vContact: Private WiFi-based Contact Tracing with Virus Lifespan

Guanyao Li, Siyan Hu, Shuhan Zhong, and S.-H. Gary Chan

Abstract—Covid-19 is primarily spread through contact with the virus which may survive on surfaces with lifespan of more than hours. To curb its spread, it is hence of vital importance to detect and quarantine those who have been in contact with the virus for sustained period of time, the so-called close contacts. The existing digital approaches for contact tracing focus only on direct face-to-face contacts. There has not been any work detecting indirect environmental contact, which is to detect people coming into an area with living virus, i.e., an area previously visited by an infected person within the virus lifespan. In this work, we study, for the first time, automatic contact detection when the virus has a lifespan. Leveraging upon the ubiquity of WiFi signals, we propose a novel, private, and fully distributed WiFi-based approach called vContact. Users installing an app continuously scan WiFi and store its hashed IDs. Given a confirmed case, the signals of the major places he/she visited are then uploaded to a server and matched with the stored signals of users to detect contact. vContact is not based on phone pairing, and no information of any other users is stored locally. The confirmed case does not need to have installed the app for it to work properly. As WiFi data are sampled sporadically, we propose efficient signal processing approaches and similarity metric to align and match signals of any time. We conduct extensive indoor and outdoor experiments to evaluate the performance of vContact. Our results demonstrate that vContact is efficient and robust for contact detection. The precision and recall of contact detection are high (in the range of 50–90%) for close contact proximity (2m). Its performance is robust with respect to signal lengths (AP numbers) and phone heterogeneity. By implementing vContact as an app, we present a case study to demonstrate the validity of our design in notifying its users their exposure to virus with lifespan.

Index Terms—contact tracing, exposure notification, proximity detection, contact detection, COVID-19, epidemic control.

1 INTRODUCTION

The outbreak of COVID-19 has had a profound impact on our lives and the global economy. COVID-19, like many other infectious diseases, is primarily spread through viral contact. Recent studies have shown that the virus has a lifespan: as airborne droplets it can last more than 10 minutes, and on surfaces it can survive from hours to days if not properly disinfected (in low temperatures it may last even longer) [1][2]. The health of any person coming into contact with the virus for a sustained period of time, say 15–30 minutes, may be at risk [3]. In order to effectively contain the disease, tracing and quarantining these close contacts as soon as possible is of paramount importance.

Traditionally, close contacts are traced manually through personal interviews with infected people by medical officers. Such manual approach is labour-intensive and slow. Due to mis-memory, the contact information may be incomplete or error prone. Furthermore, the patient would not possibly know the people in his/her proximity, and who came into an area within the virus lifespan after he/she left.

To address the above problems, we propose a novel, private, and digital contact tracing approach with virus lifespan. Anyone in contact with living virus is considered at risk. This includes those simultaneously locating with the patient, and those visiting an area within the virus lifespan after a patient. Leveraging upon ubiquitous WiFi signals everywhere, we propose an automatic and fully distributed WiFi-based approach called vContact to detect the close contacts. As far as we know, this is the first piece of work considering virus lifespan in private contact tracing. Note that though for concreteness our discussion will focus on WiFi signal, vContact can be straightforwardly extended and applied to other radio-frequency signals (such as Bluetooth) and their combination.

We illustrate the process of vContact in Figure 1. A user first installs an application (app) and turns on the WiFi sensor of the phone. The app periodically scans for WiFi, with each scan a signal vector consisting of the following two elements: 1) the signal IDs, which are the hashed (and optionally encrypted) values of the MAC addresses of the WiFi access points (APs); and 2) the corresponding received signal strength indicators (RSSIs) of the signal IDs. A signal vector is associated with a timestamp, which is the scanning/collection time of the signals. Each signal vector may be kept for a certain duration corresponding to the virus incubation period, say 14 to 28 days. Over time, the phone collects and stores a time series of the signal vectors, termed signal profile, as the user roams in the city.

Upon positive confirmation in hospital, the patient has the following two possibilities:

• With the app installed: With the consent of the patient, the health officer may access his/her signal profile (the patient may blank out or filter some parts of the
signal profile for personal reasons before sharing it with the officer). Note that due to AP MAC hashing (and possibly encryption), the officer does not know the patient geo-locations, but only clusters of anonymized IDs with different dwell time. Based on that, the officer works with the patient to identify the venues of potential health risks to the public. These anonymized IDs of risk are extracted and labelled with assessed virus lifespan, and the processed signal profile is uploaded to a secure server for other users to match in a distributed manner. Upon matching, users are alerted in private if they have close contact with the virus.

- **Without the app:** In this case, the confirmed case has to rely on his/her memory of the major venues and their visit time as the manual case. Then some staff will go to the places (the infected areas) to collect offline their WiFi information and label them with the visit time of the patient. These signals, after hashing and processing, are then uploaded and matched by the users the same way as in the case above. (Clearly, we consider the realistic condition that all WiFi signals at a position do not change drastically over some short period of time, say, days, so that the signals collected some days after the patient’s visit still reflects well the signals of the visit then.)

Works have been done on automatic digital contact tracing. Some use GPS [4] and cellular signals [5]. While effective, these approaches cannot be extended to indoor environment. They are also based on explicit user geo-locations, which raises concerns on location privacy. Some privacy-preserving approaches based on Bluetooth have attracted much attention and been implemented recently [6] [3] [7]. However, they work for only direct face-to-face contact tracing, and cannot be applied to the case with non-zero virus lifespan (environmental exposure). They are also based on phone pairing and communication, which leads to concerns on security and require a high adoption rate for effective tracing. vContact is orthogonal to them, and may be integrated with some of them (such as [3] [7]). Compared with existing works, vContact has the following strengths:

- **Contact detection with virus lifespan:** vContact is the first piece of work to capture the realistic scenario of virus lifespan. It comprehensively covers, in a single framework, those in direct face-to-face contact with an infected person and indirect environmental exposure with the areas previously visited by an infected person. The lifespan of the virus, set at the time of signal upload, may be heterogeneous depending on the frequency of disinfection operation in the venue.

- **No phone-to-phone pairing and communication:** Prior contact tracing proposals based on Bluetooth require phone pairing, which means both phones, including the infected one, have to install the app simultaneously for it to work properly. To achieve tracing effectiveness, they hence demand a high adoption rate (in the range of reportedly 40% – 70%). In contrast, each vContact phone operates independently without any pairing or communication, and does not require the confirmed case to have installed the app. This greatly reduces the adoption barrier. Furthermore, app users do not store any information of or exchange any message with the other users; it hence offers much better protection on user anonymity and attacks.

- **Data privacy:** vContact uses no personal information such as names, phone numbers, IDs, contact lists, images/videos, etc. Because phones are independent without any mutual exchange of information, no data are generated and communicated between phones, and no information of other users is kept in a phone. The collected WiFi data with hashed IDs are exclusively stored in one’s own phone. The phone data never leave local storage without the explicit consent of the user, and even so (i.e., the case of a confirmed case) the data remain anonymous at the server. Upon detection of close contact, vContact conveys such message to its user in private without any data upload.

- **Decentralization:** vContact is fully decentralized where contact is computed locally on user phones in a scalable manner without any entity (users or server) having full information. Such data fragmentation and minimization prevent data re-purposing, abuse, and mis-use. As no user data is stored anywhere else beyond one’s phone, a user may exit the system at any time by un-installation without leaving his/her data behind. The system can also be quickly dismantled through such app un-installation once the pandemic is over.

- **No GPS-based geo-location:** vContact is not based on GPS signal. Because it uses only the hashed values of WiFi MAC addresses (signal IDs), the user’s physical geo-location is non-transparent and unnecessary. This offers much stronger location confidentiality than other GPS-based geo-location approaches, because the association of all signal IDs to their physical locations takes enormously large amount of manual work (that is to visit every indoor and outdoor corner of the city and logging down the locations of all the MAC addresses encountered). Furthermore, unlike other GPS approaches, vContact can detect indoor contacts, and hence is more pervasive by covering both indoor and outdoor areas.

Detecting close contact using WiFi data is a challenging
problem because of the following issues. First, signal vectors are sampled sporadically at random discrete time, resulting in difficulties to detect contact at any arbitrary time. Moreover, signals may be sparsely sampled in the space. The scanned IDs of different users may also differ due to phone heterogeneity on antenna design and sensitivity. vContact overcomes these problems by employing a novel approach to represent the values between consecutive signal vectors and an efficient similarity metric to match signal values for contact detection.

We present in Figure 2 an overview of contact detection using WiFi in vContact. If the signal profile of the confirmed case is available, we represent the RSSIs between consecutive signal vectors of the patient as a processed vector to support vector comparison. The resultant sequence of these processed vectors is called processed profile. For the case where the confirmed case has not installed the app, the WiFi data collected by staff in the infected areas are transformed to a processed profile. Given the signal vector of a user at time $t$, it falls in the time range of the virus lifespan of a processed vector, we compare their level of matching using our proposed signal similarity metric. If the similarity is larger than a given threshold $\alpha$, the user is identified as having contact with the virus at time $t$. A user is identified as a close contact if the contact time exceeds a certain sustained period of time as specified by health officials.

The remainder of this paper is organized as follows. Section 2 discusses related works. Section 3 presents the approach of vContact. We have implemented vContact as a software development kit (SDK), and discuss the experiment setting and illustrative results on the SDK in Section 4. With the SDK, we have developed an app and present its implementation details and measurement in Section 5. We conclude in Section 6.

2 RELATED WORKS

Contact tracing has attracted much attention due to the outbreak of the COVID-19 pandemic. In this section, we present some well-known systems and schemes proposed in the industry and academia.

Google and Apple provide a toolkit for privacy-preserving contact tracing using Bluetooth. Various contact tracing systems have been deployed in some countries, such as TraceTogether in Singapore, COVIDSafe in Australia, Corowarner in Turkey, Aarogya Setu in India, Cotrack in Argentina, Covid Watch in US, etc. TraceTogether, COVIDSafe and Covid Watch use Bluetooth, Corowarner and Aarogya Setu use GPS and Bluetooth, and Cotrack uses RFID, GPRS, GPS, and telecommunication technologies. TraceTogether is the app released by the government of Singapore. Users broadcast their ID using Bluetooth and scan their nearby user’s ID. When a user is infected, the government can trace people who have close contact with the infected one based on the data. Other apps are based on the similar concept as TraceTogether. As compared with them, vContact is not based on phone pairing, and hence it breaks the requirement of simultaneous app installation. It also offers better user privacy because users are informed in private on contacts.

Contact tracing becomes a hot research topic in academia recently. Many research works focus on contact tracing using different signals. Some of them use signals which reveal user geo-location, such as GPS, cellular data, and radio frequency identification (RFID). GPS signal provides a user’s exact location, but it is usually weak and noisy in indoor environment, which limits its contact coverage. Cellular data can also be used for contact tracing to infer a user’s public transportation trips. Given the user cellular data, we can detect users who are taking the same bus, train, or subway as a confirmed case. However, the radius of the cell towers signal is large, and hence close proximity is difficult to be detected. Some researchers also propose using RFID to understand contact. However, extra devices have to be deployed. GPS data, cellular data, and RFID data can be extended to contact tracing with virus lifespan, but they may raise privacy concerns as they provide user geo-location. Compared with them, vContact uses only the hashed values of WiFi MAC addresses (signal IDs), users physical geo-location is non-transparent and unnecessary, offering much stronger location confidentiality.

To protect user location privacy, some works propose using magnetometer and Bluetooth data for contact tracing, in which user geo-location is not required. However, geomagnetism suffers from environment change. Even a small change in the environment may result in different geomagnetism signals for a location, which limits its extension to contact tracing with virus lifespan. The works focusing on Bluetooth data can be categorized into two groups. Some works rely on a third-party server for contact tracing, which raises the concern of possible data abuse. To address it, others advocate fully distributed approaches. Chan et al. propose privacy-sensitive protocols and mechanisms called PACT. Troncoso et al. propose a decentralized system called DP-3T for privacy-preserving contact tracing using Bluetooth data. User ID is encrypted and changed over time, and the user data are stored locally. Similar to DP-3T, some researchers propose PACT, which is a simple decentralized approach using smartphones for contact tracing based on Bluetooth proximity. Avitabile et al. show that the privacy issues in DP-3T are not intrinsic in any BLE-based contact tracing system, and propose a different system named Pronto-C2. Brack et al. use a distributed hash table to build a decentralized messaging
system for contact tracing. All these schemes are independently designed and very similar, apart from some minor variations on implementation and efficiency. Most of the above works focus on detecting face-to-face close contact, and they cannot be extended to the case with virus lifespan. Compared with them, we propose a private WiFi-based approach to detect close contacts with virus lifespan. To the best of our knowledge, our scheme is the first piece of work considering virus lifespan in private contact tracing using WiFi. Moreover, no phone pairing and communication are needed in our proposed scheme.

3 Contact Detection in vContact

As mentioned, we use the signal profile for contact tracing, which is a sequence of signal vectors over time. The formal definitions of signal vector and signal profile are as follows.

Definition 1. (Signal vector) The signal vector $a$ is represented as $(a_1, s_1), (a_2, s_2), ... , (a_i, s_i), ... , (a_n, s_n)$, where $a_i$ is the signal ID (hashed and possibly encrypted AP MAC address) and $s_i$ is its RSSI.

Definition 2. (Signal profile) A user’s signal profile is defined as a sequence of signal vectors over time: $W = \{(A_1, t_1), (A_2, t_2), ... , (A_i, t_i), ... , (A_n, t_n)\}$, where $A_i$ is the signal vector that is scanned at time $t_i$.

In other words, a signal vector represents the signals and RSSIs scanned by a user device at a timestamp, while a signal profile represents the signal vectors which are collected over time. As shown in Figure 2, given a user’s signal profile $W = \{(A_1, t_1), (A_2, t_2), ... , (A_i, t_i), ... , (A_n, t_n)\}$, we would like to detect if the user has contact with the virus at each $t_i$ by comparing the similarity of a signal vector at $t$ and the signal profile of a confirmed case or an infected area.

In this section, we present data processing approaches to construct processed profiles from raw signal profiles for confirmed cases who have installed the app (Section 3.3) and for the confirmed cases who have not, i.e., the infected areas case (Section 3.3). We then propose an efficient signal similarity metric to measure the signal similarity, given a user’s signal vector and a processed vector (Section 3.3). Finally, we summarize by presenting the contact detection algorithm (Section 3.4).

3.1 Signal profile processing for a patient with app

Signals are not sampled continuously but at sporadic and random intervals. We propose a data processing approach to construct continuous processed profiles from raw signal profiles for patients with installed app.

We present a toy example of signal profile processing in Figure 3. A confirmed case’s signal profile $\{(A_1, t_1), (A_2, t_2), (A_3, t_3), (A_4, t_4)\}$ consists of some signal vectors at discrete times. We aim to construct a continuous processed profile from the raw signal profile so that the signal vector at an arbitrary time can be compared. To achieve the goal, we construct processed vectors $A_i$ from any two consecutive signal vectors $A_i$ and $A_{i+1}$, and consider the virus lifespan $\tau_i$. The virus lifespan $\tau_i$ can be various for different time slots.

We define a processed vector as follows.

Definition 3. (Processed vector) A processed vector is defined as $\hat{A} = \{(a_1^s, s_1^{\min}, s_1^{\max}), (a_2^s, s_2^{\min}, s_2^{\max}), ... , (a_n^s, s_n^{\min}, s_n^{\max})\}$, where $(a_i^s, s_i^{\min}, s_i^{\max})$ denotes the RSSI range of a signal vector $a_i$ from $s_i^{\min}$ to $s_i^{\max}$.

The signal strength in a processed vector is represented as a range instead of an exact value in a signal vector. Given two consecutive signal vectors in a signal profile $A_i = \{(a_1^i, s_1^{i+1}), ... , (a_n^i, s_n^{i+1})\}$ at $t_i$ and $A_{i+1} = \{(a_1^{i+1}, s_1^{i+1}), ... , (a_n^{i+1}, s_n^{i+1})\}$ at $t_{i+1}$, the processed vector in the time range from $t_i$ to $t_{i+1}$ is denoted as $\hat{A} = \{(a_1^\gamma, s_1^{\min}, s_1^{\max}) | \gamma = 1, 2, ..., |A_i \cup A_{i+1}|\}$, where $a_1^\gamma \in A_i, a \cup A_{i+1}$, and $(s_1^{\min}, s_1^{\max})$ is denoted as $\text{min}(s_j^s, s_k^s), \text{max}(s_j^s, s_k^s)$, $a_1^\gamma \in A_i, a \cup A_{i+1}$, and $a_1^\gamma \in A_i, a \notin A_{i+1}$.

Here, $\gamma$ is a value indicating a weak signal strength, which is set to be $-100$ in our experiments. Then, we construct a continuous processed profile from a confirmed case’s signal profile considering the virus lifespan. We present a formal definition of a processed profile.

Definition 4. (Processed profile) A processed profile contains a sequence of processed vectors over time: $\hat{W} = \{(A_1, t_1^{\text{start}}, t_1^{\text{end}}), (A_2, t_2^{\text{start}}, t_2^{\text{end}}), ... , (A_m, t_m^{\text{start}}, t_m^{\text{end}})\}$, where $A_i$ is a processed vector for the time slot from $t_i^{\text{start}}$ to $t_i^{\text{end}}$, and $(t_i^{\text{start}}, t_i^{\text{end}})$ indicates the time slot of the virus lifespan.

Given a confirmed case’s signal profile $W = \{(A_1, t_1), (A_2, t_2), ... , (A_i, t_i), ... , (A_n, t_n)\}$, the processed profile is represented as $\hat{W} = \{(A_1, t_1^{\text{start}}, t_1^{\text{end}}), (A_2, t_2^{\text{start}}, t_2^{\text{end}}), ... , (A_n, t_n^{\text{start}}, t_n^{\text{end}})\}$, where $A_i$ is constructed from $A_i$ and $A_{i+1}$ and $\tau_i$ is the virus lifespan for the time slot from $t_i$ to $t_{i+1}$. Noting that $\tau_i$ is given by the health officer, and it can be various for different time slots depending on the frequency of disinfection operation in the venues.

3.2 Signal profile processing for infected areas

We consider the case where the patient has not installed the app. In this case, we need to extract the signals in the infected areas through a survey (signal collection process). We can evaluate if a user has contact with an infected area by measuring the similarity of her/his signal vector and signal vectors of each position in the area. However, collecting WiFi data for every position in the infected area is inefficient. We propose to construct the processed profile for an infected area using some sampled signal data in the area.
An infected area’s signal profile

\[ (A_1, t_1), (A_2, t_2), (A_3, t_3), (A_4, t_4) \]

An infected area’s processed profile

\[ (\hat{A}, t, t' + \tau) \]

Fig. 4. Signal profile processing for an infected area.

### TABLE 1

| Phone       | Average number of signals |
|-------------|----------------------------|
| Honor       | 75.00                      |
| Huawei Mate 30 | 128.12                     |
| OPPO        | 180.16                     |
| Huawei Nova | 92.37                      |
| Xiaomi      | 102.59                     |

Instead of collecting signal data at every position, staff walk around the area with a WiFi-on device such as a phone or a Raspberry Pi. The collected signal profile is some signal vectors over time. To generate a representative processed profile for the area, we would union all signals and their RSSIs in the signal profile. As shown in Figure 4, we merge the signal vectors in the signal profile \( \{(A_1, t_1), (A_2, t_2), (A_3, t_3), (A_4, t_4)\} \) which are collected in the infected area. We also consider the time range \([t, t']\) when a confirmed case staying in the area and the virus lifespan \(\tau\) to construct the processed profile for the infected area.

The processed profile of an area is represented as \(\hat{W} = (\hat{A}, t_{\text{start}}, t_{\text{end}})\), where \(\hat{A}\) is a processed vector and \([t_{\text{start}}, t_{\text{end}}]\) is the time range of the virus lifespan. Given the signal profile collected in the infected area \(W = \{(A_1, t_1), (A_2, t_2), \ldots, (A_n, t_n)\}\) which are collected in the infected area \([t, t']\), and the virus lifespan \(\tau\), the processed profile \(\hat{W} = (\hat{A}, t_{\text{start}}, t_{\text{end}})\) is constructed as follows: \(\hat{A} = \{(a_j, s_{j_{\text{min}}}, s_{j_{\text{max}}})| j = 1, 2, \ldots, |\bigcup^n A_i.a\}\) where \(a_j\) is a scanned signal in \(W\) (i.e., \(a_j \in \bigcup^n A_i.a\)), and \(s_{j_{\text{min}}}\) is the minimum signal strength of \(a_j\) in \(W\) while \(s_{j_{\text{max}}}\) is the maximum signal strength of \(a_j\) in \(W\); the surviving time of the virus in the infected area is from \(t_{\text{start}}\) to \(t_{\text{end}}\).

### 3.3 Signal Similarity Metric

We propose a signal similarity metric to compare the similarity of a signal vector and a processed vector for contact detection. The metric considers the signal IDs overlap ratio and the RSSI difference.

It is intuitive that the closer a user to the location of the virus, the more common signals in the user’s signal vectors and the processed vectors in the processed profile. Thus, we could use the overlap ratio of two vectors’ signal IDs to indicate their proximity. Given a user signal vector \(A\) at time \(t\) and a processed vector \(\hat{A}\), the overlap ratio is calculated as:

\[
O = \frac{|A.a \cap \hat{A}.a|}{\min(|A.a|, |\hat{A}.a|)}, \tag{2}
\]

where \(A.a\) is the Signal IDs in \(A\), \(\hat{A}.a\) is the signal IDs in \(\hat{A}\), and \(| \cdot |\) denotes the number of signal IDs.

The reason of using \(\min(|A.a|, |\hat{A}.a|)\) is to alleviate the impact of device heterogeneity. Different devices have different abilities to scan signals. Two co-located devices may scan different numbers of signals. Table 1 shows the average numbers of signals in a signal vector of various co-located phones in a shopping mall. The average number of signals is heterogeneous for different phones. The difference could be significant for some phones. In this case, using \(|A.a|, |\hat{A}.a|\) or other terms (e.g., \(|A.a \cup \hat{A}.a|\)) as the denominator will introduce more variance.

A signal could cover a large area, it is possible that two vectors with a large proportion of common signals are not in close proximity. Thus, we also consider the RSSI difference to denote the proximity. If a user stays close with the virus, the RSSI difference of the same signal in two vectors should be small. Given a user signal vector \(A = \{(a_1, t_1), (a_2, t_2), \ldots, (a_n, t_n)\}\) and a processed vector \(\hat{A} = \{(a_1, s_{1_{\text{min}}}, s_{1_{\text{max}}}), (a_2, s_{2_{\text{min}}}, s_{2_{\text{max}}}), \ldots, (a_j, s_{j_{\text{min}}}, s_{j_{\text{max}}}), \ldots, (a_m, s_{m_{\text{min}}}, s_{m_{\text{max}}})\}\), for \(a_k \in A.a \cap \hat{A}.a\), its RSSI difference is calculated as

\[
d(a_k) = \left\{ \begin{array}{ll}
s_{j_{\text{min}}} - s_i, & s_i < s_{j_{\text{min}}} \\
n_{j_{\text{max}}} - s_i, & s_i > s_{j_{\text{max}}} \\
0, & \text{otherwise}
\end{array} \right. \tag{3}
\]

The average RSSI difference at a timestamp is defined as

\[
D = \frac{\sum_{a_k \in (A.a \cap \hat{A}.a)} d(a_k)}{|A.a \cap \hat{A}.a|}, \tag{4}
\]

where \(| \cdot |\) denotes the number of signal IDs.

When a user has contact with the virus, the overlap score \(O\) (Equation 2) should be large, while the RSSI difference \(D\) (Equation 4) should be small. Therefore, we define the signal similarity of \(A\) and \(\hat{A}\) as

\[
P(A, \hat{A}) = \frac{O}{D + 1}, \tag{5}
\]

where \(0 \leq P(A, \hat{A}) \leq 1\). A larger \(P(A, \hat{A})\) indicates closer proximity.

### 3.4 Contact detection algorithm

Anyone having contact with the surviving virus may be at risk. Given a user’s signal vector \(A_i\) at \(t_i\), if the timestamp \(t_i\) is within the virus lifespan, and the similarity of \(A_i\) and the processed profile of a confirmed case or an infected area is larger than a threshold, the user will be detected as having contact with the virus at \(t_i\). The algorithm is presented in Algorithm 1.

Given a user’s signal profile \(W_i\), the signal profile of a confirmed case or an infected area \(W_2\), the virus lifespan \(\{\tau_i| i = 1, 2, \ldots, |W_2| - 1\}\) and a proximity threshold \(\alpha\), we first construct the processed profile from \(W_2\) and \(\{\tau_i| i = 1, 2, \ldots, |W_2| - 1\}\) (Line 4). Then for each signal vector \(A_i\) at time \(t_i\) in \(W_i\), if \(t_i\) falls in the time slot of a processed vector in the processed profile, we calculate the signal similarity (using Equation 5) at \(t_i\) (Line 7 ～ 9). If the similarity at \(t_i\) is larger than the given threshold \(\alpha\), the user is identified as having contact with the virus at \(t_i\) (Line 11). The algorithm evaluates the similarity of each signal vector in \(W_i\) and \(W_2\), and returns a list of detection results. The threshold \(\alpha\) depends on how we define the contact...
Algorithm 1: Contact Detection

1 Input: A user’s signal profile \( W_1 \); A confirmed case’s or an infected area’s signal profile \( W_2 \);
   \( \) Virus lifespan \( \{ \tau_i \mid i = 1, 2, ..., |W_2| - 1 \} \);
   A proximity threshold \( \alpha \).
2 Output: results of contact detection at different timestamps.
3 Initialize \( S \) to empty;
4 Construct the processed profile \( \hat{W} \) from \( W_2 \) and \( \{ \tau_i \mid i = 1, 2, ..., |W_2| - 1 \} \);
5 foreach \((A_i, t_i) \in W_1\) do
6   contact = False;
7   foreach \((A_j, t^\text{start}_j, t^\text{end}_j) \in \hat{W}\) do
8     if \( t^\text{start}_j \leq t_i \leq t^\text{end}_j \) then
9        \( s = P(A_i, A_j); \)
10       if \( s \geq \alpha \) then
11          contact = True;
12          break;
13     end
14 end
15 if contact == True then
16   Add \((True, t_i)\) to \( S \);
17 else
18   Add \((False, t_i)\) to \( S \);
19 end
20 return \( S \)

proximity for close contact. We will discuss the relationship between the signal similarity and physical proximity, and the determination for the proximity threshold \( \alpha \) in the following section.

4 Illustrative Experimental Results

We have implemented and packaged vContact as a Software Development Kit (SDK) (for app implementation as discussed in Section 5). In this section, we present illustrative experiment results on the SDK. We first introduce the experiment settings in Section 4.1. Then we study how to set the threshold \( \alpha \) in Section 4.2. We discuss the performance in different sites and for different AP numbers in Sections 4.3 and 4.4 respectively. The studies on heterogeneous devices and in-out detection of an infected area are then presented (Sections 4.5 and 4.6). We finally compare vContact with other state-of-the-art approaches in Section 4.7.

4.1 Settings

To evaluate the performance of our contact detection approach, we collect WiFi data using five mobile phones in three different sites. The brands of phones are different, including Honor, Huawei Nova, Huawei Mate30, Xiaomi, and OPPO. The three experimental sites are an office, a bus station, and a store in a shopping mall. The size of the office is around 10m × 12m. The bus station is an outdoor area, the size of which is around 2m × 15m. The area in the shopping mall for experiments is a large store with a size of 20m × 25m. The total signal numbers are 32 in the office, 109 in the bus station, and 301 in the shopping mall. The average number of signals (i.e., scanned APs) in signal vectors of the office, bus station, and shopping mall are 19.02, 24.0, 46.29, respectively.

To evaluate the detection performance for the case where the signal profiles of confirmed cases are available, we first put the five mobile devices at a location \( \ell_0 \) for 10 minutes to collect the WiFi data in each site. The WiFi signals with RSSIs scanned by a device are collected. Then we put the devices at a location \( \ell_i \) for 10 minutes for data collection, where \( i = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 \), and the distance between \( \ell_0 \) and \( \ell_i \) is \( i \) meters. The data sampling rate is set as 5s per record, so we have around 120 records of data for a device in each distance setting for each site.

To evaluate the detection performance for the case where a confirmed case’s signal profile is unavailable, we walk in the experimental sites to collect WiFi data using a mobile phone to construct the processed profiles for each site. Then we wander around and outside the area with five mobile phones collecting WiFi data for testing. The time when we were in and outside the area is recorded during the experiments.

Given the data \( D \) collected by a user’s device, we use \( D_a \) to denote the data which are collected when the user has contact with the virus (i.e., within the contact proximity with a confirmed case or in an infected area), and use \( D_b \) to denote the data which are detected as having contact with the virus. The \( D_a \) is the ground-truth data while the \( D_b \) is the detection result. Precision, recall, and F1-score are used as metrics to evaluate the contact detection results. The precision is defined as

\[
\text{precision} = \frac{|D_a \cap D_b|}{|D_b|}, \tag{6}
\]

where \(| \cdot |\) represents the data size. Similarly, recall is defined as

\[
\text{recall} = \frac{|D_a \cap D_b|}{|D_a|}. \tag{7}
\]

Based on the definition of precision and recall, F1-score is defined as

\[
F_1 = 2 \cdot \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \tag{8}
\]

4.2 Threshold \( \alpha \)

As introduced in Section 3, the contact detection algorithm relies on a threshold \( \alpha \) to identify contacts. In this section, we discuss the selection of \( \alpha \). Given the contact proximity \( km \), if the distance of a user and the virus is less than \( km \), she/he should be detected as having contact with the virus. Intuitively, \( \alpha \) is relevant to the contact proximity and it should be different for different contact proximity. We use the data collected at \( \ell_0 \) in a site as the data from confirmed cases, and detect contacts for data which are collected at \( \ell_i (i > 0) \) in the same site. When \( k \) meters is set as the contact proximity, \( D_a \) contains the data collected at \( \ell_i \) where \( i \leq k \).

Precision and recall are used as metrics, and the results of \( \alpha \) versus precision and recall for \( k = 1m, k = 2m \) and \( k = 4m \) are presented in Figure 5. As the threshold \( \alpha \) increases, the precision increases while the recall declines. The
The reason is that a larger threshold indicates closer proximity. Thus, increasing the threshold would lead to high precision. However, if the threshold is set too large, some of the data the distance of which is less than \( k \) will not be detected, resulting in a drop in recall.

The threshold can be selected according to the requirements of precision and recall for close contact detection. To balance the precision and recall, we select the intersection points, the precision and recall of which are equal for our following discussion. In Figure 5(a), the precision and recall for \( k = 1 \) are low when \( \alpha \) is set as 0.25, which indicates identifying contact within 1m is difficult. As shown in Figures 5(b), the precision and recall for \( k = 2 \) have a significant improvement when the threshold is around 0.20. The precision and recall in Figure 5(c) for \( k = 4 \) are high (around 70\%) if the threshold is around 0.17. We use the same strategy to select thresholds for other contact proximity.

### 4.3 Site study

We present the performance of contact detection in different sites in this section. We use different distance \( (k = 1 \text{m}, 2 \text{m}, 3 \text{m}, 4 \text{m}, 5 \text{m}) \) to denote the contact proximity, the threshold is set according to the discussion in Section 4.2. Results of precision versus contact proximity are shown in Figure 6(a) while results of recall versus contact proximity are shown in Figure 6(b).

In Figure 6(a) as the contact proximity increases, the precision in the three sites increases, indicating that it is easier to detect contacts within larger proximity. The precision for \( k = 1 \) is low in all sites. The result shows the difficulties in identifying whether the contact happens in 1m because the WiFi signals within a 1-m range are usually similar. However, the precision has significant improvements for larger contact proximity. The precision is high (50% – 70\%) when the proximity is 2m. The precision indoors (office and shopping mall) is better than the precision outdoors because WiFi signals is more stable indoors. The improvement is more significant in the office scenario compared with the shopping mall scenario. The recall shown in Figure 6(b) is similar to the results of precision. Our approach has a good performance on recall when \( k \geq 2 \).

### 4.4 AP number

In this part, we evaluate the impact of AP number on the performance when the contact proximity is set as \( k = 2 \text{m} \). We randomly filtering \( \sigma \% \) signals from the signal vectors for each site, and compare the signal similarity of two devices for contact detection. The filtering rate \( \sigma \% \) is set to be 10\% – 90\%. The precision and recall versus the average signal number are presented in Figures 7.

In Figure 7 as the average signal number increases, the precision increases slightly. The precision is still acceptable when the average signal number is small. Even removing 90\% signals, the precision does not drop significantly for the office and shopping mall sites. The precision outdoors (the bus station) is more stable than others. The recall shown in Figure 7 does not have obvious change as the signal number changes, demonstrating the robustness of our approach.

### 4.5 Heterogeneous devices

Different devices have different abilities to scan WiFi signals. Two co-located devices may scan different signals and RSSIs. We evaluate the performance of different devices. For each device, we compare its data at \( \ell \), with other devices’ data at \( \ell_i \) in the same site. We set the contact proximity as 1m – 5m and set the threshold following the discussion in Section 4.2. Precision and recall are used as metrics.

The precision versus contact proximity for different devices in the office site is presented in Figure 8(a). Given the contact proximity, the precision is different for distinct devices, which is consistent with our discussion. As the contact proximity increases, the precision of all devices increases. The precision of all devices significantly increases.
when $k \geq 2$ m. The recall versus contact proximity for different devices in the office is presented in Figure 8(b).

Similar to the result of precision, the performance of all devices have a large improvement in recall when $k = 2$ m. All devices achieve high recall when $k \geq 2$ m, indicating the good performance of our approach on recall. The results demonstrate that our approach is efficient and it can be applied to phones of different brands. We have similar findings in the experiments on the other two sites. We do not show the results of the other sites due to the page limitation.

4.6 In-out detection of infected areas

Contact detection for confirmed cases without the app is to detect whether a user is in or outside an infected area. We construct processed profiles for the office, bus station, and a store in a shopping mall using the collected WiFi data. Then we compare the similarity between the processed profile of the area and the data collected in and outside the area. If the similarity is larger than the threshold $\alpha$, the data is identified as being collected in the area and having contact with the virus. $\alpha$ is set as 0.2 in the experiment. Precision and recall are used as the metrics for evaluation. The results are shown in Figure 9. The detection in all the sites achieves good performances. The precision and recall are high for the three sites, illustrating vContact is very efficient for in-out detection of infected areas.

4.7 Comparison with other approaches

We have compared vContact with some other state-of-the-art approaches, which are introduced as follows,

- **Bluetooth**: It is widely used for digital contact tracing, such as schemes [29] [7] [3]. To collect Bluetooth data, two mobile devices are put at a distance of $k$ meters for 10 minutes in the three experimental sites, where $k$ is set to be \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}. We use one device as the broadcaster, and another as scanner. The scanner can scan the Bluetooth signal from the broadcaster, and the RSSI is recorded over time. For each contact proximity $k$ meters, a threshold is selected for contact detection. If a received signal strength is larger than the threshold, they are detected as having contact.

- **Jaccard similarity**: It is used to evaluate the similarity of two sets, and it is defined as the size of the intersection divided by the size of the union of two sets. If the Jaccard similarity of two signal vector is larger than a threshold, they are identified as within the contact proximity.

- **Average L-1 distance (ALD)**: It is the average L-1 norm of signal strength difference. If the ALD of two signal vectors is less than a threshold, they are identified as within the contact proximity.

- **Euclidean distance (AED)**: It is the average Euclidean distance of signal strength difference. If the AED of two signal vectors is less than a threshold, they are identified as within the contact proximity.

For the baseline approaches ALD and AED, given two signal vectors $A$ and $B$, if a signal is scanned in $A$ but not in $B$, the signal strength is set as -100 in $B$ for calculation, and vice versa. As the baseline approaches rely on a selected threshold to detect contact, for a given contact proximity,
we use the same strategy to select thresholds as discussed in Section 4.2. Precision, recall, and F1-score are used as metrics for performance comparison.

The results of precision, recall and F1-score versus proximity on the three datasets are presented in Figures 10 (the office), 11 (the bus station), and 12 (the shopping mall). In Figure 10, the precision, recall, and F-1 score of different approaches increase as the contact proximity increases. vContact always outperforms other baseline approaches on the metrics of precision and F-1 score. vContact has higher recall than others when contact proximity is less than 5 m and has similar performance to Bluetooth when the contact proximity is 5 m. The curves of precision, recall and F1-score on the other datasets have a similar trend to that on the office dataset. As shown in Figure 11(a), the precision of Bluetooth is slightly higher than vContact on the bus station dataset. But vContact has better performance than Bluetooth and other approaches with respect to recall and F1-score. As for the performance on the shopping mall dataset, vContact has higher precision than Bluetooth when contact proximity is 1 m and 2 m, but has a significant improvement on precision when contact proximity is 3 m and 4 m. In Figure 12(b), vContact has similar precision to Bluetooth and ALD. vContact always outperforms other approaches which use WiFi data for detection. Overall, vContact has a higher F-1 score than other approaches in all datasets, indicating it is more efficient for contact detection. We can also learn from the figures that vContact and other approaches have better performance in the indoor scenario, and the improvement of vContact is more significant compared with the outdoor...
5 APP IMPLEMENTATION AS A CASE STUDY

With the vContact SDK, we have implemented an Android app which notifies its user exposure duration of the virus. In this section, we first report its implementation details and user interface in Section 5.1, followed by some measurement results of the app to demonstrate and validate its design in Section 5.2.

5.1 Implementation for exposure notification

We develop an app using our approach for exposure notification. Some screens of the app are shown in Figure 13. As shown in Figure 13(a), once a user turns on the button of “Exposure data collection”, the app will scan nearby WiFi and store the data locally every 1 minute. The signal IDs (i.e., the AP MAC addresses) are encrypted when the data are stored. If one is confirmed infection, she/he could upload her/his signal profile to the server (Figure 13(b)), so that others could download the data for matching. If a user has close contact with a confirmed case, she/he will receive a notification, showing when the close contact happened and how long the contact duration is (Figure 13(c)). In the app, data are downloaded and matched automatically every day. For the purpose of testing, we also have a testing mode as shown in Figure 13(d) by which we can download the data, and trigger the detection manually during the testing.

5.2 Testing and validation

We set the contact proximity as 2m for testing. The app collects WiFi data every 1 minute. Hence, the detection approach introduced in Section 5.1 will report a detection result (i.e., true or false) for the data at each minute. In our testing, if a user stays with the virus within 2m for more than 5 minutes in a 10-minutes sliding time window, she/he will receive a possible exposure notification. Noting that the contact duration and the length of the sliding time window are parameters for the app, which can be changed according to the advice of the health officer.

We test the app in an office using five phones of different brands. The procedures are as follows. One of the phones is selected as the confirmed case, other phones are put at a location which is 2m away from the confirmed case. The button “Exposure data collection” is turned on for 15 minutes. Then, the confirmed case uploads its signal profile, and the other phones download the signal profile for matching. After that, we put other phones at a location which is 4m away from the confirmed case and repeat the testing. Each phone is selected as the confirmed case in turn. The ideal result is that a phone only receives a notification when it is 2m away from the confirmed case but no notification for 4m. The testing results are presented in Table 2 and 3. A ✓ refers that a phone receives a notification while × means it does not receive a notification.

Table 2 shows the results of exposure notification for 2m. It illustrates the good performance of our app for exposure notification. The performance of the Honor phone is not as good as other phones, indicating the different ability of phones to scan WiFi signals.

We show the results of exposure notification for 4m in Table 3. Compared with the results in Table 2, more phones are detected as having non-close contact, which is consistent with our expectation. Performance is distinct for different phones, but the overall performance is good.

6 CONCLUSION

In this work, we consider automatic digital contact tracing given that the virus has a lifespan. Leveraging upon pervasive WiFi signals, we propose a private WiFi-based approach termed vContact to detect close contacts within the virus lifespan. Our approach captures both the case of people simultaneously co-located with an infected person and the case of those coming into an area previously visited by an infected person within the virus lifetime. To the best of our knowledge, this is the first piece of work considering the virus lifespan in private contact tracing using WiFi. We propose data processing approaches and a signal similarity metric for close contact detection. We conduct extensive experiments for evaluation. The experimental results demonstrate that our approach is efficient, stable, and deployable. Our approach achieves high precision and recall (50% – 90%) when the contact proximity is 2m) for different experimental sites, and it is robust to the impact of different signal numbers and devices with different brands. We have implemented an Android app based on vContact, and demonstrated the validity of our design.

REFERENCES

[1] W. Aylin and G. Shayanne, “One chart shows how long the coronavirus lives on surfaces like cardboard, plastic, wood, and steel,” https://www.businessinsider.com/coronavirus-lifespan-on-surfaces-graphic-2020-3, 2020.
[2] K. Gray, “Covid-19: How long does the coronavirus last on surfaces?” https://www.bbc.com/future/article/20200317-covid-19-how-long-does-the-coronavirus-last-on-surfaces”, 2020.
[3] C. Pyrko, M. “Privacy in FHIR,” Hibaux, M. Salathé, J. Larus, E. Bugnion, W. Lueks, T. Stadler, A. Pyrgelis, D. Antonioli et al., “Decentralized privacy-preserving proximity tracing,” Github DP3T documents, vol. 12, 2020.
[4] A. Berke, M. Bakker, P. Vepakomma, R. Raskar, K. Larson, and A. Pentland, “Assessing disease exposure risk with location histories and protecting privacy: A cryptographic approach in response to a global pandemic,” arXiv preprint arXiv:2003.14412, 2020.
[5] A. Wesołowski, C. O. Buckee, L. Bengtsson, E. Wetter, X. Lu, and A. J. Tatem, “Commentary: containing the ebola outbreak—the potential and challenge of mobile network data,” PLoS currents, vol. 6, 2014.
[6] J. Bell, D. Butler, C. Hicks, and J. Crowcroft, “Tracesecure: Towards privacy preserving contact tracing,” arXiv preprint arXiv:2004.04059, 2020.
[7] J. Chan, S. Gollakota, E. Horvitz, J. Jaeger, S. Kakade, T. Kohno, J. Langford, J. Larson, S. Singamanilla, J. Sunshine et al., “Pact: Privacy sensitive protocols and mechanisms for mobile contact tracing,” arXiv preprint arXiv:2004.03544, 2020.
[8] Google and Apple, “Privacy-preserving contact tracing,” https://www.apple.com/covid19/contacttracing, 2020.
[9] Singapore, “TraceTogether,” https://www.tracetogether.gov.sg, 2020.
[10] Australian, “CovidSafe,” https://www.health.gov.au/resources/apps-and-tools/covidsafe-app, 2020.
[11] Turkey, “Coronawarn,” https://www.ua.com.tr/en/latest-on-coronavirus-outbreak/turkey-to-use-contact-tracing-app-to-detect-coronavirus/1804425, 2020.
[12] India, “Aarogya setu mobile app,” https://www.mygov.in/aarogya-setu-app/, 2020.
[13] Argentina, “Cotrack,” https://www.cotecn.com/en/services/government/cargo-tracking-solution-cotrack, 2020.
(a) Exposure data collection.  
(b) Share positive test result.  
(c) Possible exposure notification.  
(d) Testing mode.

Fig. 13. The app to notify user viral exposure duration.

| TABLE 2 | Result of exposure notification for a separation of 2m. |
|---------|-----------------------------------------------------|
|          | Confirmed Case | User | Honor | Mate 30 | OPPO | Huawei Nova | Vivo |
|          | Honor          | Mates30 | OPPO | Huawei Nova | Vivo |
| Honor    | x              | ✓       | ✓    | ✓        | ✓    |
| Huawei Nova | x              | ✓       | ✓    | ✓        | ✓    |
| Xiaomi   | x              | ✓       | ✓    | ✓        | ✓    |

| TABLE 3 | Result of exposure notification for a separation of 4m. |
|---------|-----------------------------------------------------|
|          | Confirmed Case | User | Honor | Mate 30 | OPPO | Huawei Nova | Vivo |
|          | Honor          | Mates30 | OPPO | Huawei Nova | Vivo |
| Honor    | x              | x       | ✓    | ✓        | ✓    |
| Huawei Nova | x              | x       | ✓    | ✓        | ✓    |
| Xiaomi   | x              | ✓       | ✓    | ✓        | ✓    |

[14] A. Sydney, Von, B.-M. Isaiah, B. Daniel, C. Jesse, F. Rhys, H. Mike, O. N. Mark, Ingle, V. Nguyen, J. Petrie, J. Schwaber, Z. Szabo, A. Veeragahnta, M. Voloshin, T. White, and H. Xue, “Slowing the spread of infectious diseases using crowdourced data,” https://www.covid-watch.org/, 2020.

[15] C. T. Nguyen, Y. M. Saputra, N. Van Huynh, N.-T. Nguyen, T. V. Khoa, B. M. Tian, D. N. Nguyen, D. T. Hoang, T. X. Vu, E. Dutkiewicz et al., “Enabling and emerging technologies for social distancing: A comprehensive survey,” arXiv preprint arXiv:2005.02816, 2020.

[16] F. Qi and F. Du, “Tracking and visualization of space-time activities for a micro-scale flu transmission study,” International Journal of Health Geographics, vol. 12, no. 1, pp. 1–16, 2013.

[17] J. K. Fitzsimons, A. Mantri, R. Pisarczyk, T. Rainforth, and Z. Zhao, “A note on blind contact tracing at scale with applications to the covid-19 pandemic,” arXiv preprint arXiv:2004.05116, 2020.

[18] L. C. Klopfenstein, S. Delpiori, C. M. Di Francesco, R. Maldini, B. D. Paolini, and A. Bogliolo, “Digital ariadne: Citizen empowerment for epidemic control,” arXiv preprint arXiv:2004.07717, 2020.

[19] L. Reichert, S. Brack, and B. Scheuermann, “Privacy-preserving contact tracing of covid-19 patients,” Sourced from, 2020.

[20] G. Li, C.-J. Chen, W.-C. Peng, and C.-W. Yi, “Estimating crowd flow and crowd density from cellular data for mass rapid transit,” in Proceedings of the 6th International Workshop on Urban Computing, Halifax, NS, Canada, 2017, pp. 18–30.

[21] G. Li, C.-J. Chen, S.-Y. Huang, A.-J. Chou, X. Gou, W.-C. Peng, and C.-W. Yi, “Public transportation mode detection from cellular data,” in Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, 2017, pp. 2499–2502.

[22] L. Isella, M. Romano, A. Barrat, C. Cattuto, V. Colizza, W. Van den Broeck, F. Gesualdo, E. Pandolfi, L. Ravà, C. Rizzato et al., “Close encounters in a pediatric ward: measuring face-to-face proximity and mixing patterns with wearable sensors,” PloS one, vol. 6, no. 2, p. e17144, 2011.

[23] M. Salathé, M. Kazandjieva, J. W. Lee, P. Levis, M. W. Feldman, and J. H. Jones, “A high-resolution human contact network for infectious disease transmission,” Proceedings of the National Academy of Sciences, vol. 107, no. 51, pp. 22 020–22 025, 2010.

[24] S. Jeong, S. Suk, and H. Kim, “A smartphone magnetometer-based diagnostic test for automatic contact tracing in infectious disease epidemics,” IEEE Access, vol. 7, pp. 20 734–20 747, 2019.

[25] C. Günther, M. Günther, and D. Günther, “Tracing contacts to
control the covid-19 pandemic,” *arXiv preprint arXiv:2004.00517*, 2020.

[26] A. Hekmati, G. Ramachandran, and B. Krishnamachari, “Contain: privacy-oriented contact tracing protocols for epidemics,” *arXiv preprint arXiv:2004.05251*, 2020.

[27] F. Sattler, J. Ma, P. Wagner, D. Neumann, M. Wenzel, R. Schäfer, W. Samek, K.-R. Müller, and T. Wiegand, “Risk estimation of sars-cov-2 transmission from bluetooth low energy measurements,” *arXiv preprint arXiv:2004.11841*, 2020.

[28] Y. Xia and G. Lee, “How to return to normalcy: Fast and comprehensive contact tracing of covid-19 through proximity sensing using mobile devices,” *arXiv preprint arXiv:2004.12576*, 2020.

[29] R. L. Rivest, J. Callas, R. Canetti, K. Esvelt, D. K. Gillmor, Y. T. Kalai, A. Lysyanskaya, A. Norige, R. Raskar, A. Shamir, E. Shen, I. Soibelman, M. Specter, V. Teague, A. Trachtenberg, M. Varia, M. Viera, D. Weitzner, J. Wilkinson, and M. Zissman, “Pact: Private automated contact tracing,” *https://pact.mit.edu/*, 2020.

[30] G. Avitabile, V. Botta, V. Iovino, and I. 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. *https://eprint.iacr.org*, Tech. Rep., 2020.

[31] S. Brack, L. Reichert, and B. Scheuermann, “Decentralized contact tracing using a dht and blind signatures,” *Last accessed: 01st May, 2020.*