Environmental Adaptive Privacy Preserving Contact Tracing System: A Construction From Public Key Rerandomizable BLS Signatures

PENGFEI WANG\textsuperscript{1}, XIANGYU SU\textsuperscript{1}, MAXIM JOURENKO\textsuperscript{2}, ZIXIAN JIANG\textsuperscript{3}, MARIO LARANGEIRA\textsuperscript{1,4}, AND KEISUKE TANAKA\textsuperscript{1}

\textsuperscript{1}Department of Mathematical and Computing Science, School of Computing, Tokyo Institute of Technology, Tokyo 152-8552, Japan
\textsuperscript{2}Department of Mathematical Informatics, Graduate School of Information Science and Technology, The University of Tokyo, Tokyo 113-8656, Japan
\textsuperscript{3}Laboratory of Plasma Membrane and Nuclear Signaling, Kyoto University, Kyoto 606-8501, Japan
\textsuperscript{4}Input Output Hong Kong (IOHK), Hong Kong

Corresponding authors: Pengfei Wang (wang.p.ae@m.titech.ac.jp) and Xiangyu Su (su.x.ab@m.titech.ac.jp)

This work was supported in part by the Input Output Cryptocurrency Collaborative Research Chair through the Input Output Hong Kong; in part by the Japan Science and Technology Agency (JST) OPERA; in part by the Japan Society for the Promotion of Science (JSPS) KAKENHI under Grant JP16H01705, Grant JP17H01695, Grant JP21H04879, and Grant JP21K11882; and in part by the JST CREST under Grant JPMJCR14D6 and Grant JPMJCR2113.

\section*{ABSTRACT} The COVID-19 pandemic has made the scientific community devise means to implement “contact tracing” mechanisms to mitigate the spread of the infection. The crucial idea is to scan and record close contacts between users using mobile device, in order to notify persons when their close contact(s) is diagnosed positive. First, the ability granted to service providers of the contact tracing systems to access user data violates user privacy, and attackers can fabricate identities and contact records in their devices, which harms the integrity of the system. Moreover, current contact tracing systems’ false-positive rate is too high to be practical as they do not filter scan results outside the range of infections, since the range of transmission for droplets is far less than the scanning range for Bluetooth Low Energy used by these systems. Furthermore, current systems neglect airborne transmission, a far cry from a tool against viruses suspended in the air. In this paper, we propose a cryptographic framework for contact tracing and provide a construction based on public key rerandomizable BLS signature, being capable of providing users of contact tracing with comprehensive privacy protection. Besides, we also implement a commitment scheme to prevent fabrication of identities and contact records. To prove the concept of our framework and to solve other problems mentioned above, we proposed a new contact tracing system, using environmental factors (temperature, humidity and airflow) to filter out results outside estimated effective transmission distance, and also take airborne transmission into consideration. Finally, we evaluate the performance of our design by implementing our algorithm on mobile devices with satisfactory results.

\section*{INDEX TERMS} Blockchain, contact-tracing, epidemiology, fluid dynamics, traceable signature, public key rerandomizable signature, BLS signature, privacy preservation.

\section*{I. INTRODUCTION}

\subsection*{A. BACKGROUND} Infectious diseases have long been one of the most deadly threats to human society. An efficient method to prevent people from being infected is to track routes of transmission of the viruses and warn potential infectees so that suspected patients can quarantine themselves before symptoms grow, avoiding further spread of the virus.

Starting from Privacy-Preserving Contact Tracing System [2] developed by Apple and Google in April 2020, various organizations or governments around the world have developed different kinds of contact tracing systems to combat the COVID-19 pandemic.

Although it has been more than two years since the outbreak, a closer look at these systems tells us that these systems have various deficiencies — lack of either precision, privacy or integrity. Therefore, current contact tracing systems cannot provide neither comprehensive security nor privacy to users.
Specifically, as an example, the privacy preserving contact tracing system developed by Apple and Google does not filter scan results by Bluetooth Low Energy (BLE) outside the transmission range. However, the transmission range of droplets is much shorter compared to the scanning range of BLE (See Section 1-B). Therefore, when all (or even only some) people who got warned try to access hospitals, the yield would be too large for local medical systems to endure.

Also, these systems do not take airborne transmission (different from droplet transmission [3]) into consideration. Droplets may remain in the air for a specific amount of time, while these floating particles may also be infectious during their “lifetime”.

Besides, the best case of privacy should be the inability of anyone to collect private user information since a secure system would better not rely on trust. However, all current contact tracing systems trust that service providers will not collect private user information. Hence, these applications intend to prevent unauthorized access, but they do not categorize service providers as “unauthorized entity”, while service providers and attackers should be treated similarly from the perspective of privacy protection.

There have been multiple data leakages reports of contact tracing database (e.g. [4]–[6]). In April 2020, a joint statement [7] signed by 177 scientists from the U.K. indicated that if there is a database, then it must not allow service providers to de-anonymize the data by any means. Blockchain, by its nature, is designed to be pseudonymous, publicly verifiable, non-repudiable, and non-modifiable.

Furthermore, all users of these existing systems allow attackers to fabricate fake contact records or fake user identities.

B. RELATED WORK
We have evaluated the existing cryptographic approaches of contact tracing systems and concluded that these systems all have similar deficiencies (See Table 1). Here, “Precision” means whether to filter out false positives (get warned but not in danger) from the scanned results; “Demo” means whether a software demonstration exists; “Confid.” means confidentiality; “Integrity” means whether someone can fabricate non-existent records or identities; “Scalability” means whether the system can be deployed in large-scale use-cases, e.g. densely populated scenarios or heavy calculations; “Flooding” means whether the system can prevent users from flooding the system with either legitimate records or illegitimate records. Additionally, “Δ” means this property exists, without being fully applied, which will be explained in the following paragraphs or sections, and “N/A” means this property is not considered as a topic by the researcher(s) in their corresponding research(es).

The idea of “privacy preserving contact tracing” is not novel. For example, Danz et al.’s work [8] tried to analyze existing schemes and propose a security framework, however it lacked a solution to potential risks, such as linkability of user identifications with contact records. Their contributions are limited to summarizing what objectives researchers should seek for a secure contact tracing system and evaluating contemporary works based on these objectives.

One of the major problems is the lack of precision. Based on our research, there is currently no contact tracing system focusing on reducing the false-positive rate of the scanning. For example, the first contact tracing system for COVID-19 is the privacy preserving Contact Tracing system designed by Apple and Google (“GAEN”) [2], using Bluetooth Low Energy (BLE) to search for surrounding devices without excluding results outside effect range of droplet transmission. Although the range of BLE varies by device, most of them are around 50–100 meters. In contrast, based on epidemiological research, the distance of droplet transmission is typically between 0–20 meters approximately [19], and can be affected by environmental factors. This means that individuals scanned outside the effective range would be false-positive cases.

Also, all current contact tracing systems ignore the fact that people do not have to contact others closely to become “close contacts” because, according to Chen [20], droplets ejected from the human body will float for a specific amount of time. Therefore, these systems do not consider the duration time for the infection tracking.

More importantly, personal privacy is not thoroughly preserved in current contact tracing systems. In this regard, GAEN is a representative model for typical contact tracing. Users hold a periodically re-generated temporary key (TEK) and generate other keys using a key derivation function. Keys derived from TEK are used to encrypt the payload with AES [21]. After being diagnosed positive, encrypted payload and TEKs will be uploaded for normal users to trace their contacts. However, for GAEN, its cryptographic design based on AES and random key generation cannot guarantee integrity, i.e., fabricating never-happened meetings or fake identities. It is worth mentioning that all current contact tracing systems do not prevent such fabrication, which would harm data integrity of contact tracing systems.

Although all of the contact tracing systems we reviewed emphasize confidentiality, which means to prevent unauthorized access, most of them have not clarified the problem about who shall be categorized as “unauthorized entities”. For example, the contact tracing application designed by N.C.S.C. of the U.K. [9] holds a public key pair for each user and stores the private key in the server. Hence, the service providers can access user data using the private key while preventing unauthorized entities from accessing it. However, from the perspective of privacy protection, service providers and unauthorized entities show no difference — both can act as threats to users’ data security, and it is not possible to prevent this from happening with traditional public-key approaches.

Systems proposed by Kim et al. [14] and An et al. [15] use functional encryption and homomorphic encryption to circumvent the aforementioned issues caused by traditional encryption. However, neither of the primitive is practical in
the real world. Moreover, these systems even collect more personally identifiable information than it is necessary, e.g., credit card information.

Canetti et al. [13] proposed two contact tracing protocols that achieve different privacy-preserving levels. However, neither of them requires signatures, thus they lack integrity on users’ claims. Furthermore, their work sticks with the BLE solution and thus cannot safely transmit and authenticate environmental factors. In fact, we have not found any contact tracing system designed to involve environmental factors in their security frameworks.

Other works, such as Liu et al.’s work [12], would fail to be implemented with heavy traffic because they require key exchange protocols and established connections. Moreover, the existing non-interactive contact tracing systems [8] [13] would lack integrity, given that users can generate arbitrary pseudonyms without verification. The authors did not provide a repository or screenshot about their implementation as well.

There are also articles mainly focused on engineering issues, such as [16]–[18]. Unlike this work which proposes new generic cryptographic frameworks for contact tracing systems, these studies focus on implementing existing technologies and building concrete applications. Moreover, they also suffer from the issues mentioned at the beginning of this section. For example, [18] uses a different approach that involves IoT, and only traces diagnosed users. Whereas [16] uses geolocation data in contact tracing but only presents possible methods to achieve privacy without analysis.

It should be specially notified that, the term “airborne-based” used in [17] refers to droplet transmission, which is not the same as the formal definition of airborne transmission, defined by CDC [22], as we describes in Section II-B.

Finally, we believe contact-tracing applications should follow the ethical guidelines proposed in [23]. However, as our goal focuses more on proposing a general cryptographic framework for contact-tracing systems rather than coding an application, we argue that our work does not violate the spirit of the guideline.

C. OUR CONTRIBUTION

Our contribution can be roughly divided into 2 parts — a new cryptographic framework, and a new contact tracing system based on such framework.

1) A NEW CRYPTOGRAPHIC FRAMEWORK

First, to solve the dilemma between anonymity and integrity, we propose a cryptographic framework for contact tracing and provide a construction based on a public key re-randomizable BLS signature scheme. The scheme is based on the original BLS signature [24] and the updatable public key mechanism [25], and it satisfies public key re-randomizability and signature aggregability. The scheme enables users to update their identities to preserve privacy while keeping the integrity with signatures to deny malicious messages. The result is that users cannot fabricate fake meeting records with other users, undermining the integrity of the bulletin board. Thus, we utilize the signature scheme as our main building block for the contact tracing system and analyze the computational complexity of the resulting scheme.

We also clarify the integration between our framework and a bulletin board, which is implemented with a blockchain (See Section IV-A3). Finally, we consider several practical issues of our blockchain-based bulletin-board, such as privacy, scalability, and storage overhead.

2) A NEW CONTACT TRACING SYSTEM

Second, based on our cryptographic framework described above, we propose a new contact tracing system, which uses environmental factors affecting particle dynamics, lifetime to filter scanning results, and lower the false positive rate, to prove the practicality of our cryptographic framework. The dynamics determines the coordinates in which users should be informed, and lifetime determines the time interval during which users should be informed. Our cryptographic framework is a variant of credential systems, and thus it can embed these arbitrary environment factors inherently. Moreover, according to new requirements (airborne transmission) in contact tracing, our system includes a new contact-tracing scheme called discrete real-time tracking. See Figure 2 for the detailed design.

Our system uses a perfect hiding commitment scheme to prevent multiple identities for the same user. Users need to commit their unique initial public keys to authorized organizations. Then, doctors need to check 1) whether the user knows the corresponding secret key and 2) the commitment is computed correctly from the public key and commitment key, without knowing either public, secret, or commitment keys.
Hence, doctors can verify whether the submitted contact records are signed by the committed public key.

Finally, we developed a demonstration on Android OS to show the efficiency of our work on mobile devices.

3) NOTES
This work is an expanded version of our previous work in CSS2021 [1]. In this paper, we have reviewed more related works, and we include preliminaries of commitment and signature scheme, and hardness assumptions in Section II. In Section IV, We add the definition of our framework. Section V is a brand new section compared to the previous work, including the definition and security proof of the building block, and a computational complexity analysis. Section VI is expanded by the construction of key management and registration, remark of ZKPoK, security proof of the construction, and expanding our security analysis. We expand the corresponding implementation and performance evaluation of our work as well, in Section VII.

Please take note that this work mildly considers real-world performance and associated concrete engineering issues. Our goal is not to provide a ready-to-distribute software, but a contact tracing solution, based on our cryptographic framework.

Please also take note that this research does not limit to the COVID-19 but takes it as a concrete example, which works on any pandemic.

D. ROADMAP
Following this section, Section II introduces environmental factors, including calculations and measurements, and cryptographic preliminaries, including commitment and signature schemes, and hardness assumptions.

Section III illustrates how our system records and transmits data using environmental factors and why we need a new approach in addition to the Bluetooth scanning. This section also specifies what data our cryptographic framework should handle. We also evaluate how our new approach of contact tracing would work in special cases.

In Section IV, we introduce our cryptographic framework for contact tracing as described in previous sections in terms of sub-protocols. We provide a concrete threat model for the framework including adversary setting and security properties. Finally, a data flow diagram is provided and explained in this section as well (Section IV-C).

In Section V, we present a novel signature scheme as our main building block according to the cryptographic framework. Hence, we provide concrete construction, security analysis, and complexity analysis in Section VI.

Section VII explains the detailed implementation of our system (with minor changes to the framework to address engineering issues), and evaluates the performance of our system based on the demonstration on physical and virtual devices.

Lastly, Sections VIII and IX discuss some unaddressed problems and summarize our work.

II. PRELIMINARIES
A. NOTATION
Table 2 introduces the notation used throughout the paper.

B. ENVIRONMENTAL FACTORS
Environmental factors play an important role in contact tracing, thus we evaluate and incorporate several epidemiological studies in our system.

The three ways of transmission for respiratory viruses, as suggested by CDC [22], are: contact transmission, droplet transmission, and airborne transmission.

In this paper, according to the definition [22], “contact transmission” means direct physical contact with infectees; “droplet transmission” refers to the infections due to viruses ejected by droplets by sneezes or coughs; “airborne transmission” means infections caused by floating liquid drops carrying viruses suspended in the air.

1) ASSUMPTIONS
We need to make some assumptions to exclude immeasurable or uncontrollable factors.

a: EXCLUDING CONTACT TRANSMISSION
Without extra wearable devices or sensors, cellphones are not able to detect direct contacts (i.e., touching), and contact tracing systems should solely rely on the most widely-used portable devices (cellphones, in this case).

b: ENCLOSED INDOOR ENVIRONMENTS
For indoor environments, it is true that ventilation is crucial for epidemic preparedness [3]. However, it is hardly possible for cellphones to sense the ventilation status of the current room. Hence, we need to decide whether to assume all indoor cases are enclosed or ventilated. If the environment is ventilated, the airflow organization will become more complex. In that case, we need to either simulate particle dynamics, which is impractical for mobile devices, or to assume indoor airflow organization models (currently not available, see Section VIII). Moreover, since an enclosed environment is more dangerous than a ventilated environment [3], as airflow can dilute viruses in the air and thus reduce their infectivity, we prefer a worst-case assumption. Consequently, we assume that all indoor environments are enclosed in our system, including balconies and opened windows.

c: END OF “LIFETIME” ≠ DEATH OF VIRUS
As the researchers indicate in their study about lifetime [20], lifetime means vaporization. However, vaporization of the droplet that carries and transmits the virus does not necessarily mean the virus’ death. The lifetime of pathogens differs from one disease to another and depends on surfaces’ materials, which are not measurable by cellphones. Therefore, we will not consider the real lifetime of pathogens.

2) OVERVIEW
Under the assumptions described above, three major measurable factors are temperature $T$, relative humidity $RH$, and
TABLE 2. Table of notations.

| Notations | Definitions                              | | Notations | Definitions |
|-----------|-----------------------------------------|---|-----------|-------------|
| $B$       | Spalding mass transfer number           | $C$ | Circle                     |
| $D$       | Binary diffusion coefficient            | $F^G$ | Gravitational force |
| $MP$      | Measured power                          | $RH$ | Relative humidity          |
| $T$       | Temperature                             | $Y$ | Mass fraction              |
| $d$       | Estimated distance (by RSSI)            | $d_0$ | Initial droplet size       |
| $f$       | Liquid water                           | $dt$ | Discrete time point        |
| $m$       | Mass of the droplet                     | $L$ | Location-stamp             |
| $t$       | Droplet Lifetime                        | $R$ | Radius of the droplet      |
| $w_0$     | Peak value of airflow velocity          | $t_r$ | Record time                |
| $v_t$     | Downward terminal velocity              | $w_1, s$ | Water at droplet surface  |
| $w_1, \infty$ | Ambient water                | $\gamma$ | Path loss exponent        |
| $\eta$    | Dynamic viscosity                       | $\kappa$ | Security Parameter        |
| $\lambda$ | Drag coefficient                        | $\xi$ | Noise term                |
| $\rho$    | Density                                 | $\sigma$ | Signage                  |
| $Z_0$     | $q$-Element Field                      | $pp$ | Public Parameter          |
| $H$       | Hash Function                           | $pk$ | Public Key                |
| $sk$      | Secret Key                              | $nym$ | Pseudonym                |
| attr$    $ | Tracing Attributes                      | $BC$ | Blockchain                |

air velocity. There are two types of transmission, transmission by droplets and by airborne. All researchers mentioned in this section have provided experimental data, respectively, with values of environmental factors staying in reasonable ranges as real-world scenarios may have.

This section is represented by the box “Discrete Tracing” and “BLE Scanning” in the design diagram (Figure 2 in Section IV).

We must emphasize that we do not intend to pay attention to the accuracy of epidemiological data and equations mentioned in this section. We believe that such researches are subject to future works of physicists and epidemiologists. Existing contact tracing approaches completely lacked consideration of such fields, and our focus here is to implement epidemiological studies in contact tracing.

a: DROPLET TRANSMISSION

According to Das et al.’s study, the influence of temperature on space-time evolution of droplets’ motion, from 0°C to 40°C, is quite limited (about 10%) [19]. Therefore, we will not take temperature into consideration of droplet transmission.

Instead, the sizes of the particles and airflow play significant roles in droplet transmission. Das et al.’s data is based on the Langevin equation [19], i.e., Equation (1).

$$\frac{4}{3} \pi R^3 \rho \frac{d^2 r_i}{dt^2} = -\lambda \frac{dr_i}{dt} + \xi(t) + F^G$$  \hspace{1cm} (1)

In order to measure the size (i.e. radius) of the droplets that determines the mass $m$ (i.e. $4\pi R^3 \rho / 3$), Han et al.’s research provides statistical data of diameter of droplets, measured at 100 ms after the ejection from the human body [26].

Together with the Langevin Equation, the impact of airflow velocity also needs to be considered, as shown in Equation (2). The impact of airflow velocity needs to be evaluated with the downward terminal velocity, $v_t$ [19].

$$v_t = \frac{2 R^2 (\rho_{\text{droplet}} - \rho_{\text{air}}) g}{9 \eta}$$  \hspace{1cm} (2)

Das et al. have explained thoroughly how to solve Equation 1 and also the effects of airflow velocity in their paper, and thus we will not cover these details in this section. Since the authors claim that their experiment data matches calculated assumptions perfectly [19], both can be implemented in our system.

We use 0.5 m/s air velocity as the threshold to determine whether the current outdoor environment is ventilated enough, as suggested by NFPA in laboratory ventilation standards [27]. If air velocity is faster than the threshold, transmission radius will be set to 0. This rule also applies to lifetime in airborne transmission, as discussed in the following section.

b: AIRBORNE TRANSMISSION

Chen’s work provides an important equation in his research as Equation (3) shows for determining lifetime of the particles. Spalding mass transfer number $B$ is computed from mass fraction $Y_{w,s}$ and $Y_{w,\infty}$. In this equation, temperature has impact on diffusion coefficient $D$ and relative humidity determines the ratio between $Y_{w,s}$ and $Y_{w,\infty}$.

$$t = \frac{\rho f d_0^2}{8 \rho_w D_w (\ln(1+B))}, \text{ where } B = \frac{Y_{w,s} - Y_{w,\infty}}{1 - Y_{w,s}}$$ \hspace{1cm} (3)

Using the equation, data in Chen’s research include particles of 1–1000 µm diameter ($d_0$), with relative humidity $RH$ from 0% to 100% and temperature at 20°C and 30°C. Both data and equation can be implemented in our system to predict lifetime.

Take note that when relative humidity approaches approximately 55.7% [20], lifetime approaches infinity, as $Y_{w,\infty}$ approaches $Y_{w,s}$, shown in Equation (3).
3) MEASUREMENT
It is crucial to distinguish indoor environments from outdoor environments for measurements, as most countries have laws or regulations on temperature/humidity in public areas. Online mapping services, such as Google Maps, are capable of categorizing place types (e.g., bank, city hall, zoo, and etc.) [33]. According to users’ location, these services can be used to infer whether the user is staying indoor or outdoor, public buildings or residential buildings. Temperature/humidity indoor can be estimated from either these laws or regulations, assuming that all public buildings keep following these laws or regulations, or users’ personal favorite ambient temperature/humidity at home. Again, we assume all indoor areas are enclosed, as stated in Section II-B1.

**TABLE 3. Regulations about temperature and relative humidity.**

| Nation     | Temperature (in Celsius) | RHI   | Reference |
|------------|--------------------------|-------|-----------|
| Japan      | 17–28                    | 40%–70% | [28]      |
| U.S.A.     | 20–24                    | 20%–60% | [29]      |
| U.K.       | ≥16 or ≥13               | N/A   | [30]      |
| France     | 19                       | N/A   | [31]      |
| China      | 26–28 or 16–20           | 40%–65% | [32]      |

In Table 3, we take five countries as examples, Japan, United States of America, United Kingdom, French Republic, and People’s Republic of China. By embedding these data (along with other countries) into the system in advance, our system can operate in different nations or regions.

Besides, outdoor temperature/humidity/airflow is easy to measure — using data from government-held organizations such as the department of meteorology or from commercial weather service providers.

Some mobile phones are also equipped with hygrometers, or humidity sensors, which can measure ambient relative humidity and should be more precise than estimation based on regulations.

**C. CRYPTOGRAPHIC PRELIMINARIES**

Now, we introduce cryptographic primitives used in this work.

1) COMMITMENT SCHEME
A commitment scheme enables a sender to commit to a value without revealing it to a receiver. The sender can then reveal the committed value and the receiver will be convinced that the value has not been changed. A commitment scheme denoted by Comm involves three algorithms (KeyGen, Commit, Reveal).

- KeyGen(pp) → ck. With public parameters pp (including a security parameter λ) as input, it outputs a commitment key ck;
- Commit(m, ck) → comm. With message m and a commitment key ck as input, it outputs a commitment comm;
- Reveal(ck, comm, ck) → m. With commitment key ck and a commitment comm as input, it outputs the message m.

The scheme should satisfy correctness, i.e., Reveal(ck, Commit(m, ck)) \( \neq m \) with negligible \( i.e., \negl(\lambda) \) probability, where ck is generated by KeyGen. Security properties are hiding and binding. Hiding indicates that the receiver cannot learn the value after being committed, whereas binding indicates that the sender cannot open the value with another key. We require a perfect hiding commitment scheme, in which the following statements hold.

- **Perfect Hiding.** For any computationally un-bounded receiver, given two messages \( m_0, m_1 \) and ck generated from KeyGen, distributions of Commit\( m_0, ck \) and Commit\( m_1, ck \) have 0 statistical difference.

**2) SIGNATURE SCHEME**
A signature scheme denoted by SIG involves three algorithms (KeyGen, Sign, Vrfy):

- KeyGen(\( \lambda \)) → (pk, sk). With security parameter λ as input, it outputs a public, secret key pair (pk, sk);
- Sign(sk, m) → σ. With signer’s secret key sk with a message m ∈ M (a message space) as input, it outputs a signature;
- Vrfy(pk, m, σ) → \{0, 1\}. It checks validity of (m, σ) pair with pk, outputs 1 if true, and 0 otherwise.

A signature scheme should be correct, i.e., \( \forall m \in M \) \( \sigma = \text{Sign}(sk, m) \), \( \text{Vrfy}(pk, m, \sigma) = 1 \) with negligible \( i.e., \negl(\lambda) \) probability, if KeyGen is honestly executed. For security, it should be secure under existential unforgeability against adaptive chosen-message attacks (EUF-CMA). That is, any adversary \( \mathcal{A} \) who is given a signing oracle and denoted by \( \mathcal{A}_{\text{Sign}(sk, \cdot)}(\cdot) \), can produce a valid \( (m^*, \sigma^*) \) pair on a fresh \( m^* \) (not queried to the signing oracle) with \( \negl(\lambda) \) probability.

We take the ECDSA scheme [36] as a concrete example as it is well adopted in blockchain-based systems.

**3) HARDNESS ASSUMPTIONS**
We use an efficiently computable non-degenerate asymmetric bilinear map \( e: G_1 \times G_2 \rightarrow G_T \) in groups \( G_1, G_2, G_T \) of prime order q. The bilinear map here and throughout the paper is type-3 [37], i.e., there exists no efficiently computable isomorphism from \( G_2 \) to \( G_1 \).

With the bilinear mapping setting, we introduce the Computational co-Diffie-Hellman (co-CDH) assumption [38] and the Symmetric eXternal Diffie-Hellman (SXDH) assumption [39].
Assumption 2.1 (co-CDH): Let \((q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)\) be an asymmetric bilinear map with random generators \(g_1 \in \mathbb{G}_1, g_2 \in \mathbb{G}_2\). The co-CDH holds if for any PPT adversary \(A\):

\[
\Pr\left[ \bar{s}_{\mathcal{S}_1} \leftarrow A(g_1, g_2, g_1^a, g_2^b) \mid a, b \stackrel{\$}{\leftarrow} \mathbb{Z}_q \right] = \text{negl}(\lambda).
\]

Assumption 2.2 (SXDH): Let \((q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)\) be an asymmetric bilinear map with random generators \(g_1 \in \mathbb{G}_1, g_2 \in \mathbb{G}_2\). The SXDH holds if for any PPT adversary \(A\), the following holds for \(i \in \{1, 2\}:\)

\[
\Pr[A(g_1, g_2, g_1^a, g_1^b, g_2^b) = 1] = a, b \stackrel{\$}{\leftarrow} \mathbb{Z}_q
\]

\[
- \Pr[A(g_1, g_2, g_1^a, g_2^b) = 1|a, b, c \stackrel{\$}{\leftarrow} \mathbb{Z}_q] = \text{negl}(\lambda);
\]

Intuitively, the SXDH assumption indicates that the Decision Diffie-Hellman (DDH) assumption holds on both \(\mathbb{G}_1\) and \(\mathbb{G}_2\).

### III. CONTACT TRACING OVERVIEW

A contact record will be generated if and only if two users meet each other in current contact tracing systems. In this case, the airborne transmission is neglected because no meeting has taken place, as we have discussed in Section II-B2.b. Therefore, we redesign the scanning mechanism to accomplish our goal.

This section is represented by box “Contact Tracing” in Figure 2 in Section IV.

#### A. ANONYMOUS DISCRETE REAL-TIME TRACKING

For users to detect whom they meet with and whether anyone has been in their current place before, our system needs to embed timestamps into records. However, in such a case, the definition of “contact records” should be completely different from the current contact tracing systems. In our design, “records” should be the discrete positions of single users with timestamps, in contrast to “contact records” of current contact tracing systems, which indicates the co-existence of two users in close distance.

In Figure 1a, point \(n\) denotes locations (previous or current) of the user where discrete records are generated. \(R_n\) represents range of transmission, as Section II-B2.a explained. Arrows merely represent the user’s track in reality, without being recorded, as all these points are treated independently by the system to avoid revealing any user’s track of movement.

For \(\text{DiscreteRecord}(t_i, L, R_n)\) of each point \(n\), time of recording \(t_i\), lifetime \(t\), location-stamp \(L\) and range \(R_n\) will be included.

The distance between points of \(n\) and \(n-1\) should equal to \(R_{n-1}\) to find a balance between leaving areas uncovered and having too many discrete records. In other words, a discrete record will be generated when the user leaves his or her last range of transmission. If the user is not moving, the system will not generate new discrete records frequently. Until a certain threshold (e.g., 30s) since the last recording is met, a new discrete record will be generated by force to ensure that the system keeps tracking the user even if one is not moving.

The client-side system will fetch all discrete points from the bulletin board without distinguishing users’ identities (ensuring anonymity). Then, the system will compare timestamps and location-stamps, together with corresponding embedded environmental factors, and thus be able to determine whether this user should be notified.

#### B. EXCEPTIONS

Besides the difference in discrete records, another major difference between our approach and other approaches is that discrete records denote the absolute position of one user. In contrast, other systems record the relative position of a pair of users.

Therefore, this will create a problem — when a group of users is moving simultaneously in the same direction and at the same speed (i.e., public transport, in this scenario), the system will misrecognize users outside the range of transmission as “close contacts” (See Figure 1b).

In Figure 1b, we take user A and B as examples. The rectangle represents a public transport vehicle, and we assume that none of the users is moving relative to the vehicle. When the vehicle moves leftward, A’s previous location will be in B’s current location, and B will be recognized as close contacts caused by the lifetime of the droplets, while none of them is within each other’s range of transmission, as suspended liquid drops containing virus are also moving with the vehicle due to inertia. Furthermore, when multiple vehicles behind this vehicle containing a positive case are using the same track, all users on the following vehicles will be mistakenly recognized as “close contacts”.

Therefore, our solution combines the traditional Bluetooth approach (relative positioning) with our discrete real-time approach (absolute positioning). When the device detects unreasonable walking speed, it will switch from discrete real-time tracking to traditional Bluetooth-based scanning.

To calculate the distance between users for Bluetooth scanning, we need to make use of signal strength, i.e., Received Signal Strength Indicator (RSSI). Barai et al.’s work [40] indicates that \(d = 10^{\frac{MP_{\text{RSSI}}}{10}}\), where \(d\) means distance.

#### C. EVALUATION

The most significant concern about our discrete real-time tracking is that gaps between discrete points \((i.e., \ n = 1 \text{ and } n = 2)\) in Figure 1a may miss some close contacts, as shown in Figure 1c.

If there exists a point \(n = 3\) between \(n = 1 \text{ and } n = 2, \) of which \(R_3 > \text{Distance}(n = 3, i)\) \((i\) denotes the intersection of \(C_1\) and \(C_2)\), there will be uncovered area which can be represented by \(C_3 \cap (C_1 \cup C_2)\).

If we take Das et al.’s data as an example, we can see that the maximum value of \(R_n\) (under the experimental environment set by the researchers) is around 5 to 6 meters. We can assume that there will not be significant changes in \(R_n\) within such a short distance, and hence we decide to leave this issue to researchers in the future.
Also, since the timestamp is fixed after record generation, if there is some significant environmental change during the lifetime, the system cannot respond. Similarly, for now, we can only assume that there is no such change in environmental factors during the lifetime.

IV. CRYPTOGRAPHIC FRAMEWORK AND SYSTEM DESIGN

We have explained every part of contact tracing in previous sections. This section will explain how contact tracing mechanisms will be incorporated into our system—starting with our cryptographic framework for the contact tracing system in terms of definition and threat model. Then, according to the framework, we present the system’s design.

A. CRYPTOGRAPHIC FRAMEWORK

There are three protocols in the tracing system: key management, recording, and tracing. For protocols, we present the formal definitions of algorithms.

1) KEY MANAGEMENT

A key management protocol is executed by users and authorized organizations. A user generates its initial public key according to parameters chosen by authorized organizations. This user updates the public key to obtain pseudonyms. This process can be done continuously alongside time slots. The key management involves the following algorithms and a key registration sub-protocol.

• Setup($1^\lambda$) → pp. Run by authorized organizations, with security parameter $\lambda$. Setup outputs a public parameter pp. pp includes information to specify input/output spaces for other system algorithms;
• KeyGen(pp) → (pk, sk, ck). Run by a user, with public parameter pp. KeyGen outputs a public and secret key pair (pk, sk) for a signature scheme and a commitment key ck for a commitment scheme;
• FormNym(pp, pk, sk, attrs) → nym. Run by a user, FormNym creates a pseudonym nym according to the input key pair (pk, sk) and time slot index $t \in$ attrs. The attributes may also contain location and other environmental factors. Users should continuously update their pseudonym.

In the key registration sub-protocol, the user commits its initial public key to authorized organizations with commitment key. An authorized organization registers the commitment if and only if the user holds the corresponding secret key and the user has never been registered before. We denote users’ real-world identities as misc, e.g., driver’s licenses.

• KeyCommit(pp, pk, ck) → comm. Run by a user, KeyCommit outputs a commitment comm from a public key pk and a commitment key ck;
• KeyReg(pp, comm, misc) → {0, 1}. Run by authorized organizations, with key commitment comm and the user’s social identity misc, KeyReg returns $b = 1$ if and only if the user:
  1) computes KeyCommit(pp, pk, ck) correctly;
  2) knows the corresponding sk;
  3) has never used its misc in KeyReg before.

We use a (perfect hiding) commitment scheme on initial public keys instead of plain-text registrations because authorized organizations may leak users’ registered public keys. Hence, a malicious user cannot take some initial public keys and generate pseudonyms on behalf of the victims without being noticed. In such a case, our system can prevent malicious users from flooding the bulletin board with fabricated contact records.

Moreover, we require users’ social identities misc in registration to prevent users from registering multiple public keys to the system. It should be noticed that misc is sensitive and should be kept private. We present the way to prove the validity of users’ public keys in Section VI.

2) RECORDING

Recording protocol is executed by users. Users create and store records of contact or discrete location. It involves the following algorithms:
• **Assert**(pp, nym, sk, attrs) → (assertion, π\textsubscript{A}). **Assert** outputs an assertion assertion and a proof π\textsubscript{A}. The assertion embeds the user’s nym and attrs. It may serve as a Bluetooth identifier when using the Bluetooth approach \cite{2} for contact tracing;

• **VrfyAssert**(pp, assertion, π\textsubscript{A}) → \{0, 1\}. **VrfyAssert** validates assertion concerning π\textsubscript{A}, returns 1 if valid, and returns 0 otherwise;

• **Save**(pp, ((assertion, π\textsubscript{A}))) → compressed Assertion. With a list of assertions and the corresponding proofs as input, **Save** runs **VrfyAssert** on each pair. It compresses and saves the valid assertions to a compressedAssertion.

Notice that the **Save** algorithm takes in assertions regardless of their ownership. A user can run **Save** on its own sets of assertions. Henceforth, our framework is capable of both contact-tracing and discrete-location-tracing.

3) TRACING

A tracing protocol is executed by users and doctors with a blockchain BC. Algorithms are as follows:

• **Show**(pp, sk, compressedAssertion) → (record, π\textsubscript{S}). Run by a user, with a locally compressed assertion compressedAssertion as input, **Show** rerandomizes it, then outputs a record record and a proof π\textsubscript{S}. The proof should also prove the user’s possession of its sk. Notice that users can run **Show** iteratively while showing multiple records;

• **VrfyShow**(pp, record, π\textsubscript{S}) → \{0, 1\}. Run by a doctor, **VrfyShow** validates record and checks if record embeds the user’s sk concerning π\textsubscript{S}. **VrfyShow** returns 1 if valid, and returns 0 otherwise;

• **Merge**(pp, \{((record, π\textsubscript{S}))\}, BC) → BC’. Run by a doctor with two sub-routines:
  - **VrfyUser**(pp, record, π\textsubscript{S}) → \{0, 1\}. With a list of records and the corresponding proof, **VrfyUser** checks if record embeds the user’s sk concerning π\textsubscript{S}. It returns 1 if true, and returns 0 otherwise;
  - **VrfyShow**(pp, record, π\textsubscript{S}) → \{0, 1\}. With a list of records and the corresponding proof, **VrfyShow** checks if record is obtained correctly from the previous algorithms. It returns 1 if true, and returns 0 otherwise.

Hence, **Merge** creates a collection of records from all the valid records and updates the blockchain with BC’ = BC||records. The **Merge** algorithm can also take in the doctor’s secret key sk\textsubscript{D} and update the blockchain with (records, σ\textsubscript{D}), where σ\textsubscript{D} ← SIG.Sign(sk\textsubscript{D}, records) from a secure signature scheme SIG;

• **Trace**(pp, sk, BC) → \{0, 1\}. Run by a user with secret key sk, we phrase the current blockchain with BC = records

\[\text{records}_0||\text{records}_1||\ldots\text{. For each } i \text{ and record } \in \text{records}, \text{ if sk is involved in record, Trace returns 1, and returns 0 otherwise. Additionally, Trace algorithm can output record and corresponding attributes.}

Remark 1 (Discrete Real-time Tracking): The Trace algorithm behaves differently between contact tracing and discrete real-time tracking. Concretely, in contact tracing, a user checks if any of its secret keys are involved in records on the blockchain. While in the discrete real-time tracking setting, records are based on the location history of diagnosed users, and other users need to check for collision concerning their history and the records. Such process only requires comparison of attributes attrs but no operations on cryptographic keys, and even no counter-party is required. Henceforth, no integrity can be achieved in the discrete real-time tracking setting.

B. THREAT MODEL

We rely only on authorized organizations in the key management protocol to register users’ public keys. Henceforth, only initial public keys are known to authorized organizations, and we therefore regard the trust of authorized organizations as “out-of-band”. We assume doctors are semi-honest, i.e., following the protocol instructions but intending to collect as much information as possible. We assume adversarial users are malicious, e.g., behaving arbitrarily for collecting sensitive data or generating fake contact records. However, as this work focuses on cryptographic attacks, we restrict the adversary from software engineering attacks, e.g., turning off BLE or deleting records from the application. According to previous works \cite{8}, \cite{12}, we define the following threat model to our system:

• **Traceability Correctness.** Users can trace all their contacts with diagnosed users by records on the blockchain;

• **Pseudonym Unlinkability.** A user’s pseudonyms from different attributes cannot be linked with each other when being used in different close contacts;

• **Trace Unlinkability.** Pseudonyms in records on the blockchain cannot be linked with any other pseudonyms of the same user by any other user, including medical doctors;

• **Integrity.** If the two users have no close contact, regardless of being diagnosed or not, one cannot trigger the other’s tracing algorithm. Integrity only works for Bluetooth contact tracing but not for discrete real-time tracking;

• **Identity Authorization & Anonymity.** Users should not be able to submit records signed with unauthorized identity (i.e., public key). In the meantime, such authorized identities should neither be personally identifiable nor be able to disclose anonymous users’ full track.

The restrictions on adversaries are crucial. If one can go offline in the recording protocol or delete records in its storage, traceability correctness is unachievable due to the loss of records in such case.

The unlinkability preserves the anonymity of users’ identities, and the integrity guarantees that no honest user’s alert can be falsely triggered. Moreover, we argue that our system has the following property.
Flooding Proof. The Merge algorithm enables doctors to run a signature scheme before merging new records to the bulletin board, thus preventing adversaries from flooding the bulletin board with unauthenticated records.

Privacy Disclosure. Hiding only users’ identities may fail to preserve privacy when only very few diagnosed users exist in a small area. In such a situation, the user can be identified by the scope of activity, i.e., according to the time and location data in its attributes.

C. DESIGN OVERVIEW

To summarize all components we have mentioned previously in Section II, III and IV, in Figure 2, we clarify their relationships and behaviors.

User’s components can be roughly divided into three categories — contact tracing, PKR-BLS (See Section V), and database. Contact tracing system is responsible for receiving assertions or generating discrete records, which contains all the attributes needed for judging whether the user is in danger or not. Assertions received will be fed into PKR-BLS, and PKR-BLS is responsible for all the cryptographic operations, such as key generation, pseudonym generation, pseudonym updating, assertion verification, etc. The database is used for storing assertions, compressed assertions, and records. Take note that “Record” is not included previously in Section IV-A — we will explain why we need a new component from the output of aggregate key update later in Section VII.

The doctor’s responsibilities include user verification, aggregate assertion verification, and merging (bulletin board).

Details of these components are explained in the following sections, and our implementation is also based on the design described in this section, which will be discussed in Section VII.

V. BUILDING BLOCKS

We first build ourselves a public key rerandomizable BLS signatures scheme (PKR-BLS). The scheme will serve as the main building block for the decentralized tracing system in Section IV-A. The construction is based on the BGLS signature scheme by Boneh et al. [24], and the updatable public keys mechanism by Fauzi et al. [25]. Our PKR-BLS scheme can also be considered as an instantiation of the “signature with flexible public key scheme” [41] with signature aggregation functionality.

A. BLS SIGNATURE SCHEME

Proposed by Boneh et al. [38], the BLS is a signature scheme that enables signature aggregation without requiring signers’ secret keys. The aggregation suffers from the “rogue public key” attack [24]. In this work, we adopt the distinct message set solution from [24]. The modified scheme is usually named by the BGLS signature, and we use “BLS” to keep consistency. Recall the needs for setup:

1) An efficiently computable non-degenerate pairing \( e : G_1 \times G_2 \rightarrow G_T \) in groups \( G_1, G_2, G_T \) of prime order \( q \). We let \( g_1 \) and \( g_2 \) be generators of \( G_1 \) and \( G_2 \) respectively.

2) A hash function \( H_0 : M \rightarrow G_1 \), where \( M \) is the message space.

A BLS signature scheme [24] involves the following algorithms:

- \( \text{KeyGen}(\lambda) \rightarrow (pk, sk) \). With a security parameter \( \lambda \) as input, KeyGen chooses a random \( sk \in \mathbb{Z}_q \) and outputs \( (pk, sk) \) where \( pk \leftarrow g_2^{sk} \in G_2 \).
- \( \text{Sign}(sk, m) \rightarrow \sigma \). Sign takes signer’s secret key \( sk \) and a message \( m \in \mathbb{M} \), it outputs \( \sigma \leftarrow H_0(m)^sk \in G_1 \).
- \( \text{Vrfy}(pk, m, \sigma) \rightarrow \{0, 1\} \). If \( e(\sigma, g_2) = e(H_0(m), pk) \), \( \text{Vrfy} \) outputs 1, otherwise outputs 0;
- \( \text{Aggre}(\sigma_i) \rightarrow \sigma \). Each signer provides a signature \( \sigma_i \in G_1 \) on distinct messages \( m_i \), and anyone can verify with \( \text{Vrfy} \) and compute \( \sigma \leftarrow \prod_{i=1}^{n} \sigma_i \) on valid signatures for the aggregate signature;
- \( \text{VrfyAggre}(pk_1, (m_i), \sigma) \rightarrow \{0, 1\} \). Given an aggregate signature \( \sigma \in G_1 \) and \( (m_i, pk_i) \) of signers who is involved in the aggregate signature, verifiers need to:
  1) ensure the messages \( m_i \) are all distinct and output 0 otherwise;
  2) compute \( h_i \leftarrow H_0(m_i) \) for all users and accept if \( e(\sigma, g_2) = \prod_{i=1}^{n} e(h_i, pk_i) \) holds.

BLS signature schemes should be correct and EUF-CMA secure. Moreover, we adopt the “aggregate chosen-key” security model from [24] for the aggregatable functionality.

1) AGGREGATION UNFORGEABILITY

In an n-user aggregate signature scheme, adversary \( A \) is provided with a single public key. Its goal is the existential forgery of an aggregate signature. The adversary can choose all of the public keys except the challenge public key. The adversary can also access a signing oracle on the challenge key. The game is between a challenger and an adversary \( \tilde{A} \).

- **Setup.** The challenger runs \( \text{KeyGen} \) to generate a key pair \( (pk, sk) \) and sends \( pk \) to the adversary.
- **Queries.** \( A \) chooses messages from \( \mathbb{M} \) adaptively and receives signatures from the challenger with \( sk \).
- **Forgery.** \( \tilde{A} \) outputs \( n - 1 \) additional keys \( pk_2, \ldots, pk_n \), \( n \) messages \( m_1, \ldots, m_n \), and an aggregate signature \( \sigma \) generated from \( n \) messages and keys (i.e., \( n \) pairs of \( (m, pk_i, sk) \)).

\( A \) wins if the aggregate signature \( \sigma \) is a valid signature on \( (m_1, \ldots, m_n) \) under public keys \( (pk, pk_2, \ldots, pk_n) \), without querying to the signing oracle with \( m_1 \) under \( pk \). We model the hash function as a random oracle.

**Definition 1 (Aggregation Unforgeability):** An aggregate signature scheme is \( (t, q_H, q_S) \)-secure against existential forgery in the aggregate chosen-key model if any adversary \( A \) runs within \( t \) time; makes at most \( q_H \) queries to the hash function (random oracle) and at most \( q_S \) queries to the
signing oracle. The following condition holds:

$$\Pr[A \text{ wins the game}] = \text{negl}(\lambda).$$

The probability takes over the randomness of $\text{KeyGen}$ and $A$.

We have the following lemma [24].

**Lemma 1:** The BLS signature scheme is aggregation unforgeable in the random oracle model if the co-CDH assumption holds on the bilinear group $(G_1, G_2)$.

**B. UPDATABLE PUBLIC KEYS**

The notion of the updatable public key (UPK) was proposed in the paper “Quisquis” [25] as a mechanism that updates public keys in a public fashion. We introduce the syntax and security of a UPK system with respect to a signature scheme: $\text{SIG}$ with algorithms $(\text{KeyGen}, \text{Sign}, \text{Vrfy})$. For explicitly, given $(pk, sk) \leftarrow \text{SIG}.\text{KeyGen}(1^\lambda)$, a UPK system involves the following algorithms:

- **Update**($pk, r$) $\rightarrow$ $pk'$. With public key $pk$, Update outputs an updated public key $pk'$ under a fresh randomness $r$;
- **VrfyKP**($pk, sk$) $\rightarrow \{0, 1\}$. VrfyKP takes in and checks the validity of a key pair ($pk, sk$);
- **VrfyUpdate**($pk', pk, r$) $\rightarrow \{0, 1\}$. With public keys $pk'$, $pk$ and randomness $r$, VrfyUpdate checks if $pk'$ is the outputs of $\text{Update}(pk; r)$.

A stand-alone UPK system [25] should be update-correct, update-indistinguishable and secret-key-unforgeable. The integration of the UPK system with a signature scheme $\text{SIG}$ enhances the indistinguishability and unforgeability adversary with accessibilities to a signing oracle. In this paper, we denote the adversary with $A_{\text{Sign}(\cdot)}$ when specifying oracle queries.

**Definition 2 (Update-Correctness):** A UPK system is update-correct if for $\text{SIG}.\text{KeyGen}(1^\lambda) \rightarrow (pk, sk)$ and for any randomness $r$:

- $\text{VrfyKP}(pk', sk) = 1$ holds for the updated key $pk' = \text{Update}(pk; r)$;
- $\text{VrfyUpdate}(\text{Update}(pk; r), pk, r) = 1$ holds.

Consider a game between a challenger and a PPT adversary: $A$ is given with a public key $pk \in (pk, sk) \leftarrow \text{SIG}.\text{KeyGen}(1^\lambda)$; With a random bit $b \xleftarrow{} \{0, 1\}$, the challenger sends $A$:

- $pk_0 \leftarrow \text{Update}(pk)$, an updated public key from $pk \in (pk, sk) \leftarrow \text{SIG}.\text{KeyGen}(1^\lambda)$;
- Or $pk_1 \in (pk_1, sk_1) \leftarrow \text{SIG}.\text{KeyGen}(1^\lambda)$, a fresh public key.

$A$ has accessibilities to a signing oracle $\text{Sign}(\cdot)$ with $sk$ (the original secret key) and $sk_b$ for the challenger’s bit $b$ (for $b = 0$, $sk_0 = sk$); $A$ wins the game if its output a bit $b^*$ that satisfies $b^* = b$ with non-negligible probability of $\lambda$ over $\frac{1}{2}$.
A PKR-BLS scheme involves the following algorithms:

1. Outputs a PKR-BLS scheme that such a modification is suitable concerning our PKR-BLS integration with a randomizable signing oracle, and we argue that this is involved in the aggregate signature, verifiers need to, compute $h_i \leftarrow H_0(h_i, p_i)$ for all users and accept if $e(\sigma, g_2) = \prod_{i=1}^{n} e(h_i, p_i)$ holds.

2. Compute \( \text{Update}(p_i) \): Sample \( r \) $\xleftarrow{\$} \mathbb{Z}_q$. Compute \( p'_{i[1]} = p_{i[1]} \), \( p'_{i[2]} = p_{i[2]} \), and output \( p'_{i} = (p_{i[0]}, p'_{i[1]}, p'_{i[2]}) \);

3. \( \text{VrfyKP}(p_i, sk_2) \): Output 1 if \( p'_{i[1]}^{sk_2} = p_{i[2]} \), otherwise output 0;

4. \( \text{VrfyUpdate}(p'_i, p_i, r) \): Output 1 if \( p'_i = \text{Update}(p_i, r) \) (on each \( p_i \in \{p_i\} \)) and computes \( \sigma' = \sigma' \). It outputs \( p'_i \), \( \sigma' \) or (\( p'_i \), \( \sigma' \)). We require such a modification due to consistency of correctness.

Definition 5 (Correctness): We define correctness over three aspects.

1. Correctness of signature and aggregate signature. For \( \text{KeyGen} \rightarrow (pk, sk_1, sk_2), \) and \( \text{Sign}(sk_1, sk_2, m) \rightarrow (\sigma, \text{Vrfy}(pk, m, \sigma)) = 1 \); For \( \{\text{KeyGen}\}, \{\text{Sign}\}, \) and \( \text{VrfyAggre}(\{pk\}, \{m\}, \sigma) = 1 \).

2. Correctness of updated signature and updated aggregate signature. For \( \{\text{KeyGen}, \text{Sign}, \text{Update} \} \rightarrow (pk, \sigma'), \text{Vrfy}(pk, m, \sigma') = \text{Vrfy}(pk', m, \sigma') \); For \( \{\text{KeyGen}\}, \{\text{Sign}\}, \text{Aggre}, \) and \( \text{Update} \) holds \( \text{VrfyAggre}(\{pk\}, \{m\}, \sigma') = \text{VrfyAggre}(\{pk\}, \{m\}, \sigma') \).

3. Update-correctness (Definition 2) holds.

Construction 1 should be correct, secure under EUF-CMA (strong EUF-CMA is unachievable due to rerandomizability), update-indistinguishable concerning a signing oracle, and aggregation unforgeable. Notice that secret key unforgeability (Definition 4) is covered by unforgeability.

Theorem 1: A PKR-BLS scheme is correct, EUF-CMA secure, update (public key) indistinguishable, aggregation unforgeable if the co-CDH assumption and the SXDH assumption hold on the bilinear group \( (G_1, G_2) \).

\( \text{a: PROOF OUTLINE} \)

Correctness is straightforward. We then show that update-indistinguishability can be reduced to the DDH assumption on group \( G_2 \), which means the SXDH assumption for the bilinear group. For EUF-CMA security and aggregation unforgeability, instead of direct proof, we use secret key unforgeability as an intermediate step. That is, we first reduce the secret key unforgeability to the DDH assumption, discrete-logarithm (DL) in fact. Then, by arguing that adversaries gain no advantages through the update mechanisms, we reduce the EUF-CMA/aggregation unforgeability to the security of the original BLS scheme, which is secure in the random oracle model under the co-CDH assumption.

The game base proofs are as follows.

Proof: Correctness is straightforward to verify.

Given a bilinear group \( (G_1, G_2) \), to prove update-indistinguishability, the SXDH challenger \( \mathcal{C}' \) first samples \( x, y \) $\xleftarrow{\$} \mathbb{Z}_q$ and a random bit \( b \) $\xleftarrow{\$} \{0, 1\}$. \( \mathcal{C}' \) sets
The above situation indicates that $h = g_1^y$, which is $s' = s$. As for $s \in \{x, y, z\}$, $\mathcal{R}_{\text{SXDH}}$ can easily check if $z = xy$ and output $b^*$, regardingly.

With secret key unforgeability, i.e., no adversary can gain advantage from the public key randomizing mechanism, we can consider $\text{SK}_1 \cdot \text{SK}_2$ as the secret key of the BLS scheme and reduce our scheme to the original one.

1) COMPLEXITY OF PKR-BLS SCHEME
In this section, we analyze the complexity of our PKR-BLS signature scheme concerning size and time.

The PKR-BLS scheme with type-3 pairings achieves aggregation unforgeability and security against EUF-CMA under the co-CDH assumption and security against public key re-randomization (update-indistinguishability and secret key unforgeability) under the SXDH assumption. A public key (an updated public key) requires one elements from group $G_1$ and three elements from group $G_2$, i.e., the size is $|G_1| + 3|G_2|$. A signature (an updated signature) is an element from group $G_1$, i.e., the size is $|G_1|$. In the output of aggregation algorithm that involves $n$ signatures, there is an aggregate signature and $n$ pairs of public key and message, which in total the size is $|G_1| + n*|G_1| + 3|G_2| + |\mathcal{M}|$. We compare our size with the original BLS signature scheme [24] in Table 4. Intuitively, we achieve update-indistinguishability with only one additional element from $G_2$ in the public key under the SXDH assumption.

For time complexity, $\text{Sign}$ requires one hashing (i.e., $H_0$) and two exponentiations; $\text{Vrfy}$ requires two pairings; $\text{Aggre}$ that involves $n$ signatures requires $n$ multiplications; $\text{VrfyAggre}$ one hashing (i.e., $H_0$), $n + 1$ pairings, and $n$ multiplications; $\text{Update}$ requires two exponentiations; $\text{VrfyKP}$ requires one exponentiation; $\text{VrfyUpdate}$ recomputes $\text{Update}$, thus it requires two exponentiations. Our algorithms share the same time complexity with the original BLS signature scheme regardless inputs being updated or not.

VI. SYSTEM CONSTRUCTION
We now present the construction of our tracing system and its security analysis.
A. SYSTEM CONSTRUCTIONS

Our construction is based on a general commitment scheme Comm, a signature scheme SIG, and the PKR-BLS scheme PKRBLBS as is follows:

Construction 2 (Contact Tracing System): Recall the system framework in Section IV-A.

- **Key management protocol:**
  - Setup: Generate parameters for the commitment scheme and the PKR-BLS scheme, denoting them with Comm.pp and PKRBLBS.pp. Choose a hash function $H_1 : \{0, 1\}^* \rightarrow Z_q$. Output $pp = (\text{Comm.pp}, \text{PKRBLBS.pp}, H_1)$ where PKRBLBS.pp = $(q, G_1, G_2, G_T, e, g_1, g_2, H_0)$;
  - KeyGen: Run PKRBLBS.KeyGen(PKRBLBS.pp) → (pk, sk₁, sk₂) and Comm.KeyGen(Comm.pp) → ck. Output (pk, sk₁, sk₂, ck), where $pk = (g_2, g_2^r, sk_2^r, sk_1^r, sk_2^r)$;
  - FormNym: Get time slot index $t$. Run $H_1(sk_1, sk_2, t) \rightarrow r$ and PKRBLBS.Update(pk; r) → pk’.
  - **Key registration sub-protocol:**
    - KeyCommit: Run Comm.Commit(Comm.pp, pk, ck) → comm and output the commitment comm;
    - KeyReg: Send comm, a proof of knowledge of relation $\text{comm} = \text{Commit}(pk, ck) \land pk[1]^b_2 = pk[2]$, and the user’s social identities misc to the authorized organizations. If misc exists, return 0.

- **Recording protocol:**
  - Assert: Run PKRBLBS.Sign(sk₁, sk₂, attrs; r) → $\sigma$, where $r$ is the randomness in FormNym. Set $\pi_A$ as a proof of relation $R_1 : \{\text{nym}[1]^b_2 = \text{nym}[2]\}$ where $\text{nym} = (g_2, g_2^r, sk_2^r, sk_1^r, sk_2^r)$. Output assertion = $(\text{nym}, \text{attrs}, \sigma)$ and $\pi_A$;
  - VrfyAsser: Phrase assertion = $(\text{nym}, \text{attrs}, \sigma)$ and run PKRBLBS.Vrfy(nym, attrs, $\sigma$). Verify relation $R_1$ according to $\pi_A$. Output $b = 1$ if both verifications return 1;
  - Save: For an assertion list $\{\text{assertion}\}$, phrase the list with $\{(\text{nym}_i), \{\text{attrs}_i\}, \sigma_i\}$, where each $\text{attrs}_i$ is unique. For each $i$, run PKRBLBS.Agre(σ₁) → $\sigma$. Output $\{(\text{nym}_i), \{\text{attrs}_i\}, \sigma\}$ as compressedAssertion.

- **Tracing protocol:**
  - Show: Phrase record = $\{(\text{nym}_i), \{\text{attrs}_i\}, \sigma\}$, sample $r' \in \mathbb{Z}_q$ and run PKRBLBS.Update $((\text{nym}_i), \sigma; r') \rightarrow (\text{nym}'_i, \sigma')$. Set $\pi_S$ as a proof of relation $R_2 : \{\text{nym}'_i \in \{\text{nym}_i\} \land \text{nym}'_i[1]^b_2 = \text{nym}_i[2]\}$ where $\text{nym}_i = (g_2, g_2^{sk_2^r}, r^{sk_1^r, sk_2^r})$. Output record' = $(\text{nym}_i'), \{\text{attrs}_i\}, \sigma'$ and $\pi_S$;
    - VrfyUser: Verify user’s proof of possession on the record, i.e., $R_2$, according to $\pi_S$ and the commitment scheme Comm;
    - VrfyShow: Verify with PKRBLBS.VrfyAggre. If both verifications return 1, drop the signatures and denote the collection of valid records as records = $\{(\text{nym}_i), \{\text{attrs}_i\}\}$, which contains all $\{\text{nym}_i\}$ and $\{\text{attrs}_i\}$ in the valid records. Run SIG.KeyGen(pp) → (pk₉, sk₉) and sign with SIG.Sign(sk₉, records) → $\sigma_D$. Output BC’ as BC’|(records, $\sigma_D$);
    - Trace: Phrase BC = records₀|| · · · . Run PKRBLBS.VrfyKP(nym, sk₂) → $b$, for each recordsᵢ = $\{(\text{nym}_i), \{\text{attrs}_i\}\} \in$ BC. If $b = 1$, output attrsᵢ for each corresponding nymᵢ; else if $b = 0$ for all nym, output 0.

Remark 2 (Zero-Knowledge Proof-of-Knowledge (ZKPoK)): Notice that we use proof-of-knowledge protocols in three algorithms of Construction 2, i.e., KeyReg, VrfyAsser, and VrfyUser. Intuitively, a ZKPoK enables provers to prove NP statements without revealing any knowledge to verifiers. In our case, a user intends to register itself to authorized organizations without revealing its public key. The user needs to prove its knowledge of the corresponding secret key when registering, contacting other users, and seeing a doctor. Thus, we prevent malicious users from spoofing honest ones. Specially, we require a set membership proof [42] in VrfyShow to prove that the user’s pseudonym is included in its own records. Such property guarantees the integrity and preserves users’ pseudonyms from being exposed to the doctor.

B. SECURITY ANALYSIS

Construction 2 is traceability-correct if for any nym ∈ BC and users’ sk₂, PKRBLBS.VrfyKP(nym, sk₂) = 1, which is guaranteed by the update-correctness of the PKR-BLS scheme.

For pseudonym unlinkability of the recording phase, users $U_1$ and $U_2$ have contact in time slots $t_1$ and $t_2$. $U_1$ asserts with different pseudonyms and attributes to generate two signatures: $\sigma^{t_1} \leftarrow \text{Sign(nym}^{t_1}, \text{attrs}^{t_1})$ and $\sigma^{t_2} \leftarrow \text{Sign(nym}^{t_2}, \text{attrs}^{t_2})$. PKR-BLS scheme’s update-indistinguishability guarantees that $U_2$ cannot distinguish nym₁ from nym₂ even after learning $\sigma^{t_1}$ and $\sigma^{t_2}$.  

### Table 4: Size comparison with the original BLS scheme [24].

|                        | Assumption | Security Model       | pk Size | $\sigma$ Size | Output Size of Aggre |
|------------------------|------------|----------------------|---------|---------------|----------------------|
| The BLS Scheme [24]    | co-CDH     | EUF-CMA              | $|G_1|+|G_2|$ | $|G_1|$         | $(|G_1|+n*(|G_1|+|G_2|+|M|))$ |
| Our PKR-BLS            | co-CDH + SXDH | EUF-CMA + Update-Ind. | $|G_1|+|G_2|$ | $|G_1|$         | $(|G_1|+n*(|G_1|+|G_2|+|M|))$ |
For trace unlinkability of the tracing phase, we require two aspects:

1) For a doctor, we take a black-box use of a zero-knowledge protocol to prove the user’s (sk₁, sk₂) is corresponding to one of the pseudonyms in its records. Also, each user updates its records before showing them to doctors, thus, preserving the trace unlinkability from doctors;

2) For other users who run the Trace algorithm, by PKR-BLS scheme’s update-indistinguishability, pseudonyms on the blockchain are indistinguishable from those stored in their local storage. Thus, our protocol is able to preserve the trace unlinkability from normal users.

For integrity, although adversary can create assertions on arbitrary attributes, to win the integrity game, i.e., to trigger an honest user’s Trace algorithm to output 1, a diagnosed adversary A has to create a record involving one of the honest user’s assertion. However, we need to notice that A and U have no contact. Thus A does not know U’s signature, nor can A forge one (by EUF-CMA). Moreover, after A learns (nym), (attrs) from BC and intends to forge a valid record record_A = (nym) ∪ nym_A, (attrs) ∪ attrs_A, A will fail with significant probability due to the aggregation unforgeability (as the adversary needs to forge on multiple signatures). Thus, integrity is preserved due to the EUF-CMA secure and aggregation unforgeability of our PKR-BLS scheme.

VII. IMPLEMENTATION AND EVALUATION

Our work is involved with cryptographic algorithms and is expected to operate on mobile devices, whose performance is comparatively limited compared to traditional cryptographic scenarios. Therefore, we decide to implement our algorithm in the Android application to demonstrate the practicality and performance of our system.

In this section, we will explain how we implement our system, as illustrated in Figure 2, with focus on some engineering issues and how they affect the framework.

A. IMPLEMENTATION

A user should hold a Database, which includes assertion, compressedAssertion, and record. We will explain why we need 3 types of asserts later in this section. Public keys need to be periodically updated to prevent the whole movement track from being exposed (i.e., FormNym).

Database is used for storing assertions either 1) generated by a signature scheme (e.g., PKR-BLS scheme in Section V) using attributes or 2) received from BLE scanning service and verified by the signature.

The assertion(id, pk, attrs, σ, g, g’) contains auto-generated device-wide unique assertion ID, pseudonym (i.e. nym), message (i.e. attrs), signature (i.e. σ), group generator (i.e. g) and contact tracing mode flag (i.e. isBLE).

Although g is contained in nym, we still need it to convert pk from primitive data type (which can be stored in database) to abstract data type used for cryptographic calculations (i.e., computing g).

Therefore, we cannot compute g directly from nym in primitive data type stored in the database, and that is why we need to make assertions contain g. Moreover, g’ is used for checking close contacts after collecting records from the blockchain.

Database also stores compressedAssertion aggregated from assertions by PKR-BLS scheme. Theoretically, as described in Section IV-A2, compressedAssertion should contain an aggregated signature (i.e. PKRBLSAggreen (σ_j) → σ), a list of pseudonyms (i.e. {nym}), a list of attributes (i.e. {attr_i}), and a list of group generators (i.e. {gi}).

Since we need to prevent duplicate data stored in the device, we prefer storing pointers in compressedAssertion to assertions using primary keys (i.e. ids of assertions).

Hence, compressedAssertion is defined as (σ, [id,]).

An infected user will update compressedAssertion using PKRBLSAggreen (See Section IV-A2), and therefore generate records, containing all associated assertions using ids, as the system needs to submit “real” data, instead of pointers.

Hence, user send (i.e. Show) a list of record and corresponding proof π_S to doctor. The doctor will use π_S, together with commitments associated with this user, which is registered to authorized organizations, to verify the user’s identity (i.e. VrfyUser). The doctor also needs to check all assertions contained in each record to verify the relationship between committed public key and key used to sign submitted assertions.

As stated in Section IV-B, because the user’s own assertions and received assertions cannot be distinguished due to security considerations, the system behaves differently for discrete and Bluetooth records. Since all discrete records are asserted by the user him/herself, the system only needs to verify all the nyms one by one. For Bluetooth assertions (i.e. isBLE = True), the user needs to periodically compress assertions received with the user’s own assertion, based on how many assertions are there left to be compressed. Therefore, if we assume the user operates compression for every N assertions received, the number of assertions belonging to the user him/herself should equal to

\[ \frac{\text{Number of assertions}}{N} \]

After identity verification, signature verification and merging have been explained explicitly. Therefore, we will not repeat them in this section.

It needs to be emphasized that, as Section IV-A2 indicates, besides pseudonym, key-pair, and database, a user also holds secret explicit randomness r to generate signature after key updating. After the user updates the public key, if this user uses the previous randomness to generate a new signature, key and updated key’s relationship can be easily referred if an attacker runs the verification function twice: to check if PKRBLSVrfy(nym, attrs, σ) = PKRBLSVrfy(nym’, attrs, σ).

1https://github.com/TaihouKai/PBK_Test
As mentioned in the Show algorithm in Section IV-A, we require users to update their stored records, i.e., users should update the pseudonym list and the corresponding aggregate signature with a newly sampled randomness. We guarantee that even if doctors misbehave and store previous users' records (thus know some pseudonyms), they cannot link those pseudonyms with any new-coming users. By the correctness of our signature scheme, doctors run VrfyAggre in their verification phase and the algorithm will output 1 if the aggregate signature is correct and is updated by the same randomness with its corresponding pseudonyms.

During tracing, our system can distinguish BLE records from discrete records using isBLE flag since BLE records and discrete records have different ways to trace. (See Section IV-A and Section III-A for tracing mechanisms.)

B. EVALUATION
We implemented our system on both Android Virtual Device (AVD) and physical devices (Samsung Galaxy Note 10+ & Google Pixel 3A) to prove its functionality and performance. Furthermore, our application utilizes the Java Pairing Based Cryptography Library (JPBC) [43], extending and modifying it based on our design, as discussed in Section V.

There are roughly two types of functions in our contact tracing system — what needs to be performed frequently and what needs to be performed less frequently.

To be specific, KeyGen only needs to be performed once per user generation; Update needs to be performed periodically (e.g., once per hour); Save needs to be performed less frequently than Update (e.g., once per day); Show and its verification only happen during diagnosis. Since time consumption of these functions is significantly shorter than time intervals between two launches, as shown in Table 5, we consider our system to be practical regarding these functionalities mentioned above.

It needs to be mentioned that time consumption of Save depends on the amount of assertions the system trying to aggregate. We provide its time consumption with different amount of assertions in Figure 5.

However, the most time-consuming functions are expected to be performed frequently, i.e., assertion generation and assertion verification. Bluetooth scanning scenario requires both generation and verification, while discrete real-time tracking requires only assertion generation (since users do not care about other users’ assertions under this scenario).

Consider an extreme case in which a user is tightly surrounded by 100 users. In such a case, the discrete scenario will take (approx.) 13.5s to generate assertions, which is acceptable in our opinion. However, the Bluetooth scenario will take (approx.) 36.3s to generate and verify assertions, which is significantly slower. However, since Bluetooth scanning only happens in public transports as explained in Section III-B, this user would have plenty of time to move together with the same group of surrounding users. At the same time, assertion generation and verification can be performed. Also, 36.3s is fast enough for the functions to be completed before the next stop.

Although Bluetooth scenario seems to take longer time to operate, as we have explained in Section IV-A2, we still expect our system to have notably better performance than current contact tracing systems in whatever scenario due to our system’s interaction-less nature, since simultaneously handshaking with 100 users would cost a significant amount of time.

VIII. FUTURE WORK
Several unsolved problems will be discussed in this section.

3We decide not to conduct a real-world physical experiment involving such amount of users to evaluate the performance of the contact tracing systems because such interaction-based scanning heavily depends on transmission environment and devices, and letting hundreds of participants sticking tightly together during the pandemic violates social distancing rule and may cause group infection which would risk participants’ personal health or even life, as COVID-19 may cause severe sequelae to infectees [44]. We cannot simply put devices together since the human body, movement and surrounding structures also interfere with Bluetooth communication. Therefore, such experiment is only held between two physical and two virtual devices and accumulated statistics is calculated using multiplication, by repeating the experiment between single pairs of devices.
First of all, our work mainly focuses on proposing a cryptographic framework, and designing a system to prove our framework to be fully practical in normal cases. However, we have not focused on concrete engineering issues (e.g., latency caused by networking problems of weather services mentioned in Section II-B3) caused by complex and unpredictable real-world scenarios. Also, as our system is only designed to be able to operate within reasonable amount of time, we have not compared the performance of our system with other systems mentioned in Section I-B. Regarding these problems, we are confident that there will be engineering solutions or improvements and we do not cover them in this paper.

Second, the impact of indoor airflow organization is neglected as we cannot measure it. We hypothesize that airflow organization can be roughly categorized by general purposes of usage and structures of the buildings. We hope researchers can do aerodynamic experiments for mobile devices to estimate airflow organization without professional equipment.

Third, the impact of outdoor airflow organization is only considered for droplet transmission. We assume that, for outdoor airborne transmission, airflow will rapidly dilute and accelerate the droplets. When droplets are diluted and accelerated, their infectivity will be largely reduced because of lower concentration and faster deactivation by ultraviolet light due to faster evaporation caused by acceleration. We hope researchers in the future can validate our assumption.

Fourth, the discrete real-time tracking does not cover the exceptions, as described in Section III-B.

Fifth, although the unlinkability of pseudo-nymns preserves users’ anonymity, the plaintext assertion of attributes may expose their identities. We thus consider a “blind-asserting” algorithm in the recording protocol (See Section IV-A2). However, without proper construction, we list it as future work.

Sixth, as described in Section IV-B, due to the nature of our approach, which requires a complete track of users (though discretely and anonymously), it may still fail if there are only a few positive cases within an area. We hope there is another way to track the lifetime of floating droplets to avoid this issue.

Finally, for Figure 5, our data records show that time consumption **approximately** decreases every time we make a new 100-time attempt, for the same x. We hypothesize that this is due to memory caching that mobile applications are not privileged enough to manage, and we hope researchers in the future may be able to figure out what causes this phenomenon. Also, if memory caching causes this, we want to know whether there is any way to use its mechanism to improve the efficiency of the operations in our system.

**IX. CONCLUSION**

In this work, we proposed a cryptographic framework for contact tracing and provide a construction based on public key rerandomizable BLS signature, with a bulletin board based on blockchain to safely distribute contact records among users. Our generic framework/construction is capable of contact tracing setting, including discrete real-time tracking, without changing any building blocks. With security properties of the underlying signature scheme (See Section V-C) and the commitment scheme (See Section II-C1, the purpose is to prevent users from generating multiple identities and having only 1 or some of them being marked as “infected” after diagnosis), our framework achieves confidentiality and partial integrity simultaneously, which have not been achieved by any reviewed related work.

In order to provide such cryptographic framework with a practical implementation, and to solve the problem of precision (false-positives), scalability and flooding, we designed a contact tracing system, using environmental factors to adjust timestamp, location-stamp, and range-stamp, influenced by temperature, relative humidity, and airflow organization, in order to provide a precise contact tracing approach. Within the system, we proposed discrete real-time tracking to fulfill the new needs of airborne transmission (See Section II-B), which current systems do not have. Our discrete real-time tracking should be combined with existing contact tracing approaches to cover more scenarios than any reviewed related work.

Based on our design, we provided a demonstration with Java on Android OS. To prove its performance, we used both physical and virtual device in our experiments and concluded that our system should be able to handle the most congested scenarios as we described in Section VII-B.

We hope our system could serve as a better alternative to existing contact tracing systems to improve both physical and cyber security.

**ACKNOWLEDGMENT**

An earlier version of this paper was presented at the 13th International Symposium on Cyberspace Safety and Security (CSS 2021), titled “Environmental Adaptive Privacy Preserving Contact Tracing System for Respiratory Infectious Diseases” [DOI: 10.1007/978-3-030-94029-4_10].

**REFERENCES**

[1] P. Wang, X. Su, M. Jourenko, Z. Jiang, M. Larangeira, and K. Tanaka, “Environmental adaptive privacy preserving contact tracing system for respiratory infectious diseases,” in Cyberspace Safety and Security, W. Meng and M. Conti, Eds. Cham, Switzerland: Springer, 2022, pp. 131–144.

[2] (Apr. 2020). Apple and Google LLC. Privacy-Preserving Contact Tracing. [Online]. Available: https://covid19.apple.com/contacttracing

[3] Scientific Brief: Sars-Cov-2 and Potential Airborne Transmission, Nat. Center Immunization Respiratory Diseases, Tech. Rep., Centers Disease Control Prevention, Atlanta, GA, USA, Oct. 2020.

[4] X. Shen. (May 2020). Personal Information Collected to Fight COVID-19 is Being Spread Online in China. Accessed: Feb. 2021. [Online]. Available: https://www.wccftech.com/article/30784804/personal-information-collected-for-fight-covid-19-is-being-spread-online-china

[5] T. Starks. (Jul. 2020). Early COVID-19 Tracking Apps Easy Prey for Hackers, and it Might Get Worse Before it Gets Better. Accessed: Feb. 2021. [Online]. Available: https://www.politico.com/news/2020/07/06/coronavirus-tracking-app-hack%e2%80%93ng-348601
[6] Z. Whittaker. (Aug. 2020). Fearing Coronavirus, a Michigan College is Tracking its Students With a Flawed App. Accessed: Feb. 2021. [Online]. Available: https://techcrunch.com/2020/08/19/coronavirus-albion-security-flaws-app%

[7] (Apr. 2020). Signees of the Joint Statement. Joint statement. Accessed: Feb. 2021. [Online]. Available: https://drive.google.com/file/d/1b4LaQHMVf4jZJfHA9SjKJuiMd3erGq/view

[8] N. Danz, O. Derwisch, A. Lehmann, W. Puentet, M. Stolle, and J. Ziemann. “Security and privacy of decentralized cryptographic contact tracing,” Cryptol. ePrint Archive, Univ. Potsdam, Potsdam, Germany, Tech. Rep. 2020/1309, 2020.

[9] I. Levy. “High level privacy and security design for NHS COVID-19 contact tracing app,” Nat. Cyber Secur. Centre, London, U.K., Tech. Rep., May 2020. [Online]. Available: https://www.nscg.gov.cn/report/nhs-covid-19-app-privacy-security-report

[10] COVID-19 Contact-Confirming Application, Ministry Health, Labour Welfare Japan, Tokyo, Japan, Tech. Rep., Dec. 2020.

[11] Notice on Online Application for Health Declaration Certificate, COVID-19 Contact-Confirming Application

[12] D. Boneh, C. Gentry, B. Lynn, and H. Shacham. “Aggregate and verifiably (Jan. 2016).

[13] L. Garg, E. Chukwu, N. Nasser, C. Chakraborty, and G. Garg. “Anonymity (Apr. 2020).

[14] Z. Whittaker. (Aug. 2020).

[15] J. Daemen and V. Rijmen. “The block cipher Rijndael,” in Cryptol. ePrint Archive, Oct. 2010, pp. 10, no. 88, Nov. 2013, Art. no. 20130560.

[16] N. Fire and P. Agency. “NFPA-45 standard on fire protection for laboratories using chemicals.” Nat. Fire Protection Assoc., Quincy, MA, USA, Tech. Rep., 2019. [Online]. Available: https://www.nfpa.org/codes-and-standards/all-codes-and-standards/list-of-codes-and-standards/detail?codesy

[17] Act on Maintenance of Sanitation in Buildings, Cabinet Japan, Japan, 1970. [Online]. Available: https://laws.e-gov.go.jp/document?lawid=345C000000000304_20191216_501C00000000183

[18] Osha Policy on Indoor Air Quality: Office Temperature/Humidity and Environmental Tobacco Smoke, United States Dept. Labor, Washington, DC, USA, Feb. 2003.

[19] Workplace (Health, Safety and Welfare) Regulations 1992, Health and Safety Executive, U.K., 1992. [Online]. Available: https://www.hse.gov.uk/pubns/books/124.htm

[20] P. François. “Code de l’énergie,” French Parliament, Paris, France, Tech. Rep., 2016. [Online]. Available: https://www.legifrance.gouv.fr/Code/id/LEGITEXT000002398320/

[21] Norms for Public Health Administration of Central Air-Conditioning Systems in Public Places, Nat. Health Commission People’s Republic China, Beijing, China, 2012.

[22] (Feb. 2, 2022). Google LLC. Place Types. [Online]. Available: https://developers.google.com/maps/documentation/places/web-service/sup%port%ed_types

[23] E. Fujisaki and T. Okamoto. “Statistical zero knowledge protocols to prove modular polynomial relations,” in Advances in Cryptology (Lecture Notes in Computer Science), vol. 1294, B. S. Kaliski, Ed. Santa Barbara, CA, USA: Springer, Aug. 1997, pp. 16–30.

[24] I. Damgård and E. Fujisaki. “A statistically-hiding integer commitment scheme based on groups with hidden order,” in Advances in Cryptology (Lecture Notes in Computer Science), vol. 2501, Y. Zheng, Ed. Queenstown, New Zealand: Springer, Dec. 2002, pp. 125–142.

[25] D. Johnson, A. Menezes, and A. Scott Vanstone. “The elliptic curve digital signature algorithm (ECDSA),” Int. J. Inf. Sec., vol. 1, no. 1, pp. 36–63, 2001.

[26] S. D. Galbraith, K. G. Paterson, and N. P. Smart. “Pairings for cryptographers,” Discrete Appl. Math., vol. 156, no. 16, pp. 3113–3121, 2008.

[27] D. Boneh, B. Lynn, and H. Shacham. “Short signatures from the Weil pairing,” in Advances in Cryptology—ASIACRYPT (Lecture Notes in Computer Science), vol. 2248, Gold Coast, QLD, Australia: Springer, 2001, pp. 514–532.

[28] L. Ballard, M. Green, B. D. Medeiros, and F. Monrose. “Correlation-resistant storage via keyword-searchable encryption,” IACR Cryptol. ePrint Arch., Johns Hopkins Univ., Baltimore, MD, USA, Florida State Univ., Tallahassee, FL, USA, Tech. Rep. 2005/417, 2005, p. 417.

[29] S. Barai, D. Biswas, and B. Sau. “Estimate distance measurement using NodeMCU ESP8266 based on RSSI technique,” in Proc. IEEE Conf. Antenna Meas. Appl. (CAMA), Dec. 2017, pp. 170–173.

[30] M. Backes, L. Hanzlik, K. Kluczniak, and J. Schneider. “Signatures with flexible public key: Introducing equivalence classes for public keys,” in Advances in Cryptology (Lecture Notes in Computer Science), vol. 11273, T. Peyrin and S. D. Galbraith, Eds. Brisbane, QLD, Australia: Springer, Dec. 2018, pp. 405–434.

[31] J. Camenisch, R. Chaabouni, and A. Shelat. “Efficient protocols for set membership and range proofs,” in Advances in Cryptology (Lecture Notes in Computer Science), vol. 5150, J. Pieprzyk, Ed. Melbourne, VIC, Australia: Springer, Dec. 2008, pp. 234–252.

[32] A. De Caro and V. Iovino. “JPBC: Java pairing based cryptography,” in Proc. IEEE Symp. Comput. Commun. (ISCC), Jun. 2011, pp. 285–855.

[33] A. De Caro and V. Iovino. “JPBC: Java pairing based cryptography,” in Proc. IEEE Symp. Comput. Commun. (ISCC), Jun. 2011, pp. 80–855.

[34] P. Wang et al. "When COVID-19 Symptoms Linger,"[Online]. Available: https://covid19.nih.gov/news-and-stories/when-COVID-19-symptoms-linger
PENGFEI WANG received the B.S. degree in computer science from Furman University, in 2016, and the M.S. degree in cybersecurity from Fordham University, USA, in 2019. He is currently pursuing the Ph.D. degree in cybersecurity with the Tokyo Institute of Technology, Tokyo, Japan. His research interests include taking cryptographic algorithms and frameworks into the industry with practical implementation, particularly for which were previously considered to be only theoretical.

XIANGYU SU received the bachelor’s degree from the Department of Information Theory and Data Science, Nankai University, China, in 2016, and the master’s degree from the Department of Mathematical and Computing Science, Tokyo Institute of Technology, Japan, in 2019, where he is currently pursuing the Ph.D. degree.

MAXIM JOURENKO received the B.Sc. and M.Sc. degrees in computer science from RWTH Aachen University, in 2013 and 2017, respectively, and the Ph.D. (Doctor of Science) degree in mathematical and computing sciences from the Tokyo Institute of Technology, in 2021. He worked at the European Organization for Nuclear Research, as a Technical Student, in 2016 and 2017. He is currently working as a Postdoctoral Researcher at the Department of Mathematical Informatics, Graduate School of Information Science and Technology, The University of Tokyo. His current research interests include cryptographic protocols and Layer-2 protocols for blockchain.

ZIXIAN JIANG received the B.S. degree in medical from Tianjin Medical University, Tianjin, China, in 2012, the second B.S. degree in medical from Nagoya University, Nagoya, Japan, in 2016, and the M.S. degree in life science from Kyoto University, Kyoto, Japan, in 2018, where he is currently pursuing the Ph.D. degree in life science. From 2016 to 2021, he was a Teaching Assistant in the seminar of Advanced Molecular and Cell Biology and Cancer Biology. His research interest includes the force inside of the cell and acting dynamics during cell circle, and has lots of skills and experiences on handling the atomic force microscope.

MARIO LARANGEIRA received the degree in computer engineering from the Technological Institute of Aeronautics (ITA), in 2004, and the M.S. and Ph.D. degrees from the Tokyo Institute of Technology, in 2008 and 2013, respectively. He is currently a Cryptographer working as a Research Fellow at the Input Output Hong Kong (IOHK), and a Specially Appointed Associate Professor of security, cryptography and blockchain systems at the Department of Mathematical and Computing Science, Tokyo Institute of Technology. His research interest includes published works on the area of blockchain-based systems, public-key cryptographic schemes, secure multiparty computation protocols, and security models for provable security.

KEISUKE TANAKA received the B.E. degree in computer science from Yamanashi University, in 1992, and the M.S. and Ph.D. degrees in computer science from the Japan Advanced Institute of Science and Technology, in 1994 and 1997, respectively. Before joining the Tokyo Institute of Technology, he was a Research Engineer at NTT Information Sharing Platform Laboratories. He is currently a Professor with the Department of Mathematical and Computing Sciences, Tokyo Institute of Technology.

* * *