

# Modelling Memory for Individual Re-identification in Decentralised Mobile Contact Tracing Applications

LUCA BEDOGNI\*, University of Modena and Reggio Emilia, Italy  
 SHAKILA KHAN RUMI\*, RMIT University, Australia  
 FLORA D. SALIM, RMIT University, Australia

In 2020 the coronavirus outbreak changed the lives of people worldwide. After an initial time period in which it was unclear how to battle the virus, social distancing has been recognised globally as an effective method to mitigate the disease spread. This called for technological tools such as Mobile Contact Tracing Applications (MCTA), which are used to digitally trace contacts among people, and in case a positive case is found, people with the application installed which had been in contact will be notified. De-centralised MCTA may suffer from a novel kind of privacy attack, based on the memory of the human beings, which upon notification of the application can identify who is the positive individual responsible for the notification. Our results show that it is indeed possible to identify positive people among the group of contacts of a human being, and this is even easier when the sociability of the positive individual is low. In practice, our simulation results show that identification can be made with an accuracy of more than 90% depending on the scenario. We also provide three mitigation strategies which can be implemented in de-centralised MCTA and analyse which of the three are more effective in limiting this novel kind of attack.

CCS Concepts: • **Security and privacy** → **Privacy protections; Social aspects of security and privacy**; • **Human-centered computing** → **Ubiquitous and mobile computing design and evaluation methods**.

Additional Key Words and Phrases: datasets, privacy preserving, digital contact tracing, mobile computing

## ACM Reference Format:

Luca Bedogni, Shakila Khan Rumi, and Flora D. Salim. 2020. Modelling Memory for Individual Re-identification in Decentralised Mobile Contact Tracing Applications. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 4 (December 2020), 20 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

## 1 INTRODUCTION

The coronavirus shocked the world in early 2020, when the COVID-19 illness affected millions of people, with hundred of thousands of deaths. The disease spread worldwide, forcing the World Health Organization to declare it a global pandemic on March 11th 2020 [32]. The coronavirus is transmitted among individuals through air droplets which are released whenever an infected person sneezes, coughs or exhales, and can therefore infect other neighboring people. To limit the contagion, governments have issued several regulations limiting the mobility of people, to make contacts with infected people as rare as possible, hence slowing down the disease

\*Both authors contributed equally to this research.

Authors' addresses: Luca Bedogni, luca.bedogni@unimore.it, University of Modena and Reggio Emilia, P.O. Box 1212, Modena, Italy, 43017-6221; Shakila Khan Rumi, shakilakhan.rumi@rmit.edu.au, RMIT University, Melbourne, VIC, Australia, 3001; Flora D. Salim, flora.salim@rmit.edu.au, RMIT University, Melbourne, VIC, Australia, 3001.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2020 Association for Computing Machinery.

2474-9567/2020/12-ART \$15.00

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

| Kind           | Match     | Data on Server                    | Example countries  | References              |
|----------------|-----------|-----------------------------------|--------------------|-------------------------|
| Centralized    | On Server | Possibly complete data            | South Korea, China | [26]                    |
| Hybrid         | On Server | Only that of positive individuals | Australia          | [22]                    |
| De-centralized | On Device | Only that of positive individuals | Germany, Italy     | [3, 11, 12, 17, 21, 38] |

Table 1. Different MCTA architectures

spread. Until a vaccine is registered and distributed to the people, social distancing and reducing contacts among individuals is the only effective way to contain the transmission of the virus [36].

To gradually return to normal mobility and social interactions, governments and health operators have the need to trace the contacts of people, so that whenever an hotbed is registered, it can promptly test and eventually put into quarantine affected people. Apart from manual contact tracing, several proposals have been developed concerning Mobile Contact Tracing Applications (MCTA), which digitally trace the contacts among individuals. In case someone of the contacts is later tested as positive, a notification is sent to the previous contacts so that they are warned and can take self-isolation measures, or being tested to confirm or negate the coronavirus infection [13, 15]. Principally two technologies were envisioned: GPS has the possibility to provide data about the location of the contact and can better understand where hotbeds may be arising; the second technology is Bluetooth Low Energy (BLE), which can detect whether two devices are in close proximity, hence recording the contact between the two individuals owners of the device. The GPS is seldom used mainly due to concerns about the privacy of the users, hence BLE is the one which is used in most countries [12], although with some concerns about its accuracy in determining the exact distance between two devices [28]. We summarize the key differences of the different proposals in Table 1, and we describe them more in details in Section 2.1.

Currently these applications are effectively utilized in several countries, and are helping to fight the disease spread by rapidly notifying people when in contact with a positive individual. However, there are also several concerns regarding the privacy issues which can arise due to their use [37]. Attackers may install cameras in places where individuals transit, along with BLE scanners. Then this data can be correlated when receiving a notification to understand who is the individual responsible for that <sup>1</sup>, by matching the time at which a specific code was heard together with the video surveillance. Clearly, apart from having the social contacts of people released, the possibility to identify who is the positive individual responsible for infecting other people is a serious concern which may flow into riots or violence [20].

Moreover, it also exists a tradeoff between the delay the server has to wait to send the notification and the privacy issues for the end user. In fact, aggregating data from multiple users together may offer better protection against attacks, at the cost of delaying the distribution of the codes of the positive individual, hence possibly not notifying in time people which may have been infected [8].

In this work we analyze a novel kind of attack, based on the memory of the individual in de-centralised MCTA. In detail, what we show is that based on the collected codes and the memory of the individuals, it is possible to reconstruct the social trace, hence identifying the positive individuals with which someone has been in contact. This kind of attack is even easier to be performed when constrained mobility is in place, such as during lockdowns or immediately after that, since the set of people someone sees in a day is low, making it easier to identify who is the positive individual. In fact, what we also show is that the sociability level of the individual, defined as the number of unique users seen in a time window in which codes remain static, is a key parameter in understanding whether this attack can take place and to what extent.

We show the potential and the risks associated with this kind of attack using two different datasets built with real world data, pertaining BLE traces collected in two different places. Our results indicate that this memory

<sup>1</sup><https://www.wired.com/story/apple-google-contact-tracing-strengths-weaknesses/>

based attack can identify positive users based on the memory of the individuals, and it also become easier if the individual has a low sociability, such as seeing co-workers and relatives and not many other people.

At the end of our work we also give some recommendations on how to limit the memory based identification of positive individuals, which can be summarized as: (i) send multiple positive codes together; (ii) send as few information as possible to notify the user, but nothing more; (iii) adopt techniques to increase the sociability of the individuals.

The rest of this paper is structured as follows: Section 2 discusses related works from literature on MCTA and privacy issues; Section 3 describes the two datasets we used for our experiments; Section 4 presents our methodology, detailing how the memory based attack can be performed; Section 5 presents the results we obtained; Section 6 concludes this paper, and discusses recommendations on how to better protect MCTA against these kind of attacks.

## 2 RELATED WORK

### 2.1 Background

To date, three different MCTA architectures were proposed: *Centralized MCTA*, in which the contacts are stored on a central server which performs the matches based on the vicinity and eventually sends the notification to the users; *Hybrid MCTA*, in which data is kept on the local device of the individual, but upon being tested positive the social trace is uploaded to the server, which can notify the people who had been previously in contact; *Decentralised MCTA*, in which data is kept on the local device of the individual, devices exchange random codes which change every 10-20 minutes, and upon being positive tested the codes from the previous days are sent to all the people with the application installed, and the match is eventually performed on the local device. Worldwide governments have adopted one of the aforementioned architectures, though many decided to adopt decentralised MCTA, as it is the one which seems to offer the best privacy for users [11, 12]. In fact, in centralised MCTA if someone has access to the data it can easily reconstruct the social interactions of any individual, with possibly dangerous privacy issues, and in Hybrid MCTA each individual is bound with a static code, hence raising again potential privacy problems.

Governments worldwide released recommendations and guidelines which help application developers and consortiums to develop and distribute MCTA. For instance the European Union released its own official document [10], stating that these apps must be (i) voluntarily installed; (ii) dismantled as soon as no longer needed; (iii) use location data only if necessary, as the purpose of MCTA is not to track people. Parallel to governments advances, also smartphone makers such as Google and Apple jointly worked together to release the Privacy Preserving Decentralized Framework inspired by the work on DP-3T [38], to monitor contacts based on Bluetooth Low Energy (BLE) [3, 21]. In the DP-3T framework, the BLE device broadcasts a temporary and random identifier to neighboring devices, and at the same time receives codes from other smartphones. All its own personal identifier, along with the codes received by others, are stored locally on the smartphone and never sent to any third party. Upon being tested positive, the health operator releases a code to the positive individual, which can upload to a server all its past codes to a central server, which will eventually aggregate those together and send them to all the devices with the app installed. All the data in the server is removed after 14 days [38]. There are many countries which adopted this framework, with minor adjustments, such as ‘Corona-Warn-App’ in Germany [16], ‘Immuni’ in Italy [17], ‘SwissCovid App’ in Switzerland [18]. Additional applications along with their architectures and model can be found in [11, 12].

### 2.2 State of the art

MCTA is a conventional method which has been successfully used in past to control emerging epidemic, e.g. SARS and Ebola outbreak effectively [25, 35]. Several technologies like Wifi, BLE, GPS, and social networks

were used to measure the spatial proximity between multiple users in MCTA. However, many of these MCTAs face various privacy related issues which were studied in the past and also rouse when MCTA was proposed to digitally trace contact during the COVID-19 outbreak.

In the past, few research were devoted to ensure privacy in MCTAs for disease surveillance or outbreak monitoring. A privacy-preserving contact tracing framework named efficient privacy-preserving contact tracing for infection detection (EPIC) was proposed in [2]. Here, the proximity contacts were identified with hybrid wireless and localization technology and matched over encrypted contents. In [29, 42], the authors proposed security protocols to ensure privacy preserved user profile matching in mobile social networking.

During the COVID-19 pandemic, several countries have deployed MCTA nationally to control the COVID-19 transmission with limited lock-down strategies [1]. A worldwide discussion and research work started, to propose protocols which could serve this purpose but without affecting the privacy of the users.

Singapore is the first country, which deployed an MCTA called 'TraceTogether' nationally to mitigate COVID-19 transmission [19]. 'TraceTogether' ensures the privacy of the contacts using a protocol, BlueTrace [4], which builds upon 4 fundamental pillars: (i) it limits the collection of personally-identifiable information, as the only information collected is a phone number; (ii) de-centralised, as the data is kept on the mobile devices; (iii) the code mobile devices send changes over time; (iv) users have control of their data, meaning that whenever they withdraw consent, all the data stored in the health authority is deleted. However in this protocol the infected users upload their traces to server which can be accessed by health authority/government. If any attack happens in the server, the traces of the infected person can be breached in this system. Different approaches have been leveraged in [9] to enhance the privacy in TraceTogether. They have discussed the privacy based on three different concepts: 1)) privacy from snoopers; 2) privacy from contacts; and 3) privacy from authorities.

An additional privacy preserving protocol has been proposed in [5] to guarantee privacy based on the above mentioned notions. The DP-3T framework is the most popular framework among the European countries[11]. In this framework, the id of the infected user is broadcasted to other contacts, hence the privacy of the infected person is not preserved to his proximity contacts [38].

In [30], the authors used zero knowledge  $\Sigma$ -protocol based contact tracing app to track the contacts in privacy-preserved way in DP-3T framework. This protocol enhances the privacy of an infected user from the health authority/governments and his close contacts. Here, the authors only focused on network-based or cryptographic attack. Two different exposure notification schemes, ReBabblers and CleverParrot were introduced in [7] to ensure privacy and integrity in decentralized MCTA. The type of attack that has been addressed in this paper is adversarial attack in users' BLE chips and central server. Another privacy focused decentralized MCTA was proposed in [6], which used a distributed hash table for contact tracing. Here, two users exchange encrypted and signed message to notify about the infection. The health authority verify the infection status of a positive user publicly using a blind signature mechanism. Three different attacker models based on malicious act of health authority, infected person and not infected person were discussed and defended in this article.

However, few of the above proposal consider this kind of attack, although not mitigating it. For instance the DP-3T framework mentions that individuals can be re-identified by other people with the help of surveillance cameras and similar tools, concluding that this kind of attack is "inherent to any proximity based notification system". In our work we present this attack in detail, and also propose some mitigation techniques which would make this kind of attack less harmful for the users.

Finally efforts were also made to understand whether BLE could actually determine the distance at which two devices stand. [28] shows that BLE distance estimation is indeed a challenging task, due to considerable RSSI variations as the orientation of the device change, or whether it is placed inside a pocket or outside. The final outcome of [28] is that distance estimation only leveraging BLE alone would be hard to realize, unless other social protocols such as placing phones on tables during meeting take place. This is also similar to the proposal found in [24], which shows a similar scenario and proposes instead an external wearable device, which would constrain

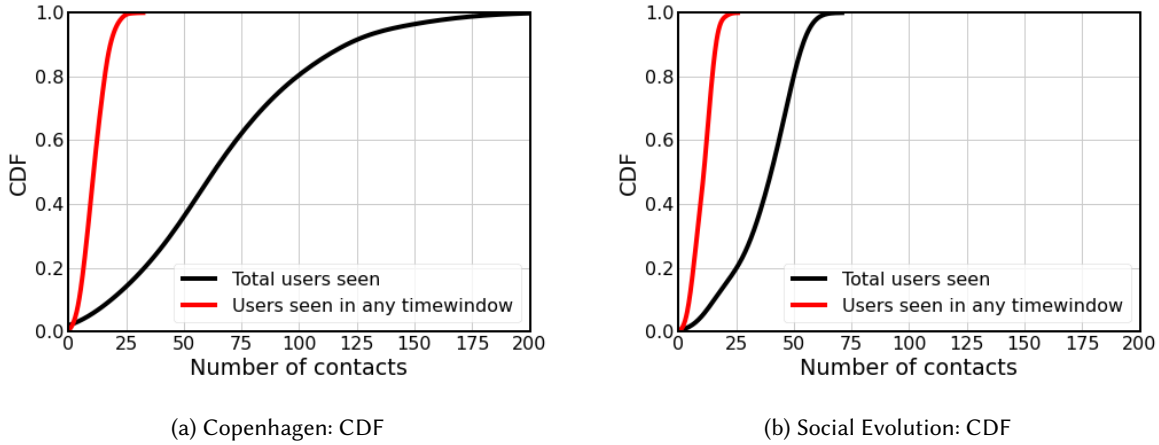


Fig. 1. Cumulative distribution function for the two datasets we used in this study. Users seen in any timewindow are similar in the two datasets, while the total number of users is much bigger in the Copenhagen dataset.

the limits of the smartphone based solution. Nevertheless, the accuracy of the BLE is still a major concern when developing MCTA leveraging this technology.

### 3 DATASET DESCRIPTION

We have collected Bluetooth proximity data from two different studies to model and analyze the decentralized MCTA. It includes Copenhagen Networks Study [34] and Social Evaluation Data [31]. The description of both datasets are as follows:

#### 3.1 Copenhagen Network Study

Copenhagen Network Study collected Bluetooth proximity data via smartphone with other data such as the network of phone call, the network of text messages, facebook friendships and others. The purpose of this study was to have a proper understanding of social systems. The dataset was collected for 706 undergraduate students at the Technical University of Denmark over one month period in 2013. The smart phone with each student in the experiment was configured as discoverable at all time. Each device scans for nearby device in every 5 minutes. Each response contains the unique identifier of all discoverable devices and the corresponding RSSI signal.

#### 3.2 Social Evolution Data

Social Network Evolution experiment collected Bluetooth proximity data using phone sensors with other datasets such as the network of phone call, the network of text messages, Wi-Fi connection and others. The purpose of this data collection was to study the adoption of different attributes such as political opinions, diet, exercise, obesity, eating habits, epidemiological contagion, depression and stress, dorm political issues, interpersonal relationships, and privacy by new dorm students. In this study, Bluetooth proximity data was collected for 70 undergraduate students at an undergraduate university hall between September 2008 to June 2010. Each instance of this dataset contains a sender user id, receiver user id and timestamp of the discovery. It also contains the probability of two persons being in the same floor. The probability is measured based on the RSSI signal strength.

### 3.3 Comparison

Figure 1 shows the Cumulative Distribution Function (CDF) for the two datasets with respect to the sociability of the users. In particular we plot with a red line the maximum number of users seen in any time window, while black lines refer to the total number of users seen in the whole measurement period. The maximum number of users in any timewindow highlights how crowded are the places the user visits, since it measures the number of users seen in a short time. The total number of users seen reflects instead how vast is the set of users seen during the whole measurement time, which in our case is set to 2 weeks. Although for the users seen in any timewindow both datasets show an almost identical behavior, regarding the total number of users the Social Evolution dataset shows a significantly lower number. This translates into a smaller circle of relative, or in other words people in the Social Evolution dataset tend to visit the same people more compared to the Copenhagen dataset.

These two datasets reflect well two different kind of mobility: while the Copenhagen dataset shows more users in the measurements, meaning a more diverse mobility, the Social Evolution datasets shows instead a more constrained and limited mobility, in which people is inclined to see similar people each day. With these two datasets it is then possible to study the impact of the memory based identification attack under different scenarios, which can better generalize our results.

## 4 METHODOLOGY

In this section we describe how we model the memory based identification attack. In our scenario, the adversary adheres to the honest-but-curious (HBC) model, in which users are participants of the communication, and do not deviate from the defined protocol. However, they will actively observe and try to understand all the possible information from the messages they receive [33]. In fact, in our scenario we do not look at users who crack the protocol or try to eavesdrop information not intended for them, but only look at what they purposely receive to match it with their visual encounters, eventually identifying positive and negative cases.

Let  $A$  be an MCTA leveraging a decentralized protocol, hence receiving every time window  $T$  codes from neighboring devices, which are then stored in  $A$  along with the time at which the code is received. At the same time, user  $U'$  can also keep track users who are close to her, either by writing them down or simply remembering encounters. Upon user  $U''$  being tested positive, the codes are uploaded to a central platform which eventually broadcast them to all users with the application installed, including application  $A$  of user  $U'$ . Thus  $U'$  can build a graph associating all the codes to the specific time window in which it was received, along with all the users encountered in the same time window. Of course at this time it is not possible to assign a specific code to a specific individual, however with simple pruning operations in the graph it is possible to simplify it, eventually identifying positive and negative individuals.

In Figure 2 we outline how attacks can be performed on sequences of BLE codes recorded. The sequence of BLE measurements of user *Alice* is built as Figure 2a shows:  $t_0$ ,  $t_1$ ,  $t_2$  and  $t_3$  represent 4 different time windows in which BLE codes remain the same for any user. During each of these time window, the code is univocal to each user, and it is broadcasted to neighboring devices. Hence *Alice* collects codes from different users, knowing only the time at which she collected them. Moreover, she does not know whether measurements from different time windows belong to the same person or not, since BLE codes are generated randomly, hence it is not possible to directly correlate measurements recorded in any two time windows. Correspondingly, *Alice* can also remember, or write down, users she sees in different time windows, which can be later correlated to the positive codes received. Clearly remembering or recording many users would be impractical, however also given the restricted mobility behavior which can be expected with such an application in place, the set of users *Alice* is going to see would remain constant for a large amount of time. Hence for the sake of this example, it is safe to say that in different time windows *Alice* is always together with *Bob* and possibly other users. Hence, if *Bob* is tested positive, his BLE codes for the past days are sent to a central server which will forward all these codes to the



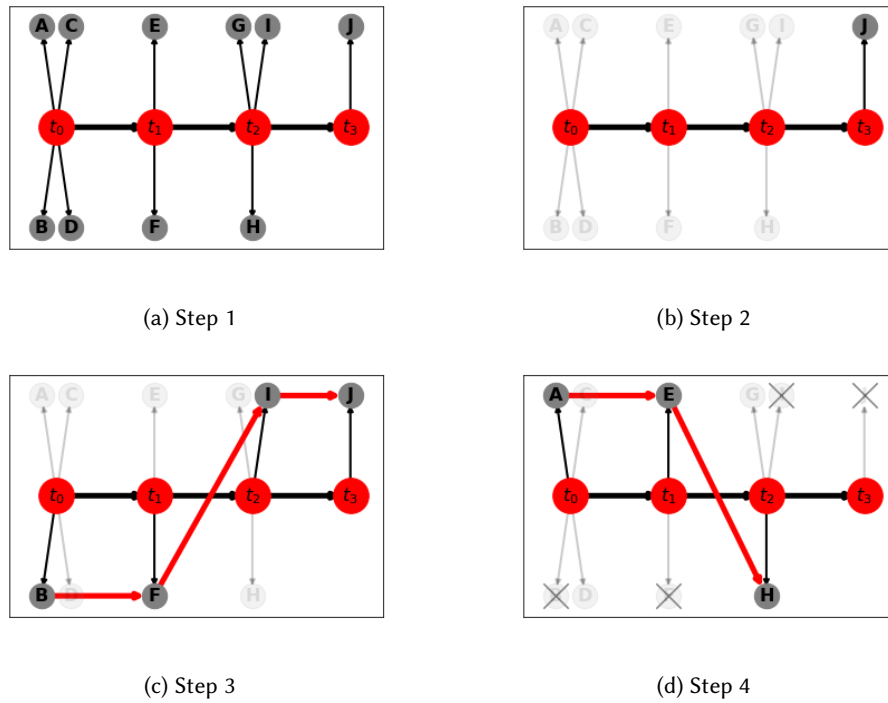


Fig. 2. Issues in Privacy Preserving Contact Tracing. In Figure 2a we show how the BLE traces is built, as each red node represents a different time window connected to every code received in  $t_0$ , and to the subsequent time window. Figure 2b shows that in case a single positive code **J** from a time window is received, then it is possible to uniquely identify the user who sent it. Figure 2c shows instead that if multiple messages received by the user who sent **J** are received, then it is also possible to correlate them, possibly using multiple devices which share their BLE data. Finally Figure 2d consequently show that this behavior may rouse a chain effect, in which also other users may be vulnerable to the identification shown in Figure 2b, in this case the user sending BLE measurements **A**, **E** and **H**.

user which installed the MCTA. It is straightforward to note that in this example *Alice* can immediately identify that *Bob* is positive, since she received the **J** code and *Bob* was the only person she met during  $t_3$ . However, *Alice* would not receive just code **J**, but all the codes from *Bob*. This is needed since the central server does not know that *Alice* received code **J**, so it has to send all the measurement from user *Bob*. Hence, in case *Alice* receives all the messages from a single user, she can quickly correlate all those measurements together as Figure 2c shows. In other words, if she receives all the positive codes sent by a single user, she just needs to correlate them to a specific individual in one time window, and she can identify *Bob*. We also note that this may also happen if different mobile devices are used to collect information, which is then merged together, though this is out of the scope of this paper. A final issue is presented in Figure 2d, which raises a chain reaction upon identifying and not considering codes identified upon receiving *Bob* codes. Since *Bob* sent codes **B**, **F**, **I** and **J**, then those codes cannot belong to any other user. Hence, in  $t_1$  there is only one code left (**E**), which in case its user is tested positive will be sent again to other users. This issue is similar to that presented in Figure 2b, in which only one code is received during a time window.

To better illustrate how the memory attack identification can be performed we provide a simple example. *Alice*, *Bob* and *Carl* meet at a certain location, and they stay there for a time equal to  $2 * t$  time windows in total. However, after the first time window  $t$  *Carl* leaves, leaving *Alice* and *Bob* together. Suppose that during the first  $t$  window *Alice* sent code **A**, *Bob* sent code **B** and *Carl* sent code **C**. During the subsequent time window, *Alice* sent code **AA**, *Bob* sent code **BB** and *Carl* sent code **CC**, though it is not received by neither *Alice* or *Bob*. At this point, if code **C** from timewindow  $t$  is received by the MCTA, no one is able to understand who has been tested positive, because it cannot be uniquely correlated to a single user. However if the notification message from the server also contains code **CC**, then **Alice** can immediately understand that the positive individual is *Carl*. This can be easily done by *Alice* since in the second time window she was alone with *Bob* and did not received from the server the **BB** code *Bob* sent out during the second time window. This also exclude *Bob* from being positive in the first time window, hence the code she received must have been sent out by *Carl*. Moreover, *Bob* can also understand that *Carl* was tested positive, since *Bob* has not received code **AA**, hence *Alice* has not been tested positive, eventually leaving only *Carl* as the possible candidate for being tested positive.

#### 4.1 Modelling contacts graph

In this section we describe in the detail the model we use within this study.

Let  $G = (V, E)$  be a graph with set of vertices  $V$  and edges  $E$ . A vertex can be of 3 kinds: T (time window), C (code) or U (user). For each time window  $t$ , the user adds to  $V$  all the codes received during  $t$ , and all the users seen. As we already stated, this can be done either by manually recording encounters, or can be modeled by leveraging memory from individuals, as we will later describe in Section 4.3.

Edges are built as follows:

$$\forall t_i, (t_{i-1}, t_i) \in E, \quad (1)$$

which connects every time  $t_i$  window with its predecessor  $t_{i-1}$ . Codes are then connected to the time window in which they have been received by the MCTA:

$$\forall c_j \in t_i, (t_i, c_j) \in E, \quad (2)$$

which connects every received code  $c_j$  with the time window  $t_i$  in which such code has been recorded. Finally, users are connected to all the codes for any time window in which that user has been seen:

$$\forall u_k, c_j \in t_i, (c_j, u_k) \in E, \quad (3)$$

which connects every user  $u_k$  seen in time window  $t_i$  with each code  $c_j$  received during such window. In this last step, we basically assign potentially each code to any user, as we do not know which user has generated which code. These edges can be pruned accordingly to a certain set of conditions, which we will describe in Section 4.2. As a final note, the more the edges between users and codes, the more it is challenging to prune them and eventually identify positive and negative users. Again, this correlates to the sociability of the individual: in case a user sees many different people, then the recorded codes may belong to many different users; in case the set of people seen is low, fewer connections can be made, hence making it easier to uniquely identify the state of any individual.

#### 4.2 Graph operations

In this section we show 3 operations which can be performed on the graph in order to simplify it. These 3 operations are denoted as:

- $I_p$ : Identify positives
- $I_n$ : Identify negatives
- $P$ : Prune edges



The  $I_p$  operation builds a set of user which are identified as positives. Whenever we are able to identify an individual  $i$  as positive, it means that the user  $j$  have received a positive code in every time window in which  $i$  and  $j$  were together. In its simplest case this can be done by looking for all the users which are connected only to positive codes. In other words, if in any  $T$  time window we can find a  $C$  node connected to only one  $U$  node, then such user must be positive. This operation can be further extended also to multiple users: a set of users is identified as positive if for any time window, the number of positive codes observed is equal to the total number of users seen during time window minus the number of those users which are identified has been already identified as negative. More formally this can be defined as follows:

$$\forall t, u_i \text{ is positive iff } |c_p(t)| = |u(t)| - |u_n(t)|, \quad (4)$$

where  $c_p(t)$  is the set of positive codes at time  $t$ ,  $u(t)$  is the set of users seen at time  $t$ , and  $u_n(t)$  is the set of negative users at time  $t$ .

Simply put, it checks whether the number of positive codes in a given time slot is equal to the number of users of that time slot which are already marked as negatives. If so, then all the other remaining users are marked as positives. The exact same operation can be applied also for the  $I_n$  operation, by changing the count of positive users with the count of the negative users. More formally this is defined as:

$$\forall t, u_i \text{ is negative iff } |c_n(t)| = |u(t)| - |u_p(t)|. \quad (5)$$

Identifying a user in any time window also enables to leverage such information in other time windows. For instance, in time windows in which there are two users, if one has already been identified either as positive or negative, then also the other user can be directly identified with the same methodology we just discussed and so on also for more numerous time windows. This is also made possible by the  $P$  operation which we will discuss later in this section.

Clearly, identification is not always possible, since there may be time windows in which it is not possible to discriminate between negative and positive users. In this scenario, we leverage the  $P$  operation, which simplifies the graph by pruning edges which are no longer valid considering the set of already identified positive and negative users.

The  $P$  operation removes edges which are not valid anymore due to newly identified positive or negative users. It removes all the edges of a positive code connecting it to a negative user, and all the edges of a negative code connecting it to a positive user. This operation is necessary since during graph building all codes were connected to all users as Equation 3 shown. However, after identifying positive and negative users, some of these connections may be removed, to simplify the graph and possibly leverage the identification of further users.

These 3 steps are iterated multiple times, until no change in the graph is observe. This happens because  $P$  may enable  $I_p$  and  $I_n$  to identify users thanks to a simpler graph, and the two identification operation allow  $P$  to prune more edges.

### 4.3 Modeling memory

In this section we describe how our model accounts for the memory of the individuals. Clearly, any human being can remember more precisely recent events compared to those which happened days before [40].

The recognition memory of individuals has been tested by several works, which show what is the probability of an individual to remember things happened in the past [27] [41]. For instance [41] reveals that users are able to recognize pictures seen one week before with an accuracy of around 90%. [27] is closer to what we need for our work, as it also measures the ability to remember things happened 2 weeks before and in intermediate time periods. Accuracies are similar to [27], stating a 75% accuracy of remembering things happened 2 weeks before, 80% at 1 week and roughly 90% at 1 day. Regardless of the exact percentage of remembering things in the past, which may also vary from individual to individual, it has been shown that human memory is triggered by specific

events, which make human being able to recognize specific things or scenarios which happened in the past [39] [23].

For our work we leverage the data from [41] and use them as probabilities of remembering or not a specific user in a given time window. Given the graph  $G$ , we remove edges according to the probabilities themselves, so that recent days have more information compared to events more distant in time. More precisely, after having built  $G$ , we keep every user in any timewindow with a probability  $p$ , where  $p$  is defined according to the following probabilities: to perform a worst case analysis, we define  $p = 0.75$  for events happened between two weeks and one week before receiving the notification,  $p = 0.80$  between 1 week and one day, and a probability  $p = 0.90$  for events happened in the current day [27][41].

We also want to note that additional tools may also be envisioned for these kind of attacks. For instance, people may also keep track of people encountered in different time windows. This can be realized by simply writing down who has been seen or not, or can even be realized by specialized apps which can help to keep track of encounters. Moreover, in places where CCTV is in place, recordings may also raise the probabilities we just presented, as it would be possible to look at encounters in specific times, if a positive code from such time window is received. Finally, already used tools such as Google Location history can be leveraged to identify places visited and exact times of visit, to better remember encounters. Hence, our analysis show to what extent the memory based identification attack can be performed when only memory from the individuals is taken into account, as other tools would raise the probabilities  $p$ , eventually raising the identification of users. We also note that scenario is related to remembering encounters of people rather than pictures or other things. However, we have also presented several tools which can be used to increase the ability to perform such recognition. However our model is general, since the ability for someone to remember encounters is only taken into account when simplifying the graph. A higher ability to remember things would keep more edges in the graph, improving the possibility to identify positive and negative individuals, while a worse memory would remove more graphs, possibly achieving worse results.

#### 4.4 Modeling mitigation

In this section we discuss different techniques and proposal which can be used in order to mitigate the risks of a memory based individual identification. Each of these proposal will be tested in Section 5, where we will also discuss under which circumstances they may work or not.

**4.4.1 Variable time windows.** Currently decentralised MCTA broadcast the whole set of codes from the positive user to all devices with the application installed. This bring a vast set of information to the receiving user, as it can disclose users' positivity or negativity based on the codes received and the different time windows. Let us make this example to be more clear: suppose *Alice*, *Bob* and *Carl* are three co-workers which work in the same room, hence with a possible risk of infection due to their closeness. On a single day  $d$ , *Carl* is tested positive, hence he is not going to work, leaving only *Alice* and *Bob* in the same room. If *Alice* receives the notification from the MCTA, she can rapidly look through the codes received to disclose whether the responsible for the notification is *Bob* or *Carl*. Looking at things from *Alice*'s perspective, if the responsible is *Bob* then she must have received the positive codes also during day  $d$  in which *Carl* was not at work. On the opposite, if the responsible is *Carl*, she must not have received any code on day  $d$ . Since she receives all the positive codes, performing this operation is simple and straightforward, and can directly identify *Bob* or *Carl* either as positive or negative.

At the same time, this may also work with strangers, i.e. people which are not part of the individual usual social network, but are seen seldom. This may include people met at the grocery store, or at the post office, or anyway in places in which we do not spend most of our time, hence contacts may be more diverse. In this case, receiving a wider set of codes may help the receiving user to rapidly exclude negative people seen more frequently, as codes during those time windows would not be present in the received codes. In this case, a large

portion of users may be quickly pruned out of the graph, leaving only users seen less frequently which can be eventually identified.

We argue that having access to a limited set of information would eventually reduce the risk of this kind of attack. Focusing our analysis on de-centralised MCTA, this can be realized either with a push or pull logic. In case of a push logic, the server should send a subset of positive codes to the client, and wait for a feedback. In case such amount of information is sufficient to issue a notification, the no further data is required, hence the communication stops. Otherwise, the client may ask for more data, until a notification is issued or no further data is available. However, this requires for the server to know the client. In case of a pull logic, the client requests to the server whether it has data to send. In the positive case, the server sends a subset of positive codes, and the client can ask for more in case those are not sufficient to issue a notification to the user. Again, the communication stops whenever there is no more data to send, or in case the data sent is enough to show a notification to the user. Of course, both the MCTA and the server must be in the same circle of trust, in other words the server must have the guarantee that the client does not ask for further information if this is not strictly needed.

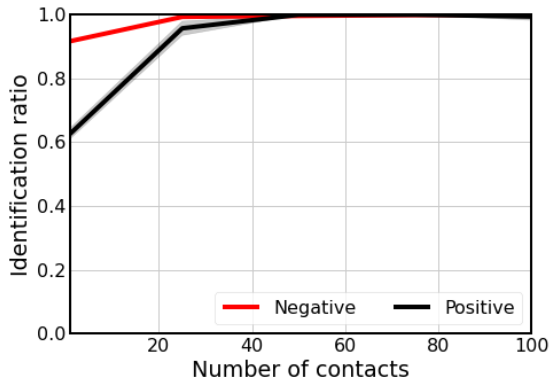
**4.4.2 Contact injection.** Another possibility to mitigate the risk of the memory based identification attack is to add codes into the report sent to the receiving users. These codes can be either false, randomly generated by the server and not belonging to any user, or they can be real, meaning that the server needs to aggregate reports from multiple positive individuals before sending them out. In the former case, the server must guarantee that the randomly generated codes do not belong to any other user. Codes are generated starting from the code used the day before. Given that these are random strings, it is unlikely that two smartphones generate the same code on the same day.

In case of adding false codes, it is a straightforward operation which can be done directly on the server. Upon receiving the  $N$  codes from the user which has just been tested positive, it adds  $k \cdot N$  codes which are randomly generated, with the guarantee that they cannot be correlated to any real user. Then the set of  $(k + 1) \cdot N$  codes is sent to all the users with the application installed, which now have  $k \cdot N$  more codes to correlate to positive users, possibly generating more noise hence more difficulty in understanding who is positive or negative. Obviously the server must guarantee that the added codes cannot be related to any other individuals, otherwise it may rise the false positives number.

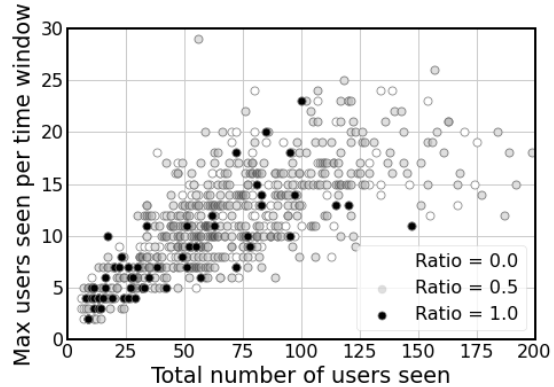
In case of aggregating real codes into the report, of course there is a trade off between the privacy preserving mechanism and the freshness of data to be reported. The server needs to have multiple reports from users before sending them out, hence it would not notify immediately users who have been in contact with the newly positive individual. Another option would be to send positive codes from users which have already been sent out, however these can be easily discarded by the MCTA, as it would easily see that some codes have already been received, hence it can be removed rapidly making this methodology not effective.

**4.4.3 Variable Contact Tracing Range.** A final possibility we envision to limit the memory based identification attack is to have a variable contact tracing range. This translates into having a variable threshold for the recognition of a close contact or a farther one. Obviously lower thresholds would extend the set of recorded contacts, eventually raising the number of false positives detected by the application, but with the benefit of increasing the sociability of the user, hence making the attack more challenging. This mitigation can be performed directly from the MCTA, which can adapt its own RSSI based on the sociability of the individuals. In fact, users with low sociability levels can be more prone to the memory based attack, while users which have a higher, more diverse sociability are more protected, as we will show in Section 5.

We analyze the distribution of re-identification risk of individuals based on different memory attack models. In particular, we explore prosecutor, journalist and marketer attacker models. Prosecutor attack is a common type of attack, which re-identifies individual user based on the background information [14]. In journalist attack, an individual user is re-identified based on the mapping to the ‘de-identified’ databases which contains



(a) Copenhagen: Positive negative identification



(b) Copenhagen: Ratio of identification

Fig. 3. Figure 3a shows the identification ratio of positive and negative contacts versus the total number of users seen. Figure 3b shows the identification ratio correlating the number of users seen in a time window versus the total number of users seen during the whole measurement period. Results are obtained with the Copenhagen dataset.

quasi-identifying and identifying variables [14]. In marketer attack, the probability of the matching records of equivalence class is measured.

In this case, we argue that changing the RSSI threshold depending on the sociability of the individual better protects users against the above mentioned attacks. In particular, low sociability users should have a lower threshold, which eventually results in having more recorded contacts hence raising their digital sociability with respect to the app. All the needed parameters can be computed by the MCTA, which can change the RSSI threshold for any time window in which the sociability level of the individual is low. In our case we do not estimate the actual distance at which two devices are located, since we are instead interested in understanding the effect of a lower threshold, which would raise the distance at which two devices are considered in contact, and a higher threshold, which would make the MCTA more accurate in determining close contacts only, at the cost of possibly missing some of them.

## 5 EXPERIMENTS

In this section we present and analyze the experiments we performed on the two datasets presented in Section 3, using the model we presented in Section 4 and the possible modeling mitigation techniques described too in Section 4.

To perform our experiments, we build the graph  $G$  for each user using the real data from the dataset, and then we run simulations to study different setups of our proposal. In any simulation, we infect one or more people from the individual social network, which eventually receives the positive codes to perform the local match and display the notification.

Since the two datasets are fundamentally different, we show the same analysis on both datasets to compare results and generalize outcomes.

Figure 3 and 4 show the identification ratio of individuals based on the Copenhagen and the Social Evolution datasets, with no model mitigation in place. More precisely Figure 3a and Figure 4a shows the identification ratio of positive and negative individuals versus the total number any two users had contact, for Copenhagen and Social Evolution respectively. Basically lower values on the x axis mean that two users have seldom seen each

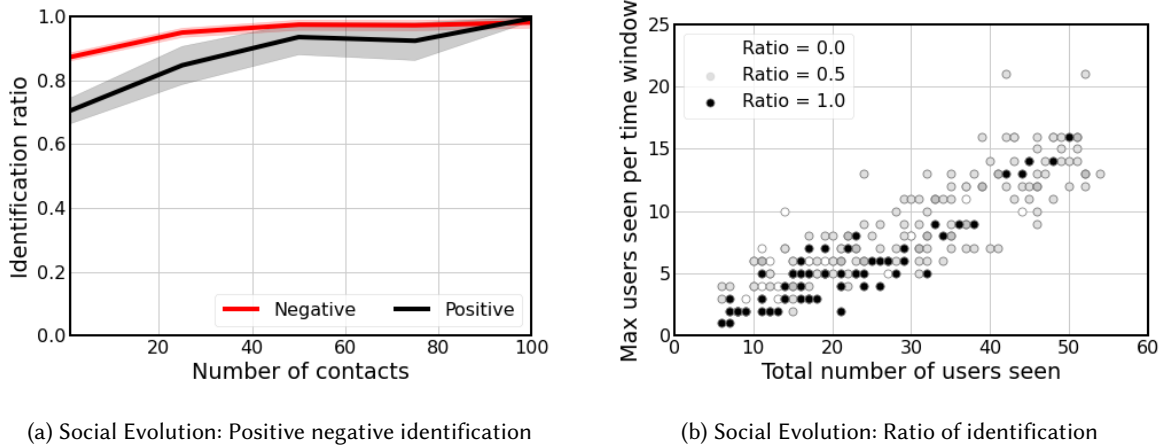


Fig. 4. Figure 4a shows the identification ratio of positive and negative contacts versus the total number of users seen. Figure 4b shows the identification ratio correlating the number of users seen in a time window versus the total number of users seen during the whole measurement period. Results are obtained with the Social Evolution dataset.

other, while higher values indicate a higher frequency of contacts. In other words, the left side of the chart shows contacts among strangers, while as we move to the right part of the chart we observe closer friends or relatives, which are seen more frequently. What Figures 3a and 4a tell us is that friends or close contacts, and in general people we see more, are easier to be identified. This happens because the more we see someone, the higher the possibility to be in contact when few people is around, or to be in contact for a longer period of time. This translates into a higher chance of identifying the individual, since there are more time windows in which it is possible to perform the detection. We can see that for this analysis, the Copenhagen and Social Evolution dataset report similar results, showing that the memory based identification attack can be performed under different conditions.

Figure 3b and Figure 4b show instead the ratio of identification, regardless whether of positive or negative individuals, accounting for the maximum number of users seen in any timewindow, which describe whether that user goes into crowded places or not, versus the total number of users seen during the whole measurement time. The three colors we plot which are white, gray and black, refer to identification ratios of 0, 0.5 and 1, respectively. Here we can see a clear pattern, which shows how users with low sociability levels, represented in bottom left part of the chart, tend to be far more identifiable compared to user with higher sociability levels, plotted in the top right part of the chart. This is due to the fact that having seen more users inherently adds noise into the graph, which in case it cannot be removed, makes the identification uncertain. This also confirms the hypothesis that under constrained mobility, such as during lock downs or immediately after, people will have less opportunities to meet people, hence the lower sociability level would make this kind of attack much easier to be performed. On the opposite, during normal mobility or in case for user with a higher sociability level, it would be more difficult to understand how is the responsible for the notification in case it is received, and in general it is harder to classify users seen as positive or negative. Again, also in this analysis the Copenhagen and the Social Evolution dataset show similar results.

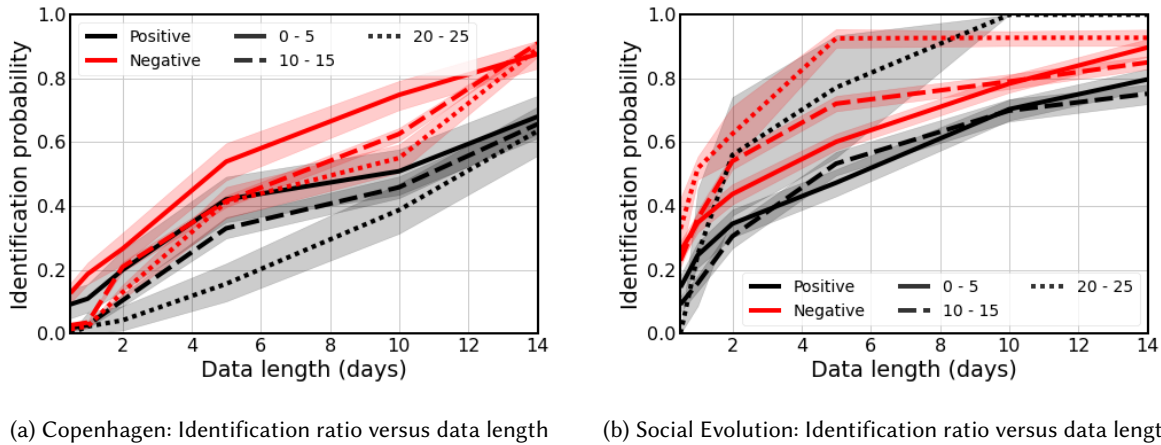


Fig. 5. Individual identification probability versus the length of the data sent to the mobile device. Results obtained with the Copenhagen datasets are shown in Figure 5a, while results obtained with the Social Evolution dataset are plotted in Figure 5b.

### 5.1 Variable time windows

We now evaluate the different mitigation techniques we discussed in Section 4. In this section we analyze how a variable time window to be reported to the user impacts the memory based identification attack.

Figure 5a and Figure 5b show the identification probability of positive and negative individuals versus the length of the data sent to the mobile application by the central server, for the Copenhagen and Social Evolution datasets, respectively. The identification ratio of negative individuals is plotted in red, while the identification of positive individuals is plotted in black. Again this confirms that negative users are easier to identify compared to positive user in both datasets. Moreover, it also confirms our hypothesis formulated in Section 4.4.1, where we stated that the longer the set of positive codes sent to the mobile application, the easier it is to identify individuals. Again, this may seem counter intuitive, so we discuss an example to clarify it. Suppose the extreme case, in which the application receives a single positive code pertaining to a single time window in which the user saw several users. It would be impossible to identify users as positive or negative individuals based solely on such information, since such code may belong to any of the users seen in such time window. Adding a code from another window may instead help to identify users based on the codes received in the first and in the second timewindow. Again, if both codes are sensed by the MCTA, then they should belong to a positive user seen in both time windows, hence users saw only in one of the two are clearly negative. On the opposite, if the MCTA sensed only one code, then the positive user must have been seen in that specific time window, but not in the other one. The more information is added, the more it is possible to perform these operations, pruning the memory graph until an identification of an individual is made possible. This shows how reducing the set of codes sent to the application helps in mitigating the memory based identification attack.

The different line styles also show the sociability levels of individuals. For sake of readability, we only show 3 sociability levels, more precisely individuals who saw at maximum between 0 and 5 users in any timewindow (0-5), users who saw at maximum between 10 and 15 users in any timewindow (10-15), and users who saw at maximum between 20 and 25 users in any timewindow (20-25). We can clearly see that among the identification of the positive individuals, lines are ordered accordingly to the sociability level they represent, with the lower sociability (0-5) being the highest for both positive and negative identification. Higher sociability follows with



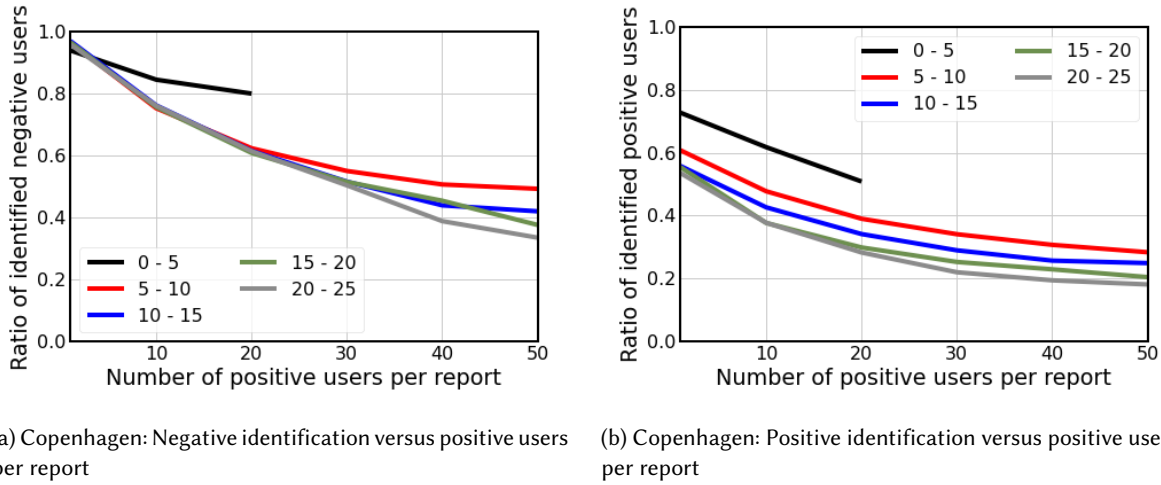


Fig. 6. Identification of negative (Figure 6a) and positive (Figure 6b) individuals versus the number of real positive users sent by the central server. Results are obtained with the Copenhagen dataset.

lower identification probabilities. In this case the Copenhagen dataset, plotted in Figure 5a, shows a more linear trend compared to the Social Evolution dataset shown in Figure 5b. This may happen due to the different granularity and methods of collecting data of the two datasets. Nevertheless, the general trend which shows that the more data is available the easier it is to perform the memory based identification attack holds for both.

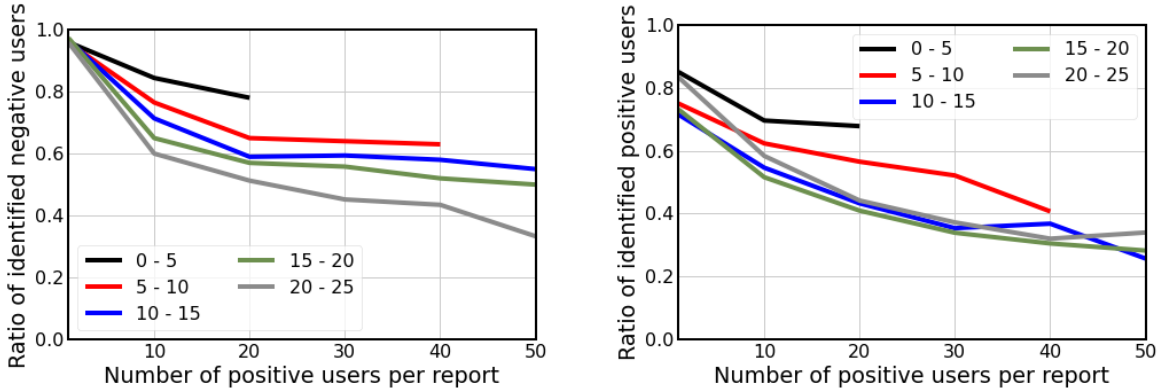
## 5.2 Contact injection

The second mitigation technique we evaluate is the contact injection, which can be performed with false codes or real positive codes.

Figure 6 and Figure 7 show the results when injecting real users into the report sent to the user which have the MCTA installed. A clear trend emerges, since an higher number of codes put in the report translates into a higher difficulty to identify positive and negative individuals. In fact, when adding codes from positive users into the report, the receiving user cannot easily identify the specific correlation between a user and the positive codes, hence the  $I_p$  and  $I_n$  operations may identify less users. This in turn makes the  $P$  operation more challenging. We also note that the line for the sociability level between 0 and 5 users stops earlier, since for such users the total number of users seen in a time window was lower compared to the others, hence we evaluated less users in total. Also in this case the Copenhagen and Social Evolution dataset present similar trends, though clearer in the Copenhagen dataset, since the higher number of users in the dataset make the trend emerge better compared to the Social Evolution dataset. However both dataset clearly show that the low sociability makes the attack easier to be performed, and that by adding positive users into the dataset it is possible to mitigate the risks associated with this kind of attack.

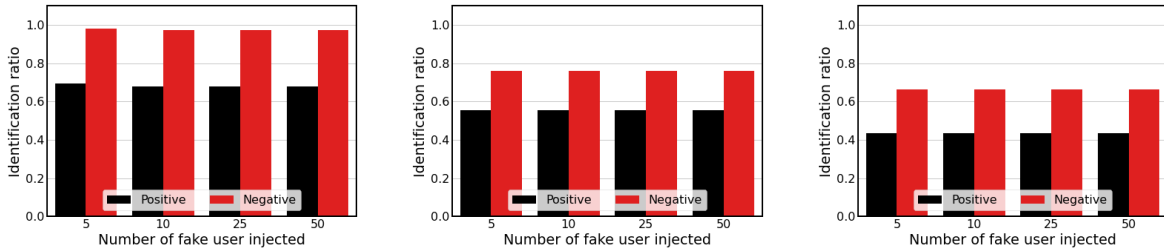
We now test whether the injection of false positive codes into the report helps in mitigating the memory based identification attack.

Figure 8 and Figure 9 show the identification ratio when adding fake users to the report sent to the users. We show this for 1 real positive user in the notification in Figure 9a, for 10 positive users added to the report in Figure 9b, and for 20 positive users in Figure 9c. Results confirm again that injecting codes from real positive users reduces the risk of being identified, while on the opposite injecting false positive codes does not change at



(a) Social Evolution: Negative identification versus positive users per report (b) Social Evolution: Positive identification versus positive users per report

Fig. 7. Identification of negative (Figure 6a) and positive (Figure 6b) individuals versus the number of real positive users sent by the central server. Results are obtained with the Social Evolution dataset.



(a) Copenhagen: Identification with fake injection - 1 positive user (b) Copenhagen: Identification with fake injection - 10 positive users (c) Copenhagen: Identification with fake injection - 20 positive users

Fig. 8. Identification with injection of fake users for different number of real positive users in the report: Figure 8a presents 1 positive user, Figure 8b presents 10 positive users, and Figure 8c presents 20 positive users. Results obtained with the Copenhagen dataset.

all the identification ratio, regardless of the number of real positive users in the report. This can be clearly seen in any of the 3 charts of Figures 8-9, since no difference can be seen with respect to the identification of positive or negative individuals in the same chart. Again also in this case both datasets show a similar trend, even though the Social Evolution datasets exhibit less differences between the identification of positive and negative individuals.

### 5.3 Variable Contact Tracing Range

The last mitigation technique which we evaluate is changing the RSSI threshold to increase the sociability of the users. We perform this test only for the Copenhagen dataset, since the Social Evolution dataset only reports that two users are close, but no RSSI information is available. We use the API of ARX data anonymization

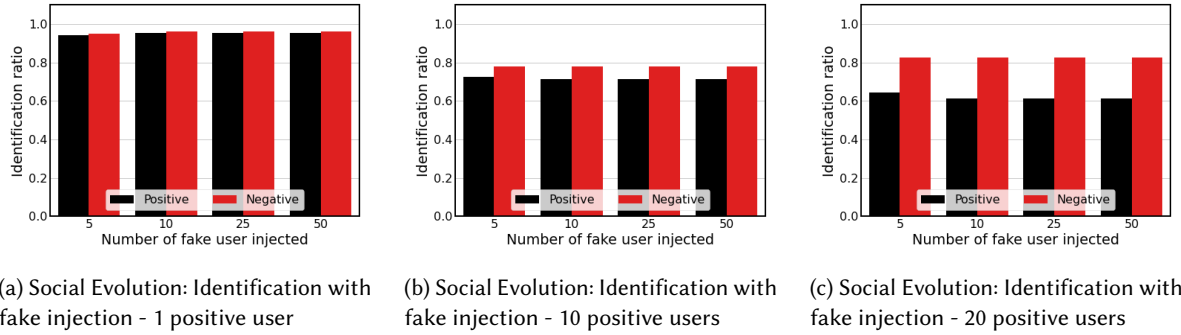


Fig. 9. Identification with injection of fake users for different number of real positive users in the report: Figure 9a presents 1 positive user, Figure 9b presents 10 positive users, and Figure 9c presents 20 positive users. Results obtained with the Social Evolution dataset.

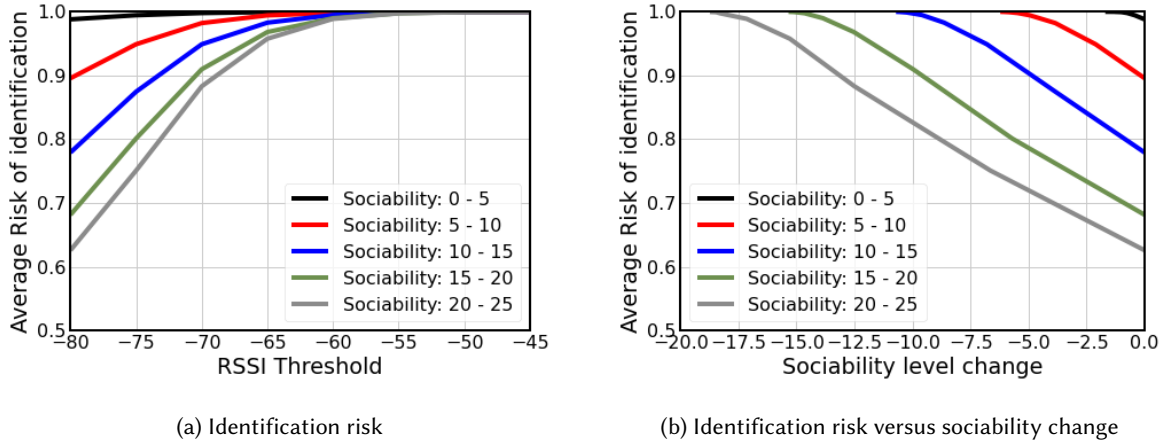


Fig. 10. Figure 10a shows the identification risk versus the RSSI threshold, while Figure 10b shows the same metric with respect to the sociability change due to a varying RSSI threshold.

tool<sup>2</sup> to measure the re-identification risk using different privacy attack models. In the following discussion, we particularly report the re-identification risk of ‘prosecutor’ attack model as it is the most common type of memory based attack.

Figure 10 shows the identification risk for an individual when changing the RSSI threshold used to trace contacts. The rationale is that a lower threshold level increases the sociability of the individual, at the cost of possibly raising the number of false positives, while a higher threshold traces only close contacts, but decreasing the sociability level of the individual, hence making it more prone to the memory based attack.

Figure 10a shows the identification risk for users with different sociability levels when varying the RSSI threshold. Again it is possible to see that users which are more social are more protect against this kind of attack. However when we increase the RSSI threshold above -60 dBm, all users are exposed to the memory based

<sup>2</sup><https://arx.deidentifier.org/>

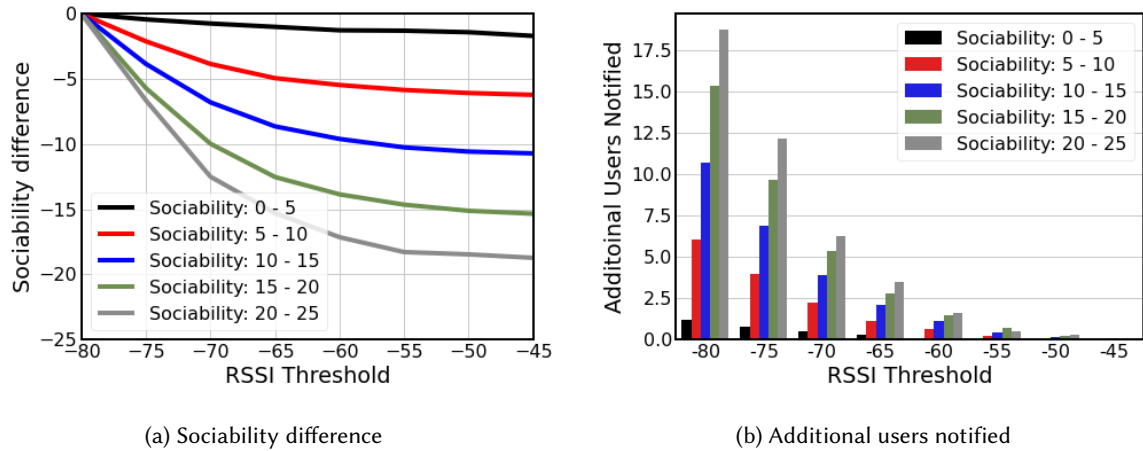


Fig. 11. Figure 11a shows the identification risk versus the RSSI threshold, while Figure 11b shows the same metric with respect to the sociability change due to a varying RSSI threshold.

identification attack regardless of their original sociability, since the set of user with which the graph is built is reduced due to the higher threshold set. This is also confirmed by Figure 10b, where we show the same metric on the y axis, while the x axis represents the sociability change compared to an RSSI threshold set to -80 dBm. Again, when we raise the threshold, we reduce the number of users traced, while on the opposite reducing it increases the sociability level. We show that when reducing the sociability level of the individual due to a higher RSSI threshold, plotted on left side of the chart, the identification risk is higher compared to the right side of the chart, where we gradually achieve the sociability level obtained when using -80 dBm as RSSI threshold.

We also show how the sociability of the individuals change due to a varying RSSI threshold, and what this would mean with respect to additional notifications sent to the users.

Figure 11a shows how the sociability of the individuals change when considering the lowest threshold we used in our experiments, which is -80 dBm. Clearly the sociability of users decreases as the RSSI threshold is raised. However, we can see that regardless of the sociability of the individuals, beyond -60 dBm no significant change is observed. Moreover, for low sociability individuals a change in the RSSI threshold increases the sociability less compared to other users. This is also confirmed by Figure 11b, where we show how many more users would be notified due to the lower RSSI threshold compared to the highest threshold we tested. We can see that only highly social individual would notify a considerable higher amount of users, while low sociability individual (i.e. those with a sociability of 0-5) would notify less than 1 user more, but increasing their sociability also raises their chances to be protected against the memory based identification attack, as we have shown in previous figures.

## 6 CONCLUSION AND DISCUSSION

In this paper we have shown the memory based identification attack, a novel privacy issue which can be performed on de-centralised mobile contact tracing applications. We have shown how it is possible to build a graph leveraging the available information obtained by the BLE codes sent by users, and how this can be used to infer the positivity or negativity of the individuals.

We have also discussed some techniques which can be used to mitigate the possibility for this attack to happen. All of these techniques can be implemented still keeping a de-centralised architecture: the Variable time window technique can be implemented in the server, with the MCTA providing feedback on whether more data

is needed to perform the match; the Contact injection can be again performed on the server, though we showed that it is effective only when injecting real positive codes and not false ones, and we discussed the trade off between information freshness and privacy protection; finally the Variable Contact Tracing Range aims at raising the sociability level of the individual, and it can be implemented directly on the MCTA, possibly accounting for a reduced probability of infection if user were farther away, but still better protecting the user against the memory based identification attack.

Our results on two different real world datasets confirmed that the memory based identification attack can be indeed performed using solely the information available in the de-centralised MCTA system, and it can lead to the identification of the individuals who are positive or negative in the network. While understanding whether a user is negative may not have an impact, identifying a positive user that might be responsible for the infection may induce violence and self-justification of the notified person. The bottom line is – this continuing practice, without consideration of the risks from memory-based re-identification, could potentially violate the privacy of individuals in the traces of decentralised MCTA systems.

Future work on this topic include personalized models dependent on the strength of ties of individuals, and the mobility patterns individuals perform. This would enable a more tailored modeling of each individual risk of being identified with a memory based attack. Moreover since our memory modeling is based on works which studied how people remember things, we also plan to collect real data which show to what extent people can remember encounters.

## REFERENCES

- [1] Nadeem Ahmed, Regio A. Michelin, Wanli Xue, Sushmita Ruj, Robert Malaney, Salil S. Kanhere, Aruna Seneviratne, Wen Hu, Helge Janicke, and Sanjay K. Jha. 2020. A Survey of COVID-19 Contact Tracing Apps. *IEEE Access* 8 (2020), 134577–134601. <https://doi.org/10.1109/access.2020.3010226>
- [2] T. Altuwaiyan, M. Hadian, and X. Liang. 2018. EPIC: Efficient Privacy-Preserving Contact Tracing for Infection Detection. In *2018 IEEE International Conference on Communications (ICC)*. IEEE, 1–6.
- [3] Apple. 2020. Exposure Notification. <https://developer.apple.com/documentation/exposurenotification>. Accessed: 2020-07-13.
- [4] Jason Bay, Joel Kek, Alvin Tan, Chai Sheng Hau, Lai Yongquan, Janice Tan, and Tang Anh Quy. 2020. BlueTrace: A privacy-preserving protocol for community-driven contact tracing across borders. *Government Technology Agency-Singapore, Tech. Rep* (2020).
- [5] James Bell, David Butler, Chris Hicks, and Jon Crowcroft. 2020. TraceSecure: Towards Privacy Preserving Contact Tracing. (2020). arXiv:2004.04059 [cs.CR]
- [6] Samuel Brack, Leonie Reichert, and Björn Scheuermann. 2020. Decentralized Contact Tracing Using a DHT and Blind Signatures. *IACR Cryptol. ePrint Arch.* 2020 (2020), 398.
- [7] Ran Canetti, Yael Tauman Kalai, Anna Lysyanskaya, Ronald L. Rivest, Adi Shamir, Emily Shenk, Ari Trachtenberg, Mayank Varia, and Daniel J. Weitzner. 2020. Privacy-Preserving Automated Exposure Notification. IACR.org. (July 2020). <https://www.microsoft.com/en-us/research/publication/privacy-preserving-automated-exposure-notification/>
- [8] Francesco Casella. 2021. Can the COVID-19 Epidemic Be Controlled on the Basis of Daily Test Reports? *IEEE Control Systems Letters* 5, 3 (Jul 2021), 1079–1084. <https://doi.org/10.1109/lcsys.2020.3009912>
- [9] Hyunghoon Cho, Daphne Ippolito, and Yun William Yu. 2020. Contact Tracing Mobile Apps for COVID-19: Privacy Considerations and Related Trade-offs. (2020). arXiv:2003.11511 [cs.CR]
- [10] EU Commission. 2020. Coronavirus: a common approach for safe and efficient mobile tracing apps across the EU. <https://ec.europa.eu/digital-single-market/en/news/coronavirus-common-approach-safe-and-efficient-mobile-tracing-apps-across-eu>. Last Update: 2020-05-15.
- [11] EU Commission. 2020. Mobile applications to support contact tracing in the EU’s fight against COVID-19: Progress reporting June 2020. [https://ec.europa.eu/health/sites/health/files/ehealth/docs/mobileapps\\_202006progressreport\\_en.pdf](https://ec.europa.eu/health/sites/health/files/ehealth/docs/mobileapps_202006progressreport_en.pdf).
- [12] Aaqib Bashir Dar, Auqib Hamid Lone, Saniya Zahoor, Afshan Amin Khan, and Roohie Naaz. 2020. Applicability of Mobile Contact Tracing in Fighting Pandemic (COVID-19): Issues, Challenges and Solutions. *Cryptology ePrint Archive, Report 2020/484*. (2020). <https://eprint.iacr.org/2020/484>.
- [13] Ken TD Eames and Matt J Keeling. 2003. Contact tracing and disease control. *Proceedings of the Royal Society of London. Series B: Biological Sciences* 270, 1533 (2003), 2565–2571.
- [14] Khaled El Emam and Fida Dankar. 2012. Re-identification risk in de-identified databases containing personal information. US Patent 8,316,054.

- [15] Luca Ferretti, Chris Wymant, Michelle Kendall, Lele Zhao, Anel Nurtay, Lucie Abeler-Dörner, Michael Parker, David Bonsall, and Christophe Fraser. 2020. Quantifying SARS-CoV-2 transmission suggests epidemic control with digital contact tracing. *Science* 368, 6491 (2020).
- [16] German government. 2020. Corona Warn App. <https://www.coronawarn.app/en/>. Accessed: 2020-07-13.
- [17] Italian government. 2020. Immuni. <https://www.immuni.it>. Accessed: 2020-07-13.
- [18] Swiss government. 2020. SwissCovid App. <https://ethz.ch/services/en/news-and-events/solidarity/pilot-swiss-covid-app.html>. Accessed: 2020-07-13.
- [19] Singaporean government. 2020. TraceTogether. <https://www.tracetgether.gov.sg/>. Accessed: 2020-07-13.
- [20] Roxana Gutiérrez-Romero. 2020. Conflict in Africa during COVID-19: social distancing, food vulnerability and welfare response. arXiv:2006.10696 [econ.EM]
- [21] Google Inc. 2020. Exposure Notification. <https://www.google.com/covid19/exposurenotifications>. Accessed: 2020-07-13.
- [22] Mahabir Prasad Jhanwar and Sumanta Sarkar. 2020. PHyCT : Privacy preserving Hybrid Contact Tracing. *IACR Cryptol. ePrint Arch.* 2020 (2020), 793.
- [23] Nattiya Kanhabua, Tu Ngoc Nguyen, and Claudia Niederée. 2014. What triggers human remembering of events? A large-scale analysis of catalysts for collective memory in Wikipedia. In *IEEE/ACM Joint Conference on Digital Libraries*. IEEE, 341–350.
- [24] Philipp H. Kindt, Trinad Chakraborty, and Samarjit Chakraborty. 2020. How Reliable is Smartphone-based Electronic Contact Tracing for COVID-19? arXiv:2005.05625 [cs.NI]
- [25] Don Klinkenberg, Christophe Fraser, and Hans Heesterbeek. 2006. The effectiveness of contact tracing in emerging epidemics. *PloS one* 1, 1 (2006), e12.
- [26] South Korea. [n.d.]. South Korea Mobile Contact Tracing. <http://global.seoul.go.kr/>
- [27] Robert E. Clark Larry R. Squire, John T. Wixted. 2007. Recognition memory and the medial temporal lobe: a new perspective. *Nature reviews. Neuroscience* 8, 11 (2007), 872–883.
- [28] Douglas J. Leith and Stephen Farrell. 2020. Coronavirus Contact Tracing: Evaluating The Potential Of Using Bluetooth Received Signal Strength For Proximity Detection. arXiv:2006.06822 [eess.SP]
- [29] Ming Li, Shucheng Yu, Ning Cao, and Wenjing Lou. 2013. Privacy-preserving distributed profile matching in proximity-based mobile social networks. *IEEE Transactions on Wireless Communications* 12, 5 (2013), 2024–2033.
- [30] Joseph K Liu, Man Ho Au, Tsz Hon Yuen, Cong Zuo, Jiawei Wang, Amin Sakzad, Xiapu Luo, and Li Li. 2020. Privacy-Preserving COVID-19 Contact Tracing App: A Zero-Knowledge Proof Approach. *IACR Cryptol. ePrint Arch.* 2020 (2020), 528.
- [31] Anmol Madan, Manuel Cebrian, Sai Moturu, Katayoun Farrahi, et al. 2011. Sensing the "health state" of a community. *IEEE Pervasive Computing* 11, 4 (2011), 36–45.
- [32] World Health Organization. [n.d.]. Opening remarks at the media briefing on COVID-19. <https://www.who.int/dg/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19-11-march-2020>. Released: 2020-03-11.
- [33] Andrew Paverd and A. Martin. 2014. Modelling and Automatically Analysing Privacy Properties for Honest-but-Curious Adversaries.
- [34] Piotr Sapiezynski, Arkadiusz Stopczynski, David Dreyer Lassen, and Sune Lehmann. 2019. Interaction data from the copenhagen networks study. *Scientific Data* 6, 1 (2019), 1–10.
- [35] Sanjay Sareen, Sandeep K Sood, and Sunil Kumar Gupta. 2018. IoT-based cloud framework to control Ebola virus outbreak. *Journal of Ambient Intelligence and Humanized Computing* 9, 3 (2018), 459–476.
- [36] Jun She, Jinjun Jiang, Ling Ye, Lijuan Hu, Chunxue Bai, and Yuanlin Song. 2020. 2019 novel coronavirus of pneumonia in Wuhan, China: emerging attack and management strategies. *Clinical and translational medicine* 9, 1 (2020), 1–7.
- [37] Manish Shukla, Rajan M A, Sachin Lodha, Gautam Shroff, and Ramesh Raskar. 2020. Privacy Guidelines for Contact Tracing Applications. (2020). arXiv:2004.13328 [cs.LG]
- [38] Carmela Troncoso, Mathias Payer, Jean-Pierre Hubaux, Marcel Salathé, James Larus, Edouard Bugnion, Wouter Lueks, Theresa Stadler, Apostolos Pyrgelis, Daniele Antonioli, Ludovic Barman, Sylvain Chatel, Kenneth Paterson, Srdjan Čapkun, David Basin, Jan Beutel, Dennis Jackson, Marc Roeschlin, Patrick Leu, Bart Preneel, Nigel Smart, Aysajan Abidin, Seda Gürses, Michael Veale, Cas Cremers, Michael Backes, Nils Ole Tippenhauer, Reuben Binns, Ciro Cattuto, Alain Barrat, Dario Fiore, Manuel Barbosa, Rui Oliveira, and José Pereira. 2020. Decentralized Privacy-Preserving Proximity Tracing. (2020). arXiv:2005.12273 [cs.CR]
- [39] Endel Tulving. 2002. Episodic memory: From mind to brain. *Annual review of psychology* 53, 1 (2002), 1–25.
- [40] Benton J Underwood. 1957. Interference and forgetting. *Psychological review* 64, 1 (1957), 49.
- [41] Bo Wang. 2014. Effect of Time Delay on Recognition Memory for Pictures: The Modulatory Role of Emotion. *PLOS ONE* 9, 6 (06 2014), 1–9. <https://doi.org/10.1371/journal.pone.0100238>
- [42] Rui Zhang, Jinxue Zhang, Yanchao Zhang, Jinyuan Sun, and Guanhua Yan. 2013. Privacy-preserving profile matching for proximity-based mobile social networking. *IEEE Journal on Selected Areas in Communications* 31, 9 (2013), 656–668.