Privacy-Preserving Protocol Based On Bluetooth Encrypted Data Sharing

Wangzhihui Mei 2019124044 Hongyi Huang 2019180029 Zijia He 2019124057 Chang Xu 2019180034
Tianyu Jin 2019180030 Zhanping Zhou 2019124060 Senmiao Liu 2019180036 Caiming Qian 2019124036
CCNU-UOW JI

Abstract

At the critical moment of resisting COVID-19, governments and health authorities are working together to find solutions to the COVID19 pandemic, to protect people and get society back up and running across the world[6] . Given the widespread use of "health codes", this paper proposes a more reasonable privacy-preserving protocol based on Bluetooth data sharing. It is mainly constructed from short-distance Bluetooth low energy beacon data-transmission and some encryption methods.

1 Introduction

At present, new coronavirus is rampant around the world, and the whole world is facing a massive challenge. Even if medical technology is quite developed, the new coronavirus discovered for the first time in large-scale infections are still somewhat helpless.

Fortunately, China has found a suitable and reasonable way to resist COVID-19, in which the government take severe administrative measures to keep distance between citizens. To effectively control the epidemic, China has not only continuously developed vaccines from the technical aspect but also adopted a powerful political means: isolation. Quarantine dramatically reduces the spread of the epidemic. Then, the economy and society need to recover to regular order. To prevent the virus from spreading again, and better track the infected, accelerate the research of the vaccine, it is essential to develop a reasonable infected tracking system.

The "health code", as a pass for returning personnel, records the individual's health and takes the form of "green code", "red code", and "yellow code" to detect data dynamically. The appearance of the "health code" relieves personnel from carrying out a series of complicated auditing methods and is a significant anti-epidemic measure.

However, the "health code" still has the following problems:

- Firstly, out of the concern of privacy protection, people would not like to share their code with others.

- Secondly, the precise location is sensitive privacy. Therefore only the data of infected people are available for the server while those uninfected citizens data should not upload to the server.
- Finally, the requirements of statistical epidemic data seem to contradict to this situation.

Therefore, we should focus on designing a protocol that enables distributed data collecting and uploading. Some tricks are done on the client-side. The core idea is that the user can share data with closed ones and can get notified when they have ever been inclosed with infected. The cloud server only collects the infected ones' data.

2 Privacy-Preserving Protocol

In this chapter, we mainly introduce the protocol framework of the virus tracking system. The protocol was based on 3 round Symmetric-key encryption and Bluetooth low energy beacon data transmission.

2.1 Demand analysis

The application scenario firstly is going to be verified this section. Generally speaking, the Wechat platform is the container of the application. User data including ID number, location history and social connection can be fetched by related department such as department of health.

The typical scenarios are:

- The personal information of users should be registered through Wechat platform, including: name, ID number, address, phone number and so on.
- when entering or leaving some specific areas current status(ID number, location, time, etc.) should be updated achieving by forcing the user to scan QR code.
- User information is uploaded and stored in the cloud server. If sometime later an infection source is pinpointed, the related information(the person within the limited range and time window) can be traced, so that the status of the related persons health code should be turned to yellow or red instead of normal green.

The cloud server is not supposed to collect privacy data, so privacy data should be kept on user devices. Based on this concern, cloud server should be treated as untrusted terminal[1]. WeChat mini program act like front-end application to verify identification of user. The data system was actually a distributed data storage system. User-to-user data exchange should be implemented to be the solution of end to end communication.

In summary:

- The information exchanged itself is anonymous;
- Even if the privacy protection requirements are met, the information can still be decrypted when needed and the contact can be located.

2.2 Proposed Protocols

2.2.1 General Procedure

As the protocol was deployed on client, so the metadata was on user's phone. We use Bluetooth to transmit user's encrypted metadata of from other proximate users, which contains no privacy data. The metadata can be the Bluetooth MAC address or pseudorandom value[8], which can vary with time, which is referred to as BLE Metadata(BLEM). BLEM is encrypted to Encrypt Metadata with Encrypt Metadata key(EMK) derived from Original key generated periodically called Periodic Rolling Exposure Key(PREK) and Interval Proximity Identifier(IPI) associated as bluetooth payload. The IPI is inturn derived from PREK and a discretized representation of time. The IPI changes at the same frequency as the Bluetooth randomized address, to prevent linkability and wireless tracking. Nonuser identifying Encrypted BLE Metadata(EBD) is associated with IPI. The broadcast metadata from a user can only be decrypted later when the user tests positive.

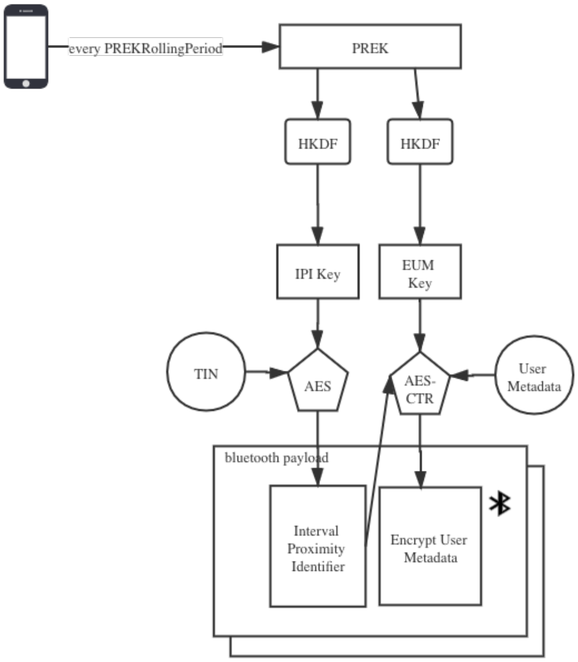

Figure 1: The 3-Phase Keygen and Encryption

In this protocol, the time is discretized in 15 minute intervals that are enumerated starting from Unix Epoch Time as period. The interval serial number is referred to as Tracking Interval Number(TIN). TIN determine which interval a timestamp is in.

Periodic Rolling Exposure Keys roll at a frequent named PREKRollingPeriod. This is usually set to 96(quarter hour), determining a key validity of 24 hours. Each key is randomly and uniquely generated with a cryptographic random number generator(CRNG). All devices sharing the same PREKRollingPeriod at the same time at the beginning of an interval whose TIN is a multiple of PREKRollingPeriod.

When a user tests positive in medical checking point, a limited set of PREK and their respective TIN(describing when their validity started) are uploaded to the Diagnosis Cloud Server. This set of PREK is limited to the time window in which the user could have been exposing other users(we usually set to 20 days). This set is called Diagnosis Key Set(DKS). If a user remains healthy and never tests positive, their PREKs will never been uploaded to server. When user refresh their health code, he will receive the DKS the diagnosis server aggregates from all positive infected. The diagnosis also distributes them to all the user clients that are participating in exposure notification.

To identify any exposures, each client periodically or proactively fetches the list of new DKS within the quarantine period from the Diagnosis Server. Because Diagnosis Keys are sets of Periodic Rolling Exposure Keys with their associated TIN, each of the clients can again derive the sequence of IPI that were broadcast over Bluetooth from users who tested positive. Then, the clients match each of the derived identifiers against the sequence they found received from Proximity Bluetooth. The Encrypted User Metadata does not have to be decrypted until a match occurs. Upon decryption, the data has to be appropriately sanitized and validated as the EUM isnt authenticated.

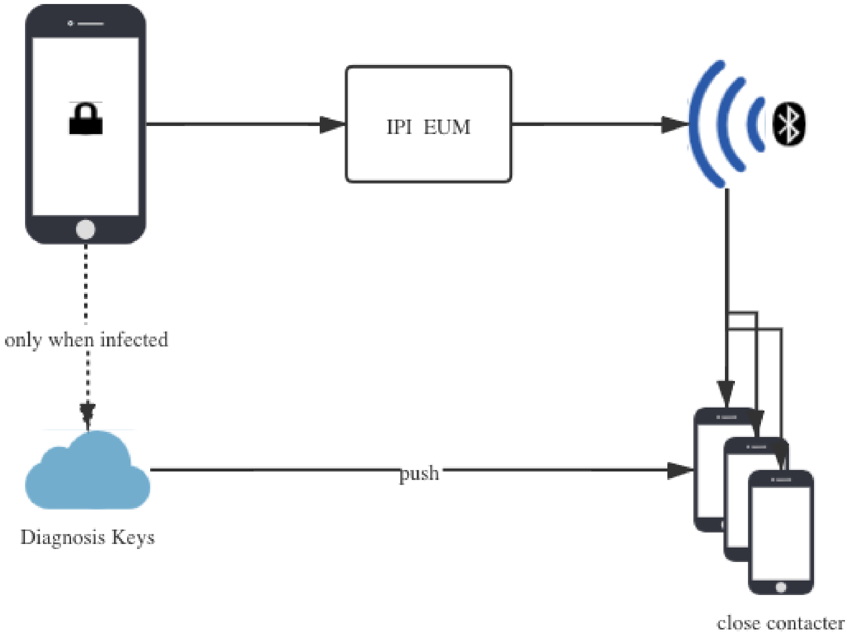

Figure 2: The 3 Code Interaction

2.2.2 Mathematic Detail

HKDF

HKDF [2]designates the HKDF function as defined by IETF RFC 5869, using the SHA-256 hash function:

$$Output \leftarrow HKDF(key, Salt, Info, OutputLength)$$

AES

AES [4]designates the encryption of a single AES-128 block

$$Output \leftarrow AES_{128}(Key, data)$$

CRNG

The CRNG [3]function designates a cryptographic random number generator:

$$Output \leftarrow CRNG(OutputLength)$$

Tracking Interval Number

Tracking Interval Number(TIN) offers a number for each 15 minute period thats shared between all devices participating in the protocol. These time windows are derived from timestamps in Unix Epoch Time. TIN is encoded as a 32-bit unsigned little-endian value.

$$TIN(timestamp) \leftarrow timestamp/(60 \times 15)$$

PREKRollingPeriod

PREKRollingPeriod(PREKP) is the duration determining the duration a Periodic Rolling Encryption Key is valid(in multiples of 15 minutes). In our protocol, PREKP is defined as 96(quarter hours), achieving a key validity of 24 hours.

Periodic Rolling Exposure Key

When setting up the device for exposure detection, the first Periodic Rolling Exposure Key(PREK) is generated on the device and associated with an TIN , corresponding to the time from which the key is valid. That value is aligned with the PREKP and is derived as follows:

$$i \leftarrow \lfloor TIN(timestamp)/PREKP \rfloor \times PREKP$$

The devices generate the 16-byte Temporary Exposure Key as follows:

$$PREK_i \leftarrow CRNG(16)$$

The key is securely stored along with i. At the end of every PREKRollingPeriod, a new key is generated.

Interval Proximity Identifier Key

Interval Proximity Identifier Key(IPIK) is derived from the Periodic Rolling Exposure Key and is used in order to derive the Rolling Proximity Identifiers[5].

$$IPIK_i \leftarrow HKDF(PREKP_i, NULL, UTF8(\text{''}IPIkey\text{''}), 16)$$

Interval Proximity Identifier

Interval Proximity Identifiers are broadcast in Bluetooth payloads with privacy-preserving functionalities.

the protocol generates a new IPI as long as Bluetooth metadata changed.

$$IPI_{i,j} \leftarrow AES_{128}(RPIK_i, 0||TIN_j)$$

Where the length of $0||TIN_j$ is 128bit.

Encrypted User Metadata Key The Encrypted User Metadata Keys(EUMK) are derived from the Periodic Rolling Exposure Keys in order to encrypt additional metadata.

$$EUMK_i \leftarrow HKDF(PREK_i, NULL, UTF8(\text{"}EUMKey\text{"}), 16)$$

Encrypted User Metadata The Encrypted User Metadata(EUM) is data encrypted along with the Interval Proximity Identifier, and can only be decrypted later if the user broadcasting it tested positive and reveals their Periodic Rolling Exposure Key[7].

$$EUM_{i,j} \leftarrow AES_{128} - CTR(EUMK_i, IPI_{i,j}, BLEMetadata)$$

2.2.3 Inter-Device Data Transmittion

In our protocol, the client uses Bluetooth low energy beacon to transmit Bluetooth payload containing Encrypted User Metadata and Interval Proximity Identifier to other proximate receivers. Mobile phone offers API for WeChat to use Bluetooth Data transmission.

Similar to WiFi device discovery, the protocol is based on advertising[10], a.k.a. Bluetooth payloads that our device sends out to anyone within reach, and scanning, which is receiving and reading other devices advertisements.

In Bluetooth communication, two devices use an association model to authenticate each other, and then exchange data securely. Bluetooth pairing is a key exchange process; however, before exchanging keys, the two devices must share internal pairing parameters including authentication requirements[9].

In order to achieve authentication, the two devices must use a certain correlation model for mutual authentication. Before Bluetooth 4.2, several authentication algorithms have fatal flaws. Hackers can use the vulnerability of MITM to pretend to establish a connection between the slave device and the master and obtain Transmit data and easily control your device by brute-forcing application layer transfer protocols.

Pairing refers to the process of authentication and key exchange between two BLE devices. The pairing process includes three stages, and LE secure connection refers to an enhanced security function introduced in the Bluetooth 4.2 specification. It uses a Federal Information Processing Standard (FIPS) algorithm (also called Elliptic Curve Diffie-Hellman (ECDH)) for generating keys and a new program for key exchange. The associated model in the BLE application refers to the model that determines the pairing method based on the input and output functions of the two BLE devices. And the Bluetooth 4.2 specification introduces a new correlation model, the digital comparison (NC).

Bluetooth pairing is divided into three stages: parameter setting, authorization and key generation, and key interaction.

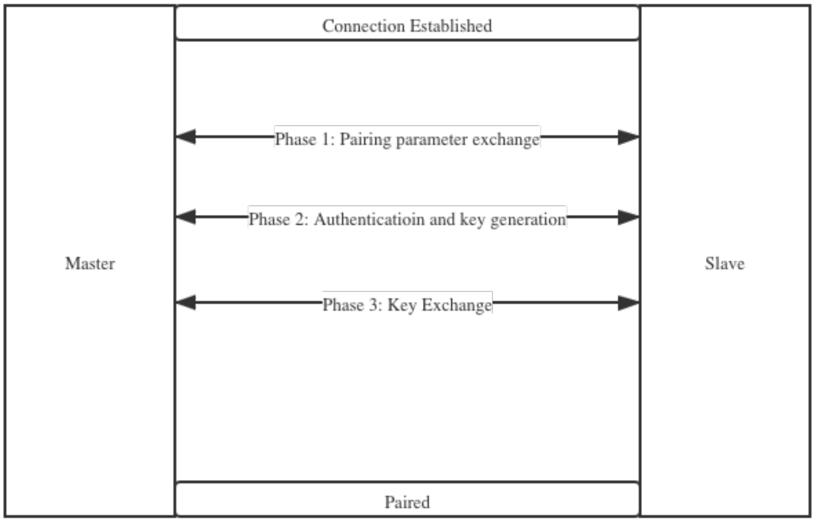
Connection Established
Phase 1: Pairing parameter exchange
Phase 2: Authenticatioin and key generation
Master
Slave
Phase 3: Key Exchange
Paired

Figure 3: Bluetooth pairing

Parameter setting

Initialize the device and respond to the device to exchange coupling parameters (such as input / output functions, certification requirements flags, encryption key size, and availability of OOB data). The first pairing stage of LE traditional pairing and LE secure connection is the same.

Response devices (such as interconnect layer slave devices) will respond by coupling response commands. The responding device can also initiate the pairing process through the security request command.

Authorization and key generation

The second pairing phase includes authentication to prevent man-in-the-middle (MITM) attacks and key generation to encrypt BLE links. In LE's traditional pairing, a temporary key (0: Just Works model, 6 digits or 20 digits: Passkey Entry model, 128 digits: OOB model) can be used to obtain a key that can encrypt the BLE link. The temporary key is random data that is not exchanged through the BLE link.

In the public key exchange phase, the initializing device and the responding device will exchange their public keys and begin to calculate the Diffie-Hellman key. You can use the Elliptic-Curve Diffie-Hellman function (P256) to generate Diffie-Hellman keys. The P256 takes as input the specific key and public key of the peer device. The Diffie-Hellman key is never exchanged, it can provide random 256-bit data. In authentication phase 1, each BLE device will perform mutual authentication to prevent man-in-the-middle (MITM) attacks (specific to BLE4.2). The authentication stage 1 of each associated model is different. Any failure during authentication phase 1 will terminate the pairing process. In authentication phase 2, each

BLE device will calculate the Long Term Key (LTK), which is used to encrypt the link. You can use the Diffie-Hellman key generated in authentication phase 1, the Bluetooth device address, and a 128-bit random number and use these devices' I / O functions to generate LTK.

Key interaction

In the key exchange phase, you can use the STK (if using LE traditional pairing) or LTK (if using LE secure connection pairing) generated in the authorization and key generation phase to encrypt the BLE link.

3 Discussion

3.1 Security

In our protocol, only infector's Periodic Rolling Exposure Key will upload to the cloud server. So it is impossible for the server to collect uninfected privacy data. For infector, their exposed Periodic Rolling Exposure Key will leak little privacy as the key schedule is fixed and defined by operating system components, preventing applications from including static or predictable information that could be used for tracking.

When the user receives others' Interval Proximity Identifiers, a Periodic Rolling Exposure Key is required to correlate between these IPIs. Because the IPIs vary within a short period. The probability of cracking the IPI is negligible, which reduces the risk of privacy loss from broadcasting the identifiers. Without the knowledge of Periodic Rolling Exposure Keys, it's computationally infeasible for an attacker to find a collision on an Interval Proximity Identifiers. This prevents a wide range of replay and impersonation attacks.

When reporting Diagnosis Keys, the correlation of Interval Proximity Identifiers by others is limited to 24 hour periods due to the use of Periodic Rolling Exposure Keys that change daily. The server must not retain metadata from clients uploading Diagnosis Keys(Periodic Rolling Exposure Keys of infectors) after including those key in the aggregated list of Diagnosis Keys per day.

The party-to-party communication a.k.a Bluetooth data transmission is based on authenticated Bluetooth pairing. It can prevent Man-In-The-Middle-Attack.

3.2 Properties and functionalities

As every mobile phone is equipped with Bluetooth low-power beacon, the protocol development is possible. The Bluetooth data transmission transmits 16 Byte data each time, the total data received is not supposed to exceed 1MB each day. It will cost no more than 20MB in a quarantine period to take up the phone's storage. Besides, the power consumption can be little, so it will not reduce the user's device battery life remarkably.

The government can pin the location of infector through distributed awareness system, which collects data of those who have ever been inclosed with infectors. With these data, it's easy to draw the movement trajectory of infector in the short term. The WeChat also offers location history. Due to the Catastrophic consequences, COVID-19 has caused, most people will be willing to accept this level of surveillance to prevent the virus from spreading temporarily. Besides, Statistical data can be obtained from Wechat client as people need to refresh their health data periodically.

4 Conclusion

In this article, we propose a privacy-preserving protocol based on BluetoothEncrypted data sharing and verified its security and functionalities. The server only records the diagnosis key of the positive infected person and does not get privacy information from it. The data exchange based on Bluetooth low energy beacon is high-efficiency and low-power. It is easy to built taking WeChat Health Code App as frontend. To a certain extent, it can play the role of epidemiological traceback and infection risk alert.

References

[1] **Berke, Alex, Michiel Bakker, Praneeth Vepakomma, Ramesh Raskar, Kent Larson, and AlexSandy' Pentland**, "Assessing disease exposure risk with location histories and protecting privacy: A cryptographic approach in response to a global pandemic," *arXiv preprint arXiv:2003.14412*, 2020.

[2] **Brief, Product**, "XIP3322B: HKDF/HMAC/SHA-256."

[3] **Datcu, Octaviana, Corina Macovei, and Radu Hobincu**, "Chaos Based Cryptographic Pseudo-Random Number Generator Template with Dynamic State Change," *Applied Sciences*, 2020, *10* (2), 451.

[4] **Gueron, Shay, Wajdi K Feghali, Vinodh Gopal, Raghunandan Makaram, Martin G Dixon, Srinivas Chennupaty, and Michael E Kounavis**, "Flexible architecture and instruction for advanced encryption standard (AES)," February 4 2020. US Patent 10,554,386.

[5] **Hirmer, Benedikt M, Kevin Bessiere, and Eric Circlaeys**, "Techniques for disambiguating clustered location identifiers," March 12 2020. US Patent App. 16/219,602.

[6] **jie Guan, Wei, Zheng yi Ni, Yu Hu, Wen hua Liang, Chun quan Ou, Jian xing He, Lei Liu, Hong Shan, Chun liang Lei, David SC Hui et al.**, "Clinical characteristics of coronavirus disease 2019 in China," *New England journal of medicine*, 2020, *382* (18), 1708–1720.

[7] **Keselman, Gleb, Yaron Sheffer, and Alon Rosen**, "Homomorphic key derivation," April 9 2020. US Patent App. 16/153,414.

[8] **Reichert, Leonie, Samuel Brack, and Björn Scheuermann**, "Privacy-preserving contact tracing of covid-19 patients," 2020.

[9] **Scarfone, Karen and John Padgette**, "Guide to bluetooth security," *NIST Special Publication*, 2008, *800* (2008), 121.

[10] **Tang, Qiang**, "Privacy-Preserving Contact Tracing: current solutions and open questions," *arXiv preprint arXiv:2004.06818*, 2020.