Alexandros Bampoulidis, Alessandro Bruni, Lukas Helminger, Daniel Kales, Christian Rechberger, and Roman Walch*

Privately Connecting Mobility to Infectious Diseases via Applied Cryptography

Abstract: Recent work has shown that cell phone mobility data has the unique potential to create accurate models for human mobility and consequently the spread of infected diseases [74]. While prior studies have exclusively relied on a mobile network operator's subscribers' aggregated data in modelling disease dynamics, it may be preferable to contemplate aggregated mobility data of infected individuals only. Clearly, naively linking mobile phone data with health records would violate privacy by either allowing to track mobility patterns of infected individuals, leak information on who is infected, or both. This work aims to develop a solution that reports the aggregated mobile phone location data of infected individuals while still maintaining compliance with privacy expectations. To achieve privacy, we use homomorphic encryption, validation techniques derived from zero-knowledge proofs, and differential privacy. Our protocol's open-source implementation can process eight million subscribers in 70 minutes.

Keywords: homomorphic encryption, COVID-19, mobile data, secure computation, differential privacy, infectious diseases

DOI 10.56553/popets-2022-0132
Received 2022-02-28; revised 2022-06-15; accepted 2022-06-16.

Alexandros Bampoulidis: Research Studio Data Science, RSA FG, Vienna, Austria, alex.bampoulidis@gmail.com
Alessandro Bruni: Katholieke Universiteit Leuven, Belgium, alessandro.bruni@kuleuven.be
Lukas Helminger: Graz University of Technology / Know-Center GmbH, Austria, lukas.helminger@iaik.tugraz.at
Daniel Kales: Graz University of Technology, Austria, daniel.kales@iaik.tugraz.at
Christian Rechberger: Graz University of Technology, Austria, christian.rechberger@tugraz.at
***Corresponding Author: Roman Walch:** Graz University of Technology / Know-Center GmbH, Austria, roman.walch@iaik.tugraz.at

1 Introduction

Human mobility plays a crucial role in infectious disease dynamics. It leads to more contact between receptive and infected individuals and may introduce pathogens into new geographical regions. Both cases can be responsible for an increased prevalence or an outbreak of an infectious disease [72]. In particular, human travel history has been shown to play a critical role in the propagation of infectious diseases like influenza [32] or measles [40]. Therefore, understanding the spatiotemporal dynamics of an epidemic is closely tied to understanding the movement patterns of infected individuals.

Until a few years ago, researchers had to rely on general data, such as relative distance and population distribution, to model human mobility. This model was then combined with a transmission model of a particular disease into an epidemiological model, which was then used to improve the understanding of the geographical spread of epidemics. Mobile phones and their location data have the unique potential to improve these epidemiological models further. Indeed, recent work [74] has shown that substituting this inaccurate mobility data with mobile phone data leads to significantly more accurate models. Integrating such up-to-date mobility patterns allowed them to identify hotspots with a higher risk of contamination, enabling policymakers to apply focused measures.

While prior studies have exclusively relied on aggregated data of all mobile network operator's subscribers' it may be preferable to contemplate aggregated mobility data of infected individuals only. Indeed, a cholera study [33] observed that although their model succeeded in showing that some mass gatherings had major influences in the course of the epidemic, it performed less well when the cumulative incidence is low. They speculated that demographic stochasticity could be one reason for the bad performance of their model. In other words, the infected individuals' mobility pattern may not be precisely reflected by the population's mobility if the prevalence is low. To mitigate this problem, we propose the usage of infected individuals' mobile phone data, which should lead to an improvement in the pre-

dictive capabilities of epidemiological models, especially in highly dynamic situations.

1.1 Privacy Goals

Our system should report a heatmap of aggregated mobile phone location data of infected individuals without revealing an individual's location or whether an individual has been infected. To that end we combine various state-of-the-art privacy-preserving cryptographic primitives to design a two-party client-server protocol for which the epidemiological researcher or a health authority inputs patients' identifiers, and the mobile network operator (MNO) inputs its subscribers' location data.

Our solution should, thereby, be able to combine both datasets without leaking the inputs to the other party. Furthermore, no party should be able to gain any information on the other dataset by cheating during protocol evaluation, e.g., by providing malicious inputs. Even if both parties follow the protocol honestly, the resulting heatmap of aggregated location data can still leak sensitive information about individuals. Thus, our protocol must also prevent this inherent output leakage while still preserving the usefulness of the resulting heatmap.

1.2 Roadmap

In Section 2, we discuss the relevant related work. Section 3 provides the necessary preliminary definitions and notations. Section 4 first states the problem we want to solve in this article. It then gradually develops a solution by introducing privacy protection mechanisms step by step. In Section 5, we perform a dedicated security and privacy analysis of our solution. Section 6 elaborates on the implementation of our solution as well as demonstrating the performance. Section 7 concludes with a discussion about legal considerations for an actual rollout and how multiple parties can be included. We defer to the appendix missing proofs of our security analysis (Appendix A) and additional material regarding differential privacy (Appendix C).

2 Related Work

Numerous research directions have previously sought to model the spread of infectious diseases. Most closely related to this paper is work connecting mobile phone data to infectious diseases.

2.1 Mobility and Infectious Diseases

Mobile phone data provides an opportunity to model human travel patterns and thereby enhance understanding of infectious diseases' transmission [72]. Location data derived from call detail records (CDRs) – phone calls and text messages – have been used to understand various infectious diseases' spatial transmission better, see Table 1. Each of the studies got their CDRs from one MNO, which most of the time had the largest market share and coverage. The common understanding is that biases such as Multi-SIM activity and different mobile phone usage across different geographical and socio-economic groups have a limited effect on general estimates of human mobility [73].

Table 1. Studies connecting mobile phone data to diseases.

	Disease	Country	Year of dataset	Subscribers (millions)	Period (months)
[69]	Malaria	Tanzania	2008	0.8	3
[74]	Malaria	Kenya	2008-09	14.8	12
[48]	HIV	Kenya	2008-09	14.8	12
[75]	Rubella	Kenya	2008-09	14.8	12
[7]	Cholera	Haiti	2010	2.9	2
[68]	Malaria	Namibia	2010-11	1.5	12
[76]	Dengue	Pakistan	2013	39.8	7
[33]	Cholera	Senegal	2013	0.1	12

The most common model was to assign an individual a daily location. More concretely, each subscriber was assigned to a study area on each day based on the cell tower with the most CDRs or the last outgoing CDR. Further, the primary study area ("home") was computed for each subscriber by taking the study area where the majority of days were spent.

All of the studies emphasized that preserving individuals' privacy is mandatory. In all cases – to the best of our understanding – the involved MNO anonymized the CDRs before handing them over to the health authority. In addition, we found that the MNO aggregated the CDRs in at least two cases. However, none of the studies discussed privacy definitions or the potential risk of de-identification, which is exceptionally high for location data [51]. Therefore, it is hard to assess if they achieved their goal of preserving individuals' privacy.

2.2 Exposure Notification

Many technological approaches were developed to help reduce the spread and impact of the Covid-19 pandemic [13, 14, 23, 39, 62, 70, 71]. Most of them focus on exposure notification, where the main challenges include privacy-friendliness, scalability, and utility. These approaches crucially rely on sizable parts of the population using smartphones, enabling Bluetooth, and installing a new app. In contrast, our proposal does not help with contact tracing, but gives potentially useful epidemiological information to health authorities without requiring people to carry around smartphones or installing an app. Indeed, any mobile phone is sufficient.

In subsequent work [45], the authors propose to use a threshold PIR-SUM protocol to allow performing privacy-preserving epidemiological modeling on top of existing contact-tracing information. Their PIR-SUM protocol is based on a multi-server private information retrieval protocol, which is not suitable for our use case where a single entity (e.g., the mobile network operator) holds all location data. While the threshold PIR-SUM protocol can in theory be built using a single-server PIR, these protocols are significantly more expensive than the multi-server PIR they use. Furthermore, their protocols require a mix-net [2] to provide unlinkability of their participants messages, which already requires multiple servers, and it is not immediatly obvious how to apply their ad-hoc MPC protocol to verify the validity of queries to a single-server PIR. For single-server PIR protocols based on homomorphic encryption, our input validation procedure from Section 4.3 might be an alternative.

2.3 PSI-CA and PSI-SUM Protocols

Several works attempt to improve contact tracing by enabling users to query a database, while learning nothing more then the number of intersections using PSI-CA (Private Set Intersection Cardinality) [24, 26, 70] protocols, or while learning nothing except the sum of the associated values of the items in the intersection using PSI-SUM [47, 57] protocols.

While a PSI-SUM protocol perfectly matches our use case in theory, an application of these PSI-SUM protocols in a straightforward fashion for our main scenario in Section 6.6 – where we want to calculate the sum of vectors of length $k = 2^{15}$ for a subset of $n = 2^{23}$ identifiers – would result in impractical communication cost (multiple TB).

In [54], the authors propose a method to build PSI-SUM from their *PIR-with-default* primitive. This approach allows one to greatly reduce the communication to be linear in the smaller set size (the size of the queried subset of the population in our scenario). They present two approaches, where the first one has an expensive setup phase (multiple GB transferred for our scenario) and then has very performant queries. However, our scenario's associated values are temporal location data and would change for new protocol executions, meaning the setup phase would have to be repeated each time. Their second approach does not rely on a setup phase and – for a database size of $n = 2^{25}$ identifiers and $t = 2^{12}$ queried elements – requires 379 MB of data transfer. However, this again only calculates the sum of a single item. A naive $k = 2^{15}$-times repetition of the approach would again result in impractical communication cost. One could investigate if the protocol can be further optimized for large associated data, since the PSI-part of the protocol does not need to be repeated.

An additional problem of the protocol in [54] is, that it cannot ensure that a query does not contain an item multiple times. Applied to our use case, this leads to problems in combination with differential privacy [27], since a larger noise needs to be added for privacy, limiting utility. The protocol in [45] solves the same issue by executing a separate MPC protocol to ensure that the queries are valid and do not contain duplicates.

2.4 Generic Multi-Party Computation

Generic Multi-Party Computation (MPC) protocols allow several parties to securely evaluate a function without having to disclose their respective inputs. Several protocols [21, 22, 37, 78] and efficient implementations for generic MPC exists [1, 49], amongst many others.

We do not use generic MPC since all efficient MPC protocols exchange data linear in the size of the computed circuit and are therefore not well suited for the large databases considered in this work. Both, secret sharing and garbled circuit based MPC, would require the (secure) transmission of the server's database (either in secret-shared form or embedded in a circuit) to the client, requiring several GB of communication (e.g., $2^{23} \times 2^{15}$ matrix of 32 bit integers has a size of 1 TB). Furthermore, the most efficient secret sharing schemes, such as the popular SPDZ [21, 22], require a so-called Beaver triple (for multiplying values) which has to be precomputed in an expensive offline phase and can not be reused. Computing enough triples for our protocol

(i.e., one triple per database entry) would require 2^{38} triples. This triple generation alone would require > 700 hours and more then 1000 TB of communication on our hardware using the MP-SPDZ library [49].

3 Preliminaries

Here we will first introduce the notations and then describe homomorphic encryption (HE) and differential privacy (DP).

We write vectors in bold lower case letters and matrices in upper case letters. We use x_i to access the i-th element of a vector $\boldsymbol{x}$. For $m \in \mathbb{N}$ and $x \in \mathbb{Z}$, let $\boldsymbol{x}^m$ be defined as the vector of powers of x: $\boldsymbol{x}^m = (1, x^1, ..., x^{m-1})$. We denote by $\boldsymbol{c} \circ \boldsymbol{d}$ the element-wise multiplication (Hadamard product) of the vectors $\boldsymbol{c}$ and $\boldsymbol{d}$. For a positive integer p, we identify $\mathbb{Z}_p = \mathbb{Z} \cap [-p/2, p/2)$. For a real number r, $\lfloor r \rceil$ denotes the nearest integer to r.

3.1 Homomorphic Encryption

Homomorphic encryption (HE) [35] allows to operate on encrypted data and, thus, has the potential to realize many 2-party protocols in a privacy-preserving manner. Compared to MPC, protocols using HE usually require less data communication and only one communication round, at the cost of more expensive computations.

Modern HE schemes [8, 9, 16, 17, 31] base their security on the learning with errors [64] hardness assumption and its variant over polynomial rings [56]. They allow to perform both addition and multiplication on ciphertexts. During the encryption of a plaintext, random noise is introduced into the ciphertext. This noise grows with each homomorphic operation, negligible for additions but significantly for homomorphic multiplications. Once this noise becomes too large and exceeds a threshold, the ciphertext cannot be decrypted correctly anymore. We call such a scheme, that allows evaluating an arbitrary circuit over encrypted data up to a certain depth, a somewhat homomorphic encryption scheme (SHE). The specific depth depends on the choice of encryption parameters, and choosing parameters for larger depths comes, in general, with a considerable performance penalty. In this work, we use the BFV [8, 31] SHE scheme to encrypt the inputs of our protocol.

Besides semantic security, two-party protocols based on HE additionally either require the notion of circuit privacy [35], or function [36] (evaluation [3]) privacy, to hide the function applied on the ciphertexts. Function privacy is often easier to achieve than circuit privacy in practice. It requires that the outputs of evaluating different circuits homomorphically on the same encrypted data need to be indistinguishable. In other words, a party decrypting the final result of a function private HE scheme can not learn anything about the circuit applied to the input data. Like many HE schemes, BFV does not naturally provide function privacy, however, it can be added by applying noise flooding [35].

3.2 Differential Privacy

Differential privacy (DP) [27] defines a robust, quantitative notion of privacy for individuals. The main idea is that the outcome of a computation should be as independent as possible (usually defined by a privacy parameter ϵ) from the data of a single input. Applied to our use case, DP makes the final heatmap independent to the contribution of any individual, preventing it from leaking sensitive information.

DP is highly compatible with existing privacy frameworks and has successfully been applied to several real-world applications. Recent work [60] showed that DP satisfies privacy requirements set forth by FERPA[1]. Even before this analysis, several businesses were already using DP. For example, Apple [5] and Google [38] have applied differential privacy to gather statistics about their users without intruding on individual's privacy. Recently, the U.S. Census 2020 uses differential privacy as a privacy protection system [11].

The most prevalent technique to achieve DP is to add noise sampled from a zero-centered Laplace distribution to the outcome of the computation. The distribution is calibrated with a privacy parameter ϵ and the global sensitivity Δq of the computation and has the following probability density function:

$$Lap(x|b) = \frac{1}{2b} e^{-\frac{|x|}{b}}, \qquad \text{with } b = \frac{\Delta q}{\epsilon}$$

To add noise to a protocol operating on integers, we discretize the Laplace distribution by rounding the sampled value to the nearest integer. For a formal definition of DP, we refer to Appendix C.

1 Family Educational Rights and Privacy Act of 1974, U.S.

4 Privacy Preserving Heatmap Protocol

We first describe our goal and then introduce each privacy protection mechanism step by step.

4.1 The Desired Functionality

Our aim is to accumulate the location data of infected individuals to create a heatmap, assisting governments in managing an epidemic. Two parties controlling two different datasets are involved: A health authority who knows which individuals are infected; and an MNO who knows the location data of their subscribers. More specifically, the MNO knows how long each of their subscribers is connected to which cell towers (CDRs are a subset of this information). Based on this mobility data, our protocol could answer several questions. One in line with epidemiological literature is to look at the individuals' mobility data in the incubation period (e.g., 5-7 days for COVID-19). The final heatmap will show areas with a higher chance of getting infected with the disease. A natural extension would be to study mobility patterns after the incubation period but before confirmation/quarantine. So our protocol is generic regarding the time unit or the granularity of location data. When discussing privacy guarantees that depend on the actual data, we will explicitly outline the chosen setting.

Protocol Description

If the MNO knows which of its subscribers is infected, it can do the following to create the desired heatmap:

- Initialize a vector $\boldsymbol{h}$ of k elements with zeros, where k is the total number of cell towers. Each element of this vector corresponds to one cell tower.
- For each infected individual, add the amount of time it spent at each cell tower to the corresponding element of the vector $\boldsymbol{h}$.
- Then the vector $\boldsymbol{h}$ contains the final heatmap, i.e., h_j contains the accumulated time spent of all infected individuals at cell tower j.

Let us rewrite this process into a single matrix multiplication. First, we encode all N subscribed individuals into a vector $\boldsymbol{x} \in \mathbb{Z}_2^N$, with $x_i \in \mathbb{Z}_2$ indicating, whether the individual i is infected ($x_i = 1$) or not ($x_i = 0$). Then we encode the location data in a matrix $Z = (\boldsymbol{z_1}, \boldsymbol{z_2}, \ldots, \boldsymbol{z_k}) \in \mathbb{Z}^{N \times k}$ such that the vector $\boldsymbol{z_j}$ contains all the location data corresponding to the cell tower identified by j. In other words, the i-th element of the vector $\boldsymbol{z_j}$ contains the amount of time individual i spent at cell tower j. Now, we can calculate the heatmap as $\boldsymbol{h} = \boldsymbol{x}^T \cdot Z$.

We depict the basic protocol, involving the health authority as a client and the MNO as a server, in Figure 1, assuming the health authority and the MNO already agreed on identifying all subscribed individuals by indices $i \in 1, ..., N$.

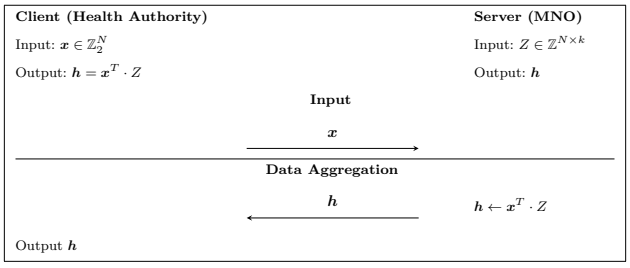


Fig. 1. Basic protocol without privacy protection.

Remark 1 (Agreeing on database indices)**.** *The protocol in Figure 1 already assumes that the two parties agree on the indices of individuals in the database. In practice, the individuals can be identified using several methods, such as phone numbers, mail addresses or government ids. We now give two options to get a mapping from a phone number to a database index, while noting that any other identifier can be trivially used instead:*

- *The MNO and health authority engage in a protocol for Private Set Intersection (PSI) with associated data (e.g., [15, 20]). In such a protocol, the health authority and the MNO input their list of phone numbers. The health authority gets as the protocol's output the phone numbers that are in both sets, as well as the associated data from the MNO. The associated data would be the index in the database in our case.*
- *The MNO sends a mapping of all phone numbers to their database index in plain. This approach is efficient and straightforward, but it discloses all subscribed individuals to the health authority. However, this is essentially a list of all valid phone numbers in random order and does not leak anything more than the validity of that number. Still, this may be an issue in some scenarios.*

While the PSI-based solution has some overhead compared to the plain one, the performance evaluation

in [20] shows that a protocol execution with 2^{22} *MNO items and* 4096 *health authority items takes about* 1.4 *seconds online (excluding a precomputable offline phase taking* 467 *seconds) with a total communication of* 8.3 *MB – a minor increase when looking at the overall protocol. While PSI-SUM protocols [47, 54, 57] could be used to calculate the final heatmap without revealing which identifiers are present in the MNO's set, their additional overhead is not worth the minor privacy gain, considering that for the type of identifier we are using (phone numbers), one can often already publicly check if a phone number is associated with a mobile network operator*[2]*. We therefore relax our setting to allow revealing which identifiers are present in the MNO's set to take advantage of the reduced communication of our approach compared to the full PSI-SUM approaches mentioned above.*

Executing the protocol described in Figure 1 would enable the MNO to learn about infected individuals, which is a massive privacy violation. On the other hand, the health authority could query a single individual's location data by sending a vector $\boldsymbol{x} = (1, 0, \ldots, 0)$, violating privacy. In the following, we describe our techniques to protect against these violations.

4.2 Adding Encryption

To protect the vector send by the health authority, and therefore who is infected and who is not, we use a HE scheme $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$. Before executing the protocol, the health authority runs KGen to obtain a secret key sk and an evaluation key evk. We assume that the MNO knows evk, which is required to perform operations on encrypted data, before running the protocol.

In the updated protocol, the health authority now uses sk to encrypt the input vector $\boldsymbol{x}$ and sends the resulting ciphertext vector $\boldsymbol{c} \leftarrow \mathsf{Enc}_{\mathsf{sk}}(\boldsymbol{x})$ to the MNO. The MNO then uses evk to perform the matrix multiplication on the encrypted input vector and sends the resulting ciphertext vector $\boldsymbol{h}^* \leftarrow \mathsf{Eval}_{\mathsf{evk}}\left(\boldsymbol{c}^T \cdot Z\right)$ back to health authority. The health authority can now use sk to decrypt the result and get the final heatmap $\boldsymbol{h} = \mathsf{Dec}_{\mathsf{sk}}(\boldsymbol{h}^*)$.

Informally, if the used HE scheme is semantically secure, then the MNO cannot learn which individuals are infected by the disease and which are not.

2 as an example, using services such as https://dexatel.com/carrier-lookup/, or often also just calling the number

4.3 Input Validation

In the simple protocol, the health authority could use a manipulated input vector $\boldsymbol{x}$ to include an individual multiple times (e.g., setting the corresponding vector entry to 100 instead of 1). Such an individual could most likely be filtered out in the final heatmap. Since the input vector is encrypted, the MNO cannot trivially check if the vector is malicious or not. Also, comparing encrypted elements is not trivially possible in most HE schemes. However, the required check can be encoded, such that it outputs 0 if everything is correct, and a random value otherwise. We then can add this value to the final output as a masking value which randomizes the MNO's response if the input vector is malicious. We describe how to generate this masking below.

Masking Against Non-Binary Query Vector

Note that the HE schemes plaintext space usually is $\mathbb{Z}_p$, i.e., the integers modulo a prime p. Therefore, the inputs to our protocol – the vector $\boldsymbol{x}$ and the matrix Z – consist of elements in $\mathbb{Z}_p$. As outlined above, it is crucial to the protocol's privacy that the input vector is binary, i.e., only contains 0s and 1s. If this is not the case, the health authority could arbitrarily modify a single person's contribution to the overall aggregated result. It is essential for DP considerations to bound the maximum possible contribution of a single individual (sensitivity).

Since the MNO only receives an encryption of the input vector, simply checking for binary values is not an option. However, we can use similar techniques to the ones used in Bulletproofs [10] to provide assurance that the query vector $\boldsymbol{x} \in \mathbb{Z}_p^N$ contains only binary elements. First, we will exploit the following general observation. Let $\boldsymbol{d} = \boldsymbol{x} - \mathbf{1}$, then $\boldsymbol{x} \circ \boldsymbol{d}$ is the zero vector iff $\boldsymbol{x}$ is binary. Note that the MNO can compute an encryption of $\boldsymbol{d}$ from the encrypted input vector. The result of the Hadamard product $\boldsymbol{x} \circ \boldsymbol{d}$ can be aggregated into a single value by calculating the inner product $\langle \boldsymbol{x}, \boldsymbol{d} \rangle$, which will again be zero if $\boldsymbol{x}$ is binary. The MNO also multiplies $\boldsymbol{x}$ with powers of a random integers y to reduce the probability of the health authority cheating by letting several entries of $\boldsymbol{x}$ cancel each other out during the inner product, which gives the mask:

$$\mu_{\mathtt{bin}'} = \langle \boldsymbol{x}, (\boldsymbol{d} \circ \boldsymbol{y}^N) \rangle \,, \tag{1}$$

where $\boldsymbol{y}^N = (1, y^1, ..., y^{N-1})$ is y's vector of powers.

For the generic case of a vector $\boldsymbol{v}$ and a randomly chosen y, $\langle \boldsymbol{v}, \boldsymbol{y}^N \rangle = 0$ will hold for $\boldsymbol{v} \neq 0$ only with

probability N/p [10]. Using a ν bit modulus p ($p \approx 2^\nu$), translates to a soundness error of $\nu - \log_2(N)$ bits. For details of this calculation see Appendix A.1. In particular, if we look at $N = 2^{23}, \nu = 60$, parameters sufficient for small nation-states (see Section 6.6), we get 37-bit statistical security. Standard literature suggest a statistical security parameter of at least 40-bit; therefore, we developed a method to enhance the statistical security without significant overhead.

Boosting Soundness

The high-level idea is that we lower the probability of cheating successfully by a random linear combination of separate masks. Intuitively, a malicious health authority would have to guess correctly for every single mask. Thus, the soundness converges to the underlying field size in the number of terms of the linear combination. For our purpose, two terms already suffice for an appropriate security level:

$$\mu_{\texttt{bin}} = \langle \boldsymbol{x}, (\boldsymbol{d} \circ \boldsymbol{y_1^N}) \rangle \cdot r_1 + \langle \boldsymbol{x}, (\boldsymbol{d} \circ \boldsymbol{y_2^N}) \rangle \cdot r_2$$

where $r_1, r_2 \xleftarrow{\$} \mathbb{Z}_p \setminus \{0\}$ are two random values. Therefore, the statistical security level increases to $\nu - 1$ bit ($= 59$ bits for $\nu = 60$). We refer to Lemma 2 in Appendix A.1 for a proof of this statement.

Applying the Mask

Once the $\mu_{\texttt{bin}}$ is calculated, it gets added to the final output of the protocol. However, if the masking value is not zero, we have to make sure that a different random value is added to each element of the output vector to prevent leaking the mask if some output vector values are known beforehand. Therefore, the final mask $\boldsymbol{\mu}$ can be calculated using a random vector $\boldsymbol{r} \xleftarrow{\$} (\mathbb{Z}_p \setminus \{0\})^k$ as follows:

$$\boldsymbol{\mu} = \mu_{\texttt{bin}} \cdot \boldsymbol{r} \tag{2}$$

The final mask $\boldsymbol{\mu}$ is now equal to $\boldsymbol{0}^k$ if $\boldsymbol{x}$ is a binary vector, random otherwise.

Remark 2 (PSI-SUM with Indices)**.** *So far the protocol securely implements a functionality dubbed PSI-SUM with Indices. For completeness, we included a description and its ideal functionality in Appendix B.*

4.4 Adding Differential Privacy

The aggregated location data can still leak information about the location data of individuals. For example, the health authority could abuse the heatmap to track an individual by just querying him or by querying him alongside individuals from a completely different area. The location data of the targeted individual would be visible as an isolated zone in the resulting heatmap. Applying DP with suitable parameters will protect against such an attack since the overall goal of DP is to decrease the statistical dependence of the final result to a single database entry. In our use case, therefore, DP achieves that it is highly unlikely to distinguish between heatmaps, in which we include a single individual in the accumulation, and heatmaps, in which we do not.

Choosing proper parameters, however, highly depends on the underlying dataset. On the one hand, the chosen ϵ should be small enough to satisfy privacy concerns; on the other hand, it should be big enough not to overflow the result with noise, creating hotspots on its own. We discuss one method to choose suitable parameters in Section 5.2.

4.5 Final Protocol

Finally, with all measures to protect privacy in place, we present the final protocol in Figure 2.

5 Security & Privacy Analysis

On the one side, the protocol provides input security against a malicious MNO, i.e., even if the MNO deviates from the protocol, it cannot determine the patient's identifiers (see Section 5.1). On the other side, individuals' location data are protected even against a malicious health authority, i.e., the health authority cannot track individuals (see Section 5.2)).

5.1 Security

Two-party protocols are usually proven secure with the real-ideal world paradigm [12]. Roughly speaking, one has to prove that the protocol does not leak any additional information than when computed with the help of a trusted third party. The trusted third party is modeled as an ideal functionality presented in Figure 3.

Semi-Honest Security

Before we discuss malicious security, we will show that our protocol achieves semi-honest security.

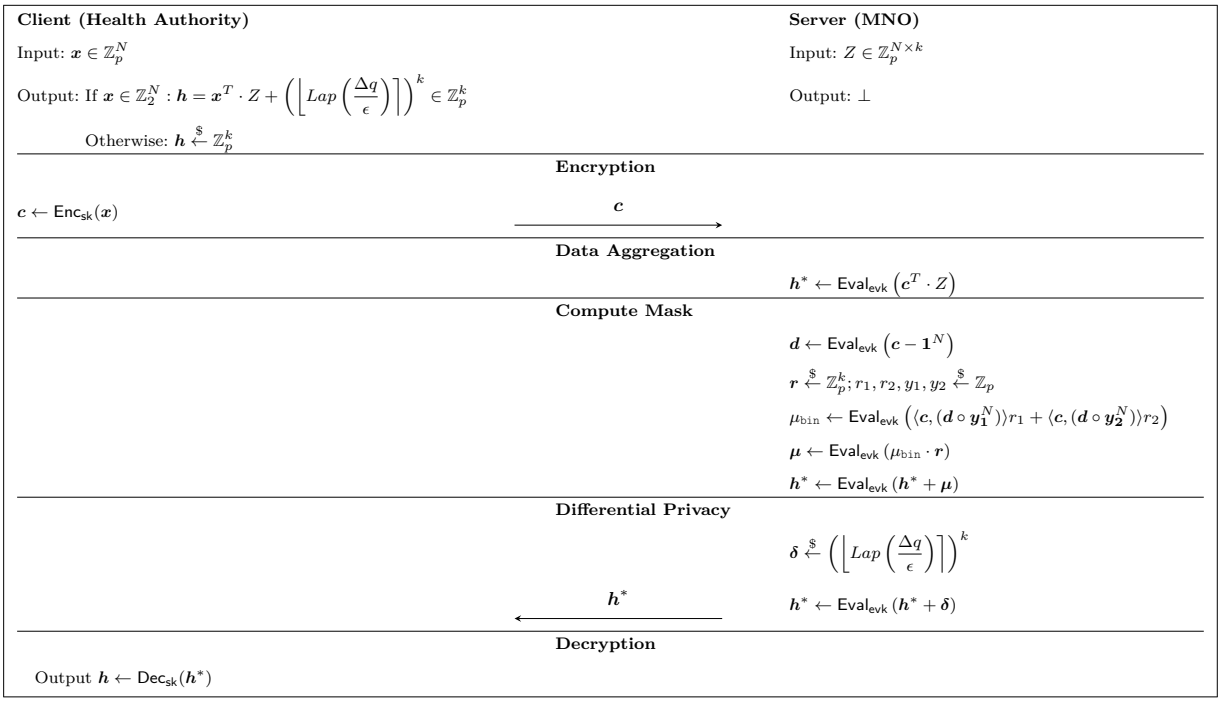


Fig. 2. Privacy preserving heatmap protocol.

Lemma 1. *Let us assume* HE *is an IND-CPA secure homomorphic encryption scheme that provides function privacy. Then the protocol in Figure 2 securely realizes* $\mathcal{F}_{Hmap}$ *against static semi-honest adversaries.*

The high-level idea is two-fold. Firstly, by the definition of semantic security, the MNO can not learn anything from encrypted data, hence, we reduce our protocol's security against the MNO's corruption to the semantic security of the underlying HE scheme. Second, function privacy guarantees that the health authority learns nothing more about the MNO's matrix, than what can be derived from the input $\boldsymbol{x}$ and the output $\boldsymbol{h}$. The formal proof can be found in Appendix A.

Malicious Security

Achieving simulation-based security against a malicious MNO would be similar to verified HE. While some theoretical constructions exist [52], they are not practical.

Instead, we show input security against a malicious MNO, also known as one-sided simulation security. This notion has been first considered in the context of oblivious transfer [59], was then formalized [44], and recently used [15] in the realm of PSI. In our protocol, one-sided simulation guarantees that the patients' identifiers are protected even in the presence of a malicious MNO (one that deviates from the protocol). For a formal definition, see Appendix A.

Theorem 1. *Let us assume* HE *is an IND-CPA secure homomorphic encryption scheme that provides function privacy. Then the protocol in Figure 2 securely realizes* $\mathcal{F}_{Hmap}$ *with one-sided simulation in the presence of a maliciously controlled MNO.*

Proof. From Lemma 1, we already know that the protocol is secure against semi-honest adversaries. The only thing left to show is input privacy of the health authority against a malicious MNO, i.e., the MNO is not able to learn any information from the health authority's input (patients' identifier). Now, due to the fact that the MNO's view only includes an encryption of the health authority's input, by the semantic security of HE, we have that the MNO learns nothing about the health authority's input. □

5.2 Privacy

The protocol's output exposes aggregate information, namely the amount of time spent by individuals at a cell tower, to the health authority. In the worst case, only one individual is present in the aggregation. Even in this case the health authority should not be able to

$\mathcal{F}_{Hmap}$

Parameters: $t, N \in \mathbb{N}, \beta \in \mathbb{R}_+$. Parties P_1 and P_2.

1. Upon receiving an input $(\mathsf{input}, sid, P_1, P_2, \boldsymbol{x})$ from a party P_1, verify that $\boldsymbol{x} \in \mathbb{Z}_p^N$, else ignore input. Next, record $(sid, P_1, P_2, \boldsymbol{x})$. Once $\boldsymbol{x}$ is recorded, ignore any subsequent inputs of the form $(\mathsf{input}, sid, P_1, P_2, \cdot)$ from P_1.
2. Upon receiving an input $(\mathsf{input}, sid, P_1, P_2, Z)$ from party P_2, verify that $Z \in \mathbb{Z}_p^{N\times *}$, else ignore input. Proceed as follows: If there is a recorded value $(sid, P_1, P_2, \boldsymbol{x}, w)$, compute $\boldsymbol{h} \leftarrow \boldsymbol{x}^T Z + (\lfloor Lap(\beta) \rceil)^k$ provided that $\boldsymbol{x} \in \mathbb{Z}_2^N$, otherwise $\boldsymbol{h} \xleftarrow{\$} \mathbb{Z}_p^k$, and send (sid, P_1, P_2, k) where k is the number of columns of Z to the adversary. Then output $(\mathsf{result}, sid, P_1, P_2, \boldsymbol{h})$ to P_1, and ignore subsequent inputs of the form $(\mathsf{input}, sid, P_1, P_2, \cdot)$ from P_2.

Fig. 3. Ideal functionality $\mathcal{F}_{Hmap}$ of the above solution.

single out any individual. To mitigate this threat, we propose to use *differential privacy* (DP).

5.2.1 Privacy-Utility Tradeoff

It is a challenge to choose the right amount of noise to protect individuals' privacy while still preserving utility. Ultimately, this tradeoff is not only technical but also has to take into account normative considerations [60]. Here, we only explore the technical tradeoff and leave the latter one to policymakers.

There has been limited research specifically addressing the technical tradeoff [50, 53, 61]. However, the methods of [50, 53] are not applicable to our protocol since they require either input from the individuals [50] or "*knowing the queries to be computed*" [53]. Therefore, we choose to follow [61]'s rigorous method to find real-world parameters for DP.

5.2.2 Choosing the Right ϵ

The model in [61] provides a principled approach to choose ϵ. It can be split into two major steps resulting in two constraints that have to be satisfied simultaneously. First, one chooses the desired utility by defining a confidence interval. The parameters of the confidence interval give the first constraint on the required minimum number w of infected individuals and ϵ. Note that if the health authority does not provide the number of infected individuals w or lies about it, privacy is not affected. Only utility cannot be guaranteed any more.

Further, the method requires setting a bound on the expected privacy harm per individual and estimating the expected cost (baseline cost) of not being part of the outcome (e.g., database breach). This leads to the second constraint on ϵ. Every pair of parameters, ϵ, and the number of infected individuals w that simultaneously fulfill both constraints, is a reasonable privacy-utility tradeoff choice.

To illustrate this method, we now provide a possible set of values for this example. Choosing these values requires a few assumptions. We want to highlight that our assumptions are, at best, educated guesses. The real-world values have to be adjusted to the concrete circumstances and be discussed by a group of privacy, ethical, legal, epidemiological and policy experts.

First, the time unit is days for consistency with previous epidemiological studies, see Section 2.1. We decided to aim for a margin of error of $\pm 5\%$ with a probability of 0.95 (confidence). In terms of privacy, the method requires us to estimate the expected base costs (harm) that arise for an individual by using the MNO's services (data breach at the MNO would leak the location data), i.e., without even being part of the computation. We assume that without performing our protocol, this probability is less than 0.00001. In the case of a leakage, we set the monetary harm inflicted to an individual to an exemplary amount of \$1000 per day. This seems reasonable since most smartphone users divulge exact location data for far less than that amount to companies. Now, we can calculate the expected baseline cost as $0.00001 \cdot \$1000 = \$0.01$ per day of leakage. We think performing the protocol is justified if the cost of participating does not exceed \$0.02. We arrive at the following two constraints (see Appendix C.1 for details)

$$\exp\left(-0.05 \cdot \frac{w\epsilon}{2}\right) \leq 0.05 \qquad \text{(utility)}$$

$$0.01 \cdot (e^{\epsilon} - 1) \leq 0.02, \qquad \text{(privacy)}$$

which are illustrated in Figure 4.

If the health authority wants to release a heatmap to inform the public about hotspots or justify their policies, it must add additional noise to the map. Otherwise, the MNO could subtract the noise, which itself added in the first place, thus removing the protection provided by DP. The addition of noise by both parties does not violate privacy because DP enjoys composability [28].

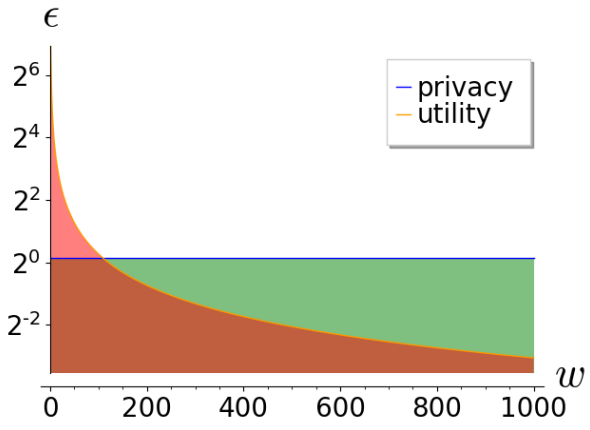


Fig. 4. Privacy-utility tradeoff: The green area are possible combination for ϵ and w (# infected individuals). Above privacy cannot be guaranteed; below utility is not satisfied.

More concretely, if the heatmap produced by the MNO is ϵ_1-differentially private and the health authority adds noise corresponding to ϵ_2 to it, then the final heatmap is $(\epsilon_1+\epsilon_2)$-differentially private. The same methodology as above should be applied to choose ϵ_2. It is crucial to find parameters such that the points (w, ϵ_1), (w, ϵ_2), and $(w, \epsilon_1 + \epsilon_2)$ are in the plot's green area.

To illustrate the trade-off of Figure 4 for a practical example, we performed experiments on the London subset of the publically available *gowalla* dataset [18]. This dataset consists of thousands of check-in's where each check-in consists of a user-id, GPS coodinates and a timestamp. To stay consistent with the methodology we discuss in the rest of the paper, we mapped all check-in locations to the locations of the nearest cell towers in London and treat multiple check-ins from the same user to the same cell tower as just one check-in. The final dataset consists of 4571 people and 9994 cell towers. Figure 5 depicts a snipped of the original heatmap and the heatmaps resulting by applying DP, having $w = 600$ randomly chosen infected people and varying ϵ. The generated figures visually confirm our expectations based on the calculations above: One can observe that the heatmap without DP (Figure 5a) is very similar to the heatmap with too little noise (Figure 5d), indicating that the noise is not enough to guarantee privacy. On the other hand, the heatmap with too much noise (Figure 5b) clearly provides no utility due to the noise creating too many hotspots. In the correctly parameterized heatmap (Figure 5c), one can observe some difference to Figure 5a due to noise, however the biggest hotspots remain the same. In other words, privacy and utility are preserved. Government officials now can use Figure 5c to set new policies (e.g., closing public locations in the hotspot areas) without the possibility to track the location of individuals.

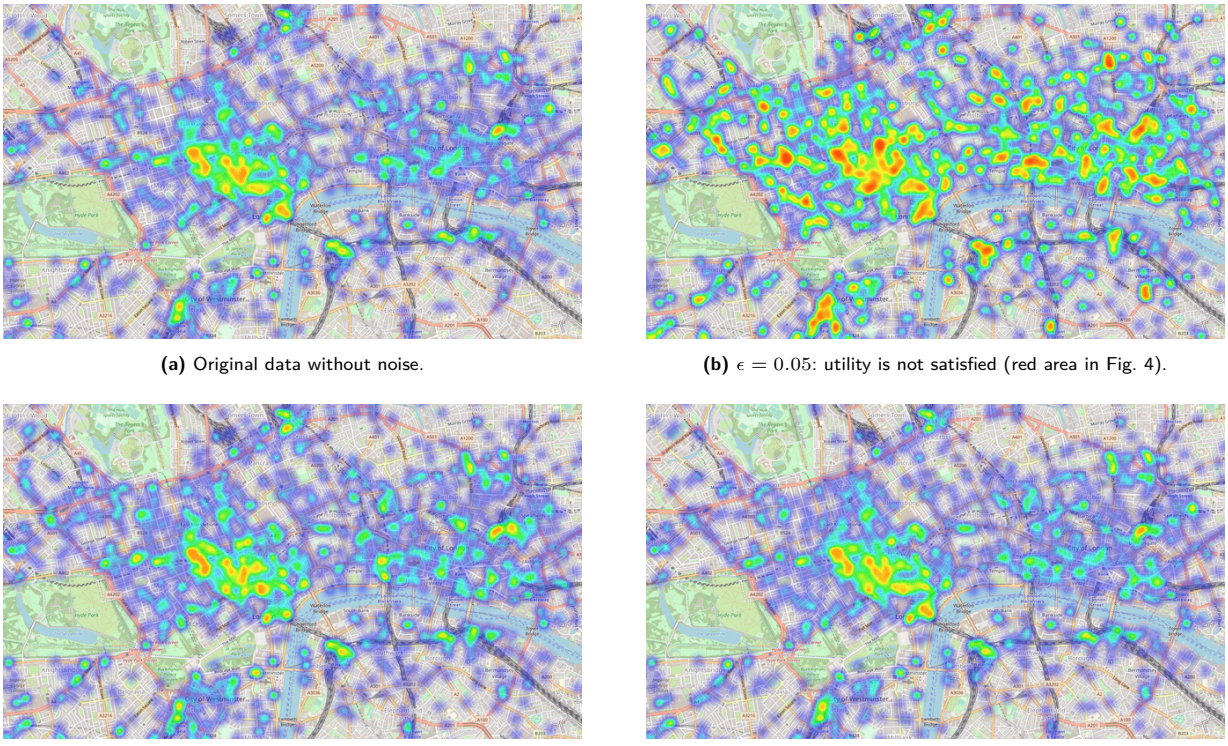
Remark 3. *Several queries could contain the same individual. Since the overall movement pattern for the same individual changes slowly over time, we model this as an identical database. Therefore the total number of queries has to be limited to the total privacy budget. For example, if we follow the values of the analysis above and the health authority queries once a week for two months (= 8 queries), the privacy budget suffices to provide utility as long as the number of infected individuals w is above 750 per week.*

5.3 Summary and Limitations

To summarize, the patients identifiers are encrypted during the whole protocol, hence, the semantic security of the HE scheme protects the privacy of the patients even if the MNO is cheating. The functional privacy of the HE scheme prevents, that the MNO's computation leaks anything about any location data to the health authority. The binary mask guarantees that each individual is only present at most once in the query and prevents that a cheating health authority can amplify the contribution of individual's location data in the final heatmap. Differential privacy then prevents that location data from individual's can be singled out from the resulting heatmap. Consequently, the location data of individuals is protected even if the health authority is cheating. Hence, all sensitive information is always kept private from other parties during the whole protocol.

Even though privacy of input data is guaranteed, the protocol has some practical limitations. The protocol cannot guarantee, that either the health authority, or the MNO use truthful data in the first place. In other words, malicious health authorities can randomly mark individuals as infected and MNO's can use fake location data to create the heatmap. The protocol then would guarantee privacy of these wrong inputs, but the produced heatmap would be useless. This dependence on the truthfullness of the input data is, unfortunately, a generic problem to *any* computation (plain and privacy preserving) and can not be prevented by cryptographic measures. We, therefore, propose that independent officials perform a yearly audit, e.g., at the end of the year, of the involved data to expose cheating parties.

Another limitation of our protocol is, that the utility of the heatmap scales with the prevalence of the disease.



(a) Original data without noise.

(b) $\epsilon = 0.05$: utility is not satisfied (red area in Fig. 4).

(c) $\epsilon = 0.6$: utility is satisfied and privacy is guaranteed (green area in Fig. 4).

(d) $\epsilon = 3$: privacy is not guaranteed (above the privacy line in Fig. 4).

Fig. 5. Influence of different ϵ values on an artificial heatmap created by mapping the *gowalla* [18] dataset onto Londoner cell towers. Figure 5a shows the unmodified heatmap providing no privacy. While Figure 5d has too little noise for privacy (practically no difference compared to Figure 5a), Figure 5b has too much noise for utility. Figure 5c provides both privacy and utility. While the noise clearly influences the image, the hotspots remain the same.

Concretely, the more people are infected, the smaller the impact of differential privacy on the final outcome. Conversely, the less people are infected the larger the impact of the noise and the utility drops. Thus, for very small prevalences it might not be possible to achieve high utility while maintainig privacy with our protocol.

6 Implementation & Performance

The data aggregation of our protocol requires only homomorphic ciphertext-ciphertext addition and homomorphic plaintext-ciphertext multiplication; however, the evaluation of the binary mask additionally requires homomorphic ciphertext-ciphertext multiplication. For our implementation we chose to use the BFV [8, 31] SHE scheme, which fulfills these requirements. More specifically we use its implementation in the SEAL v3.6 [65] library, a fast, actively developed open-source library maintained by Microsoft Research.

The computationally most expensive phase in the protocol is the Data Aggregation phase, in which the MNO multiplies a huge matrix to a homomorphically encrypted input vector. Therefore, the main objective of our implementation is to perform this huge matrix multiplication as efficiently as possible.

6.1 Packing

Modern HE schemes (including BFV) allow packing a vector of n plaintexts into only one ciphertext. Performing an operation on this ciphertext then is implicitly applied to each slot of the encrypted vector, similar to single-instruction-multiple-data (SIMD) instructions on modern CPUs (e.g., AVX). However, the ciphertext size does not depend on the exact number ($\leq n$) of encoded plaintexts. The HE schemes support various SIMD operations, including slot-wise addition, subtraction and multiplication, and slot-rotation. However, one can not directly access a specific slot of the encoded vector. We can use the SIMD encoding to speed up the matrix multiplication of our protocol significantly.

In the BFV scheme (and its implementation SEAL), the number of available SIMD slots equals the degree of the cyclotomic reduction polynomial $(x^n + 1)$; thus, it is always a power of two. In the ciphertexts, the n slots are arranged as matrix of dimensions $(2 \times n/2)$. A ciphertext rotation affects either all rows or all columns of the matrix simultaneously. Therefore, we can think of the inner matrix as two rotatable vectors, which can be swapped.

6.2 Homomorphic Matrix Multiplication

Since SEAL does not provide algorithms for plain-matrix times encrypted vector multiplication, we implement the baby-step giant-step (BSGS) optimized matrix-vector multiplication [41–43] on our own and optimize it to fully leverage all slots (i.e., both rotatable vectors) of the homomorphic ciphertexts.

BSGS Matrix Multiplication

The SIMD encoding can be used to efficiently speed up matrix multiplication by using the diagonal method introduced by Halevi and Shoup [41], and its optimized version based on the BSGS algorithm [42, 43]:

$$Z \cdot \boldsymbol{x} = \sum_{i=0}^{m-1} \texttt{diag}(Z, i) \circ \texttt{rot}(\boldsymbol{x}, i) \tag{3}$$

$$= \sum_{k=0}^{m_2-1} \texttt{rot}\left(\sum_{j=0}^{m_1-1} \texttt{diag}'(Z, km_1 + j) \circ \texttt{rot}(\boldsymbol{x}, j), km_1 \right)$$

where $m = m_1 \cdot m_2$ and $\texttt{diag}'(Z, i) = \texttt{rot}\left(\texttt{diag}(Z, i), -\lfloor i/m_1 \rfloor \cdot m_1\right)$.[3] Note, that $\texttt{rot}(\boldsymbol{x}, j)$ only has to be computed once for each $j < m_1$, therefore, Equation (3) only requires $m_1 + m_2 - 2$ rotations of the vector $\boldsymbol{x}$ in total.

Extension to Bigger Dimensions

In our protocol, we want to homomorphically evaluate $\boldsymbol{x}^T \cdot Z = (Z^T \cdot \boldsymbol{x})^T$, where $\boldsymbol{x} \in \{0, 1\}^N$ and $Z \in \mathbb{Z}_p^{N \times k}$, for big parameters N and k. As described in Section 6.1, the inner structure of the BFV ciphertext consists of two vectors of size $n/2$ each, and it does not allow a cyclic rotation over the whole input vector of size n. However, a rotation over the whole input vector is required by the BSGS algorithm. Therefore, we only can perform a BSGS multiplication with a $(n/2 \times n/2)$ matrix using this packing. Fortunately, we can use the remaining $n/2$ slots (i.e., the second vector in the inner structure of the BFV ciphertext) to simultaneously perform a second $(n/2 \times n/2)$ matrix multiplication. Therefore, after a homomorphic BSGS matrix multiplication, the result is a ciphertext c, where each of the two inner vectors encodes the result of a $(1 \times n/2) \times (n/2 \times n/2)$ vector-matrix multiplication. The sum of those two vectors can easily be obtained by rotating the columns of the ciphertext c and adding it to the first result:

$$c_{sum} = c + \texttt{rot}_{\texttt{col}}(c) \tag{4}$$

Thus, we can use one $(n/2 \times n/2)$ BSGS matrix multiplication and Equation (4) to implement a homomorphic $(1 \times n) \times (n \times n/2) = (1 \times n/2)$ vector-matrix multiplication.

Taking this into account, we split the huge $(N \times k)$ matrix into $n_v \cdot n_o$ submatrices of size $(n \times n/2)$, with $n_v = \left\lceil \frac{N}{n} \right\rceil$ and $n_o = \left\lceil \frac{2k}{n} \right\rceil$, padding the submatrices with zeros if necessary. We split the input vector $\boldsymbol{x}$ into n_v vectors of size n (padding the last vector with zeros if necessary) and encrypt each of these vectors to get n_v ciphertexts c_i. The final result of the $\boldsymbol{x}^T \cdot Z$ matrix multiplication can be computed with the following equation:

$$\tilde{c}_i = \sum_{j=0}^{n_v-1} \texttt{MatMul}(\texttt{SubMat}(Z, j, i)^T, c_j) \;\; \forall 0 \leq i < n_o \tag{5}$$

where, $\texttt{SubMat}(Z, j, i)$ returns the submatrix of Z with size $(n \times n/2)$, starting at row $n \cdot j$ and column $\frac{n}{2} \cdot i$, and $\texttt{MatMul}(Z, c)$ performs the homomorphic BSGS matrix multiplication $Z \cdot c$ followed by Equation (4).

Equation (5) produces n_o ciphertexts $\tilde{c}_i$, with the final results being located in the first $n/2$ slots of the ciphertexts. Overall, our algorithm to homomorphically calculate $\boldsymbol{x}^T \cdot Z$ requires $n_v \cdot n_o$ BSGS matrix multiplications and the total multiplicative depth is 1 plaintext-ciphertext multiplication.

6.3 Homomorphic Evaluation of the Mask

To calculate the binary vector masking value (Equation (1)), we need to calculate the inner product of two homomorphically encrypted ciphertexts c and d. After an initial multiplication $c \cdot d$, the inner product requires $\log_2(n/2)$ rotations and additions, followed by Equation (4) to produce a ciphertext, where the result is encoded in each of the n slots. Our implementation uses rejection sampling and the SHAKE128 algorithm to cryptographically secure sample all the required ran-

3 In Equation (3), $\lfloor i/m_1 \rfloor$ is equal to k.

dom values in $\mathbb{Z}_p$. The total multiplicative depth to homomorphically evaluate the final mask (Equation (2)) is 1 ciphertext-ciphertext multiplication and 2 plaintext-ciphertext multiplications.

6.4 BFV Parameters

In BFV, one can choose three different parameters which greatly impact the runtime, security, and the available noise budget (i.e. how much further noise can be introduced until decryption will fail). These parameters are the degree of the reduction polynomial $n = 2^k$, the plaintext modulus p, which needs to be prime and $p \equiv 1 \pmod{2 \cdot n}$ to enable packing, and the ciphertext modulus q. We test our implementation for a computational security level of $\kappa = 128$ bit for different plaintext moduli p using the smallest n (and its default value for q) providing enough noise budget for correct evaluation of our protocol.

6.5 Function Privacy and Noise Flooding

Function privacy can be achieved by re-randomization and noise flooding, where the MNO adds an encryption of zero with a sufficiently large noise [25, 35] to the protocol's output. Following the smudging lemma [6], one needs to add a ciphertext with noise being $\lambda_{\text{FP}} + \log_2(n) + \log_2(n_o)$ bits larger than the upper bound of our protocol's original output's noise to achieve a statistical distance of $2^{\lambda_{\text{FP}}}$ between different executions.

We implement noise flooding by creating an encryption of zero (c_0) with large noise (in practice, we set the noise as large as possible while ensuring decryption is still possible). Adding c_0 to the output of our protocol (c) results in a ciphertext which has $\lambda_{\text{FP}} = \text{NOISEBUDGET}(c) - \text{NOISEBUDGET}(c_0) - \log_2(n) - \log_2(n_o)$ statistical function privacy. In our concrete parameter sets, we ensure that $\lambda_{\text{FP}} > \nu$.

However, like most efficient instantiations of function privacy, noise flooding provides security against semi-honest adversaries only (see [25] and contained references), and so our implementation also only provides semi-honest security. Still, once available, our implementation can use efficient maliciously function-private FHE schemes instead and benefit from security against a malicious health authority.

6.6 Benchmarks

We benchmark our prototype implementation[4] on an c5.24xlarge AWS EC2 instance (96 vCPU @ 3.6 GHz, 192 GiB RAM) running Ubuntu Server 20.04 in the Region Frankfurt with a current price of $4.656 per hour.

In our benchmarks, we focus on evaluating the runtime of the Data Aggregation phase of our protocol. Since in our use cases N is much bigger than k, we implemented multithreading, such that the threads split the number of rows in the matrix (more specifically, the number of submatrices in the rows n_v) equally amongst all available threads. Therefore, each thread has to perform at most $\left\lceil \frac{n_v}{\#\text{threads}} \right\rceil \cdot n_o$ `MatMul` evaluations, which will be combined at the end by summing up the intermediate results.

The evaluation of the proving mask with its higher multiplicative depth requires BFV parameters providing a bigger noise budget, however, its actual evaluation does not impact the overall runtime of the protocol since we perform it in an extra thread in parallel to the data aggregation. Furthermore, adding DP, noise flooding, as well as the computations of the health authority (encryption and decryption), have negligible runtime.

The runtime of our protocol is $\mathcal{O}(n_v n_o)$, i.e., it scales linearly in the number of `MatMul` evaluations. This can be seen in Figure 6 in which we summarize the runtime of the homomorphic matrix multiplication for different matrix dimensions using only one thread. For real-world matrix dimensions, some added runtime has to be expected due to thread synchronization and the accumulation of the intermediate thread results.

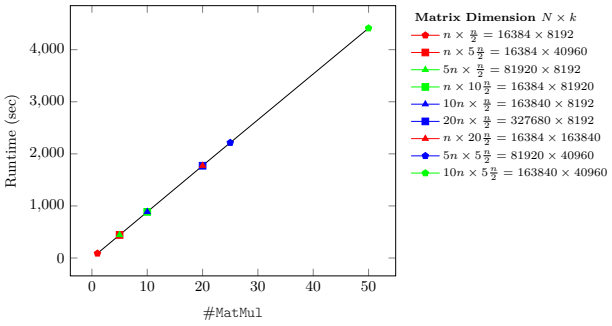


Fig. 6. Linear dependency of the runtime of the overall matrix multiplication to the number of `MatMul` evaluations. BFV parameters are: $\log_2(p) = 42$, $n = 16384$, $\kappa = 128$.

4 The source code is available at https://github.com/IAIK/CoronaHeatMap.

Real World Matrix Dimensions

In our benchmarks, we want to evaluate our protocol with parameters suitable for smaller nation states and set the matrix dimensions to N being larger then the total population of small countries, and k to be larger then the total number of cell towers in these countries. Concretely, we set $N = 2^{23}$ and $k = 2^{15}$, parameters enough to evaluate our protocol, for example, for Austria [63, 67], Singapore [34, 77], Kenya [74], New York City, Paraguay or New Zealand. In Table 2 we list the runtime for a homomorphic $(1 \times 2^{23}) \times (2^{23} \times 2^{15})$ matrix multiplication, for different BFV parameters, using (at most) 96 threads. We also provide the total number of `MatMul` evaluations and the (maximum) number of evaluations per thread. We give performance numbers for a plaintext prime p of size 42 bit, i.e., the smallest size to achieve $\nu = 41$ bit statistical privacy against malicious health authorities using our proving mask. To capture use cases, where a 42 bit plaintext modulus is not big enough, we also benchmark our protocol for a 60 bit prime p (the maximum value supported by SEAL), providing $\nu = 59$ bit statistical security. Further, we also give the achieved statistical function privacy λ_{FP} in bits for both benchmarks. As Table 2 shows, the MNO's computation takes 70 minutes for a 42 bit plaintext prime and 1 hour 25 minutes for the bigger 60 bit prime.

Table 2. Runtime for the MNO's computations for different parameters using 96 threads. $N = 2^{23}$, $k = 2^{15}$, $\kappa = 128$.

BFV			#`MatMul`	Time	AWS
$\log_2(p)$	n	λ_{FP}	total (thread)	min	price
42	16384	165	2048 (24)	69.33	$5.38
60	16384	96	2048 (24)	83.23	$6.46

Data Transmission

The data sizes which have to be transmitted between the MNO and the health authority are listed in Table 3. Each row corresponds to a different parameter set from Table 2. The sizes were obtained by storing each of the described elements on the file system on the benchmarking platform. The table lists the size of the ciphertexts (ct), the public key (pk), Galois keys (gk), and relinearization keys (rk). The public key is required for noise flooding to achieve function privacy, whereas Galois keys are required to perform homomorphic rotations. Each rotation index requires one Galois key, plus an additional key for rotating the columns. When using the BSGS algorithm, we need a key for the index 1 to calculate $\texttt{rot}(\boldsymbol{x}, j)$, and a key for the indices $k \cdot m_1$, $\forall 0 < k < m_2$. Also, for masking, we need the keys for the power-of-2 indices to calculate the inner product of two ciphertexts. The relinearization key is required to linearize the result of a ciphertext-ciphertext multiplication. We want to stress that the public key (pk), Galois keys (gk), and relinearization keys (rk) only need to be sent once before our protocol's first evaluation in a data-independent setup phase. Subsequent uses of the protocol can reuse these keys and only require transmitting the ciphertexts.

Table 3. Data transmission in MiB for parameters in Table 2.

Health Authority					MNO	Total
ct	pk[a]	gk[a]	rk[a]	Total	ct	
445.9	1.0	557.5	7.8	1012.2	1.7	1013.9
445.9	1.0	557.5	7.8	1012.2	1.7	1013.9

[a] One-time transmission (data-independent).

As Table 3 shows, health-authority-to-MNO communication is significantly more extensive than the response of the MNO. The main parts of the communication are the initial ciphertexts and the Galois keys. One reason for the size difference between the ciphertexts in the query and the response is that the parameter k is significantly smaller than N. Another reason is that our implementation performs a so-called modulus-switch after the computation, reducing the ciphertext modulus q to only one of the moduli q_i it is composed of. Further observe, that the plaintext modulus p does not affect the communication cost.

6.7 Price Estimation for Larger Countries

Here we give an estimate of the costs of evaluating our protocol to create a COVID-19 heatmap for a larger country, more specifically, for Germany. About 83 million people live in Germany, and a total of 80000 cell sites are deployed [46]. With the BFV parameters of the first entry in Table 2, i.e., $n = 16384, \nu = 41, \kappa = 128$, this corresponds to $n_v \cdot n_o = 5066 \cdot 10 = 50660$ `MatMul` evaluations.

To get $n_o = 10$ `MatMul` evaluations per thread, we would have to acquire 53 CPU's capable of handling 96 threads each. Assuming a runtime of 30 min per thread

(calculated from Table 2), and a price of \$4.656/h per CPU, we estimate the cost of evaluating the homomorphic matrix multiplication including the proving mask in a total time of 30 min to \$124 using AWS.[5] This estimate shows that it is likely very feasible to create a heatmap once a week to gain valuable insight into the spread of the disease, even for larger countries. We, however, note that care has to be taken when outsourcing this computation to cloud providers to ensure user privacy in accordance to privacy regulations.

7 Considerations and Conclusion

Our solution shows that privacy-preserving health data analytics is possible even on a national scale. We achieved this by combining three PETs. Each of them has their known limitations, but filtering out their strengths and applying them purposefully lead to a real-world cryptographic protocol. More broadly, we wanted to convey the following message: Even in times of crisis where it is tempting to lower data protection standards for purposes of big data analytics, there are technical methods to keep data protection standards high. And those technical methods are practical and available.

In the following we discuss considerations when instantiating our protocol with multiple health authorities or MNO's, as well as a summary of the key takeaways from a legal case study we conducted. More concretely, we focused on the EU General Data Protection Regulation (GDPR) [29], which is known to be on of the most strict privacy framework.

Multiple MNO's or Health Authorities

Even though it has already been shown that just using the largest MNO of a country for modelling disease dynamics is highly effective in practice [74], one might consider to use data from multiple MNO's. Our protocol can easily be extended to this setting by performing the protocol with each MNO individually and summing up the resulting heatmaps. As long as DP parameters (Section 5.2.2) are chosen, such that parameters (w_i, ϵ_i) for the i-th MNO, as well as $(\sum_i w_i, \sum_i \epsilon_i)$, fulfill the privacy-utility tradeoff, no additional information is leaked.

Multiple health authorities (e.g., for different provinces in a country) can be included using techniques from [58]. These multiple health authorities can agree on commen public keys, while keeping the decryption key hidden from all parties. After each health authority has agreed on database indices with the MNO (Remark 1), each authority can encrypt their queries using the common public key and the MNO can simply sum them up and proceed with the protocol as usual. After the protocol, the authorities proceed with the keyswitch protocol to output the final heatmap to some specified recipient (e.g., government officials). This adaptation is equivalent to the inital protocol with the same security and privacy guarantees, as long as each patient is registered with only one health authority. Otherwise, the heatmap will be a random output, due to the binary mask.

Legal Considerations

The health authority has used HE for COVID-19 positive individuals' ids, while the MNO has used DP to protected personal data. The MNO does not enter into possession of the decryption key of the health authorities data sets. Therefore, the computations performed should be considered carried out on anonymized data [66], which are data that cannot identify, directly or indirectly, the data subject. In fact, data encrypted both by the health authority and the MNO is not accessible by an entity other than the one carrying out the encryption protocol. Hence, the data should be considered anonymized data – whose processing falls out of the scope of application of the GDPR (Article 29 Working Party) – for all other entities. A similar argument holds for the aggregated location data, which are protected from singling out attacks by DP [4, 19]. Nevertheless, the processing of data by health authorities and MNO remains bound to GDPR provisions. In particular, the process to encrypt and make such data inaccessible is a processing activity under the GDPR. Thus, it should comply with legal requirements enshrined in the GDPR. In our use case, a lawful basis for processing personal data can be found in, e.g., Art. 9 (2) (i) GDPR, which deals with data processing in a public health context. It is one reasons why it is likely that there is a legal basis for our protocol.

Therefore, both MNO and health authority's processing activity protected through state-of-the-art PETs should be considered in compliance with GDPR provisions. From a legal perspective, the added value of the

5 In practice, additional costs for handling the databases, network traffic, key management, human resources, among some other costs are to be expected.

provided solution is represented on the one hand by the possibility to transform personal data into anonymized data. On the other hand, the processing activity of anonymizing data and limiting access to personal data ensure data subjects respect their fundamental rights as encoded in the EU privacy and data protection framework.

Acknowledgments

We thank our shepherd Robert Cunningham for his constructive feedback and helpful insights for improving the paper. This work was supported by EU's Horizon 2020 project Safe-DEED under grant agreement n°825225, EU's Horizon 2020 project KRAKEN under grant agreement n°871473, and by the "DDAI" COMET Module within the COMET – Competence Centers for Excellent Technologies Programme, funded by the Austrian Federal Ministry for Transport, Innovation and Technology (bmvit), the Austrian Federal Ministry for Digital and Economic Affairs (bmdw), the Austrian Research Promotion Agency (FFG), the province of Styria (SFG) and partners from industry and academia. The COMET Programme is managed by FFG.

References

[1] SCALE-MAMBA. https://github.com/KULeuven-COSIC/SCALE-MAMBA (2020)

[2] Abraham, I., Pinkas, B., Yanai, A.: Blinder - scalable, robust anonymous committed broadcast. In: CCS. pp. 1233–1252. ACM (2020)

[3] Albrecht, M.R., Chase, M., Chen, H., Ding, J., Goldwasser, S., Gorbunov, S., Halevi, S., Hoffstein, J., Laine, K., Lauter, K.E., et al.: Homomorphic encryption standard. IACR Cryptol. ePrint Arch. p. 939 (2019)

[4] Altman, M., Cohen, A., Nissim, K., Wood, A.: What a hybrid legal-technical analysis teaches us about privacy regulation: The case of singling out. Available at SSRN (2020)

[5] Apple: Differential privacy. https://www.apple.com/privacy/docs/Differential_Privacy_Overview.pdf (2020)

[6] Asharov, G., et al.: Multiparty computation with low communication, computation and interaction via threshold FHE. In: EUROCRYPT. LNCS, vol. 7237, pp. 483–501. Springer (2012)

[7] Bengtsson, L., Gaudart, J., Lu, X., Moore, S., Wetter, E., Sallah, K., Rebaudet, S., Piarroux, R.: Using mobile phone data to predict the spatial spread of cholera. Scientific reports **5**, 8923 (2015)

[8] Brakerski, Z.: Fully homomorphic encryption without modulus switching from classical gapsvp. In: CRYPTO. LNCS, vol. 7417, pp. 868–886. Springer (2012)

[9] Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (leveled) fully homomorphic encryption without bootstrapping. In: ITCS. pp. 309–325. ACM (2012)

[10] Bünz, B., et al.: Bulletproofs: Short proofs for confidential transactions and more. In: IEEE S&P. pp. 315–334. IEEE Computer Society (2018)

[11] Bureau, U.C.: Statistical safeguards. https://www.census.gov/about/policies/privacy/statistical_safeguards.html (2020)

[12] Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: FOCS. pp. 136–145. IEEE (2001)

[13] Canetti, R., Trachtenberg, A., Varia, M.: Anonymous collocation discovery: Harnessing privacy to tame the coronavirus (2020)

[14] Chan, J., Foster, D., Gollakota, S., Horvitz, E., Jaeger, J., Kakade, S., Kohno, T., Langford, J., Larson, J., Singanamalla, S., Sunshine, J., Tessaro, S.: Pact: Privacy sensitive protocols and mechanisms for mobile contact tracing (2020)

[15] Chen, H., Huang, Z., Laine, K., Rindal, P.: Labeled PSI from fully homomorphic encryption with malicious security. In: ACM Conference on Computer and Communications Security. pp. 1223–1237. ACM (2018)

[16] Cheon, J.H., Kim, A., Kim, M., Song, Y.S.: Homomorphic encryption for arithmetic of approximate numbers. In: ASIACRYPT (1). LNCS, vol. 10624, pp. 409–437. Springer (2017)

[17] Chillotti, I., Gama, N., Georgieva, M., Izabachène, M.: Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. In: ASIACRYPT (1). LNCS, vol. 10031, pp. 3–33 (2016)

[18] Cho, E., Myers, S.A., Leskovec, J.: Friendship and mobility: User movement in location-based social networks. In: ACM SIGKDD. p. 1082–1090. KDD '11, ACM, New York, NY, USA (2011). 10.1145/2020408.2020579, https://doi.org/10.1145/2020408.2020579

[19] Cohen, A., Nissim, K.: Towards formalizing the gdpr's notion of singling out. Proc. Natl. Acad. Sci. USA **117**(15), 8344–8352 (2020)

[20] Cong, K., Moreno, R.C., da Gama, M.B., Dai, W., Iliashenko, I., Laine, K., Rosenberg, M.: Labeled psi from homomorphic encryption with reduced computation and communication. In: ACM Conference on Computer and Communications Security. p. to appear. ACM (2021)

[21] Damgård, I., Pastro, V., Smart, N.P., Zakarias, S.: Multiparty computation from somewhat homomorphic encryption. In: CRYPTO. LNCS, vol. 7417, pp. 643–662. Springer (2012)

[22] Damgård, I., et al.: Practical covertly secure MPC for dishonest majority - or: Breaking the SPDZ limits. In: ESORICS. LNCS, vol. 8134, pp. 1–18. Springer (2013)

[23] Dar, A.B., Lone, A.H., Zahoor, S., Khan, A.A., Naaz, R.: Applicability of mobile contact tracing in fighting pandemic (covid-19): Issues, challenges and solutions. Computer Science Review **38**, 100307 (2020). https://doi.org/10.1016/j.cosrev.2020.100307, https://www.sciencedirect.com/science/article/pii/S157401372030407X

[24] Dittmer, S., et al.: Function secret sharing for PSI-CA: with applications to private contact tracing. IACR Cryptol. ePrint

Arch. **2020**, 1599 (2020)
[25] Ducas, L., Stehlé, D.: Sanitization of FHE ciphertexts. In: EUROCRYPT (1). LNCS, vol. 9665, pp. 294–310. Springer (2016)
[26] Duong, T., Phan, D.H., Trieu, N.: Catalic: Delegated PSI cardinality with applications to contact tracing. In: ASIACRYPT (3). LNCS, vol. 12493, pp. 870–899. Springer (2020)
[27] Dwork, C.: Differential privacy. In: ICALP (2). LNCS, vol. 4052, pp. 1–12. Springer (2006)
[28] Dwork, C., Roth, A.: The algorithmic foundations of differential privacy. Found. Trends Theor. Comput. Sci. **9**(3-4), 211–407 (2014)
[29] European Commission: General data protoction regulation. https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=celex:32016R0679 (2018)
[30] Evans, D., Kolesnikov, V., Rosulek, M.: A pragmatic introduction to secure multi-party computation. Found. Trends Priv. Secur. **2**(2-3), 70–246 (2018)
[31] Fan, J., Vercauteren, F.: Somewhat practical fully homomorphic encryption. IACR Cryptology ePrint Archive **2012**, 144 (2012)
[32] Ferguson, N.M., Cummings, D.A., Cauchemez, S., Fraser, C., Riley, S., Meeyai, A., Iamsirithaworn, S., Burke, D.S.: Strategies for containing an emerging influenza pandemic in southeast asia. Nature **437**(7056), 209–214 (2005)
[33] Finger, F., Genolet, T., Mari, L., de Magny, G.C., Manga, N.M., Rinaldo, A., Bertuzzo, E.: Mobile phone data highlights the role of mass gatherings in the spreading of cholera outbreaks. Proceedings of the National Academy of Sciences **113**(23), 6421–6426 (2016)
[34] Frost & Sullivan: Asean telecommunications towers market. https://ww2.frost.com/wp-content/uploads/2017/01/ASEAN-Telecommunications-Towers-Market_-EDT_AG_Final.pdf (2017), [Online; acc. 2021-03-30]
[35] Gentry, C.: Fully homomorphic encryption using ideal lattices. In: STOC. pp. 169–178. ACM (2009)
[36] Gentry, C., Halevi, S., Vaikuntanathan, V.: *i*-hop homomorphic encryption and rerandomizable yao circuits. In: CRYPTO. LNCS, vol. 6223, pp. 155–172. Springer (2010)
[37] Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game or A completeness theorem for protocols with honest majority. In: STOC. pp. 218–229. ACM (1987)
[38] Google: Learning statistics with privacy, aided by the flip of a coin. https://ai.googleblog.com/2014/10/learning-statistics-with-privacy-aided.html (2014)
[39] Google, Apple: Apple and google's exposure notification system. https://www.apple.com/covid19/contacttracing (2020)
[40] Grenfell, B.T., Bjørnstad, O.N., Kappey, J.: Travelling waves and spatial hierarchies in measles epidemics. Nature **414**(6865), 716–723 (2001)
[41] Halevi, S., Shoup, V.: Algorithms in helib. In: CRYPTO (1). LNCS, vol. 8616, pp. 554–571. Springer (2014)
[42] Halevi, S., Shoup, V.: Bootstrapping for helib. In: EUROCRYPT (1). LNCS, vol. 9056, pp. 641–670. Springer (2015)
[43] Halevi, S., Shoup, V.: Faster homomorphic linear transformations in helib. In: CRYPTO (1). LNCS, vol. 10991, pp. 93–120. Springer (2018)
[44] Hazay, C., Lindell, Y.: Efficient protocols for set intersection and pattern matching with security against malicious and covert adversaries. In: TCC. LNCS, vol. 4948, pp. 155–175. Springer (2008)
[45] Holz, M., Judkewitz, B., Möllering, H., Pinkas, B., Schneider, T.: PEM: privacy-preserving epidemiological modeling. IACR Cryptol. ePrint Arch. **2020**, 1546 (2020)
[46] Informationszentrum Mobilfunk: Zahl der funkanlagenstandorte in deutschland. https://www.informationszentrum-mobilfunk.de/artikel/zahl-der-funkanlagenstandorte-in-deutschland (2020), [Online; acc. 2021-03-30]
[47] Ion, M., et al.: On deploying secure computing: Private intersection-sum-with-cardinality. In: EuroS&P. pp. 370–389. IEEE (2020)
[48] Isdory, A., Mureithi, E.W., Sumpter, D.J.: The impact of human mobility on hiv transmission in kenya. PloS one **10**(11), e0142805 (2015)
[49] Keller, M.: MP-SPDZ: A versatile framework for multi-party computation. In: CCS. pp. 1575–1590. ACM (2020)
[50] Kohli, N., Laskowski, P.: Epsilon voting: Mechanism design for parameter selection in differential privacy. In: 2018 IEEE Symposium on Privacy-Aware Computing (PAC). pp. 19–30 (2018). 10.1109/PAC.2018.00009
[51] Krumm, J.: A survey of computational location privacy. Pers. Ubiquitous Comput. **13**(6), 391–399 (2009)
[52] Lai, J., Deng, R.H., Pang, H., Weng, J.: Verifiable computation on outsourced encrypted data. In: ESORICS (1). LNCS, vol. 8712, pp. 273–291. Springer (2014)
[53] Lee, J., Clifton, C.: How much is enough? choosing ϵ for differential privacy. In: Lai, X., Zhou, J., Li, H. (eds.) Information Security. pp. 325–340. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
[54] Lepoint, T., Patel, S., Raykova, M., Seth, K., Trieu, N.: Private join and compute from PIR with default. IACR Cryptol. ePrint Arch. **2020**, 1011 (2020)
[55] Lindell, Y., Pinkas, B.: Secure two-party computation via cut-and-choose oblivious transfer. J. Cryptol. **25**(4), 680–722 (2012)
[56] Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. In: EUROCRYPT. LNCS, vol. 6110, pp. 1–23. Springer (2010)
[57] Miao, P., Patel, S., Raykova, M., Seth, K., Yung, M.: Two-sided malicious security for private intersection-sum with cardinality. In: CRYPTO (3). LNCS, vol. 12172, pp. 3–33. Springer (2020)
[58] Mouchet, C., Troncoso-Pastoriza, J.R., Bossuat, J., Hubaux, J.: Multiparty homomorphic encryption from ring-learning-with-errors. Proc. Priv. Enhancing Technol. **2021**(4), 291–311 (2021)
[59] Naor, M., Pinkas, B.: Efficient oblivious transfer protocols. In: SODA. pp. 448–457. ACM/SIAM (2001)
[60] Nissim, K., Bembenek, A., Wood, A., Bun, M., Gaboardi, M., Gasser, U., O'Brien, D.R., Steinke, T., Vadhan, S.: Bridging the gap between computer science and legal approaches to privacy. Harv. JL & Tech. **31**, 687 (2017)
[61] others, J.H.: Differential privacy: An economic method for choosing epsilon. In: CSF. pp. 398–410. IEEE Computer Society (2014)
[62] Pinkas, B., Ronen, E.: Hashomer - a proposal for a privacy-preserving bluetooth based contact tracing scheme for ham-

agen. https://github.com/eyalr0/HashomerCryptoRef/blob/master/documents/hashomer.pdf (2020)
[63] Radiocells.org: Cells and wifis in austria. https://www.radiocells.org/country/at (2020), [Online; acc. 2021-03-30]
[64] Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: STOC. pp. 84–93. ACM (2005)
[65] Microsoft SEAL (release 3.6). https://github.com/Microsoft/SEAL (Nov 2020), microsoft Research, Redmond, WA.
[66] Spindler, G., Schmechel, P.: Personal data and encryption in the european general data protection regulation. J. Intell. Prop. Info. Tech. & Elec. Com. L. **7**, 163 (2016)
[67] Statista: Austria: Total population from 2015 to 2025. https://www.statista.com/statistics/263741/total-population-in-austria/ (2021), [Online; acc. 2021-03-30]
[68] Tatem, A.J., Huang, Z., Narib, C., Kumar, U., Kandula, D., Pindolia, D.K., Smith, D.L., Cohen, J.M., Graupe, B., Uusiku, P., et al.: Integrating rapid risk mapping and mobile phone call record data for strategic malaria elimination planning. Malaria journal **13**(1), 52 (2014)
[69] Tatem, A.J., Qiu, Y., Smith, D.L., Sabot, O., Ali, A.S., Moonen, B.: The use of mobile phone data for the estimation of the travel patterns and imported plasmodium falciparum rates among zanzibar residents. Malaria journal **8**(1), 287 (2009)
[70] Trieu, N., Shehata, K., Saxena, P., Shokri, R., Song, D.: Epione: Lightweight contact tracing with strong privacy (2020)
[71] Troncoso, C., et al.: Decentralized privacy-preserving proximity tracing (2020)
[72] Wesolowski, A., Buckee, C.O., Engø-Monsen, K., Metcalf, C.J.E.: Connecting mobility to infectious diseases: the promise and limits of mobile phone data. The Journal of infectious diseases **214**(suppl_4), S414–S420 (2016)
[73] Wesolowski, A., Eagle, N., Noor, A.M., Snow, R.W., Buckee, C.O.: The impact of biases in mobile phone ownership on estimates of human mobility. Journal of the Royal Society Interface **10**(81), 20120986 (2013)
[74] Wesolowski, A., Eagle, N., Tatem, A.J., Smith, D.L., Noor, A.M., Snow, R.W., Buckee, C.O.: Quantifying the impact of human mobility on malaria. Science **338**(6104), 267–270 (2012)
[75] Wesolowski, A., Metcalf, C., Eagle, N., Kombich, J., Grenfell, B.T., Bjørnstad, O.N., Lessler, J., Tatem, A.J., Buckee, C.O.: Quantifying seasonal population fluxes driving rubella transmission dynamics using mobile phone data. Proceedings of the National Academy of Sciences **112**(35), 11114–11119 (2015)
[76] Wesolowski, A., Qureshi, T., Boni, M.F., Sundsøy, P.R., Johansson, M.A., Rasheed, S.B., Engø-Monsen, K., Buckee, C.O.: Impact of human mobility on the emergence of dengue epidemics in pakistan. Proceedings of the National Academy of Sciences **112**(38), 11887–11892 (2015)
[77] Wikipedia contributors: Telecommunications in singapore — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Telecommunications_in_Singapore&oldid=1002495194 (2021), [Online; acc. 2021-03-30]
[78] Yao, A.C.: How to generate and exchange secrets (extended abstract). In: FOCS. pp. 162–167. IEEE Computer Society (1986)

A Security Proofs

We now prove security using the real-ideal-paradigm [30]. In this paradigm a protocol execution is secure if it behaves the same as when the parties send their input to a trusted third party (the ideal functionality) which does the computation and provides them with the outputs. More formally, an environment should not be able to distinguish between an observation of the protocol with a possible adversary and a simulator interacting with the ideal functionality. More specifically, most of the time, computational indistinguishability is required between the ideal and the real world. In contrast, we require (κ,ν)-indistinguishability to analyze the cheating probability more thoroughly.

Definition 1 ((κ,ν)-indistinguishability [55])**.** *Let* $X = \{X(a,\kappa,\nu)\}_{\kappa,\nu,\in\mathbb{N},a\in\{0,1\}^*}$ *and* $Y = \{Y(a,\kappa,\nu)\}_{\kappa,\nu,\in\mathbb{N},a\in\{0,1\}^*}$ *be probability ensembles, so that for any* $\kappa,\nu \in \mathbb{N}$ *the distribution* $\{X(a,\kappa,\nu)\}$ *(resp.* $\{Y(a,\kappa,\nu)\}$*) ranges over strings of length polynomial in* $\kappa+\nu$*. We say that the ensembles are* (κ,ν)*-indistinguishable if for every polynomial-time adversary* $\mathcal{A}$*, it holds that for every* $a \in \{0,1\}^*$*:*

$$|\Pr[\mathcal{A}(X=1)] - \Pr[\mathcal{A}(Y=1)]| < \frac{1}{p(\kappa)} + 2^{-\mathcal{O}(\nu)},$$

for every $\nu \in \mathbb{N}$*, every polynomial* $p(\cdot)$*, and all large enough* $\kappa \in \mathbb{N}$*.*

A.1 Binary Mask

Lemma 2. *Let* p *be a integer of bit-length* $\nu \in \mathbb{N}$*, and let* $N \leq 2^{\nu/2}$*. Further, let* $\boldsymbol{x}$ *and* $\mu_{\texttt{bin}}$ *be defined as in Section 4.3, then it holds that*

$$\Pr[\boldsymbol{x} \texttt{ not binary} \ \wedge\ \mu_{\texttt{bin}} = 0] \leq \frac{1}{2^{\nu-1}}.$$

Proof.

$$\mu_{\texttt{bin}} = \underbrace{\langle \boldsymbol{x}, (\boldsymbol{d} \circ \boldsymbol{y_1}^N)\rangle \cdot r_1}_{:=\alpha} + \underbrace{\langle \boldsymbol{x}, (\boldsymbol{d} \circ \boldsymbol{y_2}^N)\rangle \cdot r_2}_{:=\beta} = \alpha + \beta$$

We are now interested in the events when the binary mask evaluates to zero even though $\boldsymbol{x} \notin \mathbb{Z}_2^N$. This undesired behaviour can only happen in two ways, either $\alpha = \beta = 0$ or $\alpha = -\beta$. Next, we calculate the probability of these two cases.

First, since $r_1, r_2 \neq 0$ and assuming $\boldsymbol{x} \neq \boldsymbol{0}^k$ ($\boldsymbol{x} = \boldsymbol{0}^k$ is a valid input and should result in a zero mask), we

have $\Pr[\alpha = 0] = \Pr[\beta = 0] = {}^{N}/_{p}$ [10]. Hence,

$$\Pr[\alpha = \beta = 0] = \frac{N}{p} \cdot \frac{N}{p} = \frac{N^2}{p^2}. \tag{6}$$

Consequently, the probability of α being non-zero is $1 - {}^{N}/_{p}$. Further, the probability of β being $-\alpha$ is ${}^{1}/_{p}$. Combing these probabilities gives us

$$\Pr[\alpha = -\beta] = \left(1 - \frac{N}{p}\right)\frac{1}{p} = \frac{1}{p} - \frac{N}{p^2}. \tag{7}$$

We get the final probability by putting together Equation (6) and Equation (7)

$$\begin{aligned}\Pr[\alpha + \beta = 0] &= \frac{N^2}{p^2} + \frac{1}{p} - \frac{N}{p^2} < \frac{1}{p} + \frac{N^2}{p^2} \\ &\leq \frac{1}{2^\nu} + \frac{2^\nu}{2^{2\nu}} = \frac{1}{2^{\nu-1}}, \text{ because } N \leq 2^{\nu/2}.\end{aligned}$$

□

A.2 Proof of Lemma 1

π_{Hmap}

1. A party P_1 on input $(\mathsf{input}, sid, P_1, P_2, \boldsymbol{x})$ from the environment verifies that $\boldsymbol{x} \in \mathbb{Z}_p^N$, else ignores the input. Next, samples a key pair $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\kappa)$, and computes $\boldsymbol{c} \leftarrow \mathsf{Enc}_{\mathsf{pk}}(\boldsymbol{x})$. It records $(sid, P_1, P_2, \mathsf{sk})$, and sends $(sid, P_1, P_2, \mathsf{pk}, \boldsymbol{c})$ to P_2. P_1 ignores subsequent inputs of the form $(\mathsf{input}, sid, P_1, P_2, \cdot)$ from the environment.
2. On a later input of the form $(sid, P_1, P_2, \boldsymbol{h}^*)$ from P_2, P_1 computes $\boldsymbol{h} \leftarrow \mathsf{Dec}_{\mathsf{sk}}(\boldsymbol{h}^*)$, and outputs $(\mathsf{result}, sid, P_1, P_2, \boldsymbol{h})$ to the environment.
3. A party P_2 on input $(\mathsf{input}, sid, P_1, P_2, Z)$ from the environment and $(sid, P_1, P_2, \mathsf{pk}, \boldsymbol{c})$ from P_1 verifies that $Z \in \mathbb{Z}_p^{N \times k}$, else ignores the input. Next, computes the mask vector $\boldsymbol{\mu}$ and the noise $\boldsymbol{\delta}$ according to Figure 2. Then computes $\boldsymbol{h}^* \leftarrow \mathsf{Eval}_{\mathsf{pk}}(\boldsymbol{c}^T \cdot Z + \boldsymbol{\delta} + \boldsymbol{\mu})$. P_2, sends $(sid, P_1, P_2, \boldsymbol{h}^*)$ to P_1 and ignores all subsequent inputs of the form $(\mathsf{input}, sid, P_1, P_2, \cdot)$ from the environment.

Fig. 7. Formalized protocol π_{Hmap}

Proof. We use Lemma 2 to prove that to any polynomial time environment the execution π_{Hmap} with a possible adversary $\mathcal{A}$ is (κ, ν)-indistinguishable from a simulator $\mathcal{S}$ interacting with the ideal functionality $\mathcal{F}_{Hmap}$. More concretely, we claim that as long as the event that $\boldsymbol{x}$ is not binary and at the same time the mask $\boldsymbol{\mu} = \mathbf{0}^k$ does not occur, the executions of the ideal and real world are computational indistinguishable. Once we have proven this claim, we are done, since we have already shown that the probability of the above event is exponentially small in the statistical security parameter. Note that for the proof, we have rewritten the protocol in a more formal description π_{Hmap}, see Figure 7.

$\mathcal{S}_{Hmap}$

P_1, P_2 not corrupted: It starts by sampling a key pair $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\kappa)$, and sets $\boldsymbol{x} \leftarrow 0^N$. Then it computes $\boldsymbol{c} \leftarrow \mathsf{Enc}_{\mathsf{pk}}(\boldsymbol{x})$. It then instructs P_1 to send $(sid, P_1, P_2, \mathsf{pk}, \boldsymbol{c})$ to P_2. On later input of the form $(sid, P_1, P_2, \mathsf{pk}, \boldsymbol{c})$ from P_1 it samples $Z \leftarrow \mathbb{Z}_p^{N \times k}$. Then it computes $\boldsymbol{h}^* \leftarrow \mathsf{Eval}_{\mathsf{pk}}(\boldsymbol{c}^T \cdot Z + \boldsymbol{\delta} + \boldsymbol{\mu})$. It instructs P_2 to send $(sid, P_1, P_2, \boldsymbol{h}^*)$ to P_1.

P_1 not corrupted, P_2 corrupted: Similar as before but it does not have to simulate Z because it learns the input Z from P_2. Then it computes $\mathsf{Eval}_{\mathsf{pk}}(\boldsymbol{c}^T \cdot Z + \boldsymbol{\delta} + \boldsymbol{\mu})$.

P_1 corrupted, P_2 not corrupted: It learns the input $\boldsymbol{x}$ from P_1. Then it proceeds as in the first case until it has to simulate the message to P_1. In order to do this it runs a copy of π_{Hmap} internally, where it corrupts P_1. Thereby, it learns $\boldsymbol{x}^T \cdot Z + \boldsymbol{\delta} + \boldsymbol{\mu}$ and sets $\boldsymbol{h}^* \leftarrow \mathsf{Enc}_{\mathsf{pk}}(\boldsymbol{x}^T \cdot Z + \boldsymbol{\delta} + \boldsymbol{\mu})$.

P_1, P_2 corrupted: It learns the inputs $\boldsymbol{x}$ from P_1 resp. Z from P_2. It runs the protocol with the inputs, and outputs $(\mathsf{input}, sid, P_1, P_2, \boldsymbol{x})$ and $(\mathsf{input}, sid, P_1, P_2, Z)$ to the ideal functionality, which makes $\mathcal{F}_{Hmap}$ output $(\mathsf{result}, sid, P_1, P_2, \boldsymbol{x}^T \cdot Z + \boldsymbol{\delta} + \boldsymbol{\mu})$.

Fig. 8. Simulator $\mathcal{S}_{Hmap}$.

First consider a polynomial time environment which does not corrupt any of the parties. Any meaningful environment will interact with π_{Hmap} or $\mathcal{F}_{Hmap}$ in the following way.

1. It picks a vector $\boldsymbol{x} \in \mathbb{Z}_p^n$ and inputs $(\mathsf{input}, sid, P_1, P_2, \boldsymbol{x})$.
2. It sees $(sid, P_1, P_2, \mathsf{pk}, \boldsymbol{c})$.
3. It picks a matrix $Z \in \mathbb{Z}_p^{N \times k}$ and inputs $(\mathsf{input}, sid, P_1, P_2, Z)$.
4. It sees $(sid, P_1, P_2, \mathsf{pk}, \boldsymbol{h}^*)$.
5. It sees $(\mathsf{result}, sid, P_1, P_2, \boldsymbol{h})$.

Let us now assume to the contrary there is such an environment $\mathcal{E}$ that can distinguish the two systems $\pi_{Hmap} \circ \mathcal{A}$ and $\mathcal{F}_{Hmap} \circ \mathcal{S}$ with non-negligible advantage. Then we can turn $\mathcal{E}$ into a polynomial time system $\mathcal{E}'$

which wins in the IND-CPA game with non-negligible probability:

1. $\mathcal{E}'$ receives pk.
2. $\mathcal{E}'$ runs $\mathcal{E}$ to see which message $(sid, P_1, P_2, \boldsymbol{x})$ gets recorded.
3. $\mathcal{E}'$ inputs $(\boldsymbol{x}, \mathbf{0}^N)$ to the IND-CPA game and gets back an encryption $\boldsymbol{c}$, where $\boldsymbol{c}$ is either an encryption of $\boldsymbol{x}$ (if $b = 0$) or an encryption of $\mathbf{0}^N$ (if $b = 1$).
4. $\mathcal{E}'$ samples $Z \leftarrow \mathbb{Z}_p^N$. It runs $\mathcal{E}$ and provides input $(\mathsf{input}, sid, P_1, P_2, \boldsymbol{x})$, $(\mathsf{input}, sid, P_1, P_2, Z)$, $(sid, P_1, P_2, \mathsf{pk}, \boldsymbol{c})$, $(sid, P_1, P_2, \mathsf{Enc}_{\mathsf{pk}}(\boldsymbol{c}^T \cdot Z + \boldsymbol{\delta} + \boldsymbol{\mu}))$ and $(\mathsf{result}, sid, P_1, P_2, \boldsymbol{x}^T \cdot Z + \boldsymbol{\delta} + \boldsymbol{\mu})$.
5. $\mathcal{E}'$ waits until $\mathcal{E}$ outputs its guess b', then $\mathcal{E}'$ outputs b'.

If $b = 0$, then $\mathcal{E}$ observes the interaction it would see when interacting with the protocol π_{Hmap}, and if $b = 1$, then $\mathcal{E}$ observes the interaction it would see when interacting with the ideal functionality and the simulator $\mathcal{F}_{Hmap} \circ \mathcal{S}$. By assumption $\mathcal{E}$ can distinguish $\pi_{Hmap} \circ \mathcal{A}$ and $\mathcal{F}_{Hmap} \circ \mathcal{S}$ with non-negligible advantage. Therefore, $\mathcal{E}'$ will guess b with probability significantly better than $1/2$. This is a contradiction to the IND-CPA security of HE, as $\mathcal{E}'$ is polynomial time. □

A.3 One-Sided Simulation

To define one-sided simulation security, we have the notion of a protocol execution view. Let $VIEW^{\mathcal{A}}_{\pi,\mathcal{A}}(x, y)$ denoted the protocol execution view of the adversary $\mathcal{A}$, i.e., the corrupted parties' view (input, randomness, all received messages) after execution of π with input x resp. y from P_1 resp. P_2.

Definition 2. *Let $EXEC_{\pi,\mathcal{A},\mathcal{E}}$ resp. $EXEC_{\mathcal{F},\mathcal{S},\mathcal{E}}$ denote the random variables describing the output of environment $\mathcal{E}$ when interacting with an adversary $\mathcal{A}$ and parties P_1, P_2 performing protocol π, resp. when interacting with a simulator $\mathcal{S}$ and an ideal functionality $\mathcal{F}$, where only P_1 receives output. Protocol π securely realizes functionality $\mathcal{F}$ with one-sided simulation if*

1. *for any adversary $\mathcal{A}$ that controls P_2 there exists a simulator $\mathcal{S}$ such that, for any environment $\mathcal{E}$ the distribution of $EXEC_{\pi,\mathcal{A},\mathcal{E}}$ and $EXEC_{\mathcal{F},\mathcal{S},\mathcal{E}}$ are indistinguishable,*
2. *and for any adversary $\mathcal{A}$ controlling P_1 the distribution $VIEW^{\mathcal{A}}_{\pi,\mathcal{A}}(x, y)$ and $VIEW^{\mathcal{A}}_{\pi,\mathcal{A}}(x, y')$, where $|y| = |y'|$ are indistinguishable.*

B PSI-SUM with Indices

In Figure 9 we give the ideal functionality for a *PSI-SUM with Indices* primitive. Such a primitve computes the sum of the private values associated with the intersection elements of two databases and reveals the indices present in the intersection to one party. This can be seen as an relaxed version of the *Private Intersection-Sum with Cardinality* primitive introduced in [57].

$\mathcal{F}_{PSI-Sum}$

Parameters: $t, N \in \mathbb{N}$. Parties P_1 and P_2.

1. Upon receiving an input $(\mathsf{input}, sid, P_1, P_2, \boldsymbol{x})$ from a party P_1, verify that $\boldsymbol{x} \in \mathbb{Z}_p^N$, else ignore input. Next, record $(sid, P_1, P_2, \boldsymbol{x})$. Once $\boldsymbol{x}$ is recorded, ignore any subsequent inputs of the form $(\mathsf{input}, sid, P_1, P_2, \cdot)$ from P_1.
2. Upon receiving an input $(\mathsf{input}, sid, P_1, P_2, Z)$ from party P_2, verify that $Z \in \mathbb{Z}_p^{N \times *}$, else ignore input. Proceed as follows: If there is a recorded value $(sid, P_1, P_2, \boldsymbol{x}, w)$, compute $\boldsymbol{h} \leftarrow \boldsymbol{x}^T Z$ provided that $\boldsymbol{x} \in \mathbb{Z}_2^N$, otherwise $\boldsymbol{h} \xleftarrow{\$} \mathbb{Z}_p^k$, and send (sid, P_1, P_2, k) where k is the number of columns of Z to the adversary. Then output $(\mathsf{result}, sid, P_1, P_2, \boldsymbol{h})$ to P_1, and ignore subsequent inputs of the form $(\mathsf{input}, sid, P_1, P_2, \cdot)$ from P_2.

Fig. 9. Ideal functionality of PSI-SUM with Indices.

C Differential Privacy

Definition 3 (ϵ-Differential Privacy [27])**.** *A randomized mechanism $\mathcal{A}$ gives ϵ-differential privacy if for any neighboring datasets D and D', and any $S \in Range(\mathcal{A})$: $Pr[\mathcal{A}(D) = S] \leq e^{\epsilon} Pr[\mathcal{A}(D') = S]$.*

One can achieve ϵ-DP by adding noise from a zero-centered Laplace distribution to the final result of the computation. The noise is calibrated with the privacy budget ϵ and the global sensitivity Δq of the computation q: $\Delta q = \max_{D,D'} ||q(D) - q(D')||$ for all neighboring D and D'. The global sensitivity, thus, represents the maximum possible value of each element in the dataset. The Laplace distribution for a scale factor b is given as $Lap(x|b) = \frac{1}{2b} e^{-\frac{|x|}{b}}$, where usually $b = \frac{\Delta q}{\epsilon}$.

C.1 An Economic Method to Choose ϵ

We aim to provide a confidence interval for the proportion μ of individuals in the general population (or subpopulation) with a specific property. Assume a database D_N and let $g : D_N \to \mathbb{R}$ be the mechanism computing the sample mean with sensitivity $1/N$. If (for privacy reasons), we add Laplace noise ν to the outcome of g, we introduce an error source. Modeling each individual as a random variable with Bernoulli distribution allows us to bound this error by the tail bound. Hence, we can define the utility by a confidence interval with accuracy $T \in [0, 1]$, and confidence $1 - \alpha$ for $\alpha \in [0, 1]$

$$\Pr[|g(D_N) + \nu(\epsilon) - \mu| \geq T] \leq e^{\left(-\frac{TN\epsilon}{2}\right)} \leq \alpha.$$

The idea of DP is that an individual's expected harm (cost) of being in the database should be minor. Let E be the expected cost for an individual for being in the database (for a formal definition see [61]). Then the individual's cost of being in the computation g is

$$(e^{\epsilon} - 1)E.$$

Let D_w^j be the j-th column vector of the matrix Z, i.e., the location data corresponding to cell tower j. Then, we define the mechanism as

$$g(D_w^j) := \frac{\#\text{ individuals in } j}{w},$$

resulting in sensitivity $1/w$. This setup satisfies the assumption that each individual can be modeled as a Bernoulli experiment. This can be done for every cell tower, and thus covering the heatmap's area. The estimations of the expected baseline cost $E = \$0.01$ already cover the whole heatmap's area (all cell towers).