On Passive Privacy-Preserving Exposure Notification using Hash Collisions

Phillip Smith, Shamik Sarkar, Neal Patwari, and Sneha Kasera

***Abstract*—Even as the COVID-19 pandemic drove advances in contact tracing and exposure notification systems, user privacy challenges continue to plague otherwise promising approaches to contain contagions. We propose a novel, scalable approach to address privacy in contact tracing that improves utility. We apply passive WiFi scan data using two metrics suitable for estimating contact between users. We support this with real world experimental data captured across a range of environments relevant to contact tracing. To preserve privacy, we leverage properties of truncated cryptographic hashes in an adaptation unique to contact tracing. This hash collision filter allows users to share information about potential contacts with a central server without revealing sensitive information. Using an aggressive threat model, including adversarial users and a malicious server, we share how this technique can improve utility while still providing strong security protections compared to other approaches using, for example, only Bluetooth (BT) or global navigation satellite systems (GNSS). Finally, we discuss a capability of this approach that allows notification for asynchronous co-location from past contacts.**

***Index Terms*—Exposure Notification, Contact-Tracing, Hash Collision**

I. Introduction

Exposure notification plays an important role in monitoring and containing outbreaks. Viral outbreaks including H1, H2, H3 influenza subtypes, and most recently the SARS-CoV-2 coronavirus, have spurred research and global efforts to mitigate its effects [2], [21]. This led to improvements in more technologically automated solutions—sometimes referred to as *digital* contact tracing—including ambient wireless signals, Bluetooth (BT), blockchains and ultrasonic beacons [2], [20], [27], [35], [39]. Research continues to resurface important questions centered around user data privacy; existing and past approaches to contact tracing often violate individual privacy, resulting in diminished adoption rate and consequently, diminished utility [42], [46]. The ubiquitous use of personal mobile devices has enabled and expanded possibilities for contact tracing while further compounding users' privacy concerns. Even after some of the more recent, state-of-the-art approaches proposed by Apple and Google, DP-3T or PACT, challenges remain [12], [34], [40]. BT-based approaches introduce an attack surface by opening an interactive protocol to anyone within range and may need to be adapted to provide useful context such as location and time. The original BT Privacy LE MAC address rotation scheme used by these has exhibited susceptibilities to attack [4]. Despite these and other security measures, recent work has demonstrated the feasibility of tracking using imperfections of BT's physical layer RF fingerprints [14], [31]. These and other security and privacy challenges can discourage widespread use and effectiveness of these newer approaches. Finally, one important capability is the ability to detect lingering contagions from a recent spreader. If a coughing carrier in a hospital waiting room exits and others enter the same room a minute later, they will likely be exposed to contagions which would remain undetected by the prominent BT-based contact tracing approaches, for example. We emphasize this because emitted aerosols and surface contaminants, SARS-CoV-2 contagions in particular, may persist in many different environments for extended periods of time [17], [22], [25], [30], [33], [43]. Without this capability, exposure notification systems under these conditions would produce false negative results. We address these challenges and improve upon passive WiFi scanning approaches to allow asynchronous co-location contact tracing while protecting the privacy of users.

Neal Patwari is affiliated with the Electrical Engineering and Computer Science Departments at Washington University in St. Louis, St. Louis, MO 63130 USA (email: npatwari@wustl.edu).

The other authors are affiliated with the School of Computing, University of Utah, Salt Lake City, UT 84112 USA (emails: phillip.smith@utah.edu, shamik.sarkar@utah.edu, kasera@cs.utah.edu). The corresponding author is Phillip Smith.

A. Overview

Prior to the technique and implementation details we define some key terms and briefly describe a contact tracing scheme using our approach—for the purposes of this work, the terms contact tracing and exposure notification are used interchangeably unless a distinction is warranted. A deployed system consists of different actors exchanging information while observing the principle of least privilege. These actors include the ***user***, ***server***, ***care provider*** and ***key authority***. Users record WiFi power measurements throughout the day and store these, along with hashed access point (AP) service set identifiers (SSID, BSSID) in ***groups*** partitioned by time intervals. The individual hashed identifiers that comprise these groups are called ***context keys***. We use these groups of context keys to determine the likelihood of contact between users and ***carriers***, or users who have tested positive. This is done by associating sets of context keys with relative locations and identifying those common between a regular user and carriers' keys shared with the server. Carriers' past key groups are uploaded to the server by their care provider after obtaining positive test results. In order to determine whether a positive contact has occurred, a user queries the server by sharing some set of her past key groups and a unique ***authentication token*** obtained from the key authority. If the token is deemed

valid by the server authenticating with the key authority, the server accepts the user's query and responds by returning the received signal strength (RSS) values associated with any of the matching keys as well as those keys' respective ***group IDs***. This information is then used to estimate the likelihood of contact using our methods described in Section III.

B. AP Matching and Euclidean Distance

For contact determinations we compare two metrics: AP match and Euclidean distance, based on RSS. The first approach reports a metric based on the observed APs common between two users over a defined scan interval. The closer two individuals are to each other, then the higher the match metric. An empirical threshold is established to determine what match value should be associated with a contact. We include modifiers to improve robustness across environments of varying distributions of APs.

The second measurement approach associates the Euclidean distance between the RSS vectors to the physical distance between two devices (i.e., individuals). A cost factor associated with the vector length is used to increase reliability and consistency across measurements. We demonstrate the effectiveness of these approaches in selected measurement scenarios. These methods are able to distinguish distances less than three meters between users. Graphical, numerical representations of these results are presented in their respective sections.

C. Preserving Privacy

After establishing the efficacy of the proximity detection technique, we address the challenges of security and privacy. The combination of these techniques combined with present and past proximity detection become the primary advantage of our approach over others. Proximity context is inferred from WiFi access point beacons in much the same way that WiFi fingerprinting techniques can be used for localization, but with one important distinction: the ephemeral nature of radiofrequency (RF) propagation channels which normally presents a challenge to static RF maps instead enhances privacy through entropy. Additionally, through application of our hash collision filter, segregation of sensitive information and limitations of server access, we offer an efficient system, secure against many attacks by users and against server compromise. The hash-based abstraction obfuscates the identifying information (the BSSID and SSID) unique to a physical location while still allowing match and distance calculations. This AP information, is combined with a timestamp and hashed together to associate a geospatial point for proximity detection. Distance between two matching contacts' hashes can be inferred once the server provides quantized RSSI information to a user for each successful hash match submitted in a query. We analyze an aggressive threat model to identify possible attacks and avenues for information leakage.

D. Practical Implementation

After providing the theoretical foundation and analysis for our contact tracing approach, we present a system evaluation based on an implementation with synthetic user data. This demonstrates the scalability and feasibility of this approach for large-scale deployments. We conclude by discussing implementation details, focusing on trade-offs, hardware constraints and outlying cases of environmental limitations. These include potential pitfalls of mismatched, asynchronous scan intervals among different parties which would lead to false negatives, hardware or firmware limitations throttling WiFi scan rates and effects on power consumption. RF environmental limitations may occur in locations of sparse WiFi APs or unusual distribution among participants and surrounding APs.

We evaluate our WiFi proximity detection technique across a mix of 12 sensors with hundreds of hours of samples WiFi AP beacon measurements collected across different environments. Our approach reliably detects close contact between users carrying mobile devices, demonstrating the ability to distinguish close contact (20cm) from *socially distant* (3m) proximity in normal indoor environments. We prove the feasibility of our system by testing a sample implementation against more than 300M synthetic users. To summarize, the main contributions of our work consist of:

- An effective, privacy-preserving, passive WiFi-scanning technique adapted to contact tracing using our *hash collision filter*
- An improved method to detect possible exposure to lingering contagions in an environment (i.e.: asynchronous co-location)
- An implementation, evaluation, and threat analysis of our approach using collected real-world data

II. Related Work

Recent research includes many proposals for digital contact tracing. We point to some the more influential and relevant papers among these. Our WiFi-scanning solution contrasts with many similar, more mainstream BT-enabled contact tracing solutions [12], [20], [34], [40], [44], often referred to as "upload-what-you-heard" or "upload-what-you-sent" schemes. These generally share rotating, user-exchanged pseudo-random keys with a server to improve privacy for exposure notification. Variations of these approaches include the addition of location context through GPS [37] and privacy improvements for the uploader [7]. WiFi-based solutions offer different approaches, such as [45], which uses AP association logs to infer user proximity for exposure notification. Two cryptographic-based approaches to WiFi co-location for contact tracing are Enact [35] and Epic [2]. Enact more closely resembles our work through the use of hashed identifiers, but requires AP firmware modification; Epic employs an interactive, distributed protocol using multiparty homomorphic encryption. The MIT Safe Paths project's Privatekit [36] proposes a variation on a hash-based privacy-preserving WiFi co-location scheme. Many countries have offered, recommended or even mandated a contact tracing program based on these techniques. An excellent summary and comparison of these approaches can be found in [24]. Nearly all of these systems require some form of direct information exchange between users, which we contend is a security risk; our approach has no such requirement. A more

recently published, complete system, called vContact, parallels our approach in many ways and improves on the aforementioned WiFi-based approaches [24]. vContact presents a strong argument for a WiFi-based solution where virus lifespan can affect indirect environmental exposure beyond the oft-assumed face-to-face modality. Our approach primarily differs from the vContact system by focusing on the security and privacy concerns associated with collection of such data. vContact, in contrast, focuses more on the proximity analysis and only briefly addresses security concerns. This work complements the other related works proposing WiFi co-location for other purposes [10], [28]. [11] compares various WiFi co-location "matching" metrics.

Our privacy approach relates to existing work on probabilistic data structures. While these were initially proposed to improve efficiency in database lookups [5], later additions include support for approximate evaluation queries [8]. Later work adopted these techniques for differential privacy controls [13]. [26] employs a truncated hash structure to prevent system exploitation through a dictionary attack. [12] uses a Cuckoo filter in exposure notification to improve performance while strengthening privacy. Our approach favors the privacy advantages of approximate evaluation queries over the traditional goal of space-optimization.

We present a threat analysis for attacks directly relating to the privacy of our approach, while other works [7], [12], [48] provide excellent analyses on other vulnerabilities common to many digital contact tracing techniques.

III. An Unconventional Approach to Exposure Notification

In this section we explain our approach by first describing the concepts and techniques we developed to enable our digital contact tracing system, followed by a more detailed description of our implementation.

A. WiFi Proximity Inference

In contrast to manual or more recent widespread BT-enabled approaches to contact tracing, we leverage the pervasive deployment of WiFi APs to accomplish the same objective. Applying the measurements from these observations improves indoor localization where more common global navigation satellite systems (GNSS)-based options alone are ineffective due to poor signal penetration. Advances in localization and RF fingerprinting using RF signal propagation and beacon signals from WiFi APs contribute to a growing body of literature [3], [9], [18], [23], [38]. Probabilistic approaches, using advanced pre-processing techniques and an offline machine learning phase, have allowed researchers to achieve sub-meter localization accuracies [19], [32], including a recent work focused on contact tracing to include indirect environmental exposures [49]. In order to achieve these results, extensive computation and *a priori* knowledge must inform the model. In practical applications, this limits the utility of WiFi fingerprinting to the locations where an appropriate RF map was obtained. Furthermore, as the RF environment changes over time, due to weather, obstructions or relocation of APs, the model must be updated to maintain accuracy [19], [32]. These same factors which negatively impact WiFi fingerprinting either become irrelevant when applied to contact tracing or instead serve to enhance privacy by increasing entropy. This is made possible by the temporal constraint requirement of contact tracing: *the environmental conditions associated with the relative locations of two individuals is only relevant for the time period they are in close proximity to one another*. Outside of these conditions, the state of the RF environment becomes immaterial. Fundamentally, contact tracing does not require a physical coordinate to be known, and in fact a coordinate is a privacy concern. We employ the use of measurements of WiFi, BSSIDs and SSIDs for just such a purpose, which we now describe in two complementary metrics.

B. AP Match Factor

As the distance between the two individuals increases, the probability that they will continue to observe the same shared set of APs also diminishes until no common APs are observed. This concept forms the basis of what we will denote as AP match factors, defined below.

Let p and s denote two users testing for contact with one another. Let S_p represent the set of all APs observed by user p over some time interval t_p. Likewise, S_s represents the unique set of all APs observed by user s over some time interval t_s. We refer to any set of observations over the defined interval t_s as simply a *group*.

Assuming $t_s = t_p$, i.e., overlapping synchronized observation intervals, we define the match factor μ using the Jaccard Index as:

$$\mu = \frac{|S_p \cap S_s|}{|S_p \cup S_s|}. \tag{1}$$

We also define the modified match factor μ_m:

$$\mu_m = \mu \left(1 - \frac{1}{max(|S_p|, |S_s|) + 1}\right). \tag{2}$$

The introduction of the modified match factor provides a weight, penalizing the standard match factor when very few APs are present. Without this, in the case when only one AP is visible by two observers, for example, the match factor may indicate a perfect match, even when there could be a large distance between them. This possible source of quantization error diminishes as more, dispersed APs are present.

In order to ensure accuracy of the match factor, the start of WiFi scan time intervals must be synchronized among users. We accomplish this by recording and adjusting according to the offset between the phone's internal clock and its GNSS-derived clock signal. Additionally, the duration of scan interval must be long enough to capture at least one AP beacon; AP beacon intervals are typically on the order of 100 ms, so many would be captured within, for example, a 1-minute scan interval.

C. RSS Distance Factor

Our second metric augments the matching approach by introducing a distance function based on measured received power values, or quantized received power, known as received

signal strength (RSS), from surrounding APs. This technique resembles those traditionally employed for WiFi fingerprinting and localization. We use the Euclidean distance measurement for this metric. We also introduce a scale factor to compensate for variation of vector dimensions between different sets of observations. Our distance functions, Euclidean (D_e) and modified Euclidean (D_m), are then defined, respectively, as follows:

$$D_e = \Big[\sum_{i \in S_p \cap S_s} (q_i - p_i)^2 \Big]^{1/2} \tag{3}$$

$$D_m = \Big[\frac{1}{|S_p \cap S_s|} \sum_{i \in S_p \cap S_s} (q_i - p_i)^2 \Big]^{1/2}, \tag{4}$$

where p_i and q_i are the received powers of signals from AP i at the two receivers. Notably, as the accuracy of the match factor depends on the number of APs, so will the received power distance factor. Although we do not introduce a cost or penalty for this metric when fewer APs are present, this information can be inferred from the modified match factor associated with two locations. In general, our requirement is similar to that for localization for trilateration. Three APs would be the minimum requirement to disambiguously co-locate two contacts. Even in the case of three APs, if they themselves were not spatially separated, they could not reasonably be used to infer proximity.

As the physical distance between two individual observers decreases, on average, so do these proxy distance metrics due to RF propagation. Consequently, these provide useful measures to estimate the proximity or likelihood of contact for purposes of contact tracing. Section IV includes sample results we captured across different scenarios.

Sharing these measurements could allow an adversary to reconstruct a location trace and uniquely identify a user. We introduce techniques to prevent this in the next section, with a more detailed evaluation in Section IV-C.

D. The Hash Collision Filter

The foundation of our privacy technique lies in our data structure we call a *hash collision filter*, which allows a user to query a key-value database without exposing the actual data. In our application, the keys encode important context information while the response confirms membership and association with other elements. This design can prevent information leakage of potentially sensitive information by concealing the underlying information while providing ambiguous results to naïve attackers. This filter is related to hash tables and classes of approximate member query filters. Our filter uses well-known properties of a truncated hash to ambiguously encode context information in the keys while providing a key-value lookup for existence and group association in the response. It occupies a memory footprint that scales linearly with the size of the contents, while yielding no false negatives and providing a user-definable, constant false positive rate. The problem our hash collision filter aims to solve can be summarized as: *Given a set of hashed data elements (context keys) associated with one particular group G_0, does there exist within the database, any such group G_i which shares more than n hashed elements in common?*

Although our hash collision filter can be used to store other types of associated data elements, we tune the parameters to enhance its utility for sharing WiFi-based proximity context for contact tracing. In this application, elements, or *context keys*, will consist of hashed SSID/BSSID observations, a pseudo-random value and timestamps, grouped according to time interval and hence indirectly associated with a location. Users query the server by submitting their context keys. If received responses indicate that more than n context keys are found belonging to the same group in the server, then contact is presumed. This exchange allows a user to employ the AP matching technique, defined in Section III-B, to become informed of possible contacts. We employ a weak hash function to obfuscate the data underlying the individual context keys within a key group. Typically, a weak hash function is undesirable because it is vulnerable to various forms of collision attacks; the opposite is true in our case. Consequently, an adversary querying the server in an attempt to discover a user's original observations may find many unrelated matches (collisions) across different groups. Additionally, the attacker will find it computationally infeasible to identify combinations of (n or more) hashes within the same group (which would indicate a true match). Such information would be necessary, for example, to estimate a user's location history. A threshold of n common context keys among groups can be chosen based on the cost of a false positive rate versus the level of desired privacy, the expected number of APs available in an environment and, to some extent, processing overhead. We summarize the desired characteristics of a hash collision filter as follows:

Given a particular group, consisting of context keys,

1) The probability of at least one duplicate context key appearing in another group within the database must be high
2) The probability of more than n particular context keys belonging to one group also occurring within any other group in the database must be very low
3) The size of the input space, group size and hash function output length must be chosen such that a dictionary attack is not feasible

The first two conditions can be met by choosing the optimal combination of hash length and group size, as a function of the number of participants. For our implementation we use the SHA1 hash with output truncated to the desired hash output length. The basic requirement is that *the chosen hash function should produce independent, uniformly distributed values across the output space.* The mathematical derivation for these is given below, while the third is covered in Section IV-C.

Let s represent the size of the hash output space and g the group size, or number of context keys belonging to a group with $p = (s - g)/g$.

The probabilities of zero, one and more than one collision occurring among a particular group of g context keys against any other group in the database are given, respectively, by:

$$\Pr(C=0) = \binom{g}{0}\left(\frac{s-g}{s}\right)^{g} \tag{5}$$

$$\Pr(C=1) = \binom{g}{1}\left(\frac{s-g}{s}\right)^{g-1}\left(\frac{g}{s}\right) \tag{6}$$

$$\Pr(C>1) = 1-\left[\binom{g}{0}\left(\frac{s-g}{s}\right)^{g} + \binom{g}{1}\left(\frac{s-g}{s}\right)^{g-1}\left(\frac{g}{s}\right)\right], \tag{7}$$

where C is the number of collisions. If we define a random variable $X_{C=c}$ to be the number of groups with c collisions and b as the number of groups in a database, we can compute the expected number of groups with zero, one or more than one collision across all the groups (of g context keys) within a database as follows:

$$\mathbb{E}(X_{C=c}) = b\Pr(C=c) \tag{8}$$

$$\mathbb{E}(X_{C>c}) = b\Pr(C>c) \tag{9}$$

We choose $c = 1$ as the threshold for the number of collisions between groups. The parameters must be fixed for the anticipated size of the user base. Depending on the chosen parameters, hardware resources could become cost prohibitive, or systems may need to be regionally distributed rather than centralized for large populations (e.g. billions of users). For an example deployment across a population of 100M active users, we may need to store billions of context key groups groups (see Table II) if we choose a 37 bit hash length for context keys. This is derived from the equations above to yield an expected 4.37 duplicate context keys, or collisions, occurring for any one context key query across a fully populated database. The expected number of occurrences for the collision of any two arbitrary context keys in the same group across a fully populated database would be $3.0x10^{-11}$. Given our threshold, this number would also be interpreted as the false positive rate for the match factor. These parameters, group size and hash length, can be adapted to improve privacy at the cost of an increase in the false positive rate. Similar to others, the strength of this privacy mechanism depends not only on the parameters but also the foreknowledge of an adversary. We analyze its effectiveness in several threat scenarios in Section IV-C.

E. Threat Model

While many potential attack surfaces exist in a large, publicly accessible, distributed network system, we constrain our threat model as follows:

- Data in transit is protected with application-layer security and an adversary has no means to break this encryption or exploit these transmissions
- The key authority services are not compromised (i.e., remain trusted)
- Patient information at the health provider remains secure [1]
- An adversary is familiar with the system protocol and has access to publicly available information, including remote server access as an ordinary user

This primarily limits the attack surface to the protocol and server interactions. We frame the primary threat as a malicious user seeking to obtain other users' location information, identity and infection status. We also consider similar attacks from a compromised server. Other attacks, such as disruption-of-service or side-channels, are beyond the scope of this work.

[1]The health provider does not retain or submit location data but approves submission by a user

F. System Implementation

We use our implementation to demonstrate the scalability of our approach. We describe the assumptions and design decisions for one possible architecture. Our system comprises four actors: users, server(s), a care provider and the key authority. These are illustrated in Fig 1. Throughout the day, users record WiFi power measurements and hashed AP identifiers for proximity location context. The key authority sets the length of the hash function, releases daily crytographic keys and authenticates users for server queries. The key authority protects against certain classes of attacks as a complement to the hash collision filter. A new user must request access from the key authority through a trusted exchange which uniquely identifies that user. This can be similar to processes for know-your-customer transactions or an in-person exchange at, for example a healthcare provider. Once the initial exchange takes place the user can authenticate with the key authority and there is no need to retain the identity associated with that access key. After authenticating with the key authority and receiving a token, a user may query the server to determine whether likely contact occurred with a carrier. This may be on demand, but in order to efficiently manage server resources, a daily scheduled background task would be more efficient. The server stores past location contexts as hashed keys for carriers who have tested positive. The server obtains these records from an individual with authorization from a care provider after confirming test results; the care provider does not retain user location data. The server maintains these records for a set period of time appropriate for characteristics of the viral strain.

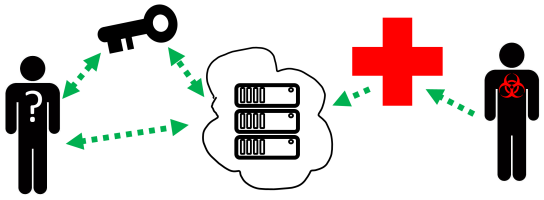


Fig. 1: System diagram depicting information exchange among, from left to right, a user, the key authority, a server, the care provider and a carrier.

Procedure

1) Each user continuously scans and records all observable AP beacons, capturing SSID, BSSID and RSS

2) Up to 10 AP beacon measurements with highest RSS observed over one minute time intervals are hashed and stored individually as context keys: $h_j = H(BSSID_j, SSID_j, k(time), time)$, where H() is a hash function and $k(time)$ is a pseudo-random value derived from the time and random daily key distributed by the key authority; time is epoch time at the resolution of the (in this case one minute) interval t_s[2]
3) Time intervals are synchronized with GNSS time, and grouped according to their time interval
4) If a user tests positive, the health provider uploads the user's n last context keys to the server, where n is defined by policy guidance relating to disease incubation period or pathology
5) Groups of context keys are stored in the server according to Group ID, which is generated for a set of associated context keys when they are uploaded[3]
6) A user, wishing to test for contact, requests a temporary token from the key authority, which authenticates and limits the rate of queries. This is computed as: $u = k_s(H(UUID), time)$, where $k_s()$ represents encryption using a symmetric key known only to the key authority and server; and H(UUID) is the hash of a unique identifier used to confirm an authorized user
7) The user then queries the server by sending u, $H(UUID)$ and up to m context keys
8) The server responds to the request by returning RSS and Group IDs for any of the context keys found in the server[4]
9) The user then computes the probability of a contact based on the AP match factor and the RSS vector distance metric

The server stores groups of associated context keys of carriers in memory as a hash table along with their group IDs and RSS values. After authentication, as users query the server with some of their own context keys, the server responds with the group IDs and RSS values associated with the user's context keys, if present. A user authenticates by presenting a unique ID to the key authority, which in turn responds with a token: the encrypted output of the current time and hash of the unique ID. The key authority maintains a record of authorized user IDs and limits token issuance over, in our case, a 24-hour period. The user presents this token and unique ID to the server performing a query. The server then computes the token for that user ID and compares the outputs; a match authenticates the user and results in a server response. This architecture provides several advantages in that it reduces the authentication overhead of computation on the server, thereby sparing resources for processing user queries, improves privacy by separately storing and processing unique identifiers and proximity context information and allows flexibility to adapt key length and security measures as required. We further examine the latter feature in Section IV-C.

Empirical data drove a quantization of the RSS value to improve the match rate, because RF fading and natural variation in RSS measurements could lead to false negative matches across two groups as we discuss in our evaluation. This quantization provides a margin in which a match may still occur. Likewise, the timestamp for each value is a group timestamp at the start of a scan interval rather than a timestamp unique to an observation.

G. Asynchronous Co-location

One of the primary advantages of our approach over the Bluetooth-based (BT) exposure notification systems is the possibility of contact tracing beyond simultaneous contacts. By expanding hash groups from a location with prior timestamps, a user would be able to query the server to identify instances where an infectious carrier had previously been. Alternatively, to reduce the number of records processed, the scan interval can be increased to span a broader temporal period. Asynchronous contacts would be useful if, for example, a particular contagion is known to survive on surfaces or in the air for a prolonged period of time — as we have learned of SARS-CoV-2 [30], [33]. Because they rely on exchanges of information between individuals present, BT contact tracing cannot detect *non-simultaneous exposure* when the carrier and the user visited the same location one shortly after the other.

IV. Evaluation

A. Proximity Measurements

Here we present a selection of results collected to evaluate the effectiveness of our techniques as they would be used for proximity estimates. We tested scenarios representative of locations that would be of interest for contact tracing — large and small publicly accessible indoor spaces. Most sample intervals of indoor locations contained at least 10 APs, but sparse outdoor locations and, for example, parking lots would often contain less. We also include outdoor measurements for comparison. Each of these figures includes distance between two contacts in meters as reported via GPS (blue) with unitless RSSI Distance and Match Factors in red. WiFi measurements were obtained across heterogeneous devices to account for variations in hardware and firmware. These include phone models HTC 2PZC5, ZTE Z558VL, ZTE 2017U and NodeMCU microcontrollers containing Qualcomm's Snapdragon 835, 210, 820 and Expressif's ESP8266 WiFI SoC, respectively. This assortment captures a range of quality including (formerly) flagship hardware, entry level and commodity electronics. For the Android devices, we forked the open source wiglnet application to automate WiFi scanning and logging. Notably, iOS does not currently have an accessible API for WiFi scanning, so a firmware update would be necessary to fully implement this system on iPhones. We also developed an automated control and collection framework in micropython for use with the ESP8266 hardware. The latter software allows bulk collection in mobile (with cellular

[2]To perform asynchronous contact tracing, duplicate RSS observations would be paired with prior, in addition to current, timestamps

[3]The health provider may upload multiple users' context keys at a time to reduce the possibility of linkage attacks by a malicious server

[4]If the server does not contain sufficient records to provide the expected number of collisions, then an appropriate ratio of false matches can be generated and returned in response to queries

hotspot tether) or static environments with programmable scan intervals, timestamp clock synchronization, automatic data uploads and over-the-air updates. Table I provides a summary of the measurements we present here.

Scenario	Reference	Description
Fixed Distance	Fig. 2	RSS and Match Factors for two devices held at a fixed distance throughout an indoor environment
Match Threshold	Fig. 3	Composite measurements illustrating RSSI variance and threshold effect on matches
Environments	Fig. 4	Six series of indoor/outdoor distance measurements on one device relative to a single point

TABLE I: Summary of Measurement Experiments

While some of the results include seemingly precise distance measurements, these are merely for reference and their accuracy would largely depend on the environment. In general the question we seek to answer is of precise distance but rather, what is the likelihood that two contacts were in close enough proximity that *transmissible* contact may have occurred. Fig. 2 shows results in an indoor environment showing *close* contact (<1m). These results display a series of measurements between two devices (with different WiFi chipsets and Android OS versions) fixed at two distances relative to one another while navigating through a typical indoor residential environment at ground level with obstructing wood/brick reinforced walls, furniture and 12 visible APs . The average match factor between the two devices were 0.59 ± 0.21 and 0.63 ± 0.20, for 20cm and 3m, respectively. The average RSS distance factors were 0.12 ± 0.05 and 0.15 ± 0.06. RSS distance factor is displayed in the top pane, while the match factor is on the bottom. The measured GPS distance between points is shown in blue in both panes with its 68% confidence interval (provided by the Android API [15]).

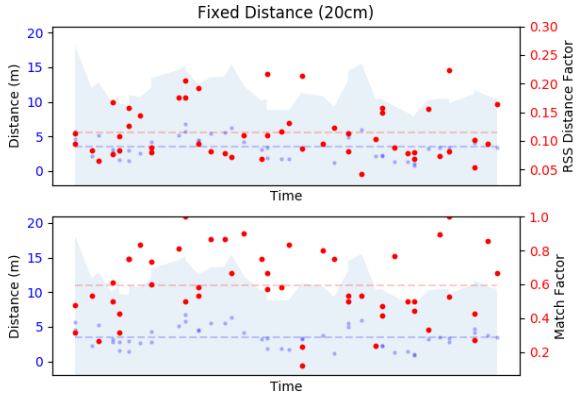


Fig. 2: GPS (blue), RSS and Match Factors (red) along an indoor path between two devices fixed at a 20cm distance.

The RSS distance measurements track the actual GPS measurements more closely over shorter, indoor distances. In contrast, for large distances, the match factor performance improves relative to the RSS distance metric. These relationships are depicted in Fig. 4. The primary difference between these environments would be the proximity and relative signal strength of measurements and, consequently, a higher number of marginally-observable APs for outdoor measurements. The other effect of increased radial distance from an AP is the increased region of ambiguity; i.e., a higher RSS constrains the physical distance estimate to a smaller circumference around an AP while a lower RSS can represent not only a larger circumference but also increased variation among the radial distance as propagation effects introduce additional error. Along shorter distances with higher RSS, there may not be enough variation or resolution between sets of observable APs to distinguish two points. This can be inferred from the Control Series of Fig. 3 (right). This curve represents composite pairwise RSSI match calculations (more than 300,000 separate comparisons) of varying thresholds across 9 static devices in close proximity (<1m) over several hours. That is, on rare occasions, two devices placed next to each other, measuring RSSI during the same 10 second interval (in this case), can vary by more than 20 dB. When comparing contacts, or matching, the RSSI threshold is key. Because even two identical devices sitting beside each other may not measure the same RSSI during the same interval, it is beneficial to further quantize the value. The quantization threshold should be large enough to capture the typical variation in RSSI but still small enough to establish some association with proximity. Variations in RSSI between devices can contribute to this challenge. For our measurements, we observed that in a controlled static co-location situation (but different orientations), the variation in RSSI measurements for the same observed AP between ESP8266 nodes, homogeneous hardware, was greater than the variation between *heterogeneous* hardware of the mobile phones used. Accordingly, we chose the low-cost ESP8266 RSSI node variance characteristics for control, as represented by the blue curve in Fig. 3. The other curves represent hours of composite pair-wise static measurements in different indoor environments for varying RSSI thresholds with physical proximity difference. From these control experiments, it would be reasonable to suggest an RSSI threshold/quantization between 10-15 dB to account for variation while preserving distance inference. This threshold could reliably, determine whether contacts are in the same room, same building or city block, for example, while avoiding false negatives that could occur if the minimum number of matches did not occur.

Fig. 3 (left) shows a representative snapshot of the number of matches between two static, co-located nodes when the measured RSSI of the same AP (meaning same Basic Service Set Identifier (BSSID) and channel) differ by less the 12 dB. Additionally, an approximately 3 hour time interval was parsed from a larger data set because it captured more than 80 unique APs. This is because the nodes were placed close enough to a road with a busy morning commute to capture many mobile/vehicular hotspots' beacons. Under normal circumstances, there would typically be only 10-12 APs visible from this location.

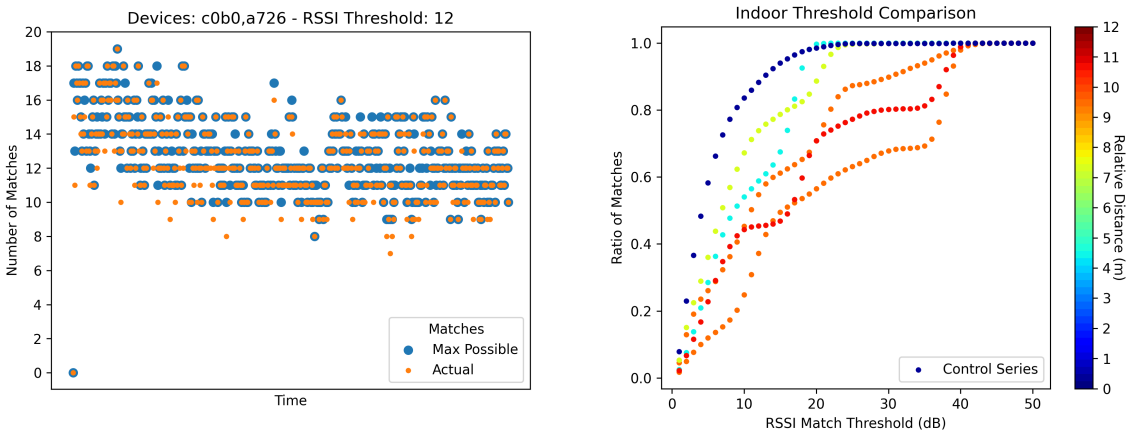


Fig. 3: The scatter plot (left) shows the number of matches between two co-located stationary sensors in the presence of more 80 access points, mostly mobile hotspots, over a period of approximately 3 hours with a 10 second measurement interval and 12 dB RSSI match threshold. The figure on the right presents the number of matches for several dispersed sensor measurement series of different distances in a complex indoor environment across a range RSSI match thresholds

B. System Performance

We demonstrate the scalability of our system by benchmarking our implementation as it may be deployed across a large user base. We measure memory, computation and throughput limitations by generating large quantities of client requests to a server using synthetic data. The first two test configurations consist of up to five machines generating equal numbers of client requests. Each single request comprises a query of 1000 records at the server. For this implementation we choose a 48-bit hash length with 32-bit length group keys — we examine the rationale for these lengths in Section IV-C. The server hosts either 200M records, consuming 22.1 GB of memory, or 4B records, consuming 434GB of memory. The third configuration, *434GB Local*, is identical to the second except that the client requests are generated on the local machine instead of remote hosts. These tests determine whether system limitations would arise due to memory, CPU, network or I/O constraints. The data are summarized in Table II below, showing the elapsed time for each test configuration and the number of clients.

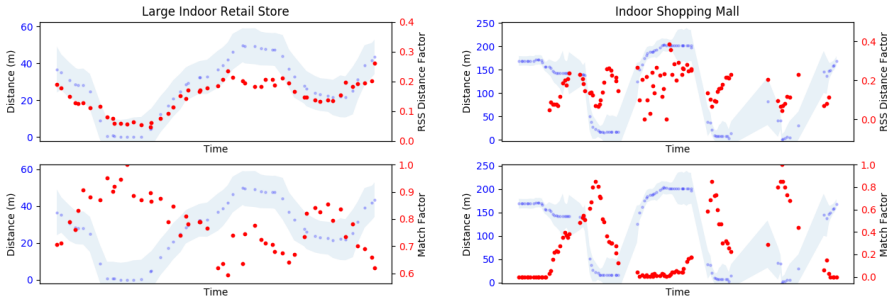
# Clients	22.1GB	434GB Remote	434GB Local
1000	3s	3s	3s
27000	43s	47s	52s
90000	152s	170s	175s

TABLE II: Shows the average elapsed time to serve increasing numbers of clients for each test configuration. Each client represents 1000 record requests.

The server records are stored in a hash-keyed dictionary. The dictionary contains a randomly generated set of keys based on the appropriate key length. A request consists of look-ups for a series of keys, which have a complexity of $O(1)$. The difference in elapsed time bears this out when comparing the first and second configurations — though we do see a slight relative increase in average response time due to secondary effects for very large number of client requests. CPU load is not separately reported as it remained constant across tests at 110%, representing utilization of slightly more than 1 of the 64 available Arm cores (of a r6g.16xlarge AWS instance). By comparing the second and third configurations we conclude the process was not bound by network constraints because the elapsed time follows a similar trend across all configurations. Eliminating these other possible constraints, we conclude that our system was I/O bound, likely due to memory access requests.

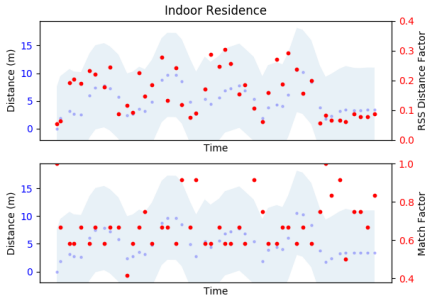
We extend these results to estimate resources for larger populations in Table III. We conservatively assume 1) a 50% adoption rate among the population, 2) a peak 3% population of active carriers, 3) every participant queries the server every day, 4) every participant records measurements for 12 hours every day, 5) 30% experiencing symptoms get tested [1] and 6) 2 keys per group are tested rather than querying all keys at once as explained in Section IV-C3.

In practice, few contact tracing applications have achieved a high (greater than 30%) adoption rate without mandatory compliance [46], [47]. Additionally, a sustained active carrier rate of 3% is overly conservative because at such a high rate, contact tracing would have long since ceased to be useful and only on rare occasions during the COVID-19 pandemic were such levels observed [1]. The server only stores data from carriers and responds to queries, so any computational load imposed by hash calculations is distributed among users. As the resource requirements following these assumptions fall within the constraints of available commodity hardware, and with the option of scaling to even larger instances as needed, we did not focus on optimizations to reduce the resource footprint or investigate backend server architectures. Nevertheless, the server load could be allocated among redundant, parallel server instances and may employ query scheduling or queues to conserve resources.

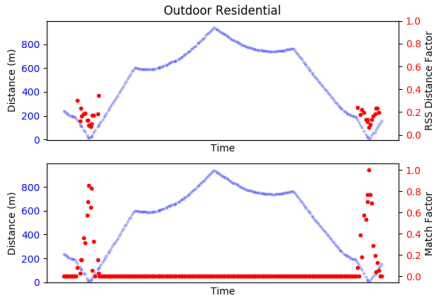


(a) Measurements along a non-overlapping perimeter path of a large indoor retail store

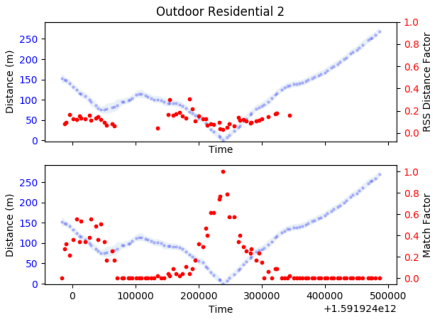
(b) Overlapping path measurements captured in an indoor mall



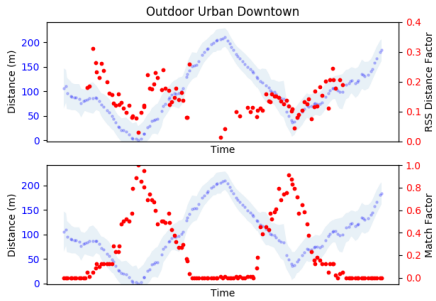
(c) Overlapping path measurements captured in a residence



(d) Measurements along an overlapping loop around a residential city block



(e) Non-overlapping path through a neighborhood



(f) Overlapping path in an outdoor urban downtown environment

Fig. 4: Distance measurements from three outdoor environments relative to a single point

Population	# Records	# Queries	Memory
100M	2.52e10	5.04e11	2.7TB
330M	8.32e10	1.66e12	9.0TB
1B	5.04e10	5.04e12	5.5TB

TABLE III: Maximum server requirements for varying populations

C. Threat Analysis

Contract tracing in any form requires an individual to disclose past locations or some form of spacial context in order to establish possible contacts. Besides the capability of past non-simultaneous contact tracing, the primary advantage of our WiFi-based contact tracing approach hinges on its favorable adaptation towards the hash collision filter privacy mechanism, which conceals the spacial context. Accordingly, the guarantees and limits of this approach necessitate more rigorous analysis.

1) Cryptographic Attacks: Privacy depends on the parameters of the hash collision filter and adversary's knowledge about the database contents. We consider three methods of cryptographic attacks: brute force, dictionary and targeted.

Brute Force. An adversary could attempt to compute all possible keys and unique groups to identify the inputs that match corresponding key groups in the server.

We define I as the set of possible inputs up to length l_I bits, G to be the set of all possible unique groups of size g consisting of unique elements of I. The total number of possible unique groups is then given by $|G| = \binom{|I|}{g}$.

An adversary would identify group matches by querying

the database over a list of inputs. This becomes infeasible to compute the groups, and hashes of each element in the group, over inputs of any useful size. For example, a small input space (e.g., 32-bit) computed over small groups (2 elements) would require computation and comparison of $\binom{2^{32}}{2} \approx 9.2x10^{18}$ groups of hashes. If an adversary knows what a match *should* look like, then a more efficient *dictionary* attack is possible.

Dictionary Attack. Statistical properties of the data such as character sets, field lengths or common identifiers can reduce the input space. Notwithstanding, if parameters for the hash collision filter are appropriately chosen, even these attacks become infeasible. Here the input space would be all possible combinations of SSID, BSSID and range of timestamps. A pre-computed dictionary attack would also be complicated by the inclusion of the key authority's rotating keys in the hash. Still, a successful attack would require associated locations to prove useful.

For a naïve dictionary attack, we calculate I, the set of possible inputs. This is then computed as:

$$\begin{aligned} I_D &= (SSID_{all})\, x\, (BSSID_{all})\, x\, (TimeSteps) \\ I_D &= \left(2^{256}\right) x\, \left(2^{48}\right) x\, (1440) \\ I_D &\approx 4.7x10^{94} \end{aligned}$$

Although still too large, with a few assumptions, the input space can be reduced. A dataset containing the 10,000 most common SSIDs from Wigl.net contains 89 unique characters with an average length of 9.4 characters. This yields $(89)^{9.4} \approx 2^{61}$ combinations for SSIDs. The current list of Organizational Unique Identifiers, which occupy the first 3 bytes of BSSIDs, contains 29508 BSSID prefixes, limiting BSSIDs to (29508 x 2^{24}) $\approx 2^{40}$. Computing the input space as above, we obtain $I_d \approx 3.7x10^{33}$:

$$\begin{aligned} I_d &= (SSID_I)\, x\, (BSSID_I)\, x\, (TimeSteps) \\ I_d &= \left(2^{61}\right) x\, \left(2^{40}\right) x\, (1440) \\ I_d &\approx 3.7x10^{33} \end{aligned}$$

These combinations would still need to be tried as group sets of context keys; a narrower target input space would be necessary.

Targeted Attack. With specific knowledge about a record that may be contained within the hash collision filter, it may be possible to confirm the existence of this record with high probability. Depending on the extent of this knowledge, we may conclude that the privacy of the record is implicitly compromised. In a linkage attack, if an adversary knows where a target may be, she may attempt to obtain and compute a local mapping of APs for some time period. Some location-based services already use such lists as inputs for coarse location estimates; open crowdsourced services may also provide such lists [6]. If we use the total number of reported APs from Wiglnet, we get the following result.

$$\begin{aligned} I_t &= (SSID - BSSID_{combo})\, x\, (TimeSteps) \\ I_t &= (721,920,161)\, x\, (1440) \\ I_t &\approx 1.0x10^{12} \end{aligned}$$

If we assume, per our implementation, a group size of 10, but at least 2 matches per group to indicate contact, then we have $|G| = \binom{|I_t|}{2} \approx 7.5x10^{14}$.

We can improve this attack with local mappings. By targeting a user assumed to be in a particular region, the input space can be restricted to APs from that locale. Data from Wigle.net suggests U.S. postal codes can contain many thousands of WiFi APs, for example [6]. Another method would segment the WiFi database into boundaries by, for example, using K-nearest neighbor with a maximum distance threshold consistent with WiFi propagation considerations. By targeting a limited region or segmenting as described, the input space for an attack could be significantly reduced but may still require express collection efforts to map and maintain accurate records for an environment. This is particularly important for our approach because the difference of even a single node in a region of interest, perhaps due to outdated source database records, could defeat this type of attack.

We also note, as others have [16], that a threshold for the level of protection afforded by a location-based-service privacy mechanism is sufficient if the amount of effort required to de-anonymize or track an individual is no less than what traditional investigative methods would require. In our targeted attack example, in order to succeed in a linkage attack against a target in an area, accurate records would be required. Physical measurement efforts (i.e., WiFi wardriving) would be necessary to ensure success. If no specific target was selected but instead the goal was an exploitation for a linkage attack from any leaked information, then the attacker would still need to deploy sensors which can unambiguously link sets of context key groups — which would be particularly challenging in crowded public areas where users' paths cross. Assuming a continuous trace were captured, the adversary would still need to query the server a sufficient number of times without being rate limited.

2) Additional Countermeasures: To reduce the effectiveness of targeted attacks, we outline several countermeasures that were not implemented in our prototype but would be desirable for deployment.

Rate Limiting. The server limits the number of queries a user can make. The key authority provides an authentication token and allows the server to recognize excessive queries from a single client. The number of queries per day are limited to two weeks' worth of observations. This is sufficient for contact tracing but far short of the number necessary to succeed in the cryptographic attacks we outlined above. Even in a collusion attack in which an adversary was able to obtain access credentials from willing participants, the number required would present a substantial effort and likely still bear hallmarks of a coordinated attack — irregular access patterns, connections from suspicious or similar IP addresses.

Daily Key Rotation. The key authority generates daily keys used by clients as inputs to the hash function. The purpose of these keys is to prevent an effective pre-computed dictionary attack. Because the keys are only released to users each day, it would impose a limit on the time available to compute hashes for a naïve dictionary attack.

Hash Length Reduction. In our implementation we choose a

hash length such that the likelihood of obtaining one collision or false match between groups is high, while obtaining more than one is low. If we relax this requirement to allow multiple matches, but still a fraction of the group size, then we mitigate these attacks. The cost of this change would be an increase in the false positive rate.

3) Server Attacks: Our prior attack scenarios involved a user exploiting public interfaces to the server to obtain information. Here we examine the risk of a compromised or malicious server to user information.

Information Leakage. If an adversary obtains access to all records uploaded to the database then the possibility for information leakage would be similar to that described in cryptographic attacks, except group ownership would already be known and the rate-limiting feature would not exist. Using the attacks mentioned before, we would assume an attacker could eventually discover all locations associated with individual uploaded records. The remaining risk to consider is the possibility of a successful linkage attack among these records, but because no association between submitted records is stored in the database this becomes a challenge — largely depending on the ratio of non-intersecting user traces. Researcher has shown that at least 4 spatio-temporal points are necessary to uniquely identify an individual among a collection of location traces [29].

De-anonymization risks would increase if the server's database is sparsely populated, containing only hash keys from a small population of users. Such a scenario could defeat the collision obfuscation approach we propose. This vulnerability can be overcome by returning false key groups in a user query when the server's database is not fully populated. Since the hash output length is configurable, and the expected number of hash collisions can be computed, the server can return a similar number of false collisions embedded in randomly generated groups–that are not associated with any real location. These key groups could also be stored on the server until replaced with true user submitted key groups to protect collected information in the event of a server compromise.

Observing Uploads. If a server operator observes uploads from users or carriers, then a linkage attack may succeed because the association between consecutive key group records may be discovered. Known location traces can be used to de-anonymize users [29], [50]. This risk can be mitigated by uploading group records combined from many carriers but is not possible for normal user queries. Instead, a user could upload one or two keys from each group. If the server responds with a match for any of these keys, then another key from the matching groups can be sent in a query. If both keys match in a group, then the user can submit the remaining keys in the matching group to confirm contact. In general this is effective because the the highest-RSS APs are used for matching and with, for example, a 1-minute scan interval it would be very unlikely that two users in close proximity would not each capture at least one of the beacon messages for such an AP; additional keys would only be uploaded to achieve the desired confidence and resolution, but only for locations where the first key matched. Variations of this approach are possible where the user (or a malicious server) may lie about matches to either discover or conceal information.

4) Longitudinal Location Privacy: In addition to some of the protective measures mentioned above, another privacy-preserving technique generally available to location-aware applications is geofencing. In the context of exposure notification and contact tracing, locations closely and uniquely associated with an individual, such as work and home, can be excluded from collection. Additionally, data collected while driving in a personal vehicle could be excluded, while travel using public transportation could be preserved. The timestamps, distances and routes could be used to automatically distinguish the two.

Other privacy-preserving approaches available to location-based services may not be applicable to the exposure notification problem, or at least our approach. For example, location obfuscation and coordinate transformation approaches could increase privacy, but at the expense of utility or false positive and false negative rates for contact events.

V. Discussion

There remain related open research questions that could further improve digital contact tracing systems. One example would be comparative performance of different systems (BT or WiFi) across different environments. Both approaches would still be susceptible to multi-path effects and other distortions but may perform differently in some environments. Could a BT signal penetrate through a crowd versus a WiFi approach with elevated APs providing a better line-of-sight to the sensors? An extensive, large-scale deployment may be necessary to answer such questions. Our future work will expand the results presented here to include quantitative benchmarks for environmental performance and power usage against other leading approaches.

A problem that merits further investigation is the role of mobile hotspots in WiFi-based contact tracing. For WiFi fingerprinting, hotspots pose a challenge because their contributions may be different or absent from the training model data set. In our contact tracing application, however, mobile hotspots may improve the results. This occurs because: (1) A mobile AP provides an additional, ephemeral signal which contributes to the uniqueness of a matched set of observations at one point in time. (2) A mobile hotstpot is less likely to be captured in a collected (e.g., "wardriving") database of WiFi APs, improving security. The first characteristic contributes to the quality of match by providing an additional AP. The more APs available in an area, the more reliable and accurate is the contact trace. The second characteristic contributes to the robustness of a contact tracing scan against offline attacks in which a user has access to a geotagged list of previously observed APs.

Limitations of mobile hardware may also impact results. BT-based contact tracing approaches required firmware updates on IOS and Android devices to accommodate the beacon exchange scheme. While not a limitation in our application, state-of-the art WiFi localization techniques yielding precise coordinates often rely on channel state information (CSI) which is not normally available on mobile phones [9], [41]. For WiFi, firmware-limited scan rates could impact performance

and the user experience. Ideally, the exposure notification application would exist as a background service, requiring little to no user interaction on a daily basis. Additionally, the application must not noticeably affect battery life on the device. The latter consideration led manufacturers in recent years to throttle certain functions, including, WiFi scan rate. This was observed by other researches who noted an impact on "war-driving" applications. An alternate approach proposed an adaptive scan technique [2] which only scans after some threshold amount of movement was detected. Although not implemented, we demonstrated the feasibility of this technique and additional benefits in terms of reduced server storage requirements and processing load.

VI. Conclusion

We introduced a new passive, WiFi-scanning approach to contact tracing, offering improvements to security and privacy through our hash collision filter. It also provides asynchronous co-location capabilities. Our evaluation and threat analysis shows its effectiveness against information-leakage attacks. We demonstrated through our collected real-world data and implementation that this system is scalable and a viable alternative to BT-based approaches for deployment on mobile devices.

References

[1] Almagor, J., and Picascia, S. Exploring the effectiveness of a covid-19 contact tracing app using an agent-based model. *Nature Scientific Reports 10* (2020), 22235.

[2] Altuwaiyan, T., Hadian, M., and Liang, X. Epic: Efficient privacy-preserving contact tracing for infection detection. In *2018 IEEE International Conference on Communications (ICC)* (2018), pp. 1–6.

[3] Bahl, P., and Padmanabhan, V. N. RADAR: an in-building RF-based user location and tracking system. In *IEEE INFOCOM 2000* (2000), vol. 2, pp. 775–784.

[4] Becker, J., Li, D., and Starobinski, D. Tracking anonymized bluetooth devices. *Proceedings on Privacy Enhancing Technologies 2019* (07 2019), 50–65.

[5] Bloom, B. H. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM 13*, 7 (July 1970), 422–426.

[6] Bobzilla, A., and Uhtu. Wigle.net: All the networks. Found by Everyone.

[7] Canetti, R., Kalai, Y., Lysyanskaya, A., Rivest, R., Shamir, A., Shen, E., Trachtenberg, A., Varia, M., and Weitzner, D. Privacy-preserving automated exposure notification. *IACR Cryptol. ePrint Arch. 2020* (2020), 863.

[8] Chazelle, B., Kilian, J., Rubinfeld, R., and Tal, A. The bloomier filter: an efficient data structure for static support lookup tables. In *SODA '04* (2004).

[9] Choi, J. Sensor-aided learning for wi-fi positioning with beacon channel state information. *IEEE Transactions on Wireless Communications 21*, 7 (2022), 5251–5264.

[10] Das, S., Chatterjee, S., Chakraborty, S., and Mitra, B. An unsupervised model for detecting passively encountering groups from wifi signals. In *2018 IEEE Global Communications Conference (GLOBECOM)* (2018), pp. 1–7.

[11] Dmitrienko, M., Singh, A., Erichsen, P., and Raskar, R. Proximity inference with wifi-colocation during the covid-19 pandemic. *ArXiv abs/2009.12699* (2020).

[12] DP3T. DP3T - Decentralized Privacy-Preserving Proximity Tracing.

[13] Erlingsson, U., Pihur, V., and Korolova, A. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security* (New York, NY, USA, 2014), CCS '14, Association for Computing Machinery, p. 1054–1067.

[14] Givehchian, H., Bhaskar, N., Herrera, E. R., Soto, H. R. L., Dameff, C., Bharadia, D., and Schulman, A. Evaluating physical-layer ble location tracking attacks on mobile devices. In *2022 IEEE Symposium on Security and Privacy (SP)* (2022), pp. 1690–1704.

[15] Google. Android Developers Reference: Location.

[16] Gruteser, M., and Grunwald, D. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services* (New York, NY, USA, 2003), MobiSys '03, Association for Computing Machinery, p. 31–42.

[17] Guo, Z.-D., Wang, Z.-Y., Zhang, S.-F., Li, X., Li, L., Li, C., Cui, Y., Fu, R.-B., Dong, Y.-Z., Chi, X.-Y., Zhang, M.-Y., Liu, K., Cao, C., Liu, B., Zhang, K., Gao, Y.-W., Lu, B., and Chen, W. Aerosol and surface distribution of severe acute respiratory syndrome coronavirus 2 in hospital wards, wuhan, china, 2020. *Emerging infectious diseases 26* (04 2020).

[18] Hashemi, H. The indoor radio propagation channel. *Proc. IEEE 81*, 7 (July 1993), 943–968.

[19] He, S., and Chan, S. . G. Wi-fi fingerprint-based indoor positioning: Recent advances and comparisons. *IEEE Communications Surveys Tutorials 18*, 1 (2016), 466–490.

[20] Hekmati, A., Ramachandran, G., and Krishnamachari, B. Contain: Privacy-oriented contact tracing protocols for epidemics. *ArXiv abs/2004.05251* (2020).

[21] Kilbourne, E. Influenza pandemics of the 20th century. *Emerging Infectious Diseases 12* (2006), 9 – 14.

[22] Kim, U., Lee, S. Y., Lee, J., Lee, A., Kim, S., Choi, O., Lee, J., Kee, S., and Jang, H.-C. Air and environmental contamination caused by covid-19 patients: a multi-center study. *Journal of Korean medical science 35* (09 2020), e332.

[23] Krumm, J., and Platt, J. Minimizing calibration effort for an indoor 802.11 device location measurement system. *Microsoft Research* (2003).

[24] Li, G., Hu, S., Zhong, S., Tsui, W. L., and Chan, S.-H. G. vcontact: Private wifi-based iot contact tracing with virus lifespan. *IEEE Internet of Things Journal 9*, 5 (2022), 3465–3480.

[25] Li, Y., Qian, H., Hang, J., Chen, X., Cheng, P., Ling, H., Wang, S., Liang, P., Li, J., Xiao, S., Wei, J., Liu, L., Cowling, B. J., and Kang, M. Probable airborne transmission of sars-cov-2 in a poorly ventilated restaurant. *Building and Environment 196* (2021), 107788.

[26] Liu, J., Asokan, N., and Pinkas, B. Secure deduplication of encrypted data without additional independent servers. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security* (New York, NY, USA, 2015), CCS '15, Association for Computing Machinery, p. 874–885.

[27] Luo, Y., Zhang, C., Zhang, Y., Zuo, C., Xuan, D., Lin, Z., Champion, A., and Shroff, N. Acoustic-turf: Acoustic-based privacy-preserving covid-19 contact tracing.

[28] Meunier, J.-L. Peer-to-peer determination of proximity using wireless network data. pp. 70 – 74.

[29] Montjoye, Y.-A., Hidalgo, C., Verleysen, M., and Blondel, V. Unique in the crowd: The privacy bounds of human mobility. *Scientific reports 3* (03 2013), 1376.

[30] Morawska, L., and Milton, D. K. It Is Time to Address Airborne Transmission of Coronavirus Disease 2019 (COVID-19). *Clinical Infectious Diseases 71*, 9 (07 2020), 2311–2313.

[31] Nikoofard, A., Givehchian, H., Bhaskar, N., Schulman, A., Bharadia, D., and Mercier, P. P. Protecting bluetooth user privacy through obfuscation of carrier frequency offset. *IEEE Transactions on Circuits and Systems II: Express Briefs 70*, 2 (2023), 541–545.

[32] Niu, J., Gu, Y., Lu, B., Cheng, L., and Jun, J. H. Wifi fingerprint localization in open space. In *LCN 2013* (2013).

[33] Noorimotlagh, Z., Jaafarzadeh, N., Martínez, S. S., and Mirzaee, S. A. A systematic review of possible airborne transmission of the covid-19 virus (sars-cov-2) in the indoor air environment. *Environmental Research 193* (2021), 110612.

[34] PACT. Pact: Private automated contact tracing.

[35] Prasad, A., and Kotz, D. Enact: Encounter-based architecture for contact tracing. pp. 37–42.

[36] PrivateKit, M. S. P. MIT Safe Paths PrivateKit.

[37] Raskar, R., Singh, A., Zimmerman, S., and Kanaparti, S. Adding location and global context to the google/apple exposure notification bluetooth api. *ArXiv abs/2007.02317* (2020).

[38] Rocamora, J. M. B., Ho, I. W. H., Mak, W.-M., and Lau, A. P. T. Survey of csi fingerprinting-based indoor positioning and mobility tracking systems. *IET Signal Process. 14* (2020), 407–419.

[39] Roomp, K., and Oliver, N. Acdc-tracing: Towards anonymous citizen-driven contact tracing, 2020.

[40] SAINZ, F. Privacy-preserving contact tracing.

[41] SCHULZ, M., LINK, J., GRINGOLI, F., AND HOLLICK, M. Shadow wi-fi: Teaching smartphones to transmit raw signals and to extract channel state information to implement practical covert channels over wi-fi. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services* (New York, NY, USA, 2018), MobiSys '18, Association for Computing Machinery, p. 256–268.

[42] SINGER, N., AND SANG-HUN, C. As coronavirus surveillance escalates, personal privacy plummets. *The New York Times*.

[43] STADNYTSKYI, V., BAX, C., BAX, A., AND ANFINRUD, P. The airborne lifetime of small speech droplets and their potential importance in sars-cov-2 transmission. *Proceedings of the National Academy of Sciences 117* (05 2020), 202006874.

[44] TCN. Temporary Contact Numbers (TCN) Coalition.

[45] TRIVEDI, A., ZAKARIA, C., BALAN, R., AND SHENOY, P. Wifitrace: Network-based contact tracing for infectious diseases using passive wifi sensing, 2021.

[46] WIERTZ, C., BANERJEE, A., ACAR, O. A., AND GHOSH, A. Predicted adoption rates of contact tracing app configurations - insights from a choice-based conjoint study with a representative sample of the uk population. *SSRN Electronic Journal* (01 2020).

[47] WYMANT, C., FERRETTI, L., TSALLIS, D., CHARALAMBIDES, M., ABELER-DÖRNER, L., BONSALL, D., HINCH, R., KENDALL, M., MILSOM, L., AYRES, M., ET AL. The epidemiological impact of the nhs covid-19 app. *Nature* (2021), 1–8.

[48] XU, H., ZHANG, L., ONIRETI, O., FANG, Y., BUCHANAN, W. J., AND IMRAN, M. A. Beeptrace: Blockchain-enabled privacy-preserving contact tracing for covid-19 pandemic and beyond. *IEEE Internet of Things Journal 8*, 5 (2021), 3915–3929.

[49] YUEN, B., BIE, Y., CAIRNS, D., HARPER, G., XU, J., CHANG, C., DONG, X., AND LU, T. Wi-fi and bluetooth contact tracing without user intervention. *IEEE Access 10* (2022), 91027–91044.

[50] ZANG, H., AND BOLOT, J. Anonymization of location data does not work: A large-scale measurement study. In *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking* (New York, NY, USA, 2011), MobiCom '11, Association for Computing Machinery, p. 145–156.