

# Privacy-Preserving Infection Exposure Notification without Trust in Third Parties

Kenji Saito, Mitsuru Iwamura \*

## Abstract

In response to the COVID-19 pandemic, Bluetooth-based contact tracing has been deployed in many countries with the help of the developers of smartphone operating systems that provide APIs for privacy-preserving exposure notification. However, it has been assumed by the design that the OS developers, smartphone vendors, or governments will not violate people's privacy.

We propose a privacy-preserving exposure notification under situations where none of the middle entities can be trusted. We believe that it can be achieved with small changes to the existing mechanism: random numbers are generated on the application side instead of the OS, and the positive test results are reported to a public ledger (e.g. blockchain) rather than to a government server, with endorsements from the medical institutes with blind signatures. We also discuss how to incentivize the peer-to-peer maintenance of the public ledger if it should be newly built.

We show that the level of verifiability is much higher with our proposed design if a consumer group were to verify the privacy protections of the deployed systems.

We believe that this will allow for safer contact tracing, and contribute to healthier lifestyles for citizens who may want to or have to go out under pandemic situations.

**Keywords:** Contact tracing, Exposure notification, Privacy,  
Blind signature, Blockchain

## 1 Introduction

### 1.1 Motivation

Contact tracing is a technique traditionally used by public health authorities to combat infectious diseases, which until now has relied primarily on manual methods. It is based on the concept of ascertaining others with whom the

---

\*The authors are with Graduate School of Business and Finance, Waseda University, email: ks91@aoni.waseda.jp

infected person has come into contact while there is a possibility of spreading the infection to others. The possibility of close contact with an infected person is notified to the person with whom the contact has been made, so that appropriate safety measures can be taken, such as self-quarantine and/or testing.

During the ongoing pandemic of COVID-19, consideration was given to applying digital communication technology to support and massively scale these efforts. By embedding proximity-detection functionality into mobile devices, it is considered possible to identify past close contacts of people who later test positive, and send them notifications with instructions on next steps. Health authorities can use this information to control the spread of the disease.

With this in mind, Bluetooth-based exposure notification (contact tracing) has been proposed. Applications have been developed in many countries for use on smartphones with the help of the developers of smartphone operating systems, namely Google and Apple, who provide APIs for privacy-preserving exposure notification. However, in the designs of these applications, it has been assumed that the OS developers, smartphone vendors, or governments will not violate people's privacy. We believe that better privacy protection without relying on the integrity of these trusted third parties is needed.

## 1.2 Contributions

Contributions of this work are as follows:

1. We identified threats that intermediaries could pose, alone or in collusion, with respect to privacy protection of the users in the existing design of the OS-assisted exposure notification.
2. We proposed a design to mitigate the threats by minimal changes to the existing design, along with an incentive design for the operation of the system in case a verifiable public ledger should be newly built independently of the government authority.
3. We showed that the level of verifiability is much higher with our proposed design if a consumer group were to verify the privacy protections of the deployed systems.

## 1.3 Organization of This Article

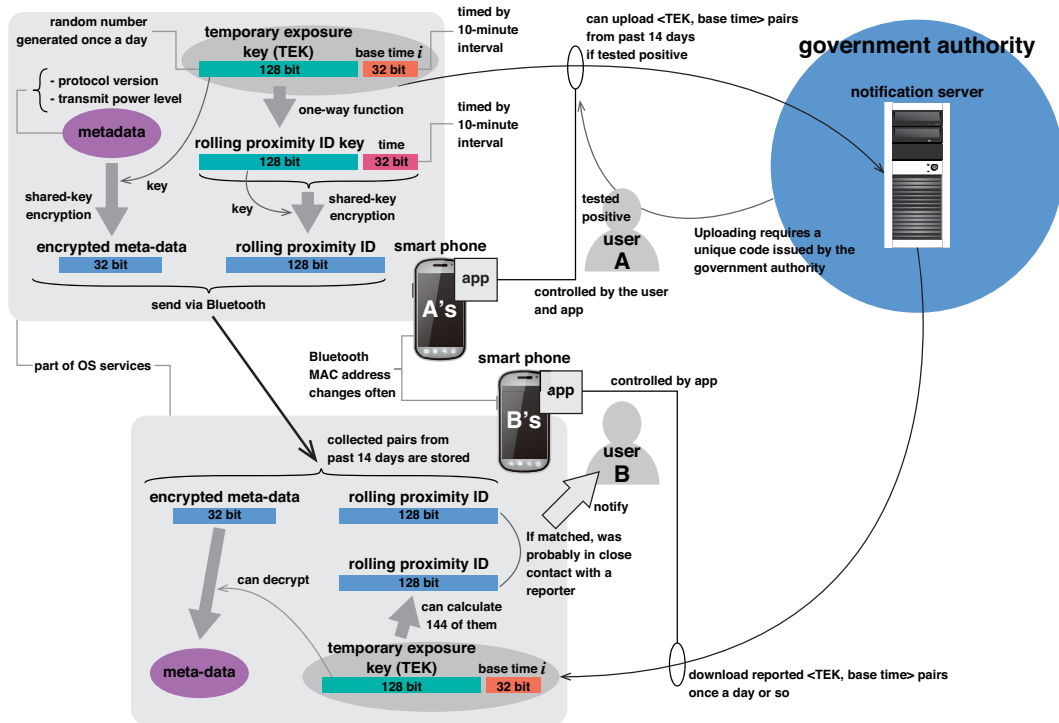
The rest of this article is organized as follows: section 2 gives brief background information to better understand our proposal: exposure notification design by Google and Apple, Merkle accumulator, blind signature, and blockchain. Those readers who are familiar with the mentioned technology

can move directly to section 3, which gives the problem statement and enumerates threats. Section 4 describes our proposal to change the existing design. Section 5 compares our proposed design with the original one with respect to verifiability of privacy protection. Section 6 explains some related work. Finally, section 7 gives conclusive remarks.

## 2 Background

### 2.1 Exposure Notification Design by Google and Apple

Google and Apple provide exposure notification as part of their OS services[3], as roughly illustrated in Figure 1. This design is often referred to as GAEN (Google Apple Exposure Notification) in literatures. We follow the convention hereafter. The cryptographic specification[2], Bluetooth specification[1] and the programming framework[10]<sup>1</sup> of GAEN have been published.



\* The application is developed by the government.

Figure 1: Overview of OS-assisted exposure notification.

<sup>1</sup>This reference is for Google Android API.

In GAEN, the time is numbered in 10-minute intervals, starting at 00:00:00 UTC on January 1, 1970 (Unix epoch).

The device generates one random *temporary exposure key (TEK)* (128 bits) every 24 hours. It is valid for 144 time units (24 hours) starting from time  $i$ . The device stores up to 14 TEKs (for two weeks) with their respective starting time  $i$ .

The device generates the *rolling proximity ID key* (128 bits) from the TEK of the day using a one-way function. Along with the timing when the MAC address of Bluetooth LE is changed, a *rolling proximity ID* (128bit) is generated by encryption from the rolling proximity ID key and the time, which is sent by Bluetooth communication as a beacon. Other smartphones listen to these beacons, storing them upon receiving them, and broadcast their own beacons as well.

The system can also attach encrypted metadata to the beacon, such as protocol version and transmit power level, which can be decrypted with (the key generated from) the TEK of the day.

The device owned by a person who tests positive and reports voluntarily sends, for example, the TEKs for the past 14 days and their starting time  $i$  to the notification server.

All users will periodically download those reported data (e.g. once a day). The user's device can recalculate the rolling proximity ID from the TEK and  $i$ . One TEK can create 144 rolling proximity IDs. If the same rolling proximity ID obtained by the calculation is stored on your device, then you have a high probability that you were in close contact.

If the device has received the encrypted metadata along with the rolling proximity ID, it can decrypt the data using the TEK, although there is no guarantee that the information is correct. This means that we must also consider the existence of other applications and/or malware that perform the same Bluetooth communication.

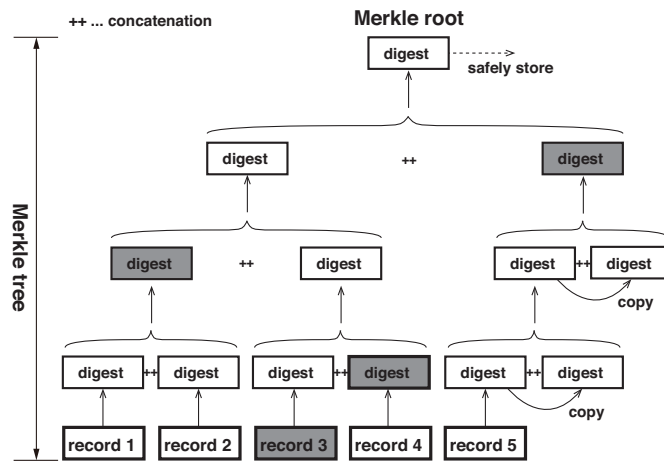
In the designs of these applications, it has been assumed that the OS developers, smartphone vendors, or governments will not violate people's privacy.

## 2.2 Merkle Accumulator

A Merkle tree[17] is a hash tree structure based on a cryptographic hash function that produces cryptographic digests. Such a tree allows representation of multiple elements with a single value, and is used for proof of existence of elements while obscuring others, as illustrated in Figure 2.

A Merkle accumulator is a Merkle tree to which elements can be added incrementally, to accumulate evidences of records.

In our work, we use a Merkle accumulator to record with verifiable evidences the reports from people tested positive.



\* In order to confirm the existence of record 3, see if the same Merkle root as safely stored can be calculated from the provided partial tree (Merkle proof) shown in gray.

Figure 2: Merkle tree and Merkle proof.

### 2.3 Blind Signature

Blind signature[6] is a technique for digitally signing hidden data as if it were signed blindfolded, as illustrated in Figure 3. It was developed to enable anonymous electronic payments.

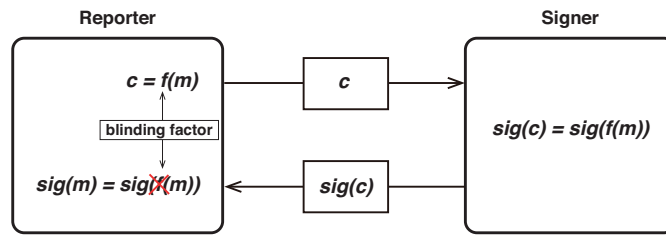
In our work, we apply blind signatures to reports from people tested positive, signed by the medical institute that performed the test.

### 2.4 Blockchain

Blockchain is a structure introduced for realization of Bitcoin[18], a digital cash system. It allows for tamper-evident storage verifiable by the public so that it can provide verifiability of digital signatures in the past[21] that is otherwise difficult because of possible compromise of private keys or signature algorithms, or expiration of public key certificates.

As Figure 4 shows, each block contains the cryptographic digest of the previous block. Such a digest must meet a certain criterion; it needs to be less than or equal to the pre-adjusted and agreed target stored in or calculated from the block. Since the digest is calculated by a one-way function whose outputs are evenly distributed, no one can intentionally configure a block to satisfy the criterion. Instead, they need to partake repetitive trials to change the values of some nonce in the block they are creating until they get a right digest.

The necessity of repetitive trials functions as a *proof-of-work* mechanism



\* The reporter wants to have message  $m$  signed by the signer without revealing  $m$ . They first wrap  $m$  with blinding factor  $f$ , and send it to the signer for signing. After receiving the signature, they remove  $f$  from it to obtain the signature for  $m$  verifiable with the signer's public key.

Figure 3: Blind signature.

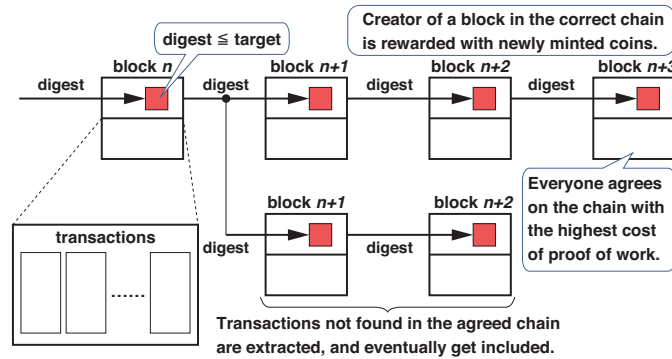


Figure 4: Blockchain based on proof of work.

intended to be a protection against falsification. A transaction itself cannot be falsified unless digital signatures are compromised. But it is conceivable to remove some transactions from a past block or to add fabricated transactions that did not exist at the time to it. If one tries so, the digest of the block is changed and is typically greater than the target. Then they would have to retry the proof of work for the block. This changes the digest stored in the next block, which in turn means that the digest of the next block is also changed and is typically greater than the target, and so on. In short, ones with a malicious intention would have to redo the proof of work from where they want to change, and outdo the ongoing process of adding blocks eventually to make the change valid, which has generally been considered highly difficult.

Such proof of work limits the number of proposed blocks at one time. But there still is a possibility of multiple participants each proposing a new block at roughly the same time, which may be accepted by different sets

of participants. Then the chain may have multiple ends that are extended independently from one another, resulting in a fork of the blockchain with multiple (and possibly contradicting) histories of blocks. If this happens, the branch that is the most difficult to produce (or rewrite) is chosen by all participants, which is the branch with the most accumulated proof of work. This mechanism, called *Nakamoto consensus*, tries to enforce that the most difficult chain branch to falsify is chosen as the single correct history<sup>2</sup>.

Ethereum[5] is a blockchain-based application platform to assure authenticity of program codes (*smart contracts*), their execution logs and the resulted states.

Both Bitcoin and Ethereum are based on proof of work, which is protected by the high power costs associated with it, but these costs are known to balance in the long run with the market value of the respective native currencies earned through block creation[13]. In other words, these blockchains are protected by the high market value of their respective cryptocurrencies.

In order to avoid the environmentally burdensome power costs of proof of work, recent blockchains such as Ethereum 2.0[9] and Polkadot[27] have adopted the idea of *proof of stake*, in which legitimate history is determined by weighted voting with a deposit of native currency. However, then again, the high market values of the respective native currencies are still the major factors that keep these blockchains safe.

Both Ethereum 2.0 and Polkadot allow multiple private ledger applications to be *anchored* to their central blockchains in the form of *shards* and *parachains*, respectively.

In our work, we store Merkle roots of reports in blockchain.

### 3 Problem

This work proposes a privacy-preserving exposure notification mechanism that tolerates situations where none of the middle entities can be trusted not to violate people's privacy. The solution must mitigate the following threats to privacy of the users that are present in GAEN, where *private data* denotes that of the phone user, such as the phone number, e-mail address, physical location, real name, etc.

1. OS developer and/or smartphone vendor alone can :
  - (a) encode private data or a marker in a TEK.
  - (b) send an arbitrary beacon containing private data or a marker.
  - (c) encrypt private data or a marker as the associated metadata.

---

<sup>2</sup>For imperfection of the design of Nakamoto consensus, readers are referred to a past work[23] by the first author of this paper.

- (d) collect identities of the close contacts with the user associated with such private data or a marker.
  - (e) notify exposures falsely to any specific users to stop or slow down their social activities.
2. The government alone can :
    - (a) collect identities of the reporters.
    - (b) stop or slow down social activities of political enemies, for example, by bringing agents close to them, and having the agents later report falsely.
  3. OS developer and/or smartphone vendor and the government can collude to :
    - (a) make fake reports from specific users, and have them downloaded by general public, to stop or slow down social activities of the close contacts with the users.

Threats 1a, 1b and 1c are possible because generation of TEKs, deriving rolling proximity IDs from them, and metadata are processed within the OS services, and the OS can send arbitrary beacons anyway, which is not detectable as applications do not know the values of TEKs until the users decide to report. Threats 1d and 1e are possible because what is performed as a result of receiving a beacon is hidden within the OS services.

Threat 2a is possible because it is the government authority that issues the unique code to the person tested positive, and it can be designed to map the code to the person. Threat 2b is possible because it is the government authority that processes the reports, and they can allow agents to bypass the normal reporting procedures to send their TEKs to the server, as the application is developed by the government, possibly with some hidden features.

Moreover, threat 3a is possible because the government authority can obtain the TEKs of a specific person from OS developers or phone vendors<sup>3</sup>, and store them on the server.

## 4 Design

### 4.1 Basic Design

We believe that the above threats can be mitigated with the following three small changes to the existing mechanism, as illustrated in Figure 5:

---

<sup>3</sup>According to Google API, applications cannot obtain TEKs without displaying a dialog that requests consent from the user. Therefore the government needs cooperations from the OS developer or phone vendor if they want to obtain the TEKs without letting the user know it.



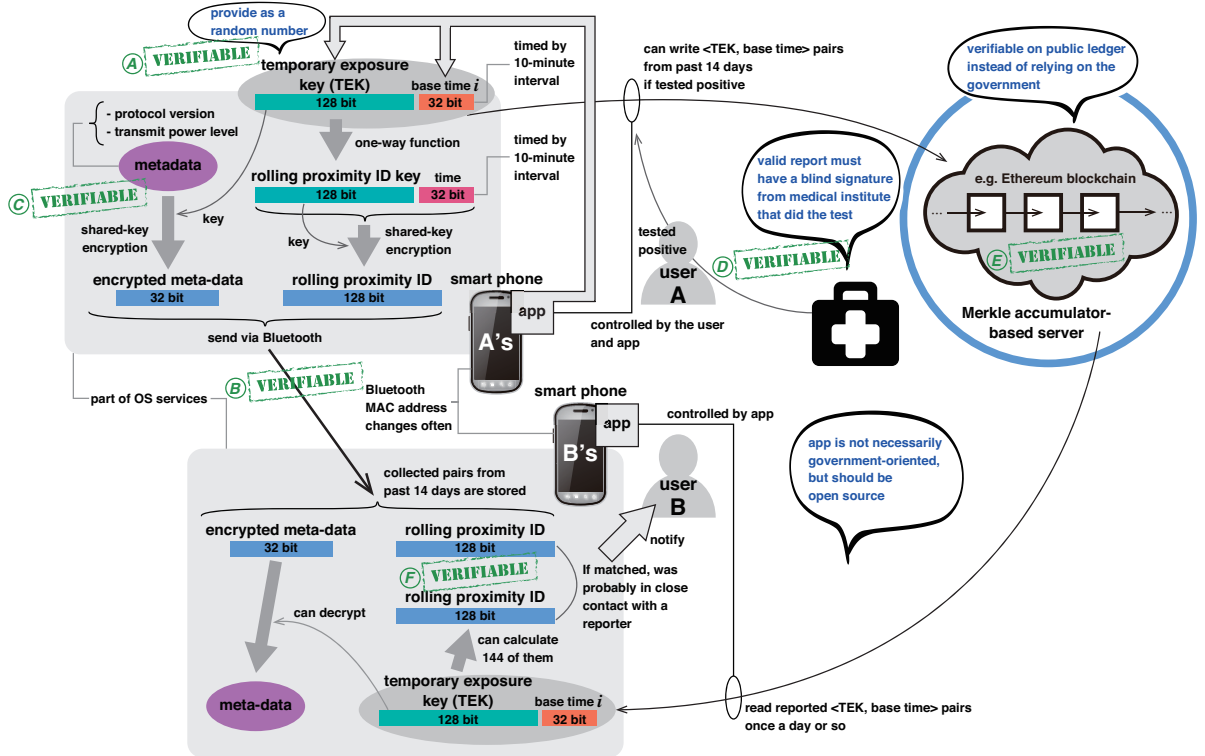


Figure 5: Overview of exposure notification without trust in third parties.

Furthermore, the applications should be open source so that they can be built and installed on smartphones by the phone users if they want.

#### 4.1.1 Generation of TEKs at the Application Side

We change the distribution of the functionality so that TEKs are generated as random numbers on the application side instead of within the OS services.

#### 4.1.2 Endorsement by Medical Institutes with Blind Signatures

The positive test results must be reported with endorsements (blind signatures) from the medical institutes who did the test. In order to avoid correlation, each  $\langle \text{TEK}, \text{base time} \rangle$  pair is blind-signed independently. The uploaded report takes the form of  $\langle \text{TEK}, \text{base time}, \text{signature}, \text{MIID} \rangle$  for each TEK, where MIID is the medical institute ID. The public key certificate used for verifying the signature can be retrieved with MIID as the search key from a public database.

Typically, when a person is informed of a positive test result at a testing facility, they are also informed of a short-term URL via a QR code, for

example. When the person accesses the URL on their smartphone, the exposure notification application sends 14 blindfolded  $\langle \text{TEK}, \text{basetime} \rangle$  pairs to the medical institute’s server, which returns blind-signed results to the application. The person can then choose to report them.

### 4.1.3 Storage of Reports in a Verifiable Public Ledger

The positive test results are reported to a verifiable public ledger (e.g. blockchain) rather than to a government server.

Assuming blockchain is used as the core of a verifiable public ledger, Merkle accumulator is applied by the server to lower the cost (transaction fees and processing time) of using blockchain. To prevent correlation, reports arriving at close timing are mixed with other reports and shuffled before being added to the accumulator. A Merkle tree is finalized every 24 hours, for example, to form a set of reports to be downloaded by the application. Our method is roughly the same as that presented in [25], using a similar or the same smart contract for Ethereum as shown in Figure 6, which returns the number of the block when the Merkle root was written, so that we can know approximately what time the root was stored.

```
contract Anchor {
  mapping (uint256 => uint) public _digests;
  constructor () public {
  }
  function getStored(uint256 digest) public view returns (uint block_no) {
    return (_digests[digest]);
  }
  function isStored(uint256 digest) public view returns (bool isStored) {
    return (_digests[digest] > 0);
  }
  function store(uint256 digest) public returns (bool isAlreadyStored) {
    bool isRes = _digests[digest] > 0;
    if (!isRes) {
      _digests[digest] = block.number;
    }
    return (isRes);
  }
}
```

- \* store() saves the current block number for a stored digest.
- \* For a given digest, getStored() returns the block number if it is stored. It returns 0 otherwise.

Figure 6: Sample Anchoring smart contract code.

Normally, an application can download the set of reports that have been added since the last time, and use all of them to create a single Merkle tree to verify that the Merkle root is stored in Ethereum (at a plausible time). If the application wants to check the existence of individual reports separately, the server needs to provide a Merkle proof (partial tree), but perhaps that would not be the case.

Table 1: Evaluation if a consumer group were to test the applications.

Point of weakness	Corresponding threats	Our proposal	GAEN
Ⓐ <i>Generation of TEKs</i>	1a, 1d	Prevented	Undetectable
Ⓑ <i>Content of beacons</i>	1b/1d	Detectable/Suspectable	Undetectable
Ⓒ <i>Metadata</i>	1c/1d	Detectable/Suspectable	Undetectable
Ⓓ <i>Reporting</i>	2a, 2b, 3a	Made more difficult	Undetectable
Ⓔ <i>Stored reports</i>	3a	Detectable	Undetectable
Ⓕ <i>Matching proximity IDs</i>	1e	Detectable	Detectable

## 4.2 Incentive Design

We also discuss how to incentivize the peer-to-peer maintenance of the public ledger, in case it is newly built and specifically used for the purpose of exposure notification. However, a private ledger would not provide sufficient proof due to a relatively large margin for malicious involvement. Therefore, it would be more appropriate to provide the new ledger in the form of an Ethereum 2.0 shard or Polkadot parachain, for example, which can be anchored firmly to existing blockchain.

Blockchain is originally designed to collect transactions, form a Merkle tree out of them, and store them in a verifiable block. By replacing transactions with reports, we obtain a straightforward design for the new ledger.

We would probably like to keep the structure of a traditional blockchain, where there is a reward for the creation of a block to incentivize its maintenance, but preferably without a fee for writing a report transaction. We also want to keep the market price of the rewards stable, given that we are protected by the high market price of the currency, although we do not need to maintain it for a long time given the expected duration of the pandemic. The authors of this work have proposed a way to stabilize the price of such a cryptocurrency and at the same time to abandon transaction fees [22], which may be applicable to the design.

## 5 Evaluation

### 5.1 Verifiability and Privacy

We evaluate our proposal by a thought experiment: if a consumer group were to verify the privacy protections of the two different systems, original GAEN and our proposed versions, the results would be as shown in Table 1.

For testing our proposed version, first, the consumer group modifies the open-source application to do the necessary logging, install it on their smartphones, and then perform testing. Below, we explain how our proposal works

for each point of weakness.

- Ⓐ **Generation of TEKs:** In our proposal, the application generates the TEKs, which prevents the OS from encoding any information into them. Naturally, the beacons cannot carry any information resulting from encoding in TEKs either.
- Ⓑ **Content of beacons:** The rolling proximity ID can be reproduced if the TEK and base time are known, to verify that the contents sent by Bluetooth are correct. If a value that is not based on the TEK is sent, it can be detected, and if it is detected, it can be suspected that a secret process is embedded in the receiver's OS.
- Ⓒ **Metadata:** Encrypted metadata in beacons sent over Bluetooth can be decrypted if the TEK is known, and detected if the correct metadata is not sent. If such is detected, it can be suspected that a secret process is embedded in the receiver's OS.
- Ⓓ **Reporting:** Reporting must be digitally signed by a medical institute, and false reports cannot be uploaded without collusion. Because of blind signature, it is not possible to identify the person who has tested positive (although identities can be inferred by the institute if only a small number of people have tested positive there during a given period).
- Ⓔ **Stored reports:** All reports will be proven for their existence, and it is not possible to insert reports retrospectively. If the application knows the TEKs, the user will be able to detect if they are compromised by the OS and a false report without the consent of the user is generated by a complicit medical institute.
- Ⓕ **Matching proximity IDs:** Suppose that a second device that is carried with the phone constantly collects beacons that would have been received by the phone. If the rolling proximity ID generated from the reported  $\langle \text{TEK}, \text{base time} \rangle$  pair is compared to the IDs in the beacons, and the user is notified of the exposure even though the IDs do not match, then we can detect that a fraud is occurring in the OS. However, this can also be detected without modifying GAEN.

### 5.1.1 Newly introduced threats?

Our proposal introduced two new parties: medical institutes and application developers<sup>4</sup>. We have to assume that these new third parties are also unreliable.

---

<sup>4</sup>Plus the public database that tells certified public keys of medical institutes, but it should be easily monitored by consumer groups and others.

If the medical institute changes the key pair for each blind signature, it can strip the reporter’s anonymity, but the institute cannot do it alone because the public keys have to be proven on a public database. By colluding with the government, threats 2a and 2b are made possible, and by colluding with OS developers or phone vendors, threat 3a is also made possible. However, the increased number of parties that must collude makes it more difficult to cheat than the original GAEN.

We propose to make the source code open to improve verifiability by consumer groups and others, and to help protect privacy by allowing careful users to build applications from the source code. On the other hand, this makes it easier for malicious application developers to introduce malware. Nevertheless, the increased risk is related to phishing and whether users install the wrong applications. The risk of malicious developers themselves running rogue applications is the same with the original GAEN.

## 5.2 Potential Performance Impact

We also roughly evaluate the extent to which the two designs can respond to an increase in the number of tested-positive reporters.

Processes such as getting blind signatures from a medical institute before reporting a positive test result, or calculating the Merkle tree from a set of downloaded reports and querying the Merkle root stored in a public ledger, are our additions to the original GAEN, but do not affect the overall performance of the system as they are processed in a distributed manner, although they do increase the power consumption of individual phones slightly. We hope that the users will consider that this is the cost of better privacy.

On the server side, processing of uploaded  $n$  reports involves the following additional calculations in our proposal: 1) verification of the signature attached to the report (cost  $O(n)$ ), 2) creation of the Merkle tree (cost  $O(n)$ ; this takes about  $2n$  digest calculations), and 3) storage of the Merkle root in the blockchain (cost  $O(1)$ ). The calculations can be parallelised and these costs can be load balanced by adding processors (Merkle tree creation requires  $O(\log n)$  sequential processing).

## 6 Related Work

### 6.1 Concerns on Exposure Notification

The general ethical concerns and guidance on exposure notification and contact tracing are well summarised in [19].

For security, [4] covers the types of attacks that are possible. [12] and [15] warn potential political use of the tool.

The effectiveness of GAEN has been measured through experiments by [26] and [14]. Meanwhile, it was announced in February 2021 that the

GAEN-based NHS COVID-19 application in UK has alerted 1.7 million contacts, and the UK government estimates approximately 600,000 cases have been prevented since September 2020 [11]. If this estimate is correct, GAEN is working effectively.

## 6.2 Enhancements to GAEN Protocol

A proposal to add GPS information to GAEN has been made by [20]. While this may improve the accuracy of contact tracing, it raises concerns about privacy.

## 6.3 Blockchain-based Contact Tracing

Some ideas on using blockchain for contact tracing have been proposed. Many have proposed defining and running their own blockchain, but a ledger system that starts small has a relatively large margin for malicious involvement in replicating the state machine, making it difficult to provide provability. Many are also trying to deal with geographic information instead of or in addition to Bluetooth proximity, which has potential difficulties in terms of privacy protection, while proximity alone is proving to be effective enough in reality.

Among such proposals, [28] and [16] incorporate geolocation information. [7] proposes a potential intervention to privacy with regard to promotion of public health. [24] proposes to track users' travel trajectories.

## 6.4 Safe Blues

Safe Blues[8] simulates and predicts actual infectious disease outbreaks by monitoring the spread of virtual viruses using technology similar to device-based contact tracing. As with actual exposure notifications, this seems feasible with privacy protection using only proximity, provided that locality is taken loosely. If this is the case, public health authorities may be able to proactively combat infectious diseases by incorporating Safe Blues functionality into exposure notification systems. Even then, it is important to ensure that the system is verifiable by the public, as we have shown in our proposal, so that it cannot be used by the authorities to unfairly control the activities of the population.

## 7 Conclusions

In this work, we have shown that minimal changes to GAEN, a working exposure notification mechanism, can protect users' privacy without trusting third parties.

Under our proposal, many of the privacy violations by OS developers, smartphone vendors, medical institutes and government authorities could be detected through verification by consumer groups and others. This is not the case if they collude, but our proposal makes this more difficult by increasing the number of actors that need to collude.

We believe that this will allow for safer contact tracing, and contribute to healthier lifestyles for citizens who may want to or have to go out under pandemic situations.

## References

- [1] Apple and Google. Exposure Notification – Bluetooth Specification. [https://blog.google/documents/70/Exposure\\_Notification\\_-\\_Bluetooth\\_Specification\\_v1.2.2.pdf](https://blog.google/documents/70/Exposure_Notification_-_Bluetooth_Specification_v1.2.2.pdf).
- [2] Apple and Google. Exposure Notification – Cryptography Specification. [https://blog.google/documents/69/Exposure\\_Notification\\_-\\_Cryptography\\_Specification\\_v1.2.1.pdf](https://blog.google/documents/69/Exposure_Notification_-_Cryptography_Specification_v1.2.1.pdf).
- [3] Apple and Google. Exposure Notification – Frequently Asked Questions. <https://www.google.com/covid19/exposurenotifications/pdfs/Exposure-Notification-FAQ-v1.2.pdf>.
- [4] Gennaro Avitabile, Vincenzo Botta, Vincenzo Iovino, and Ivan Visconti. Towards Defeating Mass Surveillance and SARS-CoV-2: The Pronto-C2 Fully Decentralized Automatic Contact Tracing System. Cryptology ePrint Archive, Report 2020/493, 2020.
- [5] Vitalik Buterin. A Next-Generation Smart Contract and Decentralized Application Platform, 2013. <https://github.com/ethereum/wiki/wiki/White-Paper>.
- [6] David Chaum. Blind signatures for untraceable payments. In *Advances in Cryptology*, pages 199–203. Springer US, 1983.
- [7] Hiten Choudhury, Bidisha Goswami, and Sameer Kumar Gurung. CovidChain: An Anonymity Preserving Blockchain Based Framework for Protection Against Covid-19, 2020. arXiv:2005.10607v1 [cs.CR].
- [8] Raj Abhijit Dandekar, Shane G. Henderson, Marijn Jansen, Sarat Moka, Yoni Nazarathy, Christopher Rackauckas, Peter G. Taylor, and Aapeli Vuorinen. Safe Blues: A Method for Estimation and Control in the Fight Against COVID-19, 2020. medRxiv 2020.05.04.20090258.

- [9] ethereum.org. Ethereum 2.0 Specifications. <https://github.com/ethereum/eth2.0-specs>.
- [10] Google. Exposure Notifications API. <https://developers.google.com/android/exposure-notifications/exposure-notifications-api>.
- [11] GOV.UK. NHS COVID-19 app alerts 1.7 million contacts to stop spread of COVID-19, Feb 2021. Press release from Department of Health and Social Care.
- [12] Jaap-Henk Hoepman. A Critique of the Google Apple Exposure Notification (GAEN) Framework, 2021. arXiv:2012.05097v2 [cs.CY].
- [13] Mitsuru Iwamura, Yukinobu Kitamura, Tsutomu Matsumoto, and Kenji Saito. Can we stabilize the price of a cryptocurrency?: Understanding the design of bitcoin and its potential to compete with central bank money. *Hitotsubashi Journal of Economics*, Vol.60(1), June 2019.
- [14] Douglas J. Leith and Stephen Farrell. Measurement-based evaluation of Google/Apple Exposure Notification API for proximity detection in a light-rail tram. *PLOS ONE*, 15(9):e0239943, Sep 2020.
- [15] Douglas J. Leith and Stephen Farrell. Google/Apple Exposure Notification Due Diligence. In *CoronaDef Workshop: Call for Innovative Secure IT Technologies against COVID-19 (NDSS 2021 Workshop)*, 2021.
- [16] Wenzhe Lv, Sheng Wu, Chunxiao Jiang, Yuanhao Cui, Xuesong Qiu, and Yan Zhang. Decentralized Blockchain for Privacy-Preserving Large-Scale Contact Tracing, 2020. arXiv:2007.00894v1 [cs.CR].
- [17] Ralph C. Merkle. A Digital Signature Based on a Conventional Encryption Function. In *Advances in Cryptology — CRYPTO '87*, pages 369–378, Berlin, Heidelberg, 1988. Springer.
- [18] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System, 2008. <http://bitcoin.org/bitcoin.pdf>.
- [19] Robert Ranisch, Niels Nijsingh, Angela Ballantyne, Anne van Bergen, Alena Buyx, Orsolya Friedrich, Tereza Hendl, Georg Marckmann, Christian Munthe, and Verina Wild. Digital contact tracing and exposure notification: ethical guidance for trustworthy pandemic management. *Ethics and Information Technology*, 2020.
- [20] Ramesh Raskar, Abhishek Singh, Sam Zimmerman, and Shrikant Kanaparti. Adding Location and Global Context to the Google/Apple Exposure Notification Bluetooth API, 2020. arXiv:2007.02317v3 [cs.CR].



- [21] Kenji Saito. Asia Internet History Projects - Fourth Decade (2010s) Section 2.3 Blockchain, September 2020.
- [22] Kenji Saito and Mitsuru Iwamura. How to make a digital currency on a blockchain stable. *Future Generation Computer Systems*, Vol.100:58–69, November 2019.
- [23] Kenji Saito and Hiroyuki Yamada. What’s so different about blockchain? - blockchain is a probabilistic state machine. In *IEEE 36th International Conference on Distributed Computing Systems Workshops (ICDCSW)*, pages 168–175, June 2016.
- [24] Jinyue Song, Tianbo Gu, Xiaotao Feng, Yunjie Ge, and Prasant Mohapatra. Blockchain Meets COVID-19: A Framework for Contact Information Sharing and Risk Notification System, 2020. arXiv:2007.10529v1 [cs.CR].
- [25] Hiroshi Watanabe, Kenji Saito, Satoshi Miyazaki, Toshiharu Okada, Hiroyuki Fukuyama, Tsuneo Kato, and Katsuo Taniguchi. Proof of Authenticity of Logistics Information with Passive RFID Tags and Blockchain, 2020. arXiv:2011.05442 [cs.CR].
- [26] Amanda M. Wilson, Nathan Aviles, James I. Petrie, Paloma I. Beamer, Zsombor Szabo, Michelle Xie, Janet McIllece, Yijie Chen, Young-Jun Son, Sameer Halai, Tina White, Kacey C. Ernst, and Joanna Masel. Quantifying SARS-CoV-2 infection risk within the Google/Apple exposure notification framework to inform quarantine recommendations, 2020. medRxiv 2020.07.17.20156539.
- [27] Gavin Wood. POLKADOT: VISION FOR A HETEROGENEOUS MULTI-CHAIN FRAMEWORK, 2016.
- [28] Hao Xu, Lei Zhang, Oluwakayode Onireti, Yang Fang, William Bill Buchanan, and Muhammad Ali Imran. BeepTrace: Blockchain-enabled Privacy-preserving Contact Tracing for COVID-19 Pandemic and Beyond, 2020. arXiv:2005.10103v2 [cs.DC].