Poster Abstract: Poirot: Private Contact Summary Aggregation

Yanping Zhang[1]*, Chenghong Wang[1]*, David Pujol[1]*, Johes Bater[1]
Matthew Lentz[1,2], Ashwin Machanavajjhala[1], Kartik Nayak[1], Lavanya Vasudevan[3,4], Jun Yang[1]
Department of Computer Science, Duke University[1], VMware Research[2],
Department of Family Medicine and Community Health, Duke University[3], Duke Global Health Institute[4]

ABSTRACT

Physical distancing between individuals is key to preventing the spread of a disease such as COVID-19. On the one hand, having access to information about physical interactions is critical for decision makers; on the other, this information is sensitive and can be used to track individuals. In this work, we design Poirot, a system to collect aggregate statistics about physical interactions in a privacy-preserving manner. We show a preliminary evaluation of our system that demonstrates the scalability of our approach even while maintaining strong privacy guarantees.

CCS CONCEPTS

• **Security and privacy → Cryptography**; **Systems security**.

KEYWORDS

COVID19, private data aggregation, Bluetooth tracing, multi-party computation, differential privacy

ACM Reference Format:
Yanping Zhang[1]*, Chenghong Wang[1]*, David Pujol[1]*, Johes Bater[1], Matthew Lentz[1,2], Ashwin Machanavajjhala[1], Kartik Nayak[1], Lavanya Vasudevan[3,4], Jun Yang[1]. 2020. Poster Abstract: Poirot: Private Contact Summary Aggregation. In *The 18th ACM Conference on Embedded Networked Sensor Systems (SenSys '20), November 16–19, 2020, Virtual Event, Japan.*

1 INTRODUCTION

COVID-19 is a contagious disease that is known to spread rapidly through person-to-person contact. To curb its spread, many state governments announced lockdowns, and offices and schools were temporarily shut down. While such measures help stabilize the infection rate, these measures cannot be implemented indefinitely.

For organizations to reopen, ideally, they would need relevant data about adherence to physical distancing [1]. This will inform their decisions on measures and interventions to reduce the risk of contracting the disease. For instance, such information can help assess the effectiveness of current policies such as opening at a reduced scale and also act as a feedback loop to inform new policies. They can also help identify *hotspots* and to better allocate sanitation resources. Individuals can also benefit from such information since it helps them understand how they are adhering to social distancing policies. On the other hand, information required to track physical distancing is very sensitive since it can reveal users' location trajectories. Thus, there is a tension between obtaining such physical distancing information and ensuring individuals' privacy.

The goal of our work, Poirot, is to collect necessary information about individuals' physical interactions in a privacy-preserving manner to provide actionable information to decision-makers and the individuals themselves. Our work measures physical interactions through the number of contact events between individuals; such measurements are directly related to the disease's spread and, thus, a crucial enabler to decision-making. To ensure privacy, we collect and release aggregate statistics of contact events that cannot be linked back to any individual. Thus, Poirot can provide utility to organizations and individuals with little to no loss of privacy.

2 DESIGN

As shown in Figure 1, the key actors involved in Poirot are users (and their smartphones), administrators, authentication servers, and some compute servers. Poirot protects sensitive data sent by individual users from the administrators who receive released statistics, any subset of the compute servers, and authentication servers. We assume that both users and servers are semi-honest, meaning they honestly participate in the protocol but may attempt to derive sensitive information. Users provide accurate information, without attempting to maliciously modify the Poirot app. Authentication servers validate user credentials and provide proof of identity to the compute servers. Finally, compute servers faithfully execute a secure multi-party protocol to evaluate queries. Compute servers may learn the total number of days for which a user has uploaded their data and which users are participating. We also assume that all but one compute server may collude with each other.

Following are the phases of Poirot workflow:

Collection. Users must first obtain *permission* to participate in the system, which happens on a recurring basis (e.g., monthly). This process follows an OAuth flow whereby the user provides an identity to Poirot via an authorization service. The app and Poirot server participate in a blind signature protocol to generate tokens for uploading data; this decouples identity from data, as the server can only validate a token but not identify the user who generated it. Tokens can encode the attributes of the user (e.g., student) based on the key used to sign the tokens (which the user can verify).

A contact event occurs when two user devices remain in close proximity for a period of time, where the distance and duration are specified by a policy (e.g., 6ft and 15min for COVID-19). To *discover contact events*, the app uses Bluetooth to exchange identifiers with nearby participating devices. Unlike existing contact

SenSys '20, November 16–19, 2020, Virtual Event, Japan

ACM ISBN 978-1-4503-7590-0/20/11…$15.00
https://doi.org/10.1145/3384419.3430603

Authors: ∗ denotes equal contribution.

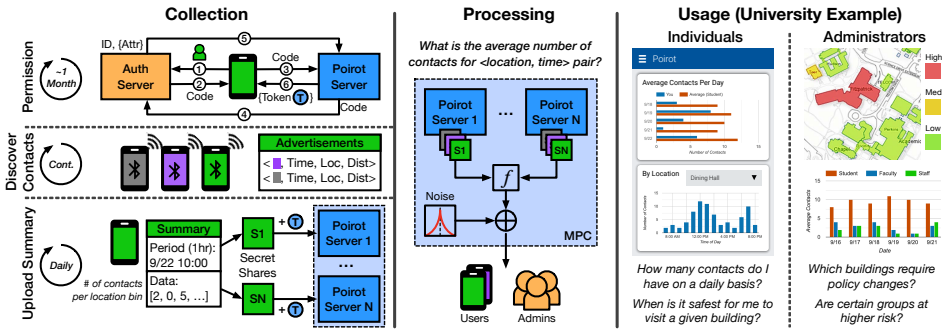

Figure 1: Overview of the Poirot workflow and architecture, broken down into three phases: collection, processing, and usage.

tracing applications, we are interested in detecting contacts regardless of whether either user has tested positive for COVID-19. To maintain long-term unlinkability of a device's identifiers over time while supporting long-running contacts (e.g., hours), identifiers are changed every epoch (nominally 15min) but both the previous and current epoch identifiers are broadcast.

Every day, the app computes and uploads aggregate summaries of all recorded contact events for that day. The aggregate summary contains a vector of counts, where each entry corresponds to a location bin (e.g., county in a state). The app generates secret-shared versions, one per compute server, which ensures that the compute servers do not learn the contents of individual summaries.

Processing. At this point, the compute servers have secret-shared contact summaries for all participating users for the previous day. The servers participate in a multi-party computation (MPC) protocol to compute aggregate statistics over the secret-shared summaries from all users. As part of the MPC protocol, Poirot adds noise to satisfy differential privacy (DP) such that the released statistics are accurate while not revealing sensitive information.

Usage. Poirot disseminates the aggregate statistics computed in the previous phase for *use* by individual users and administrators. In our example, we consider a university setting to highlight the utility in being able to answer common questions for both individuals and administrators. University administrators, for instance, may change policies to reduce crowds in certain locations according to the number of contact events. Users can understand how safe they are being, and may change their behavior to avoid visiting certain locations (e.g., dining hall) at times of high traffic.

3 EVALUATION

We have discussed how Poirot collects, processes, and utilizes aggregate contact event summaries in a privacy-preserving manner. In this evaluation, we focus on a key performance aspect: the scalability of the privacy-preserving computation of aggregate statistics.

Case studies. Our first case study captures a university setting, where members of the community (e.g., students, faculty) use the app as classes resume during the pandemic. We choose Duke University as our example, setting the number of users to 20K and the number of location bins to 256 (one per campus building). In the second case, we consider a state-wide setting, where all residents participate and the state government is interested in per-county statistics. We choose North Carolina (NC), setting the number of users to 10M and location bins 100 (one per county).

	Number of Bins			Execution Time	
Case	Location	Time	Users	App (ms)	Server (s)
Duke	256	1	20K	15.2 ± 4.5	3.9 ± 0.0
Duke	256	24	20K	366.1 ± 8.9	94.3 ± 0.4
NC	100	1	10M	6.0 ± 4.4	776.1 ± 1.7

Table 1: Performance for computing aggregate statistics.

Configuration. For the Poirot servers, we use two Google Cloud Platform servers in the us-central1-a zone running Ubuntu Linux, which are provisioned with 2 vCPUs, 32 GB memory, 10 GB storage, and 1 Gbps network bandwidth. For the app, we are using a Samsung Galaxy A20e smartphone running Android 10.

Results. In Table 1, we show the average and standard deviation for the execution time of MPC without differential privacy, both for the app (1000 trials) and the server (10 trials). The app-side computation is limited, e.g., takes 366.1ms on average for 256 bins and 24 time periods. The server-side computation is also efficient; even for 10M users, the MPC protocol completes in 776.1s. These results demonstrate the scalability for large deployment scenarios.

4 CONCLUSION

Physical distancing is key to managing the spread of contagious diseases such as COVID-19. Using a combination of MPC, DP, and other techniques, Poirot can compute aggregate statistics over contact events between individuals while guaranteeing privacy for end users and scaling to large deployment scenarios. We envision Poirot to complement contact tracing applications [2] by providing information to enable proactive (rather than reactive) actions.

ACKNOWLEDGMENTS

We thank Truls Ostbye, Gayani Tillekeratne, John Bartlett, and Dennis Clements for their helpful feedback at various phases of the project. This work is supported in part by NSF Award 2029853.

REFERENCES

[1] Kimon Drakopoulos. 2020. *The Logic Around Contact Tracing Apps Is All Wrong.* https://www.wired.com/story/opinion-the-logic-around-contact-tracing-apps-is-all-wrong/.

[2] Gabriel Kaptchuk, Daniel G. Goldstein, Eszter Hargittai, Jake M. Hofman, and Elissa M. Redmiles. 2020. *How good is good enough for COVID19 apps?* https://arxiv.org/pdf/2005.04343.pdf.