetooth ID cryptographically derived from the anonymous ID.

– IDs should be renewed for the short term to make them unlinkable to previous IDs.

– The random Bluetooth ID should be cryptographically derived from the anonymous ID so that it cannot be inferred from the random Bluetooth ID.

– Users may decide on the application or transmission of their information.

– If a user is diagnosed with the infection or is notified as having been exposed to an infector, he or she may determine whether to send his or her information to the server or health authorities.
– Data should be stored and processed only within a device except for the propagation of the random Bluetooth ID, or unless a user has consented to transmit his or her information to the server.
– Information generated in the application should be strictly limited to be used for the prevention of infectious diseases.
– Information in the application should be deleted after a period that has been proven to be safe epidemically.

Appendix I

Practical use cases for contact tracing systems

I.1 Decentralized privacy-preserving proximity tracing [b-DP^3T]

The system in [b-DP^3T] represents a decentralized contact tracing approach. The main objectives of the system are to enable a quick notification of contacts at risk and to help epidemiologists to analyse the spread of the virus.

The system is based on the broadcast of identifiers (IDs) through Bluetooth Low Energy (BLE) by the user's smartphone. Therefore, nearby users are enabled to receive and store such IDs. If an infected person is detected, their smartphone is authorized to send their IDs to the backend, which in turn broadcasts the IDs to the users of the system. Then, each receiving user compares the received IDs against the list of stored IDs, and in the case of an ID match, the application notifies the user that they have been in contact with an infected person.

The system only requires a backend server and the users' mobile devices, where the corresponding application is installed. Furthermore, the existence of a health authority is assumed. Then, the following two main processes are defined:

i) Generation and storage of ephemeral IDs (EphIDs);

ii) Proximity tracing.

I.1.1 Generation and storage of ephemeral IDs (EphIDs)

Each mobile device broadcasts changing ephemeral IDs (EphIDs), which are sent through BLE beacons (advertisements). These IDs are generated from a secret key, SK_t, where t represents the current day. Furthermore, the same key is refreshed every day by using a hash function H, in such a way that:

$$SK_t = H(SK_{t-1}).$$

This is a hash chain scheme, meaning that if a key is compromised, then all the subsequent keys are revealed, but earlier keys are not revealed. Then, SK_t is used to derive a set of EphIDs by using a pseudorandom function (PRF), say, HMAC-SHA-256, and a pseudorandom generator (PRG), say, advanced encryption standard (AES) in counter mode: $\text{EphID}_1 \parallel \ldots \parallel \text{EphID}_n = \text{PRG PRF}(SK_t, \text{"broadcast key"})$.

To avoid location tracking, each EphID has a validity period of several minutes. EphIDs are received by nearby users through BLE advertisements. Then, each EphID is stored by these users together with an exposure measurement, e.g., signal attenuation, and the day when the beacon was received. This process is shown in Figure I.1. Furthermore, each user's application locally stores their own keys SK_t that were generated during the epidemiologically relevant time, e.g., 21 days.

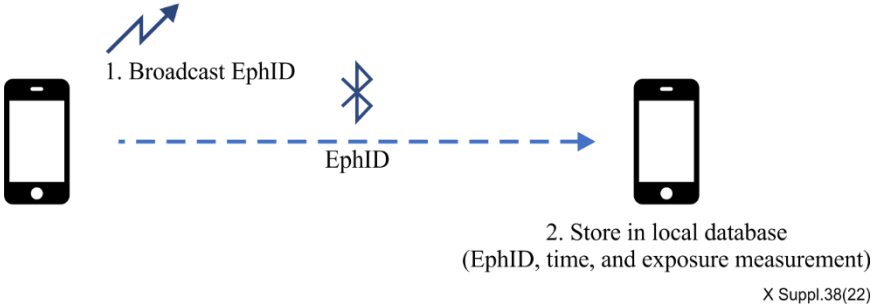

Figure I.1 – Generation and storage of ephemeral IDs (Redrawn from [b-DP^3T])

I.1.2 Proximity tracing

The process of proximity tracing illustrated in Figure I.1 is triggered when a user is diagnosed as infected by the health authority.

This authority is responsible for notifying test results, authorizing users to upload information to the backend server, and calculating the time during a patient is contagious, also known as the "contagious window". When a person is diagnosed as contagious and is certified by the health authority, the mobile device of the patient uploads the key SK_t and the first day t that they were considered to be contagious. The backend will receive a pair (SK_t, t) of each infected individual. The different (SK_t, t) pairs are periodically downloaded by registered users. It should be noted that the backend is only intended to broadcast this information, instead of processing any data. With this information, users are enabled to compute the list of EphIDs associated to a given (SK_t, t) pair. If one of these EphIDs is included in their stored list, it means the user was in contact with an infected person. Then, for each matching beacon, the data on receive time and exposure measurement is sent to an exposure estimation component, which is intended to estimate the duration of the smartphone owner's exposure to infected users in the past.

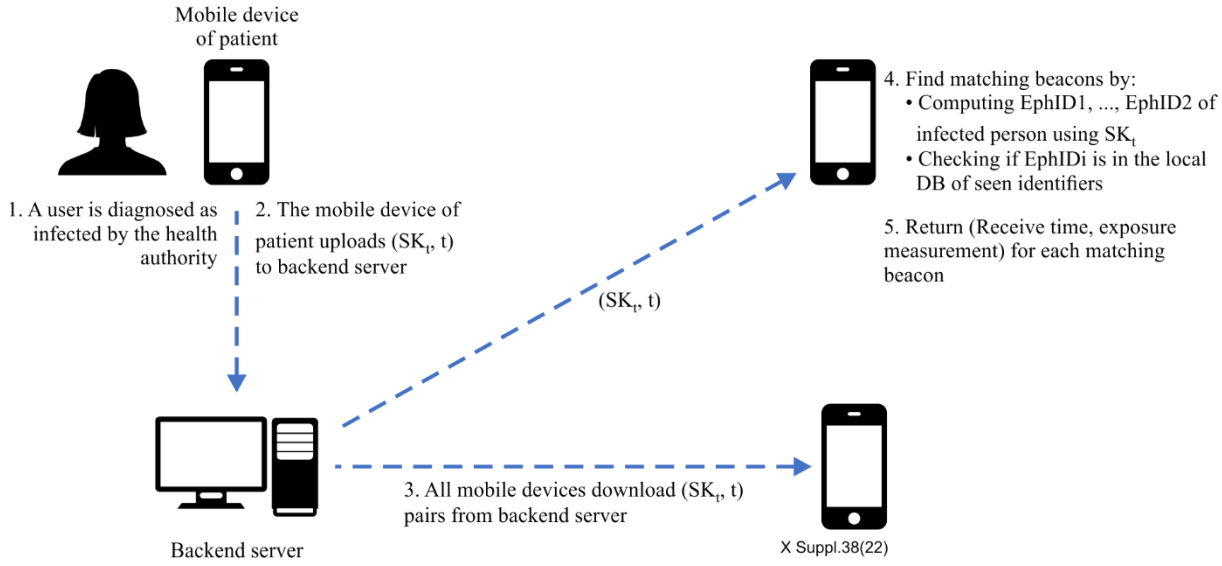

Figure I.2 – Process of proximity tracing (Redrawn from [b-DP^3T])

I.2 Centralized privacy-preserving proximity tracing [b-Martin], [b-PEPP-PT]

An example of a centralized tracing system is given in [b-PEPP-PT]. This system comprises the following components:

i) A user mobile application for proximity tracing;

ii) A backend server for generating temporary IDs used with the application and processing the data received by the application; and

iii) A push notification service to trigger the application to pull notification from the backend.

The interactions among the components are depicted in Figure I.3. These interactions are facilitated by the following protocols:

i) User registration;

ii) Proximity tracing of other mobile devices;

iii) Sharing collected proximity data with the server;

iv) Federation with other backends.

I.2.1 User registration

When a user installs this application, the backend server is always active. During user registration, a pseudonymous user ID is generated by the server and sent to the application. Since identifying attributes such as email accounts and phone numbers are not used in this scheme, a combination of a proof of work (PoW) and a CAPTCHA can be used in order to impede the mass creation of user accounts. The PoW makes registrations quite expensive and prevents denial-of-service (DoS) attacks by unauthenticated requests, while CAPTCHA requires human interaction. The registration steps are the following:

– The user requests to register to the backend;

– PoW and CAPTCHA challenges are sent to the application;

– The application computes the solution to the PoW challenge and the user solves the CAPTCHA;

– The two challenge results are sent to the backend and verified;

– The application receives client authentication credentials, i.e., random client ID and client secret key; and

– The backend server stores the application's client authentication credentials, a unique 128-bit random pseudonymous persistent user ID (PUID) and a push notification ID (PID).

After registration, when the application needs to communicate with the backend, it uses its client authentication credentials to retrieve a client access token. Then, the application uses this token to be authenticated by the backend server. The tokens are solely used for this authentication, and they are valid for a limited period of time. The authentication credentials are only used to issue access tokens. Whenever needed, the server uses the PUID to generate and send to the application one or a batch of pseudorandom temporary IDs.

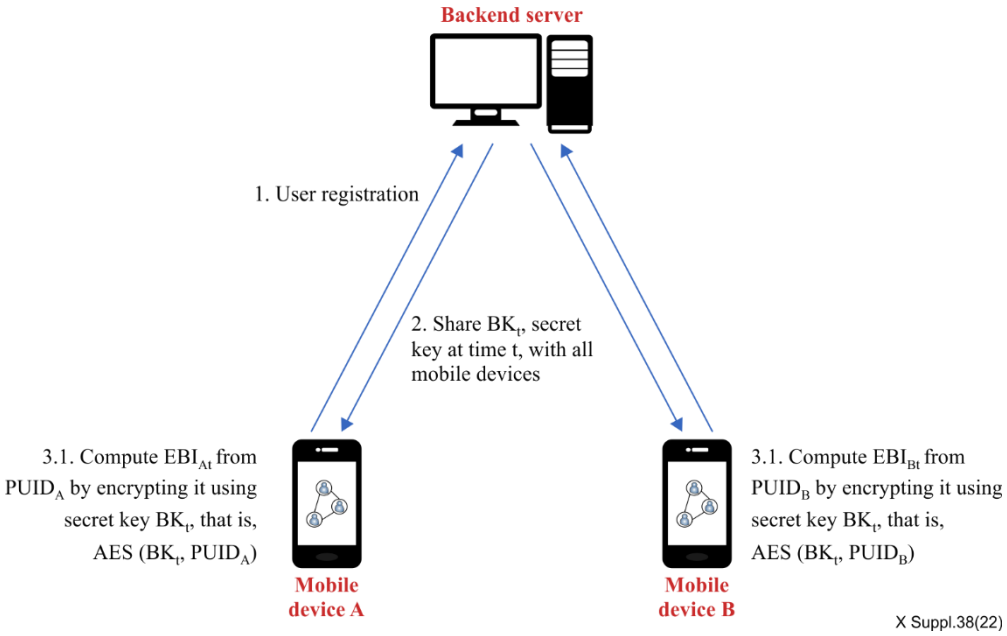

Figure I.3 – User registration and generation and storage of the encrypted pseudonymous persistent user ID(PUID) and encrypted PUID EBI_t (Redrawn from [b-PEPP-PT])

I.2.2 Proximity tracing

For every period *t*, for example, 1 h, the backend server generates a single secret key BK_t shared with all users. The backend server generates enough a set of BK_t keys to cover a larger period in the future, for example, 2 days. Then, the application generates an ephemeral BLE ID (EBID) for every *t*, by encrypting their PUID with BK_t.

$$EBID_t(\text{PUID}) = \text{AES}\ (BK_t, \text{PUID})$$

where $EBID_t$ is an encrypted PUID with BK_t using AES algorithm.

Each application broadcasts its current valid $EBID_t$ via BLE advertisements using the BLE privacy feature to prevent the tracking of users, who send out continuous BLE advertisements. Using this feature, temporary addresses instead of fixed hardware addresses are transmitted. The application implementation must use a new temporary address with every new EBID, to avoid linking of these two IDs.

Each application also constantly scans for other BLE broadcasts from [b-PEPP-PT] apps and records the received EBIDs, the current time and metadata of the BLE connection. The metadata include parameters such as the received signal strength indicator (RSSI) and outgoing and incoming signal levels (TX/RX power), which can assist in calculating the distance between the two communicating smartphones. The above data are stored only on the smartphone for as long as the user is not infected, and they are deleted after the epidemiologically relevant time, for example, 21 days.

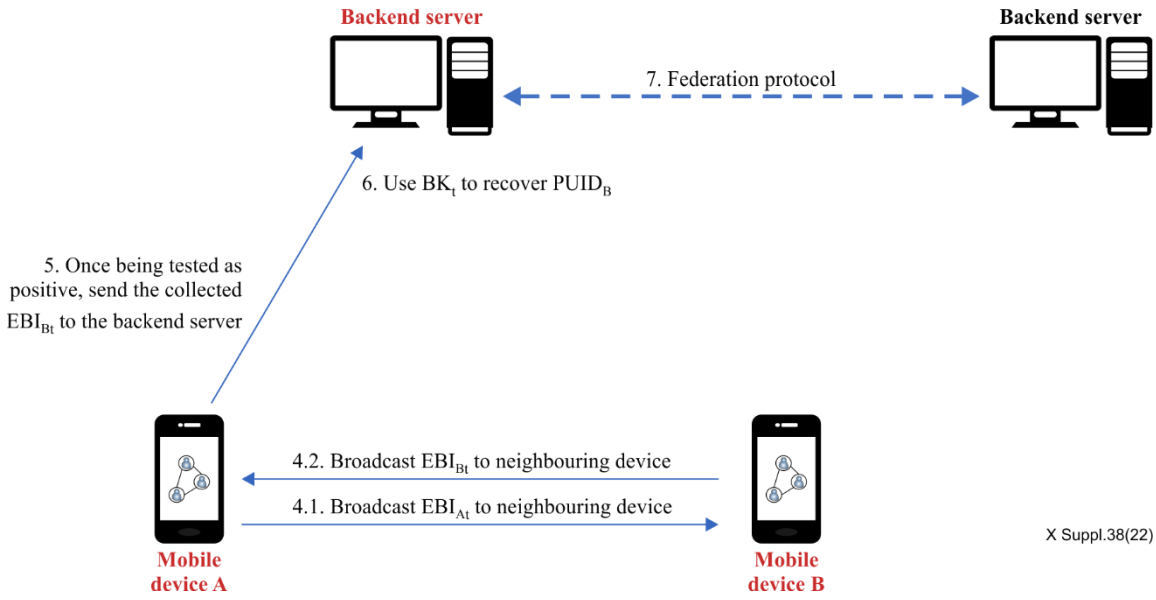

Figure I.4 – Sharing proximity data with the backend server and federation with other backend server (redrawn from [b-PEPP-PT])

I.2.3 Sharing proximity data with the server

When a user has tested positive, the collected data from adjacent mobile devices are sent to the backend server for assessing which other users are at risk and notifying them. The backend server holds these data for up to 3 weeks. To upload the data to the backend, a healthcare professional provides a transaction authentication number (TAN) to the infected user by out-of-band means.

The backend server associates each EBID received with its corresponding PUID and calculates the risk for the PUID holder.

To protect the privacy of infected users from eavesdroppers, in the German concept NTK proposed implementation PETT-PT [b-Fraunhofer AISEC], the backend server pushes notifications to infected users as well as to a random number of other user apps. The push notification acts as a trigger for the application to send a pull request to the backend. For users at risk, the pull request returns information to the user about potential infection and instructions. For the rest of the users, the exchanged messages are just "noise" and no information or instructions are provided by the app. In ROBERT (ROBust and privacy-preserving proximity Tracing protocol) [b-INRIA], a pure pull approach is followed where the application regularly inquires the backend server with its EBIDs. According to the risk assessment procedure run on the server, the application pulls a notification informing the user whether they are at risk or not.

I.2.4 Federation with other backends

To facilitate the federation of backend services, it is only necessary for a backend to recognize the originating backend of an EBID. This can be achieved by including an encrypted country code (ECC) into the EBID so that, for example, the ECC consumes 1 byte out of the 16 bytes available for the EBID. When a foreign backend receives an EBID that does not belong to it, it just forwards it to the home backend. The home backend is responsible to determine how the BK_t keys and EBID are constructed, as well as how the risk analysis is performed.

Bibliography

[b-ITU-T X.1058] Recommendation ITU-T X.1058 (2017), *Information technology – Security techniques – Code of practice for personally identifiable information protection*.

[b-ISO 7498-2] ISO 7498-2:1989 (1989), *Information processing systems – Open systems interconnection – Basic reference model – Part 2: Security architecture*.

[b-ISO 12812-1] ISO 12812-1:2017 (2017), *Core banking – Mobile financial services – Part 1: General framework*.

[b-ISO/IEC 20944-1] ISO/IEC 20944-1:2013 (2013), *Information technology – Metadata registries interoperability and bindings (MDR-IB) – Part 1: Framework, common vocabulary, and common provisions for conformance*.

[b-ISO/IEC 27000] ISO/IEC 27000:2018 (2018), *Information technology – Security techniques – Information security management systems – Overview and vocabulary*.

[b-ISO/IEC 29100] ISO/IEC 29100:2011, *Information technology – Security techniques – Privacy framework.*

[b-ETSI GR E4P 002] ETSI GR E4P 002 V1.1.1 (2021-02), *Europe for Privacy-Preserving Pandemic Protection (E4P); Comparison of existing pandemic contact tracing systems*.

[b-Apple] Apple (2020), *Apple and Google partner on COVID-19 contact tracing technology*. https://www.apple.com/newsroom/2020/04/apple-and-google-partner-on-covid-19-contact-tracing-technology/

[b-Apple & Google] Apple and Google (2020), *Exposure notification: Bluetooth Specification v1.2*. https://covid19-static.cdn-apple.com/applications/covid19/current/static/contact-tracing/pdf/ExposureNotification-BluetoothSpecificationv1.2.pdf

[b-Criddle] Criddle, C., and Kelio, L, *Coronavirus contact-tracing: World split between two types of app*. https://www.bbc.com/news/technology-52355028

[b-DP^3T] DP^3T, *Decentralized privacy-preserving proximity tracing*. https://github.com/DP-3T

[b-Fraunhofer AISEC] Fraunhofer AISEC, *"Pandemic Contact Tracing Apps: DP-3T, PEPP-PT NTK, and ROBERT from a Privacy Perspective,"* https://eprint.iacr.org/2020/489.pdf

[b-Lomas] Lomas, N. (2020), *EU privacy experts push a decentralized approach to COVID-19 contacts tracing*. https://techcrunch.com/2020/04/06/eu-privacy-experts-push-a-decentralized-approach-to-covid-19-contacts-tracing/

[b-Martin] Martin, T., Karopoulos, G., Hernández-Ramos, J.L., Kambourakis, G., and Fovino, I. N. (2020), *Demystifying COVID-19 Digital Contact Tracing: A Survey on Frameworks and Mobile Apps*, Wireless Communications and Mobile Computing, Vol. 2020, No. 8851429, pp. 29.

[b-INRIA] Institut National de Recherche en Informatique et en Automatique (INRIA), *Fraunhofer, "ROBERT – ROBust and privacy-preserving proximity Tracing protocol*," 16 April 2020, https://github.com/ROBERT-proximity-tracing.

[b-PEPP-PT] PEPP-PT, *Data protection and information security architecture*, https://github.com/pepp-pt/pepp-pt-documentation/blob/master/10-data-protection/PEPP-PT-data-protection-information-security-architecture-Germany.pdf

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	Tariff and accounting principles and international telecommunication/ICT economic and policy issues
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling, and associated measurements and tests
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	**Data networks, open system communications and security**
Series Y	Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2022