Protect Your Score: Contact Tracing With Differential Privacy Guarantees

Rob Romijnders[1], Christos Louizos[2], Yuki M. Asano[1] Max Welling[1]

[1]QUVA lab, University of Amsterdam
[2]Qualcomm AI research (an initiative of Qualcomm Technologies, Inc. and/or its subsidiaries.)
r.romijnders@uva.nl

Abstract

The pandemic in 2020 and 2021 had enormous economic and societal consequences, and studies show that contact tracing algorithms can be key in the early containment of the virus. While large strides have been made towards more effective contact tracing algorithms, we argue that privacy concerns currently hold deployment back. The essence of a contact tracing algorithm constitutes the communication of a risk score. Yet, it is precisely the communication and release of this score to a user that an adversary can leverage to gauge the private health status of an individual. We pinpoint a realistic attack scenario and propose a contact tracing algorithm with differential privacy guarantees against this attack. The algorithm is tested on the two most widely used agent-based COVID19 simulators and demonstrates superior performance in a wide range of settings. Especially for realistic test scenarios and while releasing each risk score with $\varepsilon = 1$ differential privacy, we achieve a two to ten-fold reduction in the infection rate of the virus. To the best of our knowledge, this presents the first contact tracing algorithm with differential privacy guarantees when revealing risk scores for COVID19.

1 Introduction

The COVID19 pandemic had enormous economic and societal consequences (Boden et al. 2021; Vindegaard and Benros 2020; Kim et al. 2022; Kaye et al. 2021a; Berger, Herkenhoff, and Mongey 2020). Some sources estimate the global economic impact at more than a trillion US dollars (Kaye et al. 2021b). Previous studies show that contact tracing apps can aid understanding and mitigate the early rise of the pandemic (Alsdurf et al. 2020; Baker et al. 2021; Herbrich, Rastogi, and Vollgraf 2020; Perra 2021). Most studies, however, focused on the effectiveness of the pandemic mitigation, while we argue that privacy concerns hold the deployment back (Raskar et al. 2020; Alsdurf et al. 2020). Population surveys during and after the pandemic show that mistrust and 'worries about privacy' are among the top three reasons not to use a contact tracing app (Jones, Thompson et al. 2021; Gao et al. 2022; Walrave, Waeterloos, and Ponnet 2022).

Several studies argue for the studying of privacy in contact tracing algorithms (Park, Choi, and Ko 2020; Grantz

et al. 2020; Dyda et al. 2021). We also quote an influential journal stating that most individuals would consider the privacy risks *"to be unacceptably high"* (The Lancet; Bengio et al. 2020). Yet, to the best of our knowledge, no research has been published on differential privacy when releasing a risk score to a user for the purpose of contact tracing.

Despite security measures, a contact tracing algorithm needs to assign a risk score and release the score either directly to the user or indirectly by the signal to get tested. It is precisely the communication and release of this score that an adversary can leverage to gauge the private health status of an individual. For the rest of the paper, we refer, by the name COVIDSCORE, to a risk score that a contact tracing algorithm assigns to a user and communicates to other users. Most papers about privacy and security aspects during the COVID19 pandemic center on security measures (Ahmed et al. 2020) for establishing contacts, where approaches such as hashing were studied for establishing contacts (Ali and Dyo 2021; Reichert, Brack, and Scheuermann 2021). In this work, we assume that these security precautions are adhered to and, in the presence of the security measures, identify another privacy attack on the COVIDSCORE:

> An adversary wants to determine the COVIDSCORE of a victim. The adversary installs the app and only makes contact with the victim. The next day, the adversary observes a change in their COVIDSCORE. This change is due to the victim, and the adversary reconstructs the COVIDSCORE of the victim.

The naïve approach of simply adding noise to any revealed COVIDSCORE does lead to increased uncertainty at the adversary about the score of a victim. However, adding noise naturally decreases the utility of a contact tracing algorithm. To address this conundrum, we propose a novel algorithm that, while adding noise, maintains good results in mitigating a peak of the pandemic. Moreover, we prove *differential privacy* (Dwork and Roth 2014) for the release of the COVIDSCORE and demonstrate strong performance even when $\varepsilon \leq 1$ per message.

In this paper, we make the following contributions:

1. We concretize a privacy attack in contact tracing with important implications, and we propose a novel decentralized algorithm with a differential privacy guarantee against this attack. To the best of our knowledge,

we are the first to study the differential privacy of a COVIDSCORE on top of the standard security measures.

2. The trade-off between privacy and utility is studied on two widely used simulators. The method is compared against existing methods for differential privacy, and the results show that our algorithm is Pareto optimal. For the case $\varepsilon = 1$, we show that up to the million scale, our algorithm achieves a two to ten times smaller infection rate, compared to traditional contact tracing.

3. To evaluate our algorithms' robustness across a range of realistic conditions, we also evaluate our algorithm in two challenging circumstances: under imperfect tests for COVID19 and a reduced test protocol.

The code for our method and all experiments is available at `github.com/RobRomijnders/dpfn_aaai`.

2 Related work

This section discusses the related work for our method. We discuss the current agent-based statistical contact tracing approaches and the recent research in privacy for COVID containment strategies.

Statistical contact tracing: Various approaches have been published about statistical contact tracing, especially during the COVID19 pandemic. Burdinski, Brockmann, and Maier (2022) test the efficacy of traditional contact tracing and run simulations, including self-isolation strategies. Li and Saad (2021) use a message-passing approach and analyze an isolation policy based on risk-score estimation. Herbrich, Rastogi, and Vollgraf (2020) investigate statistical contact tracing using Gibbs sampling and show results on a simulator based on stochastic block models. Braunstein et al. (2023) propose an inference model similar to belief propagation but do not test on COVID19 simulators. Most similar to ours, Baker et al. (2021) propose statistical contact tracing using belief propagation on a collapsed graph. Romijnders et al. (2023) propose another algorithm for statistical contact tracing, improving over the previous approach and comparing statistical contact tracing under constrained communication.

Privacy in COVID19 containment strategies: During the pandemic, many papers raised concerns about privacy and security in contact tracing. The first step is the design of decentralized algorithms where no central entity has the COVIDSCORE of multiple individuals (Baker et al. 2021; Herbrich, Rastogi, and Vollgraf 2020; Romijnders et al. 2023). Yet many security issues remain. Troncoso et al. (2020) provides an overview of the methods for proximity tracing and its various threat models. A paper in Nature Communications highlights the pitfalls of collecting such data from smartphones (Grantz et al. 2020) and calls for more research in privacy. Obtaining the contact graph and the various threat models for sharing GPS location are discussed in papers such as (Raskar et al. 2020; Ahmed et al. 2020; Ali and Dyo 2021; Reichert, Brack, and Scheuermann 2021). Examples of approaches that study obtaining contacts under secure and private circumstances are (Bay et al. 2020; Chan et al. 2020; Cho, Ippolito, and Yu 2020).

Differential privacy in decentralized inference: For the general purpose of statistical inference, a few but existent papers have studied differential privacy (DP). For example, DP for MCMC has been studied (Yıldırım and Ermiş 2019; Heikkilä et al. 2019). We implement and compare to a method (Wang, Fienberg, and Smola 2015; Foulds et al. 2016) that specifically tailors to Gibbs sampling (Herbrich, Rastogi, and Vollgraf 2020). Zhang et al. (2017) analyzes the computation of marginals in a message-passing approach for inference. However, that paper uses the Laplace mechanism for dealing with real-valued random variables of fixed dimensionality. In contrast, our random variables are discrete-valued and have varying degrees. Like us, Zou and Fekri (2015) use the local structure of the belief propagation message to obtain a privacy guarantee. However, they consider a different form of privacy and do not study differential privacy.

Two noteworthy approaches in contemporary literature study DP in the context of COVID19, but both methods do not relate to contact tracing. Vadrevu, Adusumalli, and Mangalapalli (2020) focuses on collecting and clustering medical records and does not mention contact tracing; Vepakomma, Pushpita, and Raskar (2021) focuses on collecting user trajectories with DP guarantees. However, these works are vulnerable to the same attack we study in this paper.

For other approaches to privacy concerning the release of a COVIDSCORE, previous research has mentioned low-bit quantization of the decentralized messages (Alsdurf et al. 2020; Apple and Google 2020; Romijnders et al. 2023). Still, these approaches have no formal guarantee pertaining to privacy. To the best of our knowledge, we are the first paper to propose an algorithm for statistical contact tracing with differential privacy guarantees.

3 Method

This method section proceeds as follows: first, we explain the model for statistical contact tracing. We discuss three existing approaches for obtaining differential privacy, which will be compared in the experimental section. Then, we propose a composite scheme for differential privacy using a recent message-passing method.

3.1 Model

We first present background on the statistical model. Both methods in the later method section use this formulation for the statistical model. This section largely follows notation from previous works (Herbrich, Rastogi, and Vollgraf 2020; Romijnders et al. 2023; Koller and Friedman 2009).

Every user on every day is modeled as a random variable that takes on one of four states, S, E, I, R. These states abbreviate for Susceptible, Exposed, Infected, and Recovered (Kermack and McKendrick 1927; Anderson and May 1992). This random variable is written as $z_{u,t}$ for user u, at time step t. The data set of observations is $D_{\mathcal{O}} = \{o_{u_i,t_i}\}_{i=1}^{O}$, which are O observations, each with an outcome $\{0,1\}$ for user u_i at time step t_i.

Test outcomes may have false positive or false negative results, with False Positive rate β (FPR) and False Negative

rate α (FNR). The model uses the observation distribution:

$$P\left(o_{u,t}|z_{u,t}\right) = \begin{cases} \alpha & \text{if } z_{u,t} = I \wedge o = 0 \\ 1 - \alpha & \text{if } z_{u,t} = I \wedge o = 1 \\ 1 - \beta & \text{if } z_{u,t} \in \{S, E, R\} \wedge o = 0 \\ \beta & \text{if } z_{u,t} \in \{S, E, R\} \wedge o = 1 \end{cases}.$$
$$(1)$$

The random variables $z_{u,t}$ are connected in two directions: over time, the variables evolve in a Markov chain $S \to E \to I \to R$; between users, a contact can influence the transition probability between states. Both interactions are summarized in the Markovian state transition:

$$P(z_{u,t+1}|z_{u,t}, z_{N(u,t)}) = \begin{cases} \psi(u, t, z_{N(u,t)}) & S \to S \\ 1 - \psi(u, t, z_{N(u,t)}) & S \to E \\ 1 - g & E \to E \\ g & E \to I \\ 1 - h & I \to I \\ h & I \to R \\ 1 & R \to R \\ 0 & \text{otherwise} \end{cases}$$
$$(2)$$

Here $\psi(\cdot)$ constitutes a noisy-OR model (Koller and Friedman 2009) that depends on states of other users:

$$\psi(u, t, z_{N(u,t)}) = (1 - p_0)(1 - p_1)^{|\{(v,u,t) \in D_c : z_{v,t} = I\}|}.$$
$$(3)$$

Here g, h, p_0, and p_1 are scalar model parameters, and they are set equal to the values from previous literature (Romijnders et al. 2023; Herbrich, Rastogi, and Vollgraf 2020). We highlight all parameter settings in Appendix C.6. $z_{N(u,t)}$ is the set of random variables of all contacts of user u at time step t. The data set of contacts, D_c, consists of a set of tuples $\{(u, v, t)\}$, where user u had a (directed) contact with user v at time step t. Equation 3 can be interpreted as a noisy-OR model, where every infected contact decreases the probability of remaining in S state.

3.2 DP contact tracing methods

We introduce three methods to obtain a differentially private COVIDSCORE, as defined in the attack model. We follow the conventional definition of (ε, δ) differential privacy (Dwork and Roth 2014; Mironov 2017) that says for every $\varepsilon > 0$, $\delta \in [0, 1]$, a mechanism $f(\cdot)$, for any outcome Φ in the range of $f(\cdot)$, and any two adjacent data sets D, D' that differ in at most one element, satisfy the following constraint:

$$p(f(D) \in \Phi) \leq e^\varepsilon p(f(D') \in \Phi) + \delta \qquad (4)$$

We define two data sets as adjacent when the COVIDSCORE of one contact differs between the data sets. The sensitivity, then, is the largest value change of a function between adjacent data sets:

$$\Delta \geq \max_{\{(D,D'):d(D,D')=1\}} \|f(D) - f(D')\|. \qquad (5)$$

Distance $d()$ is defined as the Hamming distance $d(D, D') = \sum_i \mathbf{1}[D_i \neq D'_i]$, where $\mathbf{1}[\cdot]$ is the indicator function and D_i is one of the contacts' COVIDSCORE. When the sensitivity of function $f()$ is bounded, a common mechanism is to add Gaussian noise. The Gaussian mechanism of (Dwork and Roth 2014) prescribes the noise variance for a particular sensitivity value, ε and δ.

We discuss three baseline approaches for experimental comparison: one based on traditional contact tracing, one based on previously studied Gibbs sampling, and one based on noising individual messages regardless of application.

Traditional contact tracing. In traditional contact tracing, users would test themselves when one of their recent contacts has tested positive. Many countries used this policy in the COVID19 pandemic (Baker et al. 2021). We implement this as a function that calculates the number of positive-testing contacts. If a positive test corresponds to 1 and a negative test corresponds to 0, this function has a sensitivity of 1, according to the definition in Equation 5. We use the Gaussian mechanism accordingly and release its output as the COVIDSCORE. The method is thus differentially private according to the (ε, δ) given by the Gaussian Mechanism (c.f. Appendix A of Dwork and Roth (2014)).

Gibbs sampling. Previous work proposed Gibbs sampling to estimate the COVIDSCORE in decentralized contact tracing (Herbrich, Rastogi, and Vollgraf 2020). For achieving DP, we use an existing method with ε-DP for a sample from a probability distribution with clipped likelihoods (Wang, Fienberg, and Smola 2015; Foulds et al. 2016). Inference for the model specified in Equation 2 makes estimates with Monte Carlo samples from a Gibbs chain. Therefore, if Gibbs samples were obtained under differential privacy, then the Monte Carlo estimate is DP by post-processing. The method provides an ε-DP, which is stronger than the (ε, δ)-DP of the other methods.

The Gibbs sampler has two hyperparameters: the value of B to clip the likelihoods and the number of Gibbs samples to draw. We found a value of $B = 10$ to work best. Determining the number of Gibbs samples constitutes a topic by itself (Robert and Casella 2004). Our case is even more complex as each additional sample improves the statistical estimate, but simultaneously increases the privacy bound. We find that taking 10 samples with 10 skip steps, after 100 burn-in steps, works best (Robert and Casella 2004); taking more samples would worsen the privacy bound, and taking fewer samples worsens the estimate for the COVIDSCORE.

Per-message differential privacy. As a third baseline, we compare against a form of differential privacy at the single message that is communicated between contacts, regardless of the contact tracing algorithm. As the message is a numerical value in the range $[0, 1)$, one can noise this message and consider the message-passing algorithm to be DP by the post-processing property (Dwork and Roth 2014). To message-passing algorithms are belief propagation (Herbrich, Rastogi, and Vollgraf 2020) and Factorised Neighbors (FN, Romijnders et al. (2023)). We use the latter method as a previous work shows that FN works better for these SEIR models (Romijnders et al. 2023).
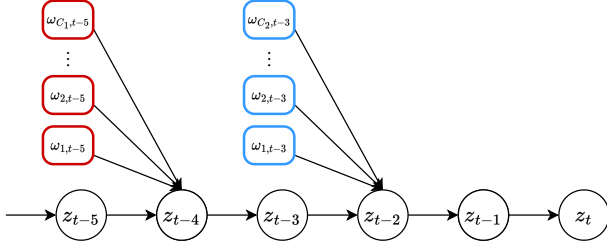
Figure 1: Example of a contact graph. This user has C_1 contacts at five time steps in the past and C_2 contacts at three time steps in the past. The released COVIDSCORE is the estimate of being in state I on time step t. Appendix C.5 generalizes the method for a general contact graph.

Dealing with the constraint that messages are in $[0, 1)$, we add noise in the logit domain. A message in the $[0, 1]$ domain corresponds to a message in $\mathbb{R}$, transformed by the logit transform $x = \log \frac{y}{1-y}$; and being calculated from the sigmoid function $y = \frac{1}{1+e^{-x}}$. If the messages are clipped to $[\gamma, 1 - \gamma]$, then the sensitivity of the mechanism in the logit-domain is $2 |\text{logit}(\gamma)|$. We use the Gaussian mechanism with this sensitivity and report results for the corresponding ε and δ values (c.f. appendix A of Dwork and Roth (2014)).

3.3 Differentially Private Factorized Neighbors

In contrast to the previous methods, we now use the structure of Factorized Neighbors (FN, (Rosen-Zvi, Jordan, and Yuille 2005; Romijnders et al. 2023)) to propose a novel algorithm. FN is a decentralized approximate inference method that calculates daily a COVIDSCORE, which represents the belief that the user is in the infected state the next day. The update equations for the model in Equation 2 were introduced in (Romijnders et al. 2023). This section analyzes the update equations and proposes a differential privacy method based on composite inputs named differentially private factorized neighbors (DPFN).

We analyze an example for one user and limited contacts here and generalize the method in Appendix C.5. Consider revealing the COVIDSCORE, $\phi_{u,t}$ for user u at day t. Let's say this user had C_1 contacts five days before and C_2 contacts three days before. Figure 1 presents an example of the corresponding contact graph. We rewrite $\omega_{u,t} = 1 - p_1\phi_{u,t}$ for reasons that will become clear shortly. Each contact, c, sends a message $\omega_{c,t} \in [0, 1]$ to user u, and this user calculates their COVIDSCORE. This version of the update equation will be referred to as $F_1(\cdot)$:

$$\phi_{u,t} = F_1(\omega_{1,t-5}, \omega_{2,t-5}, \cdots, \omega_{C_1,t-5},$$
$$\omega_{1,t-3}, \omega_{2,t-3}, \cdots, \omega_{C_2,t-3}). \quad (6)$$

F_1 makes a prediction as a function of $C_1 + C_2$ individual messages. However, when analyzing the update equations, the function F_1 only depends on messages that appear in a product term. Then one could rewrite the FN method to:

$$\phi_{u,t} = F_2(\prod_{i=1}^{C_1} \omega_{i,t-5}, \quad \prod_{i=1}^{C_2} \omega_{i,t-3}) \quad (7)$$

FN in the form of F_2 only depends on a product of messages. For this reason, we write $\omega_{u,t} = 1 - p_1\phi_{u,t}$. Thus, once such product is modified to have DP, the function F_2 will be private by the post-processing property (Dwork and Roth 2014). This was an example for two days, and in Appendix C.5, we prove that this decomposition holds for any number of days.

To derive a bound like Equation 4, we use a log-normal noise distribution for each message $\omega_{c,t}$, as the family of log-normal distributions is closed under multiplication. The log-normal distribution has a closed-form expression for its Rényi divergence, and we will prove DP via Rényi differential privacy (RDP, (Mironov 2017)).

A bound on the Rényi divergence can be converted to the ε and δ for DP (Mironov 2017). As such, we aim to bound the Rényi divergence between the two log-normal distributions that correspond to two adjacent data sets, and convert to (ε, δ)-DP later. For any two log-normal distributions, p_u and p_v, with mean parameters μ_u and μ_v, and with equal variance parameter $\sigma_*^2 = C\sigma^2$, the Rényi divergence is the following. We assume a product of C messages, each with a variance parameter σ^2. A detailed derivation is in Appendix C.3, where we also highlight the difference between this method and the Gaussian mechanism.

$$D_a(p_u|p_v) = \frac{a}{2C\sigma^2}(\mu_u - \mu_v)^2 \quad (8)$$

The divergence in Equation 8 decreases with the number of contacts C. So, the more contacts on a day, the smaller the divergence. It remains to upper bound the worst case of $(\mu_u - \mu_v)^2$ for any two adjacent data sets. In Appendix C.3 we show that for any two adjacent data sets, $(\mu_u - \mu_v)^2 \leq (\log(1 - \gamma_u p_1) - \log(1 - \gamma_l p_1))^2$. This bound is achieved by clipping every COVIDSCORE of the FN computation in the interval $[\gamma_l, \gamma_u]$. Parameter p_1 is a model parameter representing the probability that, given a contact, the virus transmits from user to user. Denoting the worst-case divergence in Equation 26 by ρ, we have a bounded Rényi divergence if the following holds:

$$\sigma^2 \geq \frac{a}{2C\rho}(\log(1 - \gamma_u p_1) - \log(1 - \gamma_l p_1))^2. \quad (9)$$

Equation 9 shows that more noise should be added whenever wider clipping values are used or when a user has fewer contacts. Experimentally, we find that tuning the clipping values could slightly improve the results, but another hyper-parameter increases the complexity of the method. Therefore, we run all experiments with $\gamma_l = 0$ and $\gamma_u = 1$.

Algorithm 1 summarizes the steps in calculating the COVIDSCORE with DPFN. The sample is from a log-normal distribution with the parameter $\mu = \omega_{*,t} - \frac{\sigma^2}{2}$. The variance parameter σ^2 follows from Equation 9 with a, ρ, and the specified number of contacts $C = |N(u, t)|$.
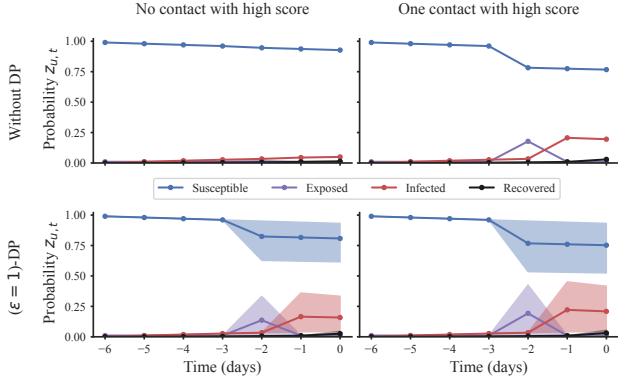
Figure 2: Showing the effect of differential privacy on the approximate inference from FN. An example user has two contacts. Both contacts have a low COVIDSCORE in the left column, while in the right column, one contact has a high COVIDSCORE. The red-shaded region indicates the 20-80 quantiles for sampling the COVIDSCORE from the DPFN mechanism of Algorithm 1. The regions overlap, which reflects privacy, but the median red line is higher in the right column, which indicates a possible infection and could inform a testing policy.

Visual example. We illustrate the effect of differential privacy on the estimates for the COVIDSCORE by FN. In this example, a user has two contacts at day -5. In the left column of Figure 2, both contacts have a low COVIDSCORE; in the right column, one contact has a high COVIDSCORE. The red line indicates the median estimate of being infected, i.e. the COVIDSCORE, and the shaded region indicates the 20-80 quantiles. Interpreting the definition of DP in Equation 4, changing the score of a single contact should not change the likelihood of an output *too much*. Whether the user has no contact with a high score (left column), or a single contact with a high score (right column), the shaded red regions in the figure overlap, which gives the contact plausible deniability against a potential adversary (Dwork and Roth 2014). The red median line, though, runs slightly higher in the right column, which is the utility needed for a contact tracing algorithm. Naturally, the added noise gives rise to a trade-off, where more noise increases privacy but decreases the utility for a subsequent testing policy and mitigation of a pandemic (Dinur and Nissim 2003). We address this trade-off in Section 4.

Optimize parameters in RDP. The bound in Equation 9 uses RDP, and depends on (a, ρ). Yet, we want to report (ε, δ)-DP. Previous literature optimizes for the optimal value a via a line search (Abadi et al. 2016). Fortunately, for our particular problem, we find a closed-form solution for optimal a and ρ, outlined in Appendix C.4. We arrive at the expression for the order a of RDP:

$$a = 1 + \frac{d + \sqrt{d(d + \varepsilon)}}{\varepsilon} \quad (10)$$

$$\rho = \varepsilon - d(a - 1)^{-1} \quad (11)$$

with $d = \log \frac{1}{\delta}$.

Algorithm 1: Differentially private factorized Neighbors

Input: Dataset of contacts' COVIDSCORE $D = \{\phi_{c,t}\}$ for all contacts c, t of user u in the set of neighbors $N(u, t)$
Parameter: Privacy parameters (ε, δ), model parameters are omitted for clarity
Output: COVIDSCORE for this user

1: Convert (ε, δ)-DP parameters to (a, ρ)-RDP parameters using Equation 10
2: Convert each $\phi_{c,t}$ to $\omega_{c,t}$ using Equation 56
3: **for** $t = -T, -T + 1, \cdots, -1$ **do**
4: $\quad \omega_{*,t} = \prod_{c \in N(u,t)} \omega_{c,t}$
5: $\quad$ Calculate $\sigma^2(a, \rho, |N(u, t)|)$ using Equation 9
6: $\quad$ Calculate $\mu(\omega_{*,t}, \sigma^2)$ using Equation 22
7: $\quad \tilde{\omega}_{*,t} \sim \text{log-normal}(\mu, \sigma^2)$
8: **end for**
9: **return** $F_2(\tilde{\omega}_{*,t=-T}, \tilde{\omega}_{*,t=-T+1}, \cdots, \tilde{\omega}_{*,t=-1})$

Assumptions on the algorithm: The inference runs for a specific time window, $t-T, t-T+1, \cdots, t-1, t$, and an estimate for the probability $p(z_{u,t} = I)$ is released to the user (i.e. using Gibbs sampling or FN). Only this COVIDSCORE is released to the user under DP, and inference is run unmodified, in an encrypted space, such as a trusted execution environment (Sabt, Achemlal, and Bouabdallah 2015). The differential privacy holds with respect to the message of a contacted user at the time step of the contact. If the user has no other contacts than an adversary, an adversary could gain more information through repeated contacts. In the worst case for K repeated contacts, the differential privacy parameters ε and δ increase K fold (Dwork and Roth 2014). We aim to investigate advanced composition bounds for this case in future work (McMahan et al. 2018). We assume to have access to a known contact graph, using methods as mentioned in the Related Work in Section 2. Finally, the attack outlined in the introduction assumes that the adversary uses a contact tracing app and does not want to get infected. Otherwise, a COVID19 test would reveal the health status.

4 Experiments and results

We test the differentially private contact tracing algorithms on two widely used simulators. We will explore the trade-off between privacy and utility in two experiments.

4.1 Simulators

The effect of a testing policy using the proposed method is tested on two simulators. These simulators are both calibrated to real-world data and account for different contact patterns based on age, profession, and type of household.

The OpenABM simulator (Hinch et al. 2021) uses a network-based process to generate contacts, and is calibrated against the UK for different age, household, and occupational networks patterns (school, work, and social network). The simulator has about 150 modifiable parameters, and we use the recommended settings – the same as used in Baker et al. (2021); Romijnders et al. (2023).
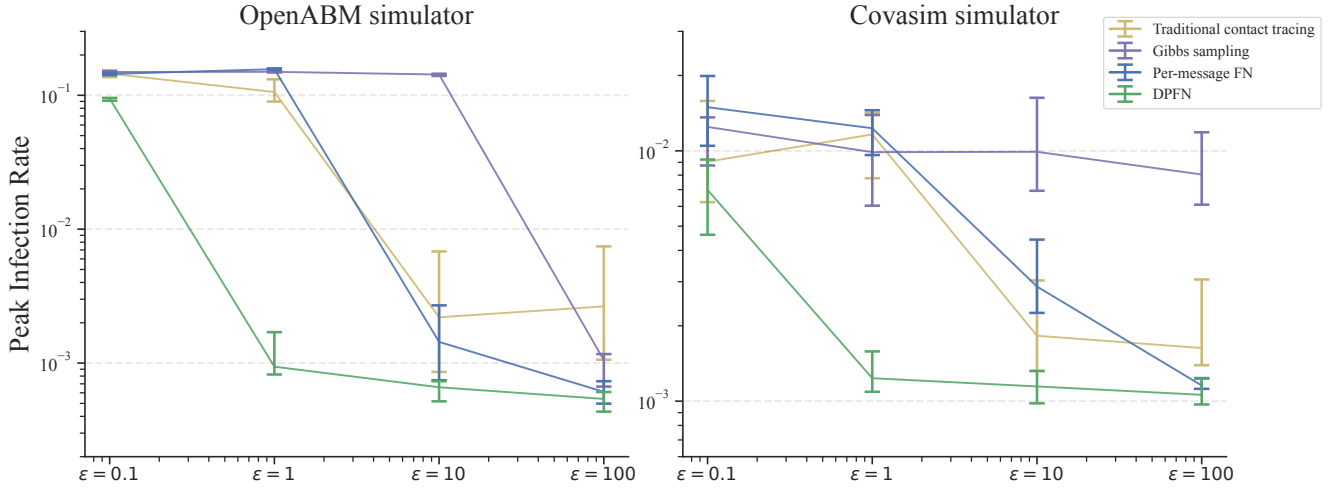
Figure 3: The privacy-utility trade-off for differentially private contact tracing. The y-axis indicates the Peak Infection rate, where lower is better. At $\varepsilon = 1$, a common setting for differential privacy, DPFN achieves a lower peak infection rate than all other methods. OpenABM and Covasim are the two most widely used simulators for COVID19. Error bars indicate 20-80 quantiles for ten random restarts.

The Covasim simulator (Kerr et al. 2021b) models different contact patterns in layers like households, schools, workplaces, and social communities. Results from this simulator were already used by policymakers (Panovska-Griffiths et al. 2020; Kerr et al. 2021a). The contacts are calibrated against a typical city in the USA, and the disease dynamics are stratified for ten age categories.

4.2 Experimental details

The experiments aim to compare the influence on the peak infection rate for the methods outlined before. Peak infection rate is a common metric to assess the capability of a protocol to mitigate the pandemic (Baker et al. 2021; Romijnders et al. 2023). The peak of the pandemic corresponds to most economic and societal consequences, as during that period the hospitals could overfill and governments might decide on a lockdown (Kaye et al. 2021a).

Test protocol. The test protocol is the same in all experiments. Each day, the decentralized algorithm predicts a COVIDSCORE per user, and users with the highest score, not currently in quarantine, receive a request to test for COVID19. Simulations on OpenABM test 10% of the population daily, and simulations on Covasim test only 2% of the population daily. Positively tested users go in isolation for ten days. We assume a 100% follow-up from users that are requested to test, Appendix A explores a scenario where the follow-up is less than 100%.

The simulation becomes increasingly challenging when tests for COVID19 have false positives and false negatives. The default FPR and the FNR are 1% and 0.1% respectively. To test the robustness to noisy tests, we increase these noise rates in an experiment similar to (Romijnders et al. 2023). The FPR increases up to a level of 25% and the FNR up to a level of 3% – these are the worst-case design specifications

as prescribed by the European centre for disease control during the COVID19 pandemic (ECDC 2021).

Simulation scale. Unless otherwise noted, we simulate a population of 100.000 users for 100 days for OpenABM and 91 days in the case of Covasim. At the start of the simulation, 25 people are infected, and interventions start after the third day of the simulation. Whenever a figure depicts an error bar or a table mentions an interval as a subscript, the number indicates the median, and the caps indicate the 20-80 quantiles of ten random restarts. The randomness between different seeds stems from the population dynamics, disease dynamics, and releasing noisy tests for the virus. The unit ‰ indicates one-per-thousand users.

Differential privacy levels. The δ forms an important parameter in differential privacy as this constitutes the probability of exceeding the ε bound. We set this value to $\frac{1}{1000}$ in all experiments. Existing literature prescribes that the δ parameter should be smaller than one divided by the data set size (Blanco-Justicia et al. 2022; Hsu et al. 2014; van Dijk and Nguyen 2023). Algorithm 1 uses a privacy bound for the contacts per day, and as our simulators have a max of 200 contacts per day, and an average of only fifteen contacts per day, $\frac{1}{1000}$ is well below the recommended standard.

4.3 Results for differential privacy

Increased privacy at higher performance. Figure 3 displays our main result, which is a trade-off for the privacy level. The x-axis varies the ε value for the differential privacy per message, as defined in Equation 4. On the y-axis, we plot the peak infection rate (PIR). A lower PIR is better, as this corresponds to fewer people simultaneously having the infection. Vice versa, a high PIR implies the occurrence of the pandemic with all its potential consequences.

Test setup	No privacy	DPFN	DPFN+
(fpr 0.0%; fnr 0.0%)	0.5 [0.5,0.6]	1.1 [0.7,1.4]	0.6 [0.5,0.7]
(fpr 1%; fnr 0.1%)	0.5 [0.4,0.6]	1.1 [0.9,1.7]	0.6 [0.5,0.6]
(fpr 10%; fnr 1%)	0.6 [0.5,0.8]	17.6 [11.4,20.4]	0.9 [0.7,1.0]
(fpr 25%; fnr 3%)	0.6 [0.5,0.8]	46.6 [40.4,48.0]	0.7 [0.6,0.8]
No testing	200 [190,212]		

Table 1: DP makes the model less robust against noisy tests (column *DPFN*), but more available tests can counteract this effect (column *DPFN+*). This result presents an important message to policymakers. The PIR can be low under private scenarios, but this requires more tests. All results are in 1 daily infection per thousand users (‰).

Figure 3 shows that for the values of $\varepsilon = 1$ per message, DPFN results in a PIR below 1%, whereas other methods, such as Traditional contact tracing, only achieve a low PIR at ten times as large value for ε. From here onwards, we focus on the $\varepsilon = 1$ case as many studies advise this setting for differential privacy (Hsu et al. 2014; Blanco-Justicia et al. 2022; Dyda et al. 2021; Wood et al. 2018).

On both simulators, one observes that our DPFN method achieves better PIR than the per-message FN method at $\varepsilon = 1$. DPFN introduces noise later in the computation, which we hypothesize maintains a better utility. In terms of PIR, Gibbs sampling does worse than all other methods. This may have two reasons: a) under differential privacy, the Gibbs chain does not converge to the correct distribution (Wang, Fienberg, and Smola 2015), or b) the high number of samples necessitates too high differential privacy budget, (ε, δ), to have good utility at low budget. The results for the Covasim simulator in the right column of Figure 3 generally display more minor differences in PIR between high and low values of ε. This stems from different modeling assumptions between the Covasim and the OpenABM simulator.

Stability of our DPFN algorithm. We also explore the relation between more noisy tests for COVID19 and the noise due to differential privacy. In Table 1, we increase the FPR and FNR, which has been highlighted as a challenging scenario for traditional contact tracing apps (Reichert, Brack, and Scheuermann 2021).

Differential privacy makes FN less robust to noisy tests, but this can be counteracted by running more tests. Compared to the *no privacy* column, the column *DPFN* of Table 1 shows that under differential privacy, at $\varepsilon = 1$, the PIR is an order of magnitude higher. At the highest FPR and FNR, the PIR increases from 0.6 ‰ to 46.6 ‰. However, adding more available tests can counteract this noise. For column *DPFN+*, we increase the daily testing budget from 10% to 15%, and the peak infection rates are again similarly

#Agents	Traditional (‰)	DPFN (‰)
50,000	123.2 [108.3,136.4]	2.5 [1.8,3.7]
100,000	121.6 [110.0,134.3]	1.5 [1.0,1.9]
500,000	134.9 [134.2,135.6]	0.5 [0.4,0.5]
1,000,000	134.4 [133.8,135.0]	0.2 [0.2,0.2]

Table 2: Evaluating our algorithm at larger population scales of the simulator. To date, the result with 1 million users is the largest simulation reported for statistical contact tracing. Even at this scale, we show that DPFN results in significantly lower PIR compared to traditional contact tracing.

low as the *no privacy* column. This shows that using more available tests can compensate for the noise resulting from differential privacy. Table 1 provides an important message to policymakers who need to make a trade-off between infection rates and privacy.

Scaling to 1M agents. Table 2 shows the simulation running with $\varepsilon = 1$ at different population scales. All simulations in this paper are run with 100.000 users, but this table shows that the benefits of DPFN continue even at the million scale. From as small as fifty thousand users to as large as one million users, the DPFN method results in a significantly lower peak infection rate compared to traditional contact tracing. We emphasize that this is the largest simulation to date for statistical contact tracing.

5 Discussion and conclusion

We propose a differentially private algorithm for releasing a COVIDSCORE that depends on decentralized communication between contacts. This algorithm protects against a newly identified privacy attack where an adversary aims to reconstruct the COVIDSCORE. Our algorithm results in a two to ten-fold decrease in the peak infection rate compared to other approaches like Gibbs sampling and traditional contact tracing. This improvement holds at $\varepsilon = 1$ per message, while other methods only achieve similar results with $\varepsilon \geq 10$. We evaluate the algorithm on two widely used simulators, and we are the first to evaluate these algorithms at a scale of a million agents, where DPFN again achieves lower PIR than traditional contact tracing.

We see two important directions for future research. First is the study of repeated contacts. Advanced composition bounds are needed to describe privacy when a user has no other contacts but repeated contacts with an adversary. Secondly, our algorithm assumes full adoption of a contact tracing app, but more research is needed into partial adoption. We discuss these and other implications of automated decision-making in Appendix B.

Contact tracing will be one of our first lines of defense to understand and mitigate a virus whenever a new pandemic arises. As argued in the introduction, studies show that privacy concerns are among the top three concerns for adopting a contact tracing app. We believe differential privacy is an essential assurance towards the safe use of contact tracing.

References

Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H. B.; Mironov, I.; Talwar, K.; and Zhang, L. 2016. Deep learning with differential privacy. In *ACM SIGSAC conference on computer and communications security*.

Ahmed, N.; Michelin, R. A.; Xue, W.; Ruj, S.; Malaney, R.; Kanhere, S. S.; Seneviratne, A.; Hu, W.; Janicke, H.; and Jha, S. K. 2020. A survey of COVID-19 contact tracing apps. *IEEE access*.

Ali, J.; and Dyo, V. 2021. Cross hashing: Anonymizing encounters in decentralised contact tracing protocols. In *IEEE International Conference on Information Networking*.

Alsdurf, H.; Bengio, Y.; Deleu, T.; Gupta, P.; Ippolito, D.; Janda, R.; Jarvie, M.; Kolody, T.; Krastev, S.; Maharaj, T.; Obryk, R.; Pilat, D.; Pisano, V.; Prud'homme, B.; Qu, M.; Rahaman, N.; Rish, I.; Rousseau, J.; Sharma, A.; Struck, B.; Tang, J.; Weiss, M.; and Yu, Y. W. 2020. COVI White Paper. arXiv:2005.08502.

Anderson, R. M.; and May, R. M. 1992. *Infectious diseases of humans: dynamics and control*. Oxford university press.

Apple; and Google. 2020. Privacy-preserving contact tracing. *apple.com/covid19/contacttracing/, (last accessed August 2023)*.

Baker, A.; Biazzo, I.; Braunstein, A.; Catania, G.; Dall'Asta, L.; Ingrosso, A.; Krzakala, F.; Mazza, F.; Mézard, M.; Muntoni, A. P.; et al. 2021. Epidemic mitigation by statistical inference from contact tracing data. *Proceedings of the National Academy of Sciences*.

Bay, J.; Kek, J.; Tan, A.; Hau, C. S.; Yongquan, L.; Tan, J.; and Quy, T. A. 2020. BlueTrace: A privacy-preserving protocol for community-driven contact tracing across borders. *Government Technology Agency-Singapore, Tech. Rep.*

Beaver, D.; Micali, S.; and Rogaway, P. 1990. The round complexity of secure protocols. In *ACM symposium on Theory of computing*.

Ben-Efraim, A.; Lindell, Y.; and Omri, E. 2016. Optimizing semi-honest secure multiparty computation for the internet. In *ACM SIGSAC Conference on Computer and Communications Security*.

Bengio, Y.; Janda, R.; Yu, Y. W.; Ippolito, D.; Jarvie, M.; Pilat, D.; Struck, B.; Krastev, S.; and Sharma, A. 2020. The need for privacy with public digital contact tracing during the COVID-19 pandemic. *The Lancet Digital Health*.

Berger, D. W.; Herkenhoff, K. F.; and Mongey, S. 2020. An seir infectious disease model with testing and conditional quarantine. Technical report, National Bureau of Economic Research.

Blanco-Justicia, A.; Sánchez, D.; Domingo-Ferrer, J.; and Muralidhar, K. 2022. A critical review on the use (and misuse) of differential privacy in machine learning. *ACM Computing Surveys*.

Boden, M.; Zimmerman, L.; Azevedo, K. J.; Ruzek, J. I.; Gala, S.; Magid, H. S. A.; Cohen, N.; Walser, R.; Mahtani, N. D.; Hoggatt, K. J.; et al. 2021. Addressing the mental health impact of COVID-19 through population health. *Clinical Psychology Review*.

Boyd, S. P.; and Vandenberghe, L. 2014. *Convex Optimization*. Cambridge University Press.

Braunstein, A.; Catania, G.; Dall'Asta, L.; Mariani, M.; Mazza, F.; and Tarabolo, M. 2023. Small-Coupling Dynamic Cavity: a Bayesian mean-field framework for epidemic inference. arXiv:2306.03829.

Burdinski, A.; Brockmann, D.; and Maier, B. F. 2022. Understanding the impact of digital contact tracing during the COVID-19 pandemic. *PLOS Digital Health*.

Chan, J.; Foster, D.; Gollakota, S.; Horvitz, E.; Jaeger, J.; Kakade, S.; Kohno, T.; Langford, J.; Larson, J.; Sharma, P.; et al. 2020. Pact: Privacy sensitive protocols and mechanisms for mobile contact tracing. arXiv:2004.03544.

Cho, H.; Ippolito, D.; and Yu, Y. W. 2020. Contact Tracing Mobile Apps for COVID-19: Privacy Considerations and Related Tradeoffs. arXiv:2003.11511.

Dinur, I.; and Nissim, K. 2003. Revealing information while preserving privacy. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*.

Dwork, C.; and Roth, A. 2014. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends in Theoretical Computer Science*.

Dyda, A.; Purcell, M.; Curtis, S.; Field, E.; Pillai, P.; Ricardo, K.; Weng, H.; Moore, J. C.; Hewett, M.; Williams, G.; et al. 2021. Differential privacy for public health data: An innovative tool to optimize information sharing while protecting data confidentiality. *Patterns*.

ECDC. 2021. Considerations on the use of self-tests for COVID-19 in the EU/EEA. *European Centre for Disease prevention and control, technical report, 17 March 2021*.

Farrand, T.; Mireshghallah, F.; Singh, S.; and Trask, A. 2020. Neither private nor fair: Impact of data imbalance on utility and fairness in differential privacy. In *Proceedings of the 2020 workshop on privacy-preserving machine learning in practice*.

Foulds, J. R.; Geumlek, J.; Welling, M.; and Chaudhuri, K. 2016. On the Theory and Practice of Privacy-Preserving Bayesian Data Analysis. In *UAI*.

Gao, G.; Lang, R.; Oxoby, R. J.; Mourali, M.; Sheikh, H.; Fullerton, M. M.; Tang, T.; Manns, B. J.; Marshall, D. A.; Hu, J.; et al. 2022. Drivers of downloading and reasons for not downloading COVID-19 contact tracing and exposure notification apps: A national cross-sectional survey. *PLOS one*.

Gil, M.; Alajaji, F.; and Linder, T. 2013. Rényi divergence measures for commonly used univariate continuous distributions. *Information Sciences*.

Grantz, K. H.; Meredith, H. R.; Cummings, D. A.; Metcalf, C. J. E.; Grenfell, B. T.; Giles, J. R.; Mehta, S.; Solomon, S.; Labrique, A.; Kishore, N.; et al. 2020. The use of mobile phone data to inform analysis of COVID-19 pandemic epidemiology. *Nature communications*.

Heikkilä, M.; Jälkö, J.; Dikmen, O.; and Honkela, A. 2019. Differentially private markov chain monte carlo. *NeurIPS*.

Herbrich, R.; Rastogi, R.; and Vollgraf, R. 2020. CRISP: A Probabilistic Model for Individual-Level COVID-19 Infection Risk Estimation Based on Contact Data. arXiv:2006.04942.

Hinch, R.; Probert, W. J. M.; Nurtay, A.; Kendall, M.; Wymant, C.; Hall, M.; Lythgoe, K. A.; Cruz, A. B.; Zhao, L.; Stewart, A.; Ferretti, L.; Montero, D.; Warren, J.; Mather, N.; Abueg, M.; Wu, N.; Legat, O.; Bentley, K.; Mead, T.; Van-Vuuren, K.; Feldner-Busztin, D.; Ristori, T.; Finkelstein, A.; Bonsall, D. G.; Abeler-Dörner, L.; and Fraser, C. 2021. OpenABM-Covid19 - An agent-based model for non-pharmaceutical interventions against COVID-19 including contact tracing. *PLoS Computational Biology*.

Hsu, J.; Gaboardi, M.; Haeberlen, A.; Khanna, S.; Narayan, A.; Pierce, B. C.; and Roth, A. 2014. Differential privacy: An economic method for choosing epsilon. In *2014 IEEE 27th Computer Security Foundations Symposium*. IEEE.

Jones, K.; Thompson, R.; et al. 2021. To use or not to use a COVID-19 contact tracing app: Mixed methods survey in Wales. *JMIR mHealth and uHealth*.

Kairouz, P.; Oh, S.; and Viswanath, P. 2015. The Composition Theorem for Differential Privacy. In *International Conference on Machine Learning, ICML*.

Kaye, A. D.; Okeagu, C. N.; Pham, A. D.; Silva, R. A.; Hurley, J. J.; Arron, B. L.; Sarfraz, N.; Lee, H. N.; Ghali, G. E.; Gamble, J. W.; et al. 2021a. Economic impact of COVID-19 pandemic on healthcare facilities and systems: International perspectives. *Best Practice and Research Clinical Anaesthesiology*.

Kaye, A. D.; Okeagu, C. N.; Pham, A. D.; Silva, R. A.; Hurley, J. J.; Arron, B. L.; Sarfraz, N.; Lee, H. N.; Ghali, G. E.; Gamble, J. W.; et al. 2021b. Economic impact of COVID-19 pandemic on healthcare facilities and systems: International perspectives. *Best Practice & Research Clinical Anaesthesiology*.

Kermack, W. O.; and McKendrick, A. 1927. A contribution to the mathematical theory of epidemics. *Proceedings of the royal society of London*.

Kerr, C. C.; Mistry, D.; Stuart, R. M.; Rosenfeld, K.; Hart, G. R.; Núñez, R. C.; Cohen, J. A.; Selvaraj, P.; Abeysuriya, R. G.; Jastrzebski, M.; et al. 2021a. Controlling COVID-19 via test-trace-quarantine. *Nature communications*.

Kerr, C. C.; Stuart, R. M.; Mistry, D.; Abeysuriya, R. G.; Rosenfeld, K.; Hart, G. R.; Núñez, R. C.; Cohen, J. A.; Selvaraj, P.; Hagedorn, B.; George, L.; Jastrzebski, M.; Izzo, A. S.; Fowler, G.; Palmer, A.; Delport, D.; Scott, N.; Kelly, S. L.; Bennette, C. S.; Wagner, B. G.; Chang, S. T.; Oron, A. P.; Wenger, E. A.; Panovska-Griffiths, J.; Famulare, M.; and Klein, D. J. 2021b. Covasim: An agent-based model of COVID-19 dynamics and interventions. *PLOS Computational Biology*.

Kim, D.; Min, H.; Nam, Y.; Song, H.; Yoon, S.; Kim, M.; and Lee, J. 2022. COVID-EENet: Predicting Fine-Grained Impact of COVID-19 on Local Economies. In *AAAI*.

Koller, D.; and Friedman, N. 2009. *Probabilistic Graphical Models - Principles and Techniques*. MIT Press.

Lam, S. K.; Pitrou, A.; and Seibert, S. 2015. Numba: A llvm-based python jit compiler. In *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*.

Li, B.; and Saad, D. 2021. Impact of presymptomatic transmission on epidemic spreading in contact networks: A dynamic message-passing analysis. *Physical Review E*.

McMahan, H. B.; Ramage, D.; Talwar, K.; and Zhang, L. 2018. Learning Differentially Private Recurrent Language Models. In *ICLR*.

Mironov, I. 2017. Rényi Differential Privacy. In *IEEE Computer Security Foundations Symposium*.

Panovska-Griffiths, J.; Kerr, C. C.; Stuart, R. M.; Mistry, D.; Klein, D. J.; Viner, R. M.; and Bonell, C. 2020. Determining the optimal strategy for reopening schools, the impact of test and trace interventions, and the risk of occurrence of a second COVID-19 epidemic wave in the UK: a modelling study. *The Lancet Child & Adolescent Health*.

Park, S.; Choi, G. J.; and Ko, H. 2020. Information Technology–Based Tracing Strategy in Response to COVID-19 in South Korea—Privacy Controversies. *Journal of the American Medical Association*.

Perra, N. 2021. Non-pharmaceutical interventions during the COVID-19 pandemic: A review. *Physics Reports*.

Raskar, R.; Schunemann, I.; Barbar, R.; Vilcans, K.; Gray, J.; Vepakomma, P.; Kapa, S.; Nuzzo, A.; Gupta, R.; Berke, A.; et al. 2020. Apps gone rogue: Maintaining personal privacy in an epidemic. arXiv:2003.08567.

Reichert, L.; Brack, S.; and Scheuermann, B. 2021. A survey of automatic contact tracing approaches using Bluetooth Low Energy. *ACM Transactions on Computing for Healthcare*.

Robert, C. P.; and Casella, G. 2004. *Monte Carlo Statistical Methods*. Springer.

Romijnders, R.; Asano, Y.; Louizos, C.; and Welling, M. 2023. No time to waste: practical statistical contact tracing with few low-bit messages. *AISTATS*.

Rosen-Zvi, M.; Jordan, M. I.; and Yuille, A. L. 2005. The DLR Hierarchy of Approximate Inference. *UAI*.

Sabt, M.; Achemlal, M.; and Bouabdallah, A. 2015. Trusted execution environment: what it is, and what it is not. In *IEEE Trustcom/BigDataSE/Ispa*. IEEE.

Troncoso, C.; Payer, M.; Hubaux, J.-P.; Salathé, M.; Larus, J.; Bugnion, E.; Lueks, W.; Stadler, T.; Pyrgelis, A.; Antonioli, D.; et al. 2020. Decentralized privacy-preserving proximity tracing. arXiv:2005.12273.

Vadrevu, P. K.; Adusumalli, S. K.; and Mangalapalli, V. K. 2020. A hybrid approach for personal differential privacy preservation in homogeneous and heterogeneous health data sharing. *High Technology Letters*.

van Dijk, M.; and Nguyen, P. H. 2023. Considerations on the Theory of Training Models with Differential Privacy. arXiv:2303.04676.

Vepakomma, P.; Pushpita, S. N.; and Raskar, R. 2021. DAMS: Meta-estimation of private sketch data structures for differentially private COVID-19 contact tracing. Technical report, Accessed: June 14, 2023. [Online]. Available: https://www.media.mit.edu/publications/dams-meta-estimation-of-private-sketch-data-structures-for-differentially-private-covid-19-contact-tracing/.

Vindegaard, N.; and Benros, M. E. 2020. COVID-19 pandemic and mental health consequences: Systematic review of the current evidence. *Brain, Behavior, and Immunity*.

Walrave, M.; Waeterloos, C.; and Ponnet, K. 2022. Reasons for nonuse, discontinuation of use, and acceptance of additional functionalities of a COVID-19 contact tracing app: cross-sectional survey study. *JMIR Public Health and Surveillance*.

Wang, Y.-X.; Fienberg, S.; and Smola, A. 2015. Privacy for free: Posterior sampling and stochastic gradient monte carlo. In *International Conference on Machine Learning*.

Wood, A.; Altman, M.; Bembenek, A.; Bun, M.; Gaboardi, M.; Honaker, J.; Nissim, K.; O'Brien, D. R.; Steinke, T.; and Vadhan, S. 2018. Differential privacy: A primer for a non-technical audience. *Vanderbilt Journal of Entertainment and Technology Law*.

Yıldırım, S.; and Ermiş, B. 2019. Exact MCMC with differentially private moves: revisiting the penalty algorithm in a data privacy framework. *Statistics and Computing*.

Zhang, J.; Cormode, G.; Procopiuc, C. M.; Srivastava, D.; and Xiao, X. 2017. PrivBayes: Private Data Release via Bayesian Networks. *ACM Transactions on Database Systems*.

Zou, J.; and Fekri, F. 2015. A belief propagation approach to privacy-preserving item-based collaborative filtering. *IEEE Journal of Selected Topics in Signal Processing*.

A   Additional results

Our paper presents two trade-offs when designing a DP contact tracing algorithm. Figure 3 presents a trade-off with the level of privacy. In this extra analysis, we look further at the financial cost of contact tracing. A contact tracing algorithm, fundamentally, decides who to get tested based on earlier test results and the contacts between users. With fewer tests daily available, the COVIDSCORE is less precise and achieving a low PIR becomes harder. Producing extra available tests comes at a cost, and restrictions may apply. To this end, we vary the number of available tests while setting DP at the conventionally accepted value of $\varepsilon = 1$, and run this simulation on the Covasim simulator. Figure 6 shows the result of this experiment. For a small number of tests, both DPFN and traditional contact tracing result in a too high peak infection rate. Achieving a result below 0.003 PIR, the traditional contact tracing requires 10,000 tests, while DPFN requires only 500 tests. We conclude from this figure that DPFN achieves a low PIR with fewer tests. Additionally, this figure shows the trade-off between peak infection rate and financial cost in the form of the number of available tests.

This section further presents three additional results.

Plotting recall and average precision: Figure 4 uses the same experimental settings as Figure 3 and plots the individual timesteps on the x-axis. The shaded region indicates the 20-80 quantiles of 20 random restarts. Recall is calculated as the ratio of the number of infectious agents in quarantine divided by the total number of infectious agents. Average precision is calculated on the COVIDSCORE against the binary indicator of being infectious – a high average precision means the COVIDSCORE can be used to discriminate between infectious and non-infectious users.

The Figure plots results for high ε values (10, 100) for comparison. These values are not considered private enough (Hsu et al. 2014; Wood et al. 2018), but show a comparison for recall and average precision. Conversely, setting $\varepsilon = 0.1$ results in too high infection rates. The curve of $\varepsilon = 1$ strikes a balance between differential privacy and low infection rates.

Scale experiment on Covasim simulator: We reproduce the scale experiments of Table 2. Table 3 shows the result. Due to the differences in dynamics across different scales, the peak infection rates at particular scales vary. However, our DPFN method achieves a lower peak infection rate across different simulation scales.

Loss to follow-up: Here we vary the loss to follow-up from a test result. All experiments assume a user isolates after receiving a positive test for COVID19. If the user ignores this signal, there are two consequences: a) a user that is likely infectious continues to interact with other users, possibly spreading the virus; b) there is an opportunity cost as the test might have been sent to another user. The Covasim simulator can simulate this scenario where a user fails to follow up on a positive test. We directly modulate the `loss-prob` parameter in their source code, and Figure 5 shows the results. One can make two observations from the figure. Firstly, up to a follow-up loss of 75%, the DPFN algo-

rithm achieves lower PIR than the traditional method. Secondly, the DPFN method shows robustness by getting lower than 0.5% PIR for a probability of loss up to 20%.

B   Algorithmic decision-making

This section provides background on three important topics in this paper.

Differential privacy. We use differential privacy to quantify the level of privacy when releasing a single covidscore. Besides the conventional definition in Equation 4, there is a more intuitive interpretation in terms of type 1 and type 2 error rates that an adversary could have if an attack is attempted (Kairouz, Oh, and Viswanath 2015). The DP noise inherently decreases the algorithm's utility. To this end, we have plotted the trade-offs of infection rate against privacy level in Figure 3 and against financial cost in the form of testing capacity in Figure 6. We hope these figures can inform policymakers about the available settings for differentially private contact tracing algorithms.

Unintended consequences and use of data. All algorithms in this paper use only contacts and COVID19 tests as input data. No other features about users are used. We assume that contacts have been established securely, for which we cite references in Section 2. We assume that the test results are processed in a trusted execution environment (Sabt, Achemlal, and Bouabdallah 2015).

Despite these restrictions on the dataset, there are open questions about the consequences of automating societal decisions. In our particular setting, we see at least three relevant issues. Firstly, unequal access to smartphones could have unintended consequences, such as disparate infection rates in particular groups with unequal access to smartphones. Secondly, even under equal access, partial app adoption could reduce performance due to the lack of information and whether a contact occurred. We studied a somewhat related effect in Figure 5, but more research on this topic is necessary. Lastly, previous research has shown that DP can exacerbate biases existing in the data. This effect has been studied in Farrand et al. (2020) and should be investigated in the decentralized setting of contact tracing. We believe that studying these effects holistically is important before deploying algorithmic decision-making in the real world.

Data retention. The data retention of a potential app should be limited in scope. To this end, all algorithms in this paper use the data for only a fourteen-day window, and the data is deleted automatically afterward. Our implementation of DPFN is purely stateless. This means no state about a person or health status is maintained; data can be removed at any time during the time window, and removed contacts will not influence any prediction being made the next day. Appendix Section C.6 and Table 4 report on the impact of setting such a window and show that any window length longer than fourteen days only slightly improves the results.

C   Additional details on the method

C.1   Notation

In this paper and the rest of the appendix, we use the following notation:
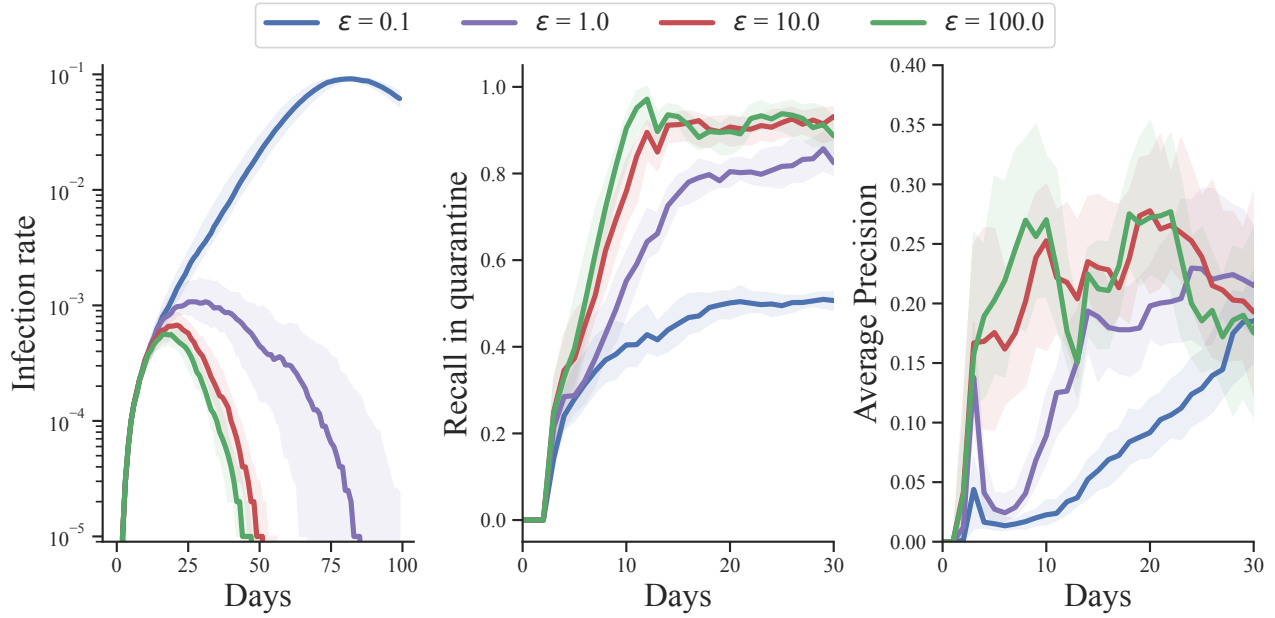
Figure 4: Recall and average precision during a simulation on OpenABM. The shaded regions indicate the 20-80 quantiles of twenty random restarts. The peak infection rate of the curve $\varepsilon = 1$ is in between lower and higher privacy levels. This effect is observed as well in intermediate values for the recall and average precision during the crucial first month of the epidemic simulation. The recall and average precision diagrams only plot 30 days of simulation, which is the crucial phase for a pandemic (Perra 2021).
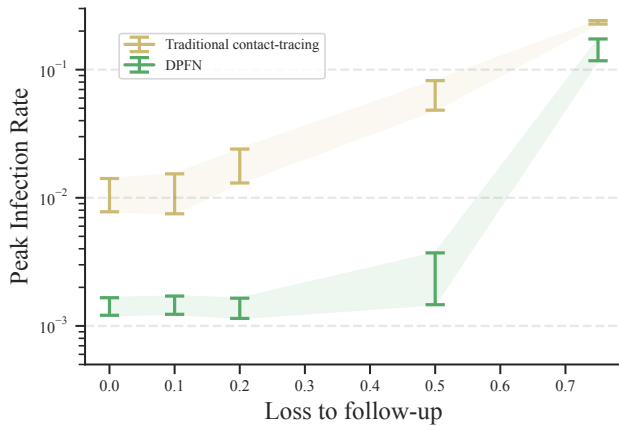

Figure 5: This figure explores the scenario where a user tests positive, but does not isolate. 'Loss to follow-up' is the probability that a user ignores the request to isolate after a positive test. The DPFN method achieves lower PIR than traditional contact tracing across a wide range of the 'loss to follow-up' probability.
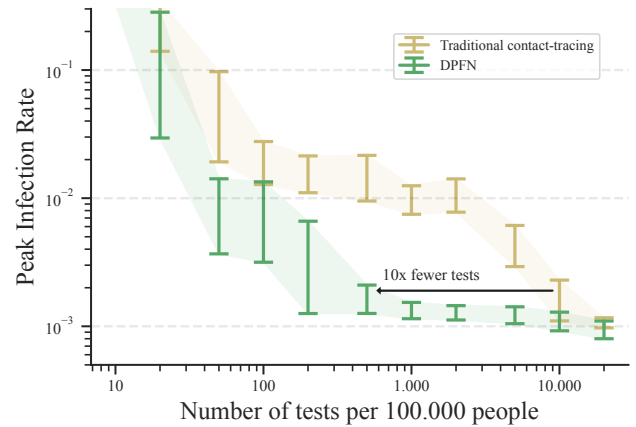

Figure 6: We vary the number of tests daily available in the simulation. Generally, more available tests provide more information to the contact tracing algorithm – c.f. Table 1. Considering a peak infection rate of 0.003, the traditional contact tracing method achieves this low PIR with 10,000 tests, while DPFN only requires 500 tests for the same PIR. This figure is made with $\varepsilon = 1$.

- $z_{u,t} \in \{S, E, I, R\}$: The random variable which takes on value for the states Susceptible, Exposed, Infectious, Recovered;

- $N(u,t)$: the set of neighbors of user u at timestep t. When it is clear from context $N(u)$ or $N(t)$ might be used to abbreviate for $N(u,t)$;

- $z_{N(u,t)}$: the set of random variables that are in the neighborhood of $z_{u,t}$, these are the states of contacts of user u at time step t;

- $D_{\mathcal{O}} = \{o_{u_i,t_i}\}_{i=1}^{O}$: The data set of observations with O observations, each with an outcome $\{0, 1\}$ for user u_i at time step t_i. These observations have false positives and false negatives with respect to the underlying state; this observation model is mentioned in Equation 1.

- $\phi_{c,t} \in [0, 1)$: General COVIDSCORE for a contact c that occurred on time t;

- $\omega_{c,t} \in (1 - p_1, 1]$: Message from a contact c at time step t. Related to $\phi_{c,t}$ by Equation 56;

- $C \in \mathbb{N}$: capital letter C is generally used to indicate the number of contacts, $C_t = |N(t)|$ for a particular user;

- $p_0, p_1 \in [0, 1)$: model parameters, p_0 is the probability for spontaneous infection, p_1 is the probability of the transmission of the virus given the occurrence of a contact and is used in Equation 3;

- $g, h \in (0, 1)$: model parameters, g is the probability of the state transition $E \rightarrow I$, h is the probability of the state transition $I \rightarrow R$;

- $b_u(z_u)$: FN belief over random variable z_u;

- $B_{N(u)}$: joint beliefs over a set of random variables, in this case, $N(u)$ is the set of neighbors, contacts, of user u;

- $\mu, \sigma^2 \in \mathbb{R}^+$: general parameters for the log-normal distribution;

- $a > 1, \rho > 0$: a is the order of the Rényi divergence, ρ is the upper bound on the Rényi divergence;

- $D_a(\cdot|\cdot)$: the divergence of order a between two probability distributions, in this paper we solely use Rényi divergence;

- $\varepsilon > 0, \delta \in [0, 1]$: parameters for (ε, δ) Differential privacy, defined in Equation 4; for brevity, normal letter d refers to $\log \frac{1}{\delta}$;

- $F(\cdot), F_1(\cdot), F_2(\cdot)$: these functions all refer to the Factorized Neighbor function, this approximate inference method was introduced by (Rosen-Zvi, Jordan, and Yuille 2005) and the update equations for an SEIR epidemic model were derived by (Romijnders et al. 2023);

- α, β: False Positive and False negative rate for the COVID19 tests, these parameters are assumed to be known by the statistical model during inference;

- γ generally indicates a clipping value. For the per-message methods, values are clipped in $[\gamma, 1 - \gamma]$, for the DPFN, messages are clipped in $[\gamma_l, \gamma_u]$

C.2 Properties of the log-normal distribution

Our method makes extensive use of the log-normal distribution and we outline a few essential properties in this section. We define as log-normal distribution:

$$p(x) = \frac{1}{x\sigma\sqrt{2\pi}} \quad \exp\{\frac{-1}{2\sigma^2}(\log(x) - \mu)^2\} \qquad (12)$$

This distribution has mean parameter $\mu \in \mathbb{R}$, and variance parameter $\sigma \in \mathbb{R}^+$. The domain is $x \in \mathbb{R}^+$, and the expected value is

$$m = E[x] = \exp\{\mu + \frac{\sigma^2}{2}\}. \qquad (13)$$

We will refer to μ as the mean parameter and refer to m as the expected value. Similarly, we will refer to σ^2 as the variance parameter.

One can rewrite Equation 13 for the μ parameter:

$$\mu = \log(m) - \frac{\sigma^2}{2}. \qquad (14)$$

The Rényi divergence between two probability distributions, p_u and p_v, and respective parameters, μ_u, σ_u^2, μ_v, σ_v^2 (Gil, Alajaji, and Linder 2013) is the following:

$$D_a(p_u|p_v) =$$
$$\log\left(\frac{\sigma_v}{\sigma_u}\right) + \frac{1}{2(a-1)}\log\left(\frac{\sigma_v^2}{\sigma_*^2}\right) + \frac{a}{2}\frac{(\mu_u - \mu_v)^2}{\sigma_*^2} \quad (15)$$

$$\sigma_*^2 = a\sigma_v^2 + (1-a)\sigma_u^2. \qquad (16)$$

The family of log-normal distributions is closed under multiplication. If $X_j \sim \text{log-normal}(\mu_j, \sigma_j^2)$ are n independent, log-normally distributed variables, then:

$$Y = \prod_{j=1}^{n} X_j \sim \text{log-normal}\left(\sum_{j=1}^{n} \mu_j, \sum_{j=1}^{n} \sigma_j^2\right). \qquad (17)$$

C.3 Rényi Differential Privacy of Algorithm 1

This section derives the Rényi differential privacy for the method outlined in Algorithm 1. Previous work established a relation between bounding the Rényi divergence and converting the bound to the ε and δ for DP (Mironov 2017). As such, we bound the Rényi divergence between the product terms that are required each day for the computation of $F_2(\cdot)$ (defined in Section 3). The conversion from (a, ρ)-RDP to (ε, δ) is discussed in Appendix C.4.

Appendix C.2 overviews properties of the log-normal distribution. We know that the log-normal distribution is closed under multiplication, and its Rényi divergence has a closed-form expression (Gil, Alajaji, and Linder 2013). We will bound the Rényi divergence of two product terms, $\omega_{*,t}$, under the worst-case adjacent data sets. In Section 3 we showed that FN computation depends on a product of messages $\omega_{c,t}$. The value of $\omega_{u,t}$ represents a function of the

#Agents	Traditional (‰)	FN (‰)
50,000	7.6 [5.8,9.9]	1.1 [0.8,1.4]
100,000	11.6 [7.8,14.1]	1.0 [1.0,1.4]
500,000	2.1 [0.5,2.7]	0.2 [0.2,0.3]
1,000,000	1.9 [1.0,2.6]	0.2 [0.2,0.2]

Table 3: Reproducing Table 2 on the Covasim simulator. These simulators have different dynamics. Still, across scales, our differentially private decentralized algorithm achieves lower peak infection rates.

COVIDSCORE that a contact communicates by the relation $\omega_{c,t} = 1 - p_1\phi_{c,t}$. This is explained after Equation 56.

We start by writing the Rényi divergence of this product term. Take two adjacent data sets, according to Equation 5, and name the distributions over their product p_u and p_v, the Rényi divergence is:

$$D_a(p_u|p_v) = \underbrace{\log\left(\frac{\sigma_v}{\sigma_u}\right) + \frac{1}{2(a-1)}\log\left(\frac{\sigma_v^2}{\sigma_*^2}\right)}_{\text{equals } 0} \quad (18)$$

$$+ \frac{a}{2\sigma_*^2} \cdot (\mu_u - \mu_v)^2 \quad (19)$$

$$= \frac{a}{2\sigma_*^2}(\mu_u - \mu_v)^2 = \frac{a}{2C\sigma^2}(\mu_u - \mu_v)^2 \quad (20)$$

The first two terms in Equation 18 are zero as each dataset for the same number of contacts has the same variance parameter $\sigma_*^2 = \sigma_v^2 = \sigma_u^2 = C \cdot \sigma^2$. The variance parameter for a single $\omega_{c,t}$-term is written as σ^2. As per the property of the log-normal distribution, the σ^2 parameters are additive when the random variable is multiplied, and $C \cdot \sigma^2$ is the variance parameter of the product.

The Rényi divergence depends on the difference in the mean parameters, $(\mu_u - \mu_v)^2$, and we want to bound this for the worst-case adjacent data sets. A log-normal distribution has mean $m = \exp\{\mu + \frac{\sigma^2}{2}\}$. As such, we define the μ parameter of the log-normal distribution as:

$$\mu_{c,t} = \log\omega_{c,t} - \frac{\sigma^2}{2}. \quad (21)$$

Each message is sampled from a log-normal distribution like $s_{c,t} \sim \text{log-normal}(\mu_{c,t}, \sigma^2)$, with $\mu_{i,t} = \log\omega_{i,t} - \frac{\sigma^2}{2}$. The product is $\omega_{*,t} = \prod_{i=1}^{C}\omega_{i,t}$, which has a distribution $s_{*,t} \sim \text{log-normal}(\mu = \sum_{i=1}^{C}\mu_{c,t}, C \cdot \sigma^2)$.

The μ parameter of a single message is defined as Equation 21. Therefore, the μ parameter of the product follows from:

$$\mu_{*,t} = \sum_{i=1}^{C}\mu_{c,t} = \left(\sum_{c \in N(u,t)}\log[\omega_{c,t}]\right) - \frac{C\sigma^2}{2} \quad (22)$$

Using FN for decentralized contact tracing, we can clip the COVIDSCORE $\phi_{c,t}$ in the range $[\gamma_l, \gamma_u]$. If so,

the messages $\omega_{c,t}$ will be bounded as well between $[1 - p_1\gamma_u, 1 - p_1\gamma_l]$.

The sum in Equation 22 is monotone in each message $\omega_{c,t}$. We aim to bound the worst-case value for $\mu_u - \mu_v$ and thus we have two cases. Firstly, each message is at the lower clip value γ_l, against the adjacent data set where one message is at the upper clip value γ_u. Secondly, each message is at the upper clip value γ_u, against the adjacent data set where one message is at the lower clip value γ_l.

In the first case, we have

$$(\mu_u - \mu_v) = \underbrace{C(\log(1 - \gamma_l p_1) - \frac{\sigma^2}{2})}_{\text{all } \gamma_l}$$
$$- \left(\underbrace{(C-1)(\log(1 - \gamma_l p_1) - \frac{\sigma^2}{2}) + \log(1 - \gamma_u p_1) - \frac{\sigma^2}{2}}_{\text{one switch to } \gamma_u}\right)$$
(23)

$$= \log(1 - \gamma_l p_1) - \log(1 - \gamma_u p_1).$$

In the second case, we have

$$(\mu_u - \mu_v) = \underbrace{C(\log(1 - \gamma_u p_1) - \frac{\sigma^2}{2})}_{\text{all } \gamma_u} -$$
$$\left(\underbrace{(C-1)(\log(1 - \gamma_u p_1) - \frac{\sigma^2}{2}) + \log(1 - \gamma_l p_1) - \frac{\sigma^2}{2}}_{\text{one switch to } \gamma_l}\right)$$
(24)

$$= \log(1 - \gamma_u p_1) - \log(1 - \gamma_l p_1).$$

After squaring, we have in both cases

$$(\mu_u - \mu_v)^2 = (\log(1 - \gamma_u p_1) - \log(1 - \gamma_l p_1))^2. \quad (25)$$

Equation 25 is established for the worst-case adjacent data sets. Therefore, we have that for any adjacent data sets

$$D_a(p_u|p_v) \leq \frac{a}{2C\sigma^2}(\log(1 - \gamma_u p_1) - \log(1 - \gamma_l p_1))^2 = \rho$$

for any two adjacent data sets. (26)

In Equation 26, we write the upper bound as ρ. Therefore, we have (a, ρ)-RDP whenever σ^2 has at least the following value

$$\sigma^2 \geq \frac{a}{2C\rho}(\log(1 - \gamma_u p_1) - \log(1 - \gamma_l p_1))^2 \quad (27)$$

Difference from Gaussian mechanism in the log-domain: Other than RDP with a log-normal distribution, one could also use the Gaussian mechanism from (Dwork and Roth 2014) in the log-domain. Such a method would apply the Gaussian mechanism to the logarithm of each score and consider the exponent function as post-processing. We experimented with this method and found this method to achieve strictly worse PIR results. One reason could be the form of Equation 21. Due to Jensen's inequality, we know that the

mean of a concave function (e.g. the logarithm) is smaller than or equal to the concave function applied to the mean. The factor $-\frac{\sigma^2}{2}$ could be considered to counteract this 'bias,' and we hypothesize that this is the reason the Gaussian mechanism in the log domain achieves worse results than RDP with the log-normal distribution.

Clipping with public knowledge: The log-normal distribution assumes a value on $\mathbb{R}^+$. In the product of Equation 7, C COVIDSCORE are clipped to $[\gamma_l, \gamma_u]$, and each message is calculated according to Equation 56. As such, it is public knowledge that the product of messages should lie in $[(1 - \gamma_u p_1)^C, (1 - \gamma_l p_1)^C]$. Therefore, after the sampling in line 5 of Algorithm 1, we clip the messages to this known interval.

C.4 Optimize for parameters of RDP

Section 3.3 presents an algorithm that achieves differential privacy for contact tracing via Rényi differential privacy. Although Rényi differential privacy has better composition properties (Mironov 2017), the analysis introduces a new hyperparameter a. This hyperparameter could be optimized via an experimental parameter sweep (Abadi et al. 2016). Fortunately, for our particular problem, we find a closed-form solution for this hyperparameter. This allows one to convert (a, ρ)-RDP to (ε, δ)-DP. In this section, we write out the system of equations and show four closed-form solutions to the resulting polynomial.

We reduce the equations in Section 3.3 to an optimization problem and use the KKT conditions to find a stationary point. Equation 27 shows that the noise scale grows linearly with $\frac{a}{\rho}$. Arguably, any lower noise scale implies less noise on the COVIDSCORE and less noise in whichever users get tested and, as a result of a positive test, get quarantined. As such, the optimization objective is to minimize the fraction $\frac{a}{\rho}$. The constraint for this problem is that ε is fixed non-linearly for a given δ, ρ, and a. From (Mironov 2017) we know that $\varepsilon = \rho + \frac{\log \frac{1}{\delta}}{a-1}$.

We can write the optimization problem as follows:

Optimization-problem 1.

$$\min_{a, \rho} \frac{a}{\rho}$$

Such that:

$$\rho + \frac{\log \frac{1}{\delta}}{a-1} - \varepsilon = 0$$

The search space is constrained to $a > 1$ and $\rho > 0$. For the Lagrangian, $\mathcal{L} = \frac{a}{\rho} + \nu(\rho + \frac{\log \frac{1}{\delta}}{a-1} - \varepsilon)$, we write the

optimality conditions (Boyd and Vandenberghe 2014):

$$\nabla_a \mathcal{L} = \frac{1}{\rho} + \nu\left(\frac{-\log \frac{1}{\delta}}{(a-1)^2}\right) = 0 \qquad \text{stationarity in } a \quad (28)$$

$$\nabla_\rho \mathcal{L} = \frac{-a}{\rho^2} + \nu = 0 \qquad \text{stationarity in } \rho \quad (29)$$

$$\rho + \frac{\log \frac{1}{\delta}}{a-1} = \varepsilon \qquad \text{Primal feasibility} \quad (30)$$

For clarity, we rewrite the constant $d = \log \frac{1}{\delta}$. We know $d > 0$, which will be used later. The system of equations results:

$$\frac{1}{\rho} = \nu\left(\frac{d}{(a-1)^2}\right) \tag{31}$$

$$\frac{a}{\rho^2} = \nu \tag{32}$$

$$\rho = \varepsilon - \frac{d}{a-1}. \tag{33}$$

Substituting Equation 31 in Equation 32 yields:

$$a\frac{\nu^2 d^2}{(a-1)^4} = \nu. \tag{34}$$

This means that either $\nu = 0$, which is infeasible (Boyd and Vandenberghe 2014), or

$$\nu = \frac{(a-1)^4}{ad^2}. \tag{35}$$

Substitute Equation 35 in Equation 32 with Equation 33:

$$a = \nu\rho^2 = \frac{(a-1)^4}{ad^2}\left(\varepsilon - \frac{d}{a-1}\right)^2 \tag{36}$$

$$a = \frac{(a-1)^2}{ad^2}(\varepsilon(a-1) - d)^2 \tag{37}$$

$$a^2 d^2 = (a-1)^2(\varepsilon(a-1) - d)^2 \tag{38}$$

In Equation 37 we use that the order a is larger than 1.

Equation 38 is a fourth-order polynomial in a. For clarity, we rename $x = a - 1$:

$$(x^2 + 2x + 1)d^2 = x^2(\varepsilon x - d)^2 \tag{39}$$

$$d^2 x^2 + 2d^2 x + d^2 = x^2(\varepsilon^2 x^2 - 2\varepsilon dx + d^2) \tag{40}$$

$$d^2 x^2 + 2d^2 x + d^2 = \varepsilon^2 x^4 - 2\varepsilon dx^3 + d^2 x^2 \tag{41}$$

$$\varepsilon^2 x^4 - 2d\varepsilon x^3 - 2d^2 x = d^2 \tag{42}$$

This polynomial could be rewritten.

$$(d + \varepsilon x^2)(\varepsilon x^2 - 2dx - d) = 0 \tag{43}$$

This polynomial has zeroes:

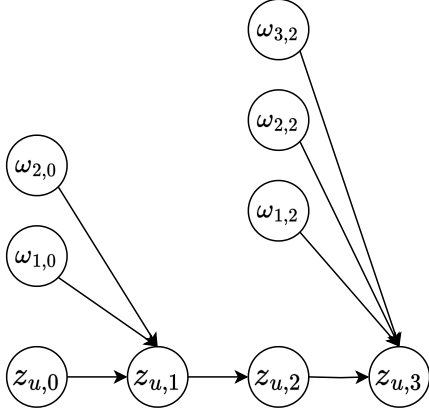Figure 7: A contact graph with five contacts on two days, as discussed in Appendix Section C.5.

$$x_1 = i\frac{\sqrt{d}}{\sqrt{\varepsilon}} \tag{44}$$

$$x_2 = -i\frac{\sqrt{d}}{\sqrt{\varepsilon}} \tag{45}$$

$$x_3 = \frac{d + \sqrt{d(d+\varepsilon)}}{\varepsilon} \tag{46}$$

$$x_4 = \frac{d - \sqrt{d(d+\varepsilon)}}{\varepsilon} \tag{47}$$

Variable i indicates the imaginary number. Solutions x_1 and x_2 are imaginary and have the real part being 0, which is primal infeasible. Solution x_4 is always negative for $\varepsilon > 0$, $\delta > 0$, which is primal infeasible. Therefore, we use solution x_3 and arrive at:

$$a = 1 + \frac{d + \sqrt{d(d+\varepsilon)}}{\varepsilon} \tag{48}$$

The value for ρ follows from Equation 30.

C.5 FN as a function of products

This subsection writes the FN update equations to show that FN can be considered a function with inputs of only the product of messages. First, we work out the updates for a particular small graph. Second, we generalize this for arbitrary contact graphs. (Romijnders et al. 2023) derived the update equation for an SEIR model and a general number of timesteps and contacts. For illustration, we write the update equations for a SIR model on four timesteps, with two contacts at day 0, $\omega_{1,0}$ and $\omega_{2,0}$, and with three contacts on day 2, $\omega_{1,2}$ and $\omega_{2,2}$ and $\omega_{3,2}$. The contact graph is illustrated in Figure 7. As COVIDSCORE for this example, we consider the belief of being in state I at day $t = 3$ which is indicated by $b(z_3 = I)$. The user u subscripts are omitted for clarity, i.e., z_t refers in the following to $z_{u,t}$.

Equation 49 is the general FN update from (Rosen-Zvi, Jordan, and Yuille 2005) and Equation 50 writes out the conditional distribution using the product rule for probabilities. Equation 51 simplifies the summation over random variables. As the Markov chain only considers transitions $S \to I \to R$, we can reduce computation from the sum that grows exponentially to a sum over traces that grows quadratically or cubically for an SEIR model. This observation has been made in (Herbrich, Rastogi, and Vollgraf 2020). Equation 52 writes the summation in full.

$$b(z_3 = I)$$
$$= \mathbb{E}_{B_{N(u)}}[p(z_3 = I|z_{N(u)})] \tag{49}$$
$$= \mathbb{E}_{B_{N(u)}}[\sum_{z_2}\sum_{z_1}\sum_{z_0} p(z_3 = I|z_2, z_{N(u)})$$
$$\cdot p(z_2|z_1, z_{N(u)})p(z_1|z_0, z_{N(u)})p(z_0)] \tag{50}$$
$$= \sum_{(z_0,z_1,z_2)\in\{SSS,SSI,SII,III\}} \mathbb{E}_{B_{N(u)}}[p(z_3 = I|z_2, z_{N(u)})$$
$$\cdot p(z_2|z_1, z_{N(u)})p(z_1|z_0)p(z_0)] \tag{51}$$
$$= \mathbb{E}_{B_{N(u)}}[p(z_3 = I|z_2 = S, z_{N(u)})$$
$$\cdot p(z_2 = S|z_1 = S)p(z_1 = S|z_0 = S, z_{N(u)})p(z_0 = S)]$$
$$+ \mathbb{E}_{B_{N(u)}}[p(z_3 = I|z_2 = I, z_{N(u)})$$
$$\cdot p(z_2 = I|z_1 = S)p(z_1 = S|z_0 = S, z_{N(u)})p(z_0 = S)]$$
$$+ \mathbb{E}_{B_{N(u)}}[p(z_3 = I|z_2 = I, z_{N(u)})$$
$$\cdot p(z_2 = I|z_1 = I)p(z_1 = I|z_0 = S, z_{N(u)})p(z_0 = S)]$$
$$+ \mathbb{E}_{B_{N(u)}}[p(z_3 = I|z_2 = I, z_{N(u)})$$
$$\cdot p(z_2 = I|z_1 = I)p(z_1 = I|z_0 = I, z_{N(u)})p(z_0 = I)] \tag{52}$$
$$= \mathbb{E}_{B_{N(u)}}[p(z_3 = I|z_2 = S, z_{N(u)})$$
$$\cdot (1 - p_0)p(z_1 = S|z_0 = S, z_{N(u)})(1 - p_0)]$$
$$+ \mathbb{E}_{B_{N(u)}}[gp_0 p(z_1 = S|z_0 = S, z_{N(u)})(1 - p_0)]$$
$$+ \mathbb{E}_{B_{N(u)}}[ggp(z_1 = I|z_0 = S, z_{N(u)})(1 - p_0)]$$
$$+ \mathbb{E}_{B_{N(u)}}[gggp_0] \tag{53}$$
$$= (1 - (1 - p_0)(1 - p_1\phi_{1,2})(1 - p_1\phi_{2,2})(1 - p_1\phi_{3,2}))$$
$$\cdot (1 - p_0)(1 - p_0)(1 - p_1\phi_{1,0})(1 - p_1\phi_{2,0})(1 - p_0)$$
$$+ gp_0(1 - p_0)(1 - p_1\phi_{1,0})(1 - p_1\phi_{2,0})(1 - p_0)$$
$$+ gg(1 - (1 - p_0)(1 - p_1\phi_{1,0})(1 - p_1\phi_{2,0}))(1 - p_0)$$
$$+ gggp_0 \tag{54}$$
$$= (1 - (1 - p_0)\omega_{1,2}\omega_{2,2}\omega_{3,2}))$$
$$\cdot (1 - p_0)(1 - p_0)\omega_{1,0}\omega_{2,0}(1 - p_0)$$
$$+ gp_0(1 - p_0)\omega_{1,0}\omega_{2,0}(1 - p_0)$$
$$+ gg(1 - (1 - p_0)\omega_{1,0}\omega_{2,0})(1 - p_0)$$
$$+ gggp_0. \tag{55}$$

In Equation 53 we replace the known scalars for prior belief p_0, and probability $p(z_{t+1} = I|z_t = I, z_{N(u,t)}) = g$. In Equation 54, we replace the conditional distribution under the local belief according to Equation 8 of (Romijnders et al. 2023).

Equation 55 makes explicit that the output from FN only depends on a product of scores. From Equation 54 we write the messages in shorthand. If the user receives a message $\phi_{c,t}$ from contact c on time t, then rewrite the message to:

$$\omega_{c,t} = 1 - p_1\phi_{c,t}. \tag{56}$$

With this transform, FN depends on products of the $\omega_{c,t}$.

While this example was specific for an SIR in the graph in Figure 7, previous work showed the FN update for an SEIR model in general. In full generality, the messages, $\omega_{c,t}$, occur in each conditional probability distribution and can in each case be rewritten to the product $\omega_{*,t}$, similar to Equation 55. Such product term would only ever appear in a conditional distribution $p(z_{u,t+1}|z_{u,t} = S, z_{N(u)})$, which are the transitions $S \to S$ or $S \to E$. Observations appear as a distinct factor and do not influence the above product. Therefore, FN on a contact graph with arbitrary contacts and arbitrary observations can be rewritten to depend on the product $\omega_{*,t}$, and Algorithm 1 describes its differentially private implementation.

C.6 Experimental details

In addition to the experimental details in Section 4.2, we expand on a few more points pertaining to our experiments.

The most important difference between the Covasim simulator and the experiments using OpenABM is that in Covasim, each day 10% of the population may be tested and in OpenABM, each day 2% of the population may be tested. We found that in Covasim, when testing too many agents daily, all methods would achieve an equally low peak infection rate, i.e. no pandemic. Such setting would not allow for a comparison of the methods. In both simulators, isolation is for ten days, and the first isolation only starts on the fourth day; the first three days are used to start the simulation.

The simulator settings for OpenABM follow previous literature (Baker et al. 2021; Romijnders et al. 2023). Each simulation starts with 25 users in infectious state; scale experiments with 500.000 and more users start with 50 users in infectious state. The simulator settings for Covasim follow the recommended tutorials. The simulator is set with dates February 1st 2020 to May 1st 2020, which accounts for 91 days (whereas OpenABM runs for 100 days). Following recommendations by the authors, population type 'hybrid' is used. Each simulation starts with 25 users in infectious state; scale experiments with 500.000 and more users start with 100 users in infectious state.

The OpenABM simulator uses ten health states, Uninfected (0), Pre-symptomatic (1), Mild pre-symptomatic (2), Asymptomatic (3), Symptomatic (4), Mild symptomatic (5), Hospitalised (6), Critical (7), Recovered (8), Death (9). A user can test positive from state one to seven. We report infection rate as occurrence of state three to seven. The Covasim simulator models with five states, Susceptible, Exposed, Infectious (which is a subset of Exposed), Recovered, and Death. A user can test positive in the Exposed state. We report infection rate as the occurrence of the Infectious state.

The model, as defined in Section 3 has four model parameters, and we have set them based on insights from the literature. Model parameters are set at $p_0 = \frac{1}{1000}$, $p_1 = 0.05$, $g = 0.99$, $h = 0.10$. This can be interpreted as that the typical infection lasts eleven days, where the user is not considered to spread the virus on the first day of infection. Only for the visualization in Figure 2, the parameter p_1 is exaggerated to 0.25 for illustration purposes.

Hyperparameter setting	PIR (‰)
number of days = 7	133.5 [131.8,137.3]
number of days = 10	17.9 [12.2,21.1]
number of days = 14	1.0 [0.8,1.3]
number of days = 21	1.0 [0.9,1.5]
number of days = 30	1.1 [0.9,1.4]
$p_1 = 0.001$	3.9 [1.7,7.6]
$p_1 = 0.010$	1.1 [0.9,1.7]
$p_1 = 0.050$	1.0 [0.8,1.3]
$p_1 = 0.100$	1.0 [0.9,1.2]

Table 4: Two hyperparameter sweeps for important parameters of our algorithm. Numbers indicate the median peak infection rate (PIR) as one infection per thousand (‰). The subscripts indicate 20-80 quantiles of ten random restarts.

Parameters specific to each method. In Gibbs sampling, we take ten samples, each separated by ten skip steps, and each chain is burned in for 100 steps. The likelihoods are clipped at a norm of 10. In per-message differential privacy, the messages are clipped between $[0.01, 0.99]$, and the Gaussian mechanism is run in the logit domain. In the *Traditional contact tracing* method, a sampled COVIDSCORE could be negative due to the tails of the Gaussian distribution. However, we clip negative values to zero as it is public knowledge that a COVIDSCORE must be non-negative. Experiments with FN use the aforementioned model parameters, and assumes that the false positive and false negative rates are known. In future, one could experiment with different noise rates between noisier tests, e.g. self testing, and less noisy tests, e.g. testing in lab environments.

All experiments run inference in a time window of fourteen days. This setting is in accordance with previous works (Baker et al. 2021; Romijnders et al. 2023). The prior for each window is similar to (Herbrich, Rastogi, and Vollgraf 2020): $p(z_{u,0} = S) = 1 - p_0$, $p(z_{u,0} = E) = p_0$, and $p(z_{u,0} = I) = p(z_{u,0} = R) = 0$.

Decentralization: All methods that we study in this paper are decentralized. No (central) entity knows about the COVIDSCORE of multiple individuals. Both DPFN, Gibbs sampling, and Traditional contact tracing operate by sending messages between users. One step in our simulation that needs further clarification is the selection of which people to signal for a COVID19 test. Currently, the simulation selects the users with the highest COVIDSCORE for a test. This computation can happen encrypted (Beaver, Micali, and Rogaway 1990; Ben-Efraim, Lindell, and Omri 2016), or differentially private (Dwork and Roth 2014). A different scenario that one could experiment with is that the tests are based on a threshold of the COVIDSCORE but making more tests reduces PIR, so this makes the results hard to compare. Therefore, to compare the contact tracing algorithms, we use the ranking approach for the testing policy.

Hyperparameters: During this research, we have made hyperparameter sweeps for two important parameters, and we report the results of that sweep here.

The duration of the inference window changes how many

days of past contacts influence a particular COVIDSCORE. Increasing this window makes the statistical inference more accurate but would create a possibly worse composition for privacy and increased computing costs. The computation scales cubically with the size of the window (Romijnders et al. 2023). Table 4 displays the result of this sweep, which is run with $\varepsilon = 1$. We have chosen the value of fourteen, which strikes a balance between the privacy and compute cost on the one hand and the PIR on the other hand.

The second hyperparameter sweep concerns the value of p_1 in our experiments. This parameter is paramount, as it appears in both the statistical model in Equation 3 and the privacy bound in Equation 9. The value is estimated by population studies at 0.01 (Hinch et al. 2021), and misspecifying the value has the problem of model misspecification during inference of the COVIDSCORE. On one side, a low value of p_1 requires less noise to satisfy DP per Equation 9. On the other side, a slightly higher value achieves a lower, better peak infection rate. We hypothesize this effect is due to a higher value of p_1 emphasizing contact tracing and introducing a stronger connection between a user and its contacts' scores. Table 4 displays the results of this sweep at $\varepsilon = 1$ and window size 14. We pick a value $p_1 = 0.05$, which has the lowest median PIR, also taking into consideration the quantiles, which are slightly lower for $p_1 = 0.05$.

Technical details: For the OpenABM simulator, we use code at github.com/BDI-pathogens/OpenABM-Covid19. For the Covasim simulator, we use code at github.com/InstituteforDiseaseModeling-/covasim. The parameter files for OpenABM can be found at github.com/BDI-pathogens/OpenABM-Covid19/tree/master/tests/data/*.csv.

Our experiments run on a 32-core CPU node with less than 60GB of memory. Only the experiments with over a half million users require a compute node with 120GB memory or more. Experiments with 100.000 users generally finish in under three hours each. We are also releasing an implementation in C++, which finishes such experiment in about 30 minutes for 16 parallel simulations. The experiments with belief propagation generally take ten times longer, which amounts to about a whole day for 100.000 users. Due to compound computing requirements in the inference algorithm, the experiments with one million users can take up to forty hours. The python implementation uses Numba runtime jit-compilation (Lam, Pitrou, and Seibert 2015).

In Figure 5, the experiment for the loss to follow-up directly modulates this parameter in the Covasim simulator. This parameter is defined as `loss_prob` in `covasim/interventions.py`.

Python implementation:
`github.com/RobRomijnders/dpfn_aaai`.
C++ implementation:
`github.com/RobRomijnders/dpfn_cpp`.