Privacy-Preserving Multi-Operator Contact Tracing for Early Detection of Covid19 Contagions

Davide Andreoletti[1], Omran Ayoub[2], Silvia Giordano[1], Massimo Tornatore[2], and Giacomo Verticale[2]

[1]Networking Laboratory, University of Applied Sciences of Southern Switzerland, Manno, Switzerland, Email: {name.surname}@supsi.ch
[2]Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milano, Italy, Email: {name.surname}@polimi.it

***Abstract*—The outbreak of coronavirus disease 2019 (covid-19) is imposing a severe worldwide lock-down. Contact tracing based on smartphones' applications (apps) has emerged as a possible solution to trace contagions and enforce a more sustainable selective quarantine. However, a massive adoption of these apps is required to reach the critical mass needed for effective contact tracing. As an alternative, geo-location technologies in next generation networks (e.g., 5G) can enable Mobile Operators (MOs) to perform passive tracing of users' mobility and contacts with a promised accuracy of down to one meter. To effectively detect contagions, the identities of positive individuals, which are known only by a Governmental Authority (GA), are also required. Note that, besides being extremely sensitive, these data might also be critical from a business perspective. Hence, MOs and the GA need to exchange and process users' geo-locations and infection status data in a privacy-preserving manner. In this work, we propose a privacy-preserving protocol that enables multiple MOs and the GA to share and process users' data to make only the final users discover the number of their contacts with positive individuals. The protocol is based on existing privacy-enhancing strategies that guarantee that users' mobility and infection status are only known to their MOs and to the GA, respectively. From extensive simulations, we observe that the cost to guarantee total privacy (evaluated in terms of data overhead introduced by the protocol) is acceptable, and can also be significantly reduced if we accept a negligible compromise in users' privacy.**

***Index Terms*—Mobile Operators, Privacy, Covid19**

I. Introduction

Following recent surge of the coronavirus disease (Covid-19) epidemic, various governmental and organizational bodies are expressing strong interest in employing mobile-communication technologies to early detect contagions. With the term 'early detection', we refer to the identification of positive individuals before they show any symptoms. This generally happens during the incubation period of the virus (around 5 days for Covid-19 [1]), or even during the entire course of the disease. As asymptomatic people unknowingly diffuse the virus, early detection is fundamental to drastically limit virus spread [1].

Several smartphone-based apps for early detection are already available [2]–[4]. These apps allow a user to know whether she encountered positive individuals or not (e.g., by correlating her mobility with that of known positive cases). In most countries, to comply with strict local privacy regulations, these apps are developed with privacy as a primary design constraint. However, app-based approaches suffer from several drawbacks. First, it is hard to reach the critical mass needed for an effective contact tracing (a typical safe number is 60% of population, a very challenging target [3]). In addition, these apps require the continuous use of data acquisition technologies (e.g., the GPS and the Bluetooth) that extensively consume devices' batteries. We also note that people is less likely to keep such apps installed on their smartphones at the very beginning of the epidemics, when early detection is decisive to contain the diffusion of the virus.

Contact tracing exploiting users' mobility data collected by mobile telecom operators (MOs) is regarded as a promising alternative to app-based solutions [5]. In the upcoming years, once 5G will have consolidated its penetration, MOs will possess technologies to perform a continuous and accurate tracking of users' devices. For instance, Ref. [6] shows that the average accuracy of device positioning in ultra-dense 5G networks will be on the sub-meter order. A great advantage of a passive and continuous positioning is the very limited involvement of the final users. Users are not required to install any application on their smartphones, but only to give an explicit consent to track their position (explicit consent that is currently commonly granted to several apps, such as [2], [4]), therefore more easily reaching the critical mass. Even though users (and governments) are becoming more concerned regarding possible violation of the privacy of positioning data by the MOs [7] (e.g., a MO might sell them to third parties), we remark that MOs already estimate subscribers' positions to improve their services [8], and that the proposed approach guarantees that MOs do not get any sensitive data beyond it.

To effectively obtain an early detection of infections through mobility tracing, in addition to users' mobility data discussed above, the identities of positive individuals, which are only known to a Governmental Authority (GA) through the nation medical institutions, are required, and this is (a possibly even more)-sensitive information that must not be exposed. Therefore, MOs and the GA are required to collect, exchange and process this mobility data (from MO) and infection data (from the GA) in a secure and a privacy-preserving manner.

In this work, we propose a privacy-preserving protocol that enables GA and MOs to securely share and process users' data, such that each user is guaranteed to be the only person who knows the number of contacts with positive individuals she had (henceforth referred to as user's *score*). The protocol is built on consolidated privacy-enhancing strategies (e.g., secure

secret sharing and homomorphic encryption) that guarantee total privacy to users, i.e., the mobility and the infection status of a user are only known to her MO and to the GA, respectively.

This privacy is achieved at an acceptable cost in terms of data overhead exchanged among MOs and GA, as shown from extensive simulations. With slight modifications, the proposed protocol can also be employed i) to make this user discover how many positive people were in her same locations, but not necessarily in her close proximity (e.g., in a pub) and ii) to make the GA know only the identities of the users with a score above a given threshold. The identification of these users would make it possible to more easily stop the diffusion of the virus, but it poses a privacy dilemma and does not comply to several privacy regulations. In this work, we only provide the technical means to realize such identification in a privacy-preserving manner. Note also that, in case the number of these users is high, such procedure requires the exchange of a significant data overhead. However, we also show that this overhead can be heavily reduced at a negligible reduction of users' privacy.

The rest of the paper is structured as follows: in Section II we briefly review some existing approaches for privacy-preserving contact tracing. Section III describes the involved entities and their privacy requirements. We present the building blocks of the proposed protocol and the protocol itself in Sections IV and V, respectively. In Section VI we show some illustrative results obtained by simulation. Finally, Section VII concludes the paper.

II. Related Work

Existing solutions for contact tracing are generally based on smartphone apps of two main types: i) *location-based* (e.g., PrivateKit from MIT [2]) in which user's locations are acquired (e.g., with the GPS technology) and correlated with the locations of positive individuals; ii) *token-based* (e.g., TraceTogether [3] and Immuni [4]) that exchange anonymized tokens with smartphones in the proximity of the user (i.e., by exploiting the Bluetooth), and successively match the received tokens with those of known positive persons. A user who is tested positive can deliberately share her data (either location or received tokens) with a trusted authority, who then broadcasts it to all the others. Based on this, the app returns if a user has been in contact with a positive individual [2]. As operations are done on users' devices, privacy is mostly preserved, i.e., users' location and contacts are not exposed to the authority.

However, several privacy issues are still pending. For instance, in location-based apps each user receives the location data of a positive person, whose identity might be obtained from re-identification attacks [3], [9]. In TraceTogether [3] users send their phone numbers and all the received tokens to the authority, which in turn sends a message to those users who met some positive person. As users' contacts are exposed to the authority, this solution would hardly be adopted in countries with strict privacy laws, and several solutions are proposed to solve this issue [3]. For example, users might send to the authority only their tokens, and then perform anonymized queries to know if they met some positive person. However, other malicious behaviors are possible given that users obtain the tokens of persons in their proximity. Specifically, a user can craft a query to discover if the person that she met at a given time is positive [3].

In this work, we exploit consolidated privacy-preserving techniques (e.g., as those employed in [10], [11]) to compute the number of contacts that a user had with positive people, while guaranteeing that users' contacts, locations and infection status are not disclosed to illegitimate parties (see subsection III-B for further details). Differently from token-based solutions that only detect users' proximity, our protocol allows to compute also the number of positive persons within a given place. Unlike existing location-based apps, however, we assume that users' locations are estimated by MOs without any involvement of the users (e.g., using techniques for accurate geo-localization from cellular signals, such as those proposed in [6]). In this respect, authors in [5], [12] argue that MOs might play a decisive role in fighting the spreading of a virus, provided that users' privacy is guaranteed.

III. Modeling of Involved Entities

In this Section, we formally define the concept of users' scores, and we describe the role and privacy requirements of the entities involved in their computation. Before doing that, we introduce the concepts of *contact* and *infection status*. We say that two persons $user_i$ and $user_j$ have a contact iff the distance between them is below a given threshold th. We encode this information in the binary variable $c_{ij}^{(t)} = 1$ if $Dist(loc_i^{(t)}, loc_j^{(t)}) < th$, and 0 otherwise, where $loc_i^{(t)}$ and $loc_j^{(t)}$ refer to the geo-location (e.g., latitude and longitude) of $user_i$ and $user_j$ at time t, respectively, while $Dist$ is a measure of geographical distance. Concerning the infection status, we then introduce the binary variable $s_i^{(t)} = 1$ in case $user_i$ is considered positive at time t, and 0 otherwise. $Score_i$ is the number of contacts that $user_i$ has, during a given period of time, with positive individuals. In formulas:

$$Score_i = \sum_t \sum_{j:c_{ij}=1} s_j^{(t)} \tag{1}$$

Similarly, $Score_i^{(Loc)}$ is the number of positive persons that were in a certain location Loc at the same time of $user_i$, and is computed as $Score_i^{(Loc)} = \sum_{j \in Loc} s_j^{(t)}$, where the considered locations are assumed to be chosen by $user_i$ herself.

A. Role of Involved Entities

The GA is an entity established by the government to monitor the infection status of individuals within a certain region and, specifically, to collect from medical institutions the identities of positive individuals willing to share this data.

MOs are instead telecom companies that provide mobile connectivity within the considered region. Without loss of generality, we assume that each user is served by only one

MO, and that the whole area is covered by all the MOs. Then, we also assume that MOs estimate the locations of their users at time t, i.e., $\hat{loc}_i^{(t)}, \forall i$ from cellular signals received by users' devices (e.g., as done in [6]).

B. Privacy Requirements and Security Models

We assume that the GA and the MOs are *honest-but-curious*, i.e., they honestly execute the protocol but also try to violate other parties' privacy from the received data. Privacy requirements for each type of data are illustrated below.

1) Users' Locations: estimates of a user's locations should only be known to her MO.

2) Users' Contacts: information regarding contacts between two users should only be known to their MOs. In addition, if these users are subscribers of different MOs, each MO should not know anything neither about the identity of the other MO's user, nor about the number of contacts between its users and any other user of its competitors (e.g., how many contacts $user_i$ and $user_j$ have during a given period).

3) Users' Infection Status and Scores: The infection status of a user should only be known to the GA and to the user herself (say $user_i$). $Score_i$ and $Score_i^{(Loc)}$ should only be known to $user_i$, except when $Score_i$ is greater than a threshold χ. In this case, $Score_i$ and the identity of $user_i$ might also be known to the GA (see subsection V-C for the details).

IV. Building Blocks of the Privacy-Preserving Protocol

A. Existing Privacy-Preserving Building Blocks

1) Shamir Secret Sharing: A Shamir Secret Sharing (SSS) scheme [13] allows to securely distribute a secret s among a set of participants in such a way that s can only be recovered if a sufficient number of them cooperate. The piece of secret s that each participant receives is called *share*, and it is referred to as $[\![s]\!]$. In this work, we employ a $(2, 2)$ SSS, i.e., s is reconstructed only if 2 out of the 2 considered participants cooperate. SSS has several homomorphic properties, i.e., each participant can perform several operations on the shares that result in the same operations over the corresponding secrets (e.g., linear combinations). Then, participants can compute $[\![s_1 \cdot s_2]\!]$ using the **Mult** protocol presented in [14], or they can use the **EQ** and **Comp** protocols [14] to perform the equality check and the comparison operations. In the latter, participants input their shares $[\![s_1]\!]$ and $[\![s_2]\!]$ and obtain the share $[\![b_{eq}]\!]$ (resp., $[\![b_{ge}]\!]$), where $b_{eq} = 1$ (resp., $b_{ge} = 1$) iff $s_1 = s_2$ (resp., $s_1 \geq s_2$) and 0 otherwise.

2) Paillier Cryptosystem: Paillier [15] is a secure cryptosystem with the following properties: i) it is asymmetric, i.e., anyone can encrypt a message, but only the owner of the private key can decrypt it; ii) it is probabilistic, i.e., two encryptions of the same plaintext yield different ciphertexts and iii) it is homomorphic with respect to the summation of two ciphertexts (computed as $Enc(m_1 + m_2) = Enc(m_1) \cdot Enc(m_2)$) and to the product between a ciphertext and a plaintext (computed as $Enc(m_1 \cdot m_2) = Enc(m_1)^{m_2}$).

B. New Privacy-Preserving Primitives based on SSS

*1) **Secure Square Distance**:* the *Secure Square Distance* module takes in input the shares of the coordinates of points i and j, i.e., $[\![x_i]\!], [\![y_i]\!], [\![x_j]\!], [\![y_j]\!]$ and returns $[\![d_{ij}^2]\!]$, where d_{ij} is the euclidean distance between these points. This module is based on the Mult subroutine.

*2) **ObliviousTransfer**:* the ObliviousTransfer module (OT) allows a sender to deliver some data to a receiver without knowing which data has been transmitted. OT inputs i) a set of $2N$ shared elements arranged into a table with N rows and two columns (namely, *attribute* and *value*) and ii) the share $[\![attribute_x]\!]$. This module is based on the Mult and EQ subroutines and outputs the share $[\![value_i]\!]$ if the attribute at row i is equal to $attribute_x$, and $[\![0]\!]$ otherwise. This value is computed as $[\![value_i]\!] = \sum_{j=1}^{N} [\![eq_{jx} \cdot value_j]\!]$, where $eq_{jx} = 1$ if $attribute_x = attribute_j$, and 0 otherwise.

V. The privacy-preserving protocol

The proposed protocol works in three main phases, namely *contact tracing*, *score computation* and *communication with users*. We describe these phases in the following subsections. We refer to the generic users $user_i$ and $user_j$ as subscribers of MO_k and $MO_{k\prime}$, respectively, but the described operations are valid for each user and MO.

A. Privacy-Preserving Contact Tracing

In this phase, MO_k obtains the binary value c_{ij} encoding the information about its generic $user_i$'s contacts, $\forall i$. Firstly, MO_k estimates the current location of $user_i$, i.e., $(\hat{lat}_i^{(t)}, \hat{long}_i^{(t)})$ by analyzing cellular signals coming from her device [6]. From this data, the MO_k can independently assess the contacts among its subscribers, but not with other MOs' users (since a free exchange of users' mobility data is prohibited by the considered privacy requirements). Hence, we propose to perform the privacy-preserving computation of c_{ij} as follows.

MO_k and $MO_{k\prime}$ compute the projections of their users' estimated positions on an euclidean plane (e.g., $\hat{x}_i^{(t)}, \hat{y}_i^{(t)}$) and exchange these values among them in form of secret shares. Then, they execute the Secure Square Distance module and obtain $[\![d_{ij}^2]\!]$, being d_{ij}^2 the squared euclidean distance between the generic $user_i$ and $user_j$. The Comp module is then employed to compare $[\![d_{ij}^2]\!]$ with the threshold $[\![th^2]\!]$ and obtain $[\![c_{ij}]\!]$. The MOs finally exchange these shares and recover the secret c_{ij} (that is 1 if $user_i$ and $user_j$ has a contact, and 0 otherwise). A representation of this phase is depicted in Fig. 1.

B. Secure Computation of Users' scores

In this phase, MO_k securely computes the score values of $user_i$. To do so, the GA sends to MO_k the infection status (in encrypted form) of $user_i$ during a considered period (e.g., in the last day), i.e., $Enc_{GA}(s_i^{(t)}), \forall t$. At each time instant of the considered period, MO_k and $MO_{k\prime}$ obtain $c_{ij}^{(t)}$ as described in the previous subsection. If this value is 1 (i.e., there is a contact between these users at time t), MO_k and $MO_{k\prime}$ exchange

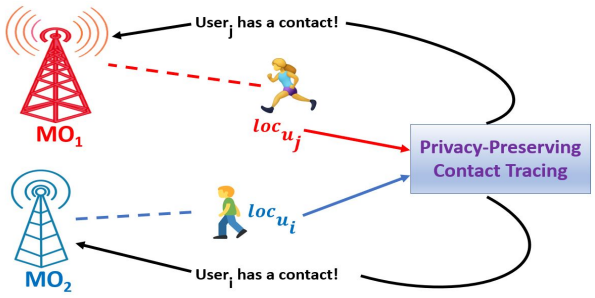

Fig. 1: Positioning process and privacy-preserving contact tracing performed by a pair of MOs

TABLE I: Data of subscribers of MO_k

Index	Identity	Score
1	$Id_1 = Name_1 \| \| PhoneNumber_1$	$Enc_{GA}(score_1)$
...	...	...
i	$Id_1 = Name_i \| \| PhoneNumber_i$	$Enc_{GA}(score_i)$
...	...	...
N_k	$Id_{N_k} = Name_{N_k} \| \| PhoneNumber_{N_k}$	$Enc_{GA}(score_{N_k})$

with each other $Enc_{GA}(s_i^{(t)})$ and $Enc_{GA}(s_j^{(t)})$. Then, MO_k computes $Enc_{GA}(Score_i)$ by homomorphically executing the summation in Eq. 1. Similarly, MO_k computes $Score_i^{(Loc)}$ by homomorphically summing the encrypted infection status of all users within area Loc at a given time, which are asked to all the remaining MOs.

The obtained data are then arranged by MO_k in a table that we represent in Table I. Such table has N_k rows (one of each subscriber of MO_k) and three columns, which are *Index*, *Identity* and *Score*. The first refers to the index of the row at which a certain user's data is stored. Without loss of generality, we assume that $user_i$'s data is stored at the i-th row. The second one stores the identities of the users (e.g., anything allowing to univocally identify them, such as full names and telephone numbers). The third represents the $Score$ values of users in encrypted form.

C. Communication with users

In this phase, we show how to distribute users' scores only to the legitimate entity (i.e., either the user herself or the GA). We consider the scenarios of *User-Triggered Communication* and *GA-Triggered Communication*. In the former, scores are requested by $user_i$ herself, and are kept secret to any other entity. In the latter, the GA identifies only the users with a score greater than a given threshold χ.

1) User-Triggered Communication: $user_i$ directly asks to MO_k the values $Enc_{GA}(score_i)$ and $Enc_{GA}(score_i^{(Loc)})$, for any location she is interested in. Then, $user_i$ exploits the homomorphic properties of the Paillier cryptosystem to compute $Enc_{GA}(Score_i \cdot Token_i)$, where $Token_i$ is a random value known only to her. $Enc_{GA}(Score_i \cdot Token_i)$ is then sent to the GA, which deciphers it and sends $Score_i \cdot Token_i$ back to $user_i$. Finally, $user_i$ removes the mask $Token_i$ and obtains $Score_i$. A similar computation is performed to obtain $score_i^{(Loc)}$. We represent this phase in Fig. 2.

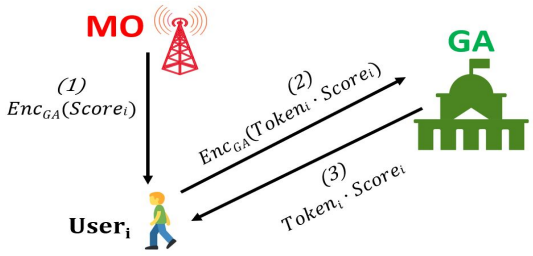

Fig. 2: Representation of the User-triggered communication

2) GA-Triggered Communication: MO_k sends to the GA $index_x$ and $Enc_{GA}(score_x), \forall x$. Then, the GA deciphers $Enc_{GA}(score_x)$ and obtains $(index_x, score_x), \forall x$. In case $\exists i : score_i \geq \chi$, the GA and MO_k jointly execute the OT subroutine described in subsection IV-B2. To do so, MO_k sends to the GA $[\![index_x]\!]$ and $[\![identity_x]\!], \forall x$, while the GA sends to MO_k $[\![index_i]\!]$. With these values in input, the OT module returns to MO_k and to the GA their shares $[\![identity_i]\!]$. Finally, MO_k sends its share $[\![identity_i]\!]_{MO_k}$ to the GA, which combines it with $[\![identity_i]\!]_{GA}$ and recover the identity of $user_i$. In the next subsection, we show how the proposed protocol fulfills the considered privacy requirements under the honest-but-curious security model.

D. Fulfillment of Privacy Requirements

1) Users' Locations: during the contact tracing phase, estimated users' locations are distributed among pairs of MOs as secret shares. As SSS is proven information-theoretic secure [13], no information about locations is obtained from the single shares owned by each MO.

2) Users' Contacts: at each execution of the contact tracing phase, pairs of MOs distribute to each other new shares of their users' locations. This prevents a leakage of users' identities (which cannot be inferred from locations' shares), as well as from counting the number of contacts between two users. Then, during the score computation phase, $MO_{k'}$ can homomorphically compute $Enc_{GA}(s_j + 0)$ (which yields a different ciphertext without altering the hidden infection status), in such a way that MO_k cannot count the number of contacts between $user_i$ and $user_j$.

3) Users' Infection Status and Scores: MO_k computes $user_i$'s scores by performing homomorphic summations on values encrypted by the GA but, since it does not know the private encryption key, it does not discover any plaintext. Then, in the user-triggered communication scenario, $user_i$ sends $Enc_{GA}(s_i \cdot Token_i)$ to the GA. As the latter does not know $Token_i$, it cannot obtain the actual values of the scores. Finally, in the GA-triggered communication scenario, GA and MO_k execute the OT module. From this execution, the GA learns the identity of $user_i$ and MO_k learns nothing. As the GA is considered a honest-but-curious entity, we assume that it executes the OT module only to identify users with the highest chance to be positive (i.e., if $score_i \geq \chi$). Clearly, the GA might execute this module regardless of the value of

$score_i$ and learn the identity and scores of all the users. In the next subsection, we discuss a possible extension of the protocol to cope with this malicious behaviour of the GA.

E. Extension of the protocol for dishonest participants

We now describe how the protocol can be improved to address two malicious schemes. In the first one, the GA tries to obtain the identity of $user_i$ when $Score_i < \chi$. The proposed solution works as follows: MO_k selects two random variables τ_1 and τ_2 and computes $Enc_{GA}(\tau_1 \cdot score_x + \tau_2), \forall x$. These values are then sent to the GA in form of secret share, i.e., $[\![Enc_{GA}(\tau_1 \cdot score_x + \tau_2)]\!], \forall x$ and given in input to the OT module. From its execution, MO_k and the GA obtain $[\![Enc_{GA}(\tau_1 \cdot score_i + \tau_2)]\!]$. MO_k sends its share to the GA, which can then recover the secret $Enc_{GA}(\tau_1 \cdot score_i + \tau_2)$ and, from it, the plaintext $\tau_1 \cdot score_i + \tau_2$. Finally, the GA sends to MO_k both $\tau_1 \cdot score_i + \tau_2$ and $score_i$. Since the GA never obtains the values τ_1 and τ_2, it cannot counterfeit a $score_i \geq \chi$ and a corresponding valid $\tau_1 \cdot score_i + \tau_2$. MO_k detects a cheat if $score_i < \chi$ or the actual $\tau_1 \cdot score_x + \tau_2$ cannot be computed from $score_i$. If the GA does not cheat, the OT module is executed again as previously described, and the GA obtains $identity_i$.

In the second malicious scheme, MO_k counterfeits the encryption infection status of $user_i$. To address this issue, the GA sends to MO_k the infection status of users multiplied by a constant, e.g., $Enc_{GA}(s_i \cdot Token_{GA})$, where $Token_{GA}$ is known to the GA only. The GA detects a cheat if the ciphered score computed by MO_k does not decrypt to a multiple of $Token_{GA}$ (i.e., $Score_i \cdot Token_{GA}$).

VI. Illustrative Numerical Results

A. Simulation Settings

We perform our experiments considering a population of $N = 1.5$ millions users, whose mobility is traced every 20 seconds within an overall period of 1 hour. The initial position of the generic $user_i$ is given by $x_i = R_i cos(\theta_i), y_i = R_i sin(\theta_i)$, being R_i and θ_i two random variables that follow the Gaussian distribution (with zero mean and standard deviation equal to $3800m$) and the uniform distribution defined over $[0, 2\pi]$, respectively. Users move following the Gauss-Markov model [16] (40% of them at an average speed of 0.01m/s, 40% at 1m/s and the remaining 20% at 14m/s). The region occupied by the population is $1900km^2$ large, and is covered by $K \in [2, 5]$ MOs, who have the same number of subscribers $\frac{N}{K}$. 1% of the whole population is assumed to be currently positive.

B. Data Overhead

We now show the overhead generated in each phase of execution of the protocol, being b the bit-length of the shares exchanged by participants (in our simulations $b = 25$ bits).

1) Contact tracing phase: We assume that two users have a contact if their distance is below $th = 2$m. The overhead generated to evaluate if there is a contact is $18b^2 + 10b$. The number of these evaluations depend on the number of users currently located within a given area, which in turns depends on its size. To avoid comparisons among users with a negligible probability to meet, we assume that contacts are searched within non-overlapping squares of size l. In Fig. 3, we show the average and maximum overhead generated by each MO to execute the contact tracing phase over an area of size $l \in \{10, 35, 60, ..., 285\}$ meters. From this figure, we observe a super-linear increase of both the average and maximum overhead per area with increasing l. We also notice that the overhead is higher when decreasing the number of involved MOs K. While the average overhead is always less than 6.8 Mbytes, the maximum overhead grows significantly with l. As an example, when $K = 2$ the maximum overhead goes from 0.01 to 134 Mbytes when l goes from 10 to 285 meters.

2) Score Computation phase: With 4096 bit-long ciphertexts [17], the overhead at each execution of the score computation phase (every 1 hour in our simulations) is 768 Mbytes from the GA to the MOs (i.e., obtained by delivering the infection status of users), and 153 Mbytes among MOs (i.e., obtained by exchanging the infection status of their users in case of contact).

3) Communication with users phase: The overhead generated in the user-triggered communication is negligible (i.e., 1.5 Kbytes/user). On the other hand, the GA-triggered communication generates a total overhead of $14N_\chi N_k b^2 + 2N_\chi N_k b + Nb$ bits, where N_χ is the number of users whose score is $\geq \chi$. For instance, for $K = 5$ and $\chi = 10$ the overhead is 4650 Mbytes. Although this value can be considered acceptable, we note that it would be much higher if longer periods and a higher number of users were considered. To reduce this overhead, the GA sends to the MO both the share $[\![index_i]\!]$ and a range $[index_i - \eta_-, index_i + \eta_+]$ that indicates the rows of Table I in which the identity of $user_i$ should be searched.

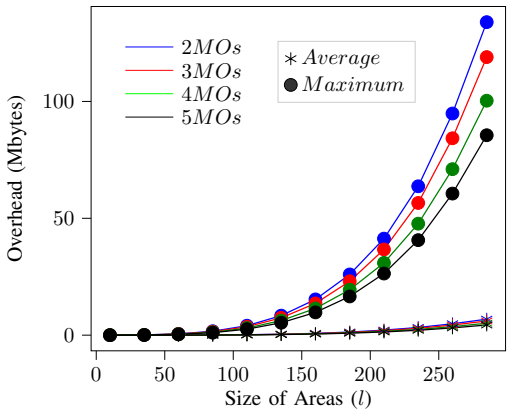

Fig. 3: Overhead of data exchanged by each MO, during the contact tracing phase, within areas of size l

TABLE II: Time needed to perform the contact tracing phase within a given subarea

	Size of Subareas l (meters)											
	10	35	60	85	110	135	160	185	210	235	260	285
Avg Timing (seconds)	$2 \cdot 10^{-5}$	$2.4 \cdot 10^{-2}$	$1.8 \cdot 10^{-1}$	$6.7 \cdot 10^{-1}$	1.7	3.8	7.2	12.4	20.2	30.8	45.7	64.1
Max Timing (seconds)	$7 \cdot 10^{-2}$	1	5.1	15.6	37.7	78.3	143.5	244.1	388.5	599.1	892.7	1260.6

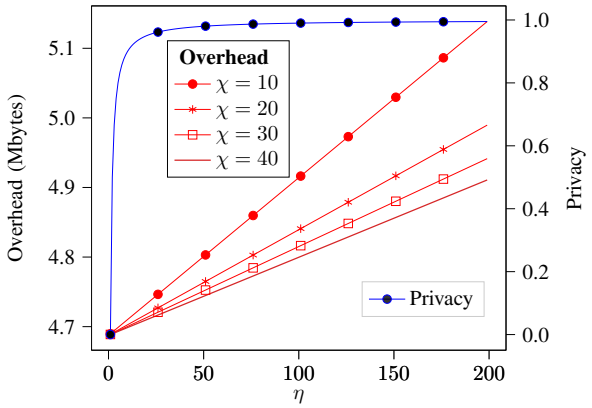

Fig. 4: Trade-off between data overhead and privacy with varying the range η, for several values of the threshold χ and $K = 5$

Since the GA-triggered communication is issued only for users with a score $\geq \chi$, the MO discovers that one among its users with index $\in [index_i - \eta_-, index_i + \eta_+]$ in Table I has a higher-then-average chance to be positive. Hence, there is a trade-off between overhead and $user_i$'s privacy. We measure privacy as the probability that her MO discovers that $user_i$ has a score $\geq \chi$, i.e., $privacy_i = 1 - \frac{1}{\eta}$, where $\eta = \eta_- + \eta_+$. In Fig. 4, we show the trade-off between overhead and privacy with varying $\eta \in [1, 200]$, for $\chi \in [10, 20, 30, 40]$. From this figure, we observe that a very high level of privacy can be reached at a remarkable reduction of the overhead. For instance, for $\chi = 10$ the overhead drops from 4650 to 5 Mbytes if we accept 99.5% of the total privacy.

C. Computational Time

In Table II we show the average and maximum time needed to compute the contacts among users, for several values of l. We note that l should not exceed 85 meters to allow a sampling of users' mobility every 20 seconds. Then, the GA-triggered communication for a single user takes $\tau \cdot \eta$, with $\tau = 6ms$ on a Intel Core I7 computer. When $K = 5$, the identities of users with score ≥ 10 are obtained in 66 minutes if total privacy is considered (i.e., if $\eta = 3 \cdot 10^5$). This value drops to 2.64 seconds if the 99.5% of privacy is considered sufficient.

VII. Conclusion

We proposed a privacy-preserving protocol that enables a GA (owning users' infection status) and several MOs (owning accurate estimations of users' positions) to compute the number of contacts that users have with positive persons, during a considered period. The protocol guarantees that such measure is only obtained by the legitimate user, and that her infection status and mobility data are known, respectively, only to her MO and to the GA. The protocol can also be employed i) to make a user know the number of positive people who stayed in her same area (even thought not in close contact with her) and ii) to make the GA discover the identities only of users with the highest chance to be positive. We evaluated the cost of privacy in terms of overhead generated by the protocol. From extensive simulations, we observed that the overhead is acceptable, and can further be reduced at a negligible reduction of users' privacy.

References

[1] L. Ferretti *et al.*, "Quantifying dynamics of sars-cov-2 transmission suggests that epidemic control and avoidance is feasible through instantaneous digital contact tracing," *medRxiv*, 2020.

[2] R. Raskar *et al.*, "Apps gone rogue: Maintaining personal privacy in an epidemic," *arXiv preprint arXiv:2003.08567*, 2020.

[3] H. Cho *et al.*, "Contact tracing mobile apps for covid-19: Privacy considerations and related trade-offs," *arXiv preprint arXiv:2003.11511*, 2020.

[4] E. Santoro, "Covid-19: il tracciamento dei contatti e il supporto delle nuove tecnologie," *Ricerca & Pratica*, vol. 37, no. 2, pp. 78–81, 2020.

[5] N. Oliver *et al.*, "Mobile phone data and covid-19: Missing an opportunity?" *arXiv preprint arXiv:2003.12347*, 2020.

[6] M. Koivisto *et al.*, "Joint device positioning and clock synchronization in 5g ultra-dense networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 5, pp. 2866–2881, 2017.

[7] T. Scantamburlo *et al.*, "Covid-19 and contact tracing apps: A review under the european legal framework," *arXiv preprint arXiv:2004.14665*, 2020.

[8] D. Andreoletti *et al.*, "Discovering the geographic distribution of live videos' users: A privacy-preserving approach," in *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2018, pp. 1–6.

[9] M. Maouche *et al.*, "Ap-attack: a novel user re-identification attack on mobility datasets," in *Proceedings of the 14th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, 2017, pp. 48–57.

[10] D. Andreoletti *et al.*, "A privacy-preserving protocol for network-neutral caching in isp networks," *IEEE Access*, vol. 7, pp. 160 227–160 240, 2019.

[11] D. Andreoletti *et al*, "An open privacy-preserving and scalable protocol for a network-neutrality compliant caching," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 2019.

[12] S. S. Sathya *et al.*, "Privacy-protective mobile big data analytics and covid-19 response: Challenges and opportunities for telecommunication companies."

[13] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[14] T. Turban, "A secure multi-party computation protocol suite inspired by shamir's secret sharing scheme," Master's thesis, Institutt for telematikk, 2014.

[15] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *International conference on the theory and applications of cryptographic techniques*. Springer, 1999, pp. 223–238.

[16] T. Camp *et al.*, "A survey of mobility models for ad hoc network research," *Wireless communications and mobile computing*, vol. 2, no. 5, pp. 483–502, 2002.

[17] C. Jost *et al.*, "Encryption performance improvements of the paillier cryptosystem." *IACR Cryptology ePrint Archive*, vol. 2015, p. 864, 2015.