A BLUEPRINT FOR EFFECTIVE PANDEMIC MITIGATION

Rahul Singh[1], Wenbo Ren[2], Fang Liu[3], Dong Xuan[4], Zhiqiang Lin[5], Ness B. Shroff[6]
[1]Department of ECE, Indian Institute of Science, Bangalore, [2]CSE Department, The Ohio State University, [3]ECE Department, The Ohio State University, [4]CSE Department, The Ohio State University, [5]CSE Department, The Ohio State University, [6]ECE and CSE Departments, The Ohio State University,

NOTE: Corresponding author: Ness B. Shroff, shroff.11@osu.edu

Abstract – Traditional methods for mitigating pandemics employ a dual strategy of contact tracing plus testing combined with quarantining and isolation. The contact tracing aspect is usually done via manual (human) contact tracers, which are labor-intensive and expensive. In many large-scale pandemics (e.g., COVID-19), testing capacity is resource limited, and current myopic testing strategies are resource wasteful. To address these challenges, in this work, we provide a blueprint on how to contain the spread of a pandemic by leveraging wireless technologies and advances in sequential learning for efficiently using testing resources in order to mitigate the spread of a large-scale pandemic.

We study how different wireless technologies could be leveraged to improve contact tracing and reduce the probabilities of detection and false alarms. The idea is to integrate different streams of data in order to create a *susceptibility* graph whose nodes correspond to an individual and whose links correspond to spreading probabilities. We then show how to develop efficient sequential learning based algorithms in order to minimize the spread of the virus infection. In particular, we show that current contact tracing plus testing strategies that are aimed at identifying (and testing) individuals with the highest probability of infection are inefficient. Rather, we argue that in a resource constrained testing environment, it is instead better to test those individuals whose expected impact on virus spread is the highest. We rigorously formulate the resource constrained testing problem as a sequential learning problem and provide efficient algorithms to solve it. We also provide numerical results that show the efficacy of our testing strategy.

Keywords – contact tracing, COVID-19, selective testing

1. INTRODUCTION

The outbreak of COVID-19 has unfolded an unprecedented worldwide health, economical, and social crisis. Today, COVID-19 has spread to 188 countries, infected nearly 30 million people globally, and resulted in close to one million deaths. The International Monetary Fund (IMF) has predicted that the global economy will shrink by 3% this year, the worst decline since the Great Depression of the 1930s [2]. Today, millions of workers have been laid off, and the tourism or hospitality industry has been hurt particularly hard.

The COVID-19 outbreak and the mixed successes that nations have had in controlling the virus has underscored the need for the development of technological tools for pandemic mitigation. This paper provides a blueprint on how technologies should be used in conjunction with smart testing techniques in order to contain a pandemic such as COVID-19. The first step for pandemic mitigation is to identify or trace the close contacts who might have been exposed to the disease from a contagious individual. Contact tracing, an old technique, has been used as effective tools to battle pandemics for many years, and some countries do use aggressive contact tracing to successfully contain COVID-19. However, traditionally, contact tracing is a manual approach, relying on a human being's memory. Such an approach cannot scale to large and rapidly moved populations today. Meanwhile, manual tracing may result in delays, which could limit its utility. Therefore, recently numerous digital contact tracing systems have been developed and deployed across the globe, by using a wide variety of sources to track "encounters" including CCTV footage, records of credit card transactions [1], locations measured using cellular networks or WiFi hotspots [26], locations via GPS, and cryptographic tokens exchanged via Bluetooth Low Energy (BLE) or acoustic channels [15]. For a recent survey of works on contact tracing and privacy-aware contact tracing, see [9, 11, 10, 23, 24, 28, 7, 19]. Also see [14, 25] for some interesting recent works in this area.

However, currently contact tracing system appears to be uncoordinated individual efforts focused on individual technologies and one type of data stream (e.g., camera based systems, phone apps, etc.). Further, a number of these solutions suffer from a variety of technological limitations including a lack of coverage, privacy and security concerns, high missed detection and/or false alarm rates. For instance, increasingly Bluetooth-based contact tracing has gained mainstream use particularly with Apple/Google's support. However, our recent analysis [29] with the released COVID-19 contact tracing apps shows that most Bluetooth-based contact tracing apps use just the received signal strength indicator (*RSSI*) of the Bluetooth for distance measurements. Unfortunately, in practice, numerous factors can affect the *RSSI* that can make the distance measurement inaccurate, such as the power of the antenna used for broadcasting (i.e., the *TxPower*) and the obstacles blocking transmission paths. Moreover, Bluetooth-based proximity tracing can also raise false positives because of the potential misinterpretation of various scenarios. For example, a proximity tracing system may interpret two users have contact even if they are separated by a solid wall, where the risk of infection is much lower than the risk indicated by the measured distance.

Therefore, we would like to propose an improved approach by combining as many data sources as possible in an integrated way, with the key objective of minimizing false positives and false negatives in the contact tracing and meanwhile protecting user's privacy. The contact tracing data we can collect includes (1) multiple channels including both Bluetooth and ultrasound (using both microphones and speakers available in the smartphone), and multiple sources including (2) WiFi and (3) cellular networks if they are available. We show how we can use improved methodology to collect data that is privacy aware, transparent, and integrated in Section 2.

Similarly, while testing followed by quarantining/isolation is a powerful tool against a pandemic such as COVID-19, testing capacity remains an issue, especially in hard hit areas where testing results could take multiple days, even up to a week, to arrive. While traditional approaches have focused on testing individuals who exhibit symptoms or have come in contact with other infected individuals, these approaches miss out many potential areas of outbreak where asymptomatic or presymptomatic super-spreaders seed the virus, which gets detected only after it has already spread significantly. Thus, testing capacity needs to be used judiciously to prevent widespread outbreaks. In Section 4, we argue that the current myopic approach to testing focuses on identifying individuals with the highest probability of being infected, which does not help minimize the overall number of infected individuals.

Organization. The rest of this paper is organized as follows: In Section 2, we describe an improved and integrated methodology to collect contact tracing data. In Section 3, we describe techniques that enable us to efficiently integrate the data collected from various streams. This allows us to reduce the "error probabilities" associated with false alarms or missed detection of diseases, and generate a dynamic *susceptibility graph*. In Section 4, we address the practical problem of testing under constraints on resources. We perform simulations to show necessities of contact tracing and building a contact graph in Section 5, and conclude in Section 6.

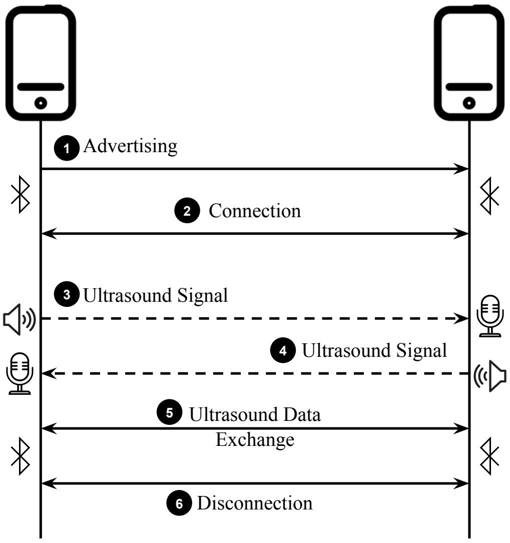


Fig. 1 – A Simplified Protocol for Improved Data Exchange.

2. IMPROVED DATA COLLECTION

There are two fundamental objectives that a good digital contact tracing system must satisfy: (1) it should be effective in tracking an individual (e.g., few false positives and missed detections), and (2) it should protect the privacy of users. For example, a CCTV footage would be highly effective if (1) were the sole objective, however it does not meet (2) since it is too much privacy-invasive. Therefore, we must look for effective and privacy-aware digital contact tracing techniques.

Since the outbreak of Covid-19, numerous techniques based on Bluetooth, WiFi, and cellular networks have been developed for contact tracing. Each technique has its own pros and cons. For instance, Bluetooth based solutions can achieve reliable communication and a low energy operation, but these suffer from a high rate of false positives due to a long communication range. WiFi based solutions do not require installation of apps on mobile phones, and they rely heavily on access point (AP) deployment, and its coverage. Therefore, in this paper, we aim to present an integrated approach that improves the accuracy of a Bluetooth-based approach with additional channels, and combines WiFi and cellular information if they are available. Furthermore by utilizing clever algorithms that are provably optimal, it aims to increase the efficiency with which infected individuals are contained early without infecting too many of their "neighbors".

Improving Bluetooth-based Contact Tracing with Ultrasound Signals. Since Bluetooth-signals can penetrate obstacles such as solid walls, and also have a long transmission range, we would like to leverage other sensors in a smartphone to improve its proximity accuracy. In particular, we can use the inaudible ultrasound generated from the speaker and recorded by the microphone for distance measurement. This feature can be used very easily. Ultrasound-based distance measurement is promising, and it has been shown to achieve a centimeter-level accuracy, e.g. BeepBeep [16]. It can help to improve the performance of mobile contact tracing systems. According to the CDC guidelines [6], for the purpose of Covid transmission, it can be assumed that two individuals are in close contact (and hence likely to spread infection) if the distance between them is within 6 feet. Clearly, with an accuracy of a few centimeters, ultrasound is a very reliable and accurate technique for the purpose of mobile contact tracing.

Note that ultrasound cannot penetrate solid walls, and hence it can also help us to rule out those "false contacts" that have been declared by the Bluetooth connections occurring across walls.Using ultrasound for distance measurement requires a pair of devices to exchange data between each other [16]. Fortunately, Bluetooth-based mobile contact tracing systems have already implemented such an information exchange mechanism, and hence developers can easily extend these systems in order to support ultrasound distance measurement. This means that we can conveniently deploy both ultrasound and Bluetooth technologies.

An illustration of how we can integrate ultrasound and Bluetooth for improved contact tracing is presented in Fig. 1. The scheme is composed of six steps for the improved data exchange. In particular, when two smartphones encounter, they first use Bluetooth to discover each other and initiate connections. At the beginning, in Step ❶, one smartphone keeps advertising BLE packets to nearby devices. Meanwhile, another smartphone scans its environment for any possible advertising packets from its nearby devices, and initiates the connection in Step ❷. After a connection has been established, two devices emit ultrasound signals in turns (Step ❸ and ❹). When both devices have received signals from the other device, they record the information that is required by algorithms in order to measure distances, and then exchange this information amongst each other via Bluetooth in Step ❺. For example, when using the algorithm from BeepBeep [16] for distance measurement, in Step ❸, alongside emitting an ultrasound signal, the phone (P_A) will record the timestamp T_{A1} when it senses the signal sent by itself, and the other phone (P_B) will also store the timestamp T_{B1} when it receives such a signal. Similarly, in Step ❹, phone P_B records the timestamp for signal emitting as T_{B2} and phone P_A stores the timestamp T_{A2} when receiving signal. Next, in Step ❺, phone P_A has to send both T_{A1} and T_{A2} to phone P_B, and phone P_B also needs to share its two timestamps (i.e., T_{B1} and T_{B2}) with P_A.

$$Distance(P_A, P_B) = \frac{C}{2} \times ((T_{A2} - T_{A1}) - (T_{B2} - T_{B1})) + D_{B,B} + D_{A,A} \tag{1}$$

When each phone has these four timestamps, it can calculate the distance by using Equation *(1)*, where C is the sound speed, and $D_{A,A}$ or $D_{B,B}$ is the distance from one phone's speaker to its microphone respectively[1]. Finally, in Step ❻, two devices disconnect Bluetooth when the transmissions of other types of data for contact tracing finish.

Integrated Contact Tracing with WiFi.
Besides the above mobile phone based contact tracing approaches, we can also use WiFi logs. Different from the above approaches, the WiFi based solution does not require app installation on mobile phones, and relies upon widely deployed WiFi access points. It also does not require active involvement of mobile phones for exchanging the required information. The basic principle is described as follows: WiFi networks log all the associations and disconnections of devices connected to access points. We can analyze these WiFi logs to know where and when the devices (and hence their users) are close to each other. This provides information about the contacts of device users.

One of the advantages of such a WiFi-based solution is that the WiFi log data is always available as long as WiFi networks are active. Such networks allow both a) reactive and b) proactive techniques for contact tracing. Let's take the example of a university campus in order to illustrate this. In reactive contact tracing, once a student is confirmed to be infected, the university health administration can use the WiFi MAC addresses of the student's mobile phone and his/her other computing devices such as laptop and Apple watch etc. to search in the WiFi logs of campus networks, thereby determining the locations where the student visited during a certain time period, and also his/her contacts at these locations. In proactive contact tracing, the university health administration can proactively analyze WiFi logs to identify potential high risk users such as super spreaders, and hot-spots (such as big gathering) in the campus. The university health administration can proactively pull WiFi logs and determine if the number of students in a gathering exceeds the limit that social distancing allows, and take appropriate measures.

The WiFi based contact tracing technique described above has its own limitations, for example the AP association logs can generate false alarms. In order to overcome these, we might consider using Received Signal Strength Indicator (RSSI) and Channel State Information (CSI) to reduce the errors. WiFi based solution cannot be applied in areas without WiFi connectivity. We should consider enhancing contact tracing by integrating multiple solutions such as Bluetooth, WiFi and acoustic symbiotically, where one helps or replaces the other depending on user preferences, environmental dynamics, and resource availability. For example, in a WiFi-AP dense area such as a campus academic building, the WiFi-based solution can play a dominant role, while the mobile app running Bluetooth and acoustic

[1] For more details, please refer to BeepBeep [16]

based contact tracing can be automatically switched off for better energy savings. Similarly, if the density of WiFi AP is less, then Bluetooth and acoustic based solutions could be activated. They can help each other in a crowded and occluded (e.g., by walls and other types of obstacles) environment for accurate and reliable contact tracing. Note that Bluetooth gives relative location information whereas WiFi gives absolute location information. There might be a mismatch in the locations as identified by WiFi data modality in conjunction with Bluetooth data. In order to resolve this issue, we can model the location of the user using probabilistic techniques, and then use filtering techniques in order to derive a more accurate location. Readers interested in more details about these techniques can refer to Kalman filtering and related topics in [17, 13].

Integrated Contact Tracing with Cellular Network. The location of a smartphone can also be identified from its communication with nearby cell towers. Since a phone has to connect with cell towers in order to send and receive data through cellular network, it constantly searches nearby cell towers and initiates connections during movements. In each established connection, a cell tower not only knows which phone is trying (each phone has a unique identifier) to connect at which time, but can also calculate the distance from itself to such a phone (e.g., using the time elapsed between a ping command, and the corresponding reply). As such, having access to the locations of cell towers, as well as the distance of a phone from each of the involved towers, we can use a "triangulation" technique to pinpoint the location of a phone. However, in practice, such techniques often can only locate a smartphone in an area instead of an exact position.Moreover, using this technique for location tracking could raise privacy concerns, in that, it requires access to the identifier of each phone that may disclose user identify as well. Therefore, when only having the corresponding permissions, cellular network can be used for contact tracing, and meanwhile the user identity must also need to be protected.

How to collect the encounter records. Even though there are distributed models for contact tracing which allow each user to individually control whether or not to disclose its own encounter records, we advocate a centralized model in which each individual user's contact is collected by a central agency, and then stored at a central backend. This is bound to raise privacy concerns, and hence we need to introduce privacy-preserving mechanisms.To this end, we will generate pseudonyms for each user periodically and the linkage between a pseudonym and the real user is only resolved at the trust authority. The authority is only allowed to link pseudonyms to real users when the pseudonyms belong to (*i*) infected individuals that are confirmed by healthcare authorities or (*ii*) individuals who have close contact with infected ones. As such, the privacy of individuals who have no risk of infection will be preserved. A similar approached has been proposed and adopted in ROBERT [3].

Moreover, privacy concerns might arise from using the cell tower information for locating users because the identifier of a user's phone needs to be accessed. In order to mitigate these, we can also link such identifiers with pseudonyms. Similar approaches can also be applied in WiFi positioning. Therefore, the entry of data for upload involves self pseudonym, encounter pseudonym, timestamp, Bluetooth proximity, ultrasound proximity.

After processing each upload data entry, the output of this improved data collection procedure is data entries that involve pseudonyms of two encounters, the timestamp, the adjusted proximity, and the infection risk. In particular, the adjusted proximity is the weighted average from combining proximity measured from different sources (i.e., Bluetooth, ultrasound, WiFi, and cell tower), and the infection risk can be obtained by using environment detecting heuristics. For example, when there is no proximity measurement from ultrasound and the WiFi proximity indicates encounters are in different rooms, the infection risk can be adjusted to a low level. Besides the above privacy-preserving data collection methods, we can also apply tools from the field of differential privacy [8]. These utilize different kinds of data processing and noise injection methods, thereby making it difficult for any party to determine whether or not a particular individual is in the original data records and providing privacy protection to the users. Such a guarantee on the privacy would encourage more users to join the system.

3. DATA INTEGRATION AND SUSCEPTIBILITY GRAPH

Here, the goal will be to create a "susceptibility graph" that describes compactly the different ways in which disease is likely to spread. We begin by introducing this graph, and then also describe how to construct this graph by integrating the data from multiple sources. The graph would be time-variant.

3.1 Graph Structure

A basic version of the graph would contain the following components, and the designer is free to make reasonable modifications on it.

- **Nodes.** Each node represents an individual that could be potentially infected. Individuals that are isolated will be removed from the graph. Also, we can remove individuals who have recovered from the virus from the graph. However, since recovered individuals lose their antibodies for most viruses (including COVID-19), re-infections are possible after a period of time, so they would have to be reintroduced into the graph after some time. We use $\mathcal{N}$ to denote the set of nodes (individuals).

- **Node infection state.** For each individual i and time t, we use $X_i(t)$ to denote its infection state, where $X_i(t) = -1$ means that this individual does not have the disease, $X_i(t) = 0$ denotes that it has the virus but cannot spread the virus, and $X_i(t) = 1$ denotes that it has the virus and is able to spread the virus. For individual i and time t, we use $U_i(t)$ to denote its test results at time t. $U_i(t) = 0$ means individual i does not take a test at time t, $U_i(t) = -1$ means it is tested as negative at time t, and $U_i(t) = 1$ means it is tested as positive at time t.

- **Edges.** For every two nodes i and j, If persons i, j have direct contact, then there is an undirected edge (i, j) between them. We are free to choose the way in which we define "contact": for example if these people are staying less than 6 feet apart for at least a certain duration of time co-occurrence in a narrow space (e.g., a room and a bus), or participating in the same event. The contact information can be deduced by the techniques introduced in Section 2. We use $\mathcal{E}$ to denote the set of undirected edges and that (i, j) is in $\mathcal{E}$ means there is a contact between i and j.

- **Base infection probabilities.** Given the fact that we cannot test every individual, each untested individual has a base probability of being infected. This probability can be helpful for some tasks like finding a suspected infected individual. For instance, a person who contacted 500 people yesterday could be more likely to be infected than someone who was in contact with a confirmed positive person; and we can use the base infection probabilities to deduce this probability. The simulations in Section 5 also indicate that take the base infection probabilities into account can find and isolate more infected people. The base infection probability could be time-varying (e.g. abrupt changes due to certain events), and we use $p_b(t)$ to denote the base infection probability at time t.

- **Spreading probabilities.** In case two individuals i, j have been in contact, and one of them, say user i, was a positive case, then there is a chance that individual j got infected by the contact. This chance may also be time-variant. We let the spreading probability be denoted as $p_{i \to j,s}(t)$, which is the probability that j got infection from a contact with i. The calculation of the probability will be discussed later.

- **Time.** The time can either be continuous or discrete. Continuous time better fits the reality, but such an assumption also needs more storage and computation power to process the graph. Besides, given the fact that there are delays, or the occurrence time of events or contacts are not known precisely, how to construct an accurate timely graph needs to be investigated. If time is discretized, then the duration of a discrete time-slot could be anywhere from several minutes to one day. Using time slots can help reduce the storage and computation resources required.

3.2 Graph Construction

As discussed earlier, the graph $\mathcal{G}$ consists of a set of nodes and a set of edges, where each node holds a base infection probability and each edge holds a spreading probability. The nodes, edges, and the probabilities all need to be deduced from the data, and the data can be multi-sourced, for example wifi access logs of all users, CCTV cameras, or Bluetooth scanning based contact tracing. More details on how to construct such a graph are as follows:

- **Individual identification.** Identifying the individuals and avoiding duplication are necessary for the success of graph construction. How to do these may depend on the data collection methods. For instance, in the university WiFi logging system, an individual has and only has one access ID, and thus, this ID can be used to identify an individual. However, in general WiFi systems, an individual may have multiple devices, and removing the duplication is significant in this case. One method is restricting the tracking to one type of device such as mobile phones. When using the Bluetooth contact tracing, we can use the IDs of the mobile phones to identify the individuals, which is also applicable when using Bluetooth contact tracing and WiFi logging simultaneously.

- **Edge detection.** If there is a possible contact between two individuals, then an edge should be generated to connect these two individuals. The contact can have multiple types. For instance, the contact can be staying less than 6 feet, co-occurring in the same room at some time period, or connected to the same access point during some time period. This information can be deduced from the collected data.

- **Base infection probabilities.** The base infection probability can be deduced from the positive rate per test or the number of confirmed positive cases per randomly tested individuals. For instance, a university randomly tested 1,000 students and found 20 positive cases, then we can assume that each student of the university has 2% probability to be positive. If we do not have this information, we can use the number of newly detected infections in a period with a multiplier as the estimate.

- **Spreading probabilities (link probabilities).** Deducing the spreading probabilities is a relatively harder task, which can be divided into two steps.

The first step is to infer the type or level of contacts between two individuals. Have they stayed closer than 6 feet or stayed in the same room for a while? The second step is to deduce the link probability. Accurate characterizations of the link probabilities could come from exposure data studies to the virus. However, a reasonable model would be to use a concave function of time to estimate the link probability.

- **Testing results.** For an individual, if this individual has taken a virus test and got the result, then we know whether this individual has the disease or not (with a certain confidence).

3.3 Data Integration

In real-world scenarios, there are multiple data sources. For example, different contact tracking data sources as described earlier (Bluetooth or ultrasound contact tracing data, WiFi logs, GPS, etc.) could be integrated to greatly improve the quality of contact tracing. The integration could be done by using filtering techniques, in which we compute the probability of an edge conditioned on the (multi-source) information available to us. We would typically rely upon generative models of the data in order to compute these conditional probabilities. With multiple data sources, we need to deal with inconsistent data. For instance, Bluetooth gives relative location information whereas WiFi gives absolute location information, and the information of two sources may be inconsistent. We can deal with this issue by assuming that the data collections of the sources are random and independent and assign probability distributions to them. Probabilistic description allows "soft recovery" of data after we use filtering algorithms. Bayesian updates can be used to merge or pool information from various source. We can use Kalman filtering or some other filtering algorithm.

Such an integration can yield us the following kind of improvements:

- Reduced inaccuracies and better estimates of the link probabilities. Consider for example the case when people could have social contact by virtue of being located in a crowded facility such as students in the same classroom or people in the same flight. However, data sources such as building information, WIFI access might be noisy. In this case, one could combine GPS data (collected from probably smartphone usage) in order to yield an accurate estimate of social contacts.

4. TESTING UNDER RESOURCE CONSTRAINTS

The goal here is to leverage the information contained in the susceptibility graph in order to sequentially choose individuals for testing so as to minimize the spread of the pandemic. *Note that this objective is quite different from focusing on testing individuals with the highest probability of infection*, which is what current systems try to do. Rather our focus must be *on testing individuals that have the highest expected impact on viral spread.* Consider the following example as an illustration.

Example: Assume that two individuals i and j are infected with probabilities 0.1 and 0.3, respectively. However, assume that the expected number of individuals that i encounters is 50 times larger than the expected number of individuals that j comes in contact with. In this case, it makes more sense to prioritize testing individual i over individual j. This is another reason why we should test healthcare workers more often, because of their frequent contact with a large number of individuals. Based on this key insight, our goal will be to:

- Develop learning based approaches that result in smart testing capabilities which balance the exploration and exploitation subject to testing constraints. Isolate individuals who have been tested positive and quarantine their contacts.

- Our model will also incorporate practical issues such as inaccurate estimates, testing errors, pool testing, etc.

- Develop efficient rules of thumb that can be easily implemented in practice. This could mean testing asymptomatic individuals who have not encountered a confirmed infected person, but have made a large number of contacts.

4.1 Suspicious Infection Inference

One significant task is to find the most likely infected individuals from the partial observations, i.e., the test results of some individuals. To do this, one way is to interpret the probability that a person is infected given the partial observations, such as ("noisy") contact graph or test results of a few individuals from the graph etc. These algorithms could be based upon the susceptibility graph constructed by using the methods stated in Section 3.

4.1.1 Partial Observed Markov Decision Process (POMDP)

We formulate the problem of sequential testing for COVID-19 as a Markov Decision Process (MDP). Population is composed of N individuals, and the state evolves at discrete times $t \in [1, T]$. Let $X_i(t) \in \{0, 1\}$ denote the hidden state of individual i at t, where $X_i(t) = 0$ means that i is free of disease at t and $X_i(t) = 1$ indicates that i is infected. We use the vector $X(t) := (X_1(t), X_2(t), \dots, X_N(t)) \in \{0, 1\}^N$ to represent the state of the entire system. Let $\mathcal{X} := \{0, 1\}^N$ denote the state-space of the network. Note that the state vector $X(t)$ is never fully revealed to the learner[2].

Test and Quarantine: At each time $t \in [1, T]$, the learner has a unit budget to choose an individual $i \in [1, N]$ in order to "sample" (test for infection). Sampling an individual i at t reveals the state $X_i(t)$. We let $U(t) \in [0, N]$ denote the sampling decision at time t. In case no one is sampled at t, we let $U(t) = 0$. We let $Y_i(t)$ denote the test result at time t; $Y_i(t) = +1$ means the person tested positive, $Y_i(t) = -1$ means the test was negative, and $Y_i(t) = 0$ means that the individual was not tested at time t. The vector comprising of observations $Y_i(t)$ is denoted $Y(t)$.

If sampled individuals are found to be infected, then they are isolated, i.e., kept out of the population, and hence cannot spread the disease to their neighbors. We let $Q(t)$ denote the set of those individuals who are isolated at time t.

State Transition: Let us now look at the *controlled* transition probabilities of the controlled Markov process $X(t)$. We first introduce some notations. For $x, y \in \mathcal{X}$, define

$$\Delta_1(x, y) = \mathbb{1}\left\{\sum_{i=1}^{N} |x_i - y_i| = 1\right\} \quad \text{and} \tag{2}$$

$$\Delta_2(x, y) = \begin{cases} i \text{ if } x_i \neq y_i \text{ and } \Delta_1(x, y) = 1, \\ \emptyset \text{ otherwise .} \end{cases} \tag{3}$$

Clearly, $\Delta_1(y, x)$ assumes value 1 only if x and y differ in a single position; since disease can spread to only one more person during two consecutive times, this function is 0 if x cannot evolve to y in one single time-step. Δ_2 provides us with the node that "transitioned" to diseased state when the system evolved in a unit step from x to y. Thus, the single-step transition probabilities are given as

$$P_t(x, y) = \Delta_1(x, y) p \sum_{i \in \mathcal{V}'_t} w'_t(i, \Delta_2(y, x)). \tag{4}$$

Objective: Let $\mathcal{F}_t := \cup_{s=1}^{t} (U(s), Y(s), \ell(s))$ be the observation history of the learner [18]. Then, the policy π is a sampling decision at t on the basis of $\mathcal{F}_{t-1}$, i.e., $\pi : \mathcal{F}_{t-1} \mapsto U(t),\ t \in [1, T]$. Our goal is to find a policy that solves the following problem,

$$\min_{\pi}\ \mathbb{E}_{\pi}\left(\sum_{t=1}^{T} \|X(t)\|_1\right), \tag{5}$$

$$\text{s.t. } \mathbb{E}_{\pi}\left(\sum_{t=1}^{T} \mathbb{1}\left(U(t) \neq 0\right)\right) \leq C, \tag{6}$$

where $\|\cdot\|_1$ denotes the L_1 norm and C is the total testing capacity. The instantaneous cost $\|X(t)\|_1$ encourages the policy to keep the infections as low as possible in an as early as possible manner. The capacity constraints are crucial because not many testing kits are available during epidemics. An alternative, somewhat equivalent and simpler objective is to remove the capacity constraints altogether and include a cost for using testing kits,

$$\min_{\pi} \mathbb{E}_{\pi}\left(\sum_{t=1}^{T} \|X(t)\|_1 + \lambda \mathbb{1}\left(U(t) \neq 0\right)\right), \tag{7}$$

where $\lambda > 0$. We now briefly discuss how to solve the above discussed MDP. Theoretical results on the existence of optimal policies, and methods to solve constrained MDPs, or POMDPs can be found in [4, 21, 22]. In case the parameters describing the environment are unknown, we can use machine learning techniques developed in [20].

Belief State MDP: We introduce a belief state, which is a posterior distribution over the state space $\mathcal{X}$. This transforms the POMDP to a continuous-state MDP on the belief state. We denote the belief state by $\mathcal{J}(t) = \{\mathcal{J}(t, x)\}_{x \in \mathcal{X}}$, where $\mathcal{J}(t, x) := \mathbb{P}\left(X(t) = x | \mathcal{F}_t\right)$. By Bayes' Rule, the terms $\mathcal{J}_t(x)$ are computed recursively as

$$\mathcal{J}_{t+1}(x) = \sum_{y \in \mathcal{X}} \mathcal{J}_t(y) \mathbb{P}\left(Y_{U(t)} | X(t) = y\right) P_t(y, x), \tag{8}$$

where the state transition probabilities $P_t(y, x)$ are as discussed in *(4)*.

Optimal Policy: The optimal sampling policy can be obtained by solving the following set of non-linear Dynamic Programming equations [12],

$$\begin{aligned} V_t(\mathcal{J}_t) = &\sum_{x \in \mathcal{X}} \|x\|_1 \mathcal{J}_t(x) \\ &+ \min_{u \in [0, N]} \left(\mathbb{E}\ V_{t+1}(\mathcal{J}_{t+1}) + \lambda \mathbb{1}\{u \neq 0\}\right), \end{aligned} \tag{9}$$

$$V_T(\mathcal{J}) = \sum_{x \in \mathcal{X}} \|x\|_1 \mathcal{J}(x),\ \forall \mathcal{J} \in \Delta(\mathcal{X}), \tag{10}$$

where $\Delta(\mathcal{X})$ denotes simplex on $\mathcal{X}$ and $\mathcal{J}_t$ denotes representative belief state at time t. Optimal sampling action at time t in state $\mathcal{J}_t$ corresponds to a minimizer of the r.h.s. in the above equation. However, equations *(9)*, *(10)* are computationally intractable because the number of required computations is $O(2^N)$. Thus, we propose tractable provably approximate solutions next.

4.1.2 Hidden Markov Model

Since we might not observe the susceptibility graph $\mathcal{G}$, we can model it as a hidden Markov process [17]. Assume that the system evolves at discrete time-instants $t = 1, 2, \ldots, T$. Each slot could be of a duration equal to half hour, one hour, or one day. For each time slot t, we use $X_i(t)$ to denote the infection state of the i-th individual. In this section we slightly tweak the binary-valued state model that was described earlier, and allow it to assume 4 values. This allows us to design an algorithm that is more accurate. Thus, we let $X_i(t) = -1$ if

[2] So this problem is a partially observable MDP (POMDP), which is non-trivial to solve in general case.

individual i is not infected by the virus at time t, and $X_i(t) = 0$ if individual i is infected by the virus but cannot spread the virus, i.e., is in incubation, $X_i(t) = +1$ if individual i is infected by the virus and can spread the virus, and finally $X_i(t) = -2$ if they have recovered from the virus or have been isolated already. We call $X_i(t) = 0$ "inactive" and $X_i(t) = 1$ active.

The rest of the discussion in this section makes the following simplifying assumptions:

- States $X_i(t)$ and decisions $U_i(t)$ do not change within a slot.
- Upon becoming an active spreader, an individual can spread the disease only after the current time-slot ends. This might seem to be restrictive, but is justifiable since our modeling procedure already introduces "noise" due to erroneous tracing and testing.
- The spreading probability, denoted as p_s, is a constant that is independent of other parameters such as the values of the states, the number of days one has been infected, etc.
- After becoming an inactive infected person, at each time slot the individual becomes active with a probability equal to $p_{0,1}$.
- For an active infected person, for every time slot, this individual has a constant probability to get removed. We use $p_{1,-2}$ to denote this probability.
- For an individual at state $X_i(t) = a$, it has a constant probability $p_{a\to b}$ to be tested to be state $U_i(t) = b$.

Let $\mathcal{F}_t$ be the filtration generated by $(X_i(s), Y_i(s), i \in \mathcal{N}, s \le t)$. With the above assumptions in place, we can write the "dynamics" or transition probabilities governing $X(t) = \{X_i(t))\}_{i=1}^N$ as follows,

$$\begin{aligned}
&\mathbb{P}\{X_i(t+1) = 2 \mid \mathcal{F}_t\} \\
&\quad = \mathbb{1}_{X_i(t)=-2} + \mathbb{1}_{X_i(t)=1} \cdot p_{1,-2}, && (11)\\
&\mathbb{P}\{X_i(t+1) = 1 \mid \mathcal{F}_t\} \\
&\quad = \mathbb{1}_{X_i(t)=1}(1-p_{1,-2}) + \mathbb{1}_{X_i(t)=0} \cdot p_{0,1}, && (12)\\
&\mathbb{P}\{X_i(t+1) = 0 \mid \mathcal{F}_t\} \\
&\quad = \mathbb{1}_{X_i(t)=0} \cdot (1-p_{0,1}) \\
&\quad + \mathbb{1}_{X_i(t)=-1}\Big(1 - \prod_{j:(i,j)\in\mathcal{E}, X_j(t)=1} (1-p_s(t))\Big), && (13)\\
&\mathbb{P}\{X_i(t+1) = -1 \mid \mathcal{F}_t\} \\
&\quad = 1 - \mathbb{P}\{X_i(t+1) = 1, 0, \text{ or } -2 \mid \mathcal{F}_t\}, && (14)
\end{aligned}$$

and the dynamics or transition probabilities of $U(t) = \{U_i(t)\}_{i=1}^N$ as follows,

$$\begin{aligned}
&\mathbb{P}\{U_i(t) = 1 \mid \mathcal{F}_t\} \\
&\quad = \mathbb{1}_{U_i(t)\neq 0}\Big(\sum_{s=-1,0,1} \mathbb{1}_{X_i(t)=s} \cdot p_{s\to 1}\Big), && (15)\\
&\mathbb{P}\{U_i(t) = 1 \mid \mathcal{F}_t\} \\
&\quad = \mathbb{1}_{U_i(t)\neq 0}\Big(\sum_{s=-1,0,1} \mathbb{1}_{X_i(t)=s} \cdot p_{s\to -1}\Big), && (16)
\end{aligned}$$

where whether $U_i(t) = 0$ or not is determined by the tracing or testing algorithms.

Clearly, $X(t) = \{X_i(t)\}_{i\in\mathcal{N}}$ is a Markov process, and if we are provided with the values of $U_i(0), U_i(1), U_i(2), ...$, then our goal is to find the most likely values of $X(t)$. In order to do this, we might use Markov chain Monte Carlo (MCMC) algorithms such as Gibbs Sampling. Readers may refer to [27] for a review of MCMC algorithms.

4.1.3 Graph Embedding

Computing the infection probabilities of individuals directly will be computationally cumbersome, and we can utilize graph embedding [5] techniques in order to find suspicious infected individuals. These techniques map the nodes of a graph to points in $\mathbb{R}^d$, where d is a natural number. If the graph embedding algorithm is properly chosen, then if two points are close in the space $\mathbb{R}^d$ then they are also close in the susceptibility graph, so that the probability that the virus spreads from one individual to the other is high. Note that in the graph each node may have up to $|\mathcal{N}|$ edges, but in the embedded graph, each node only has d coordinates. Since the number of edges might be much more than d, performing computations with the embedded coordinates is much more efficient than directly working with the original graph.

5. SIMULATIONS

In this section, we use a simulation to indicate the necessities of contact tracing and building a contact graph. Contact tracing is an essential technique for finding potential infectious people. A commonly employed naive contact tracing technique is to trace and test only those who have had contact with a confirmed positive person. We call this simple and intuitively appealing contact tracing policy as Policy 1. However, this method may not be optimal, especially under circumstances when a sizeable proportion of the population is infected. To see why this might be the case, consider the scenario when two people are waiting to get tested. The first person had a close contact with a confirmed infected person, while the second person did not have any such close contact with a confirmed infected person; but did closely contact 500+ untested people (for example this person works in a supermarket). Policy 1 will suggest to us to test the first person; however, when a significant proportion of the population (e.g., 1%) are positive, in

expectation, the second person would have been in close contact with more than one infected people. Thus, taking the number of contacts with all people into consideration could significantly improve the tracing. The simulation in this section also confirms our claim.
These two policies can be mathematically described as follows.

- Policy 1: Fix a time frame of a duration say 2 weeks, and call the time duration composed of the previous two weeks as the "tracing window". At any given time we only take into account those contacts that occurred during the tracing window. Let p_s be the probability that the virus spreads from an infected person to a healthy person during a contact. We assume that p_s is constant and known. For any person s, given that person s contacted m confirmed infected persons during the tracing window, we use

  $$\mathbb{P}\left(\{s \text{ got infected}\}\right) = 1 - (1 - p_s)^m \tag{17}$$

  to measure the risk that person s is infected. We then choose to test those persons who have the highest probabilities of being infected.

- Policy 2: It additionally utilizes the contact graph, and checks the number of contacts of each person $s \in \mathcal{S}$. Hence, if n is the number of contacts of s in the tracing window, we let

  $$\mathbb{P}(\{s \text{ got infected}\}) = 1 - (1 - p_s)^m (1 - p_b p_s)^n \tag{18}$$

  in order to measure the risk that person s is infected. Over here, p_b is the so-called base infection probability, which can either be a constant or depend on the proportion of confirmed cases of the population (i.e., adaptive). Note that we are assuming that the infection status of these n people are unknown.

Our simulations results are depicted in Fig. 2, and clearly show the superior performance of Policy 2 as compared to that of Policy 1. More details on the simulation setup are as follows:

- The population size is 1000 people, and a single person (that is chosen uniformly at random from the population) is infected by the virus at day zero.

- Regarding the transmission capability of the virus, we assume that a person will be able to spread the virus 1 day after getting infected. Moreover, a person remains infected for at least 7 days. After this duration, on each day the person will change his state (to either isolated due to its symptoms, recovered, or deceased) with a probability of 1/7. Thus, in expectation, the virus lasts for 14 days. A person whose state has changed to removed, will not spread the virus or get infected.

- During any particular day, any two people in the city meet each other with a probability of 0.01. Thus, on average, a person meets around 10 people per day in expectation. When two people meet, and one of them is infected while the other is not, the virus will spread with a probability of $3/(14 \times 10)$; hence an infected person spreads the virus to an average of $R = 3$ people before being removed.

- Each day the community chooses 20 people to quarantine by using its policy. If quarantined persons are found to be infected, then they will be isolated until they are removed. Otherwise, they will be quarantined for 14 days, and then will be back to the normal schedule.

- We assume that the community as a whole knows all the contacts between all of its people, and whenever a person is removed the community gets to know this information at the beginning of the next day. Also, we assume that the spreading probability p_s is known to the community. We note that, with the knowledge of p_s and assuming the value of p_b, the community is able to compute Eq. *(17)* and Eq. *(18)*.

We perform simulations for 150 consecutive days, and record the cumulative infections in the population for the following 5 policies and parameters:

- No contact tracing of any sort is utilized.
- Policy 1 (Eq. *(17)*).
- Policy 2 with $p_b = 0.02$, where 0.02 is a well tuned value.
- Policy 2 with $p_b = 0.2$, where 0.2 is an example of a not well tuned value of p_b.
- Policy 2 with adaptive $p_b = N_{\text{rr}}/1000$, where "rr" denotes "recently removed" and N_{rr} means the number of people removed in the tracing window (i.e., the last two weeks).

Our simulation results are summarized in Fig. 2. We explicitly state the number of total infections in Table 1.

tracing policy	parameter p_b	total infections
no tracing	—	987
Policy 1	—	617
Policy 2	0.02	540
Policy 2	0.2	669
Policy 2	adaptive	569

Table 1 – Total number of infections of the virus under different tracing policies.

We summarize our findings as follows.

- Contact tracing and quarantine facilities are essential in order to control the spread of virus. Without these, the total number of infections are around 987,

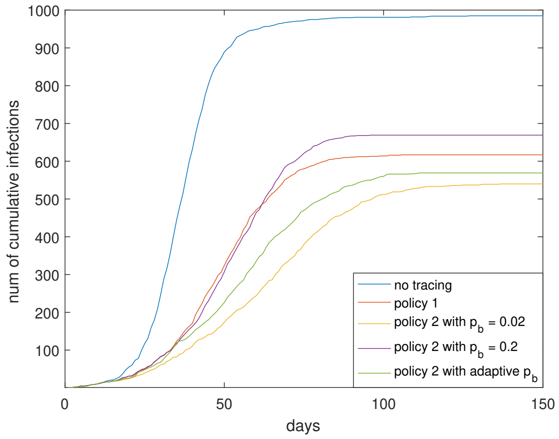


Fig. 2 – Simulation of the spreading of the virus under different tracing policies.

which is approximately the entire population. Even a simple contact tracing technique such as Policy 1, reduces the total infections to 617 (which is a 37% reduction). Thus, it is worth the effort to establish a system that can track the contacts of the people in order to better detect the infections.

- Compared to Policy 1 that only considers the contacts with confirmed infected people, Policy 2 also takes contacts with untested (but probable to be infected) people into account, and hence it has a better performance. This is clearly demonstrated in its superior performance in all of the three experiments, for example when $p_b = 0.02$, we get a gain of around 12%.

- When p_b is tuned properly (e.g., $p_b = 0.02$), Policy 2 performs better. However, the tuning effort is substantial, and might be deemed infeasible in practice. For instance, when $p_b = 0.2$, the performance of Policy 2 is worse than Policy 1. Thus, Policy 2 with adaptive values of p_b is a good option in practice.

- If we are to use only contact tracing and quarantine facilities, our performance is not very good. Even if tracing is possible for 2% of the population per day, the majority of the population will get infected after a few months. Hence, it is necessary to combine contact tracing and quarantines with other policies, e.g., avoiding contacts to reduce the number of contacts, and also wearing masks to reduce the virus spreading probability.

6. CONCLUSION

In this paper, we have provided a detailed blueprint on how to contain the spread of a pandemic by integrating the use of various wireless technologies with sequential learning based techniques. In particular, we show how different wireless technologies could be leveraged to improve contact tracing efforts and reduce the probabilities of detection and false alarms. The idea is to use possibly disparate wireless data streams for data collection, then integrate this data to improve coverage, reduce probabilities of errors and false alarms and create a *susceptibility* graph that could be used for intelligent testing. Based on this susceptibility graph, we show how to develop efficient sequential learning based algorithms in order to minimize the spread of the virus infection. Another contribution is that we develop provably optimal algorithmic solutions that rely upon the theory of partially observable Markov decision processes. In particular, we show that current contact tracing plus testing strategies that are aimed at identifying (and testing) individuals with the highest probability of infection are inefficient. Instead, we find that it is better to test those individuals whose expected impact on virus spread is the highest. We formulate the testing problem as a Partially Observable Markov Decision Process whose goal is to minimize the expected spread of the virus subject to testing capacity constraints. We provide efficient algorithmic solutions to this problem and show via numerical results that our solution substantially reduces the spread of the virus.

ACKNOWLEDGEMENT

This work was supported in part by NSF grants CNS-1618520 and CNS-2028547.

REFERENCES

[1] Covid-19 contact tracing: a briefing. `https://www.bmj.com/content/369/bmj.m1859`. (Accessed on 07/02/2020).

[2] The global economy is expected to shrink by 3% this year. `https://www.economist.com/graphic-detail/2020/04/14/the-global-economy-is-expected-to-shrink-by-3-this-year`.

[3] Robert – robust and privacy-preserving proximity tracing protocol. `https://github.com/ROBERT-proximity-tracing`. (Accessed on 09/17/2020).

[4] Eitan Altman. *Constrained Markov decision processes*, volume 7. CRC Press, 1999.

[5] Hongyun Cai, Vincent W Zheng, and Kevin Chen-Chuan Chang. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Transactions on Knowledge and Data Engineering*, 30(9):1616–1637, 2018.

[6] CDC. How to protect yourself & others. `https://www.cdc.gov/coronavirus/2019-ncov/prevent-getting-sick/prevention.html`.

[7] Aaqib Bashir Dar, Auqib Hamid Lone, Saniya Zahoor, Afshan Amin Khan, and Roohie Naaz. Applicability of mobile contact tracing in fighting pan-

demic (covid-19): Issues, challenges and solutions. *Computer Science Review*, page 100307, 2020.

[8] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

[9] Luca Ferretti, Chris Wymant, Michelle Kendall, Lele Zhao, Anel Nurtay, Lucie Abeler-Dörner, Michael Parker, David Bonsall, and Christophe Fraser. Quantifying sars-cov-2 transmission suggests epidemic control with digital contact tracing. *Science*, 368(6491), 2020.

[10] Stephen M Kissler, Christine Tedijanto, Edward Goldstein, Yonatan H Grad, and Marc Lipsitch. Projecting the transmission dynamics of sars-cov-2 through the postpandemic period. *Science*, 368(6493):860–868, 2020.

[11] Mirjam E Kretzschmar, Ganna Rozhnova, Martin CJ Bootsma, Michiel van Boven, Janneke HHM van de Wijgert, and Marc JM Bonten. Impact of delays on effectiveness of contact tracing strategies for covid-19: a modelling study. *The Lancet Public Health*, 5(8):e452–e459, 2020.

[12] Vikram Krishnamurthy. *Partially Observed Markov Decision Processes*. Cambridge University Press, 2016.

[13] Panqanamala Ramana Kumar and Pravin Varaiya. *Stochastic systems: Estimation, identification, and adaptive control*. SIAM, 2015.

[14] Douglas J Leith and Stephen Farrell. Coronavirus contact tracing: Evaluating the potential of using Bluetooth received signal strength for proximity detection. 2020.

[15] Yuxiang Luo, Cheng Zhang, Yunqi Zhang, Chaoshun Zuo, Dong Xuan, Zhiqiang Lin, Adam C. Champion, and Ness Shroff. Acoustic-turf: Acoustic-based privacy-preserving covid-19 contact tracing, 2020.

[16] Chunyi Peng, Guobin Shen, Yongguang Zhang, Yanlin Li, and Kun Tan. Beepbeep: a high accuracy acoustic ranging system using cots mobile devices. In *Proceedings of the 5th international conference on Embedded networked sensor systems*, pages 1–14, 2007.

[17] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[18] Albert N Shiryaev. *Optimal stopping rules*, volume 8. Springer Science & Business Media, 2007.

[19] Viktoriia Shubina, Sylvia Holcer, Michael Gould, and Elena Simona Lohan. Survey of decentralized solutions with mobile devices for user location tracking, proximity detection, and contact tracing in the covid-19 era. *Data*, 5(4):87, 2020.

[20] Rahul Singh, Abhishek Gupta, and Ness B Shroff. Learning in Markov decision processes under constraints. *arXiv preprint arXiv:2002.12435*, 2020.

[21] Richard D Smallwood and Edward J Sondik. The optimal control of partially observable Markov processes over a finite horizon. *Operations research*, 21(5):1071–1088, 1973.

[22] Edward J Sondik. The optimal control of partially observable markov processes over the infinite horizon: Discounted costs. *Operations research*, 26(2):282–304, 1978.

[23] Qiang Tang. Another look at privacy-preserving automated contact tracing. *arXiv preprint arXiv:2010.13462*, 2020.

[24] Qiang Tang. Privacy-preserving contact tracing: current solutions and open questions. *arXiv preprint arXiv:2004.06818*, 2020.

[25] Amee Trivedi and Deepak Vasisht. Digital contact tracing: technologies, shortcomings, and the path forward. *ACM SIGCOMM Computer Communication Review*, 50(4):75–81, 2020.

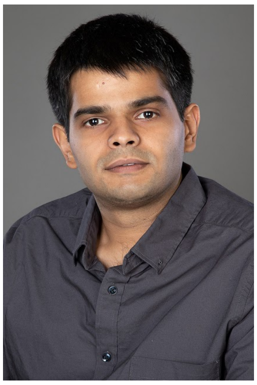
[26] Amee Trivedi, Camellia Zakaria, Rajesh Balan, and Prashant Shenoy. Wifitrace: Network-based contact tracing for infectious diseases using passive wifi sensing, 2020.

[27] Don Van Ravenzwaaij, Pete Cassey, and Scott D Brown. A simple introduction to markov chain monte–carlo sampling. *Psychonomic bulletin & review*, 25(1):143–154, 2018.

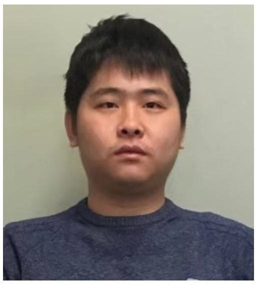
[28] Haohuang Wen, Qingchuan Zhao, Zhiqiang Lin, Dong Xuan, and Ness Shroff. A study of the privacy of covid-19 contact tracing apps. In *International Conference on Security and Privacy in Communication Networks*, 2020.

[29] Qingchuan Zhao, Haohuang Wen, Zhiqiang Lin, Dong Xuan, and Ness Shroff. On the accuracy of measured proximity of Bluetooth-based contact tracing apps. In *International Conference on Security and Privacy in Communication Networks*, 2020.

AUTHORS



Rahul Singh received his B. Tech. degree in Electrical Engineering from Indian Institute of Technology, Kanpur, India, in 2009, M.Sc. degree in Electrical Engineering from University of Notre Dame, South Bend, IN, in 2011, and his Ph.D. degree in Electrical and Computer Engineering from the Department of Electrical and Computer Engineering Texas A&M University, College Station, TX, in 2015. Currently he is an Assistant Professor at the Department of Electrical Communication Engineering, the Indian Institute of Science. His research interests include stochastic control, machine learning, applied probability, networks and large-scale complex cyber physical systems. He has earlier worked as a Postdoctoral Researcher at the Laboratory for Information Decision Systems (LIDS), Massachusetts Institute of Technology, at Intel, Santa Clara as a Machine Learning Engineer, and as a Data Scientist at Encored Technologies.



Wenbo Ren received his B.S. degree in Electronic Information Engineering from University of Science and Technology of China in 2016. Since then, he has been a Ph.D. student in Computer Science and Engineering at The Ohio State University. His research interests include machine learning theory and algorithms, with a focus on ranking, active learning, sequential decision making, recommender systems, etc. He has made publications or served as reviewers for top-tier machine learning conferences including ICML, NeurIPS, and AISTATS.

Fang Liu received his B.S. degree in Information Engineering from Shanghai Jiao Tong University, China, and his Ph.D. degree in Electrical and Computer Engineering from The Ohio State University. In 2020, he joined Facebook Inc. as a research scientist. His research interests include machine learning theory and systems, with a focus on sequential decision making under uncertainty. Dr. Liu has served as a program committee member or reviewer for many top-ranked machine learning conferences, such as ICML, NeurIPS, ICLR, and AAAI.

Dong Xuan received his B.S. and M.S. degrees in Electronic Engineering from Shanghai Jiao Tong University, China, and his Ph.D. degree in Computer Engineering from Texas A&M University. In 2001, he joined the Department of Computer Science and Engineering at The Ohio State University, where he is currently a full professor. He was on the faculty of Electronic Engineering at Shanghai Jiao Tong University from 1993 to 1998 and a visiting scholar in Computer Science at City University of Hong Kong from 1997 to 1998. He was a research assistant/associate in the Real-Time Systems group in the Department of Computer Science at Texas A&M University from 1998 to 2001. His research interests include computer networking and mobile systems. Dr. Xuan has served as an editor for IEEE TPDS and ACM ToSN, and has been a TPC member for a number of IEEE and ACM flagship conferences such as IEEE INFOCOM, ICDCS, ICNP, and ACM MobiHoc. He is a recipient of the National Science Foundation CAREER Award and the College of Engineering Lumley Research Award at The Ohio State University.

Zhiqiang Lin received his Ph.D. degree in Computer Science from Purdue University in 2011. He was a faculty member in the Computer Science Department at University of Texas at Dallas between 2011 and 2017. Since 2018, he has been a faculty member in the Department of Computer Science and Engineering at Ohio State University. His research interests are systems and software security, with an emphasis on binary analysis and vulnerability discovery. Dr. Lin is currently an associate editor of ACM TOPS, IEEE TDSC, and IEEE TMC. He has also served as a TPC member for numerous systems security conferences including IEEE S&P, ACM CCS, USENIX Security, and NDSS. Dr. Lin received the AFOSR Young Investigator award (2014) and the NSF CAREER award (2015). He is also a recipient of VMware Faculty Research Award (2012) and the Outstanding Junior Faculty Research Award at UT Dallas (2013).

Ness B. Shroff received his Ph.D. degree in electrical engineering from Columbia University in 1994. He joined Purdue University immediately thereafter as an Assistant Professor with the School of Electrical and Computer Engineering. At Purdue, he became a Full Professor of ECE and the director of a university-wide center on wireless systems and applications in 2004. In 2007, he joined The Ohio State University, where he holds the Ohio Eminent Scholar Endowed Chair in networking and communications, in the departments of ECE and CSE. He holds or has held visiting (chaired) professor positions at Tsinghua University, Beijing, China, Shanghai Jiaotong University, Shanghai, China, and IIT Bombay, Mumbai, India. He has received numerous best paper awards for his research and was listed in Thomson Reuters' on The World's Most Influential Scientific Minds, and has been noted as a Highly Cited Researcher by Thomson Reuters in 2014 and 2015. He has served on numerous editorial boards and chaired various major conferences and workshops. He currently serves as the steering committee chair for ACM Mobihoc, and Editor in Chief of the IEEE/ACM Transactions on Networking. He received the IEEE INFOCOM Achievement Award for seminal contributions to scheduling and resource allocation in wireless networks.