Can Two Walk Together: Privacy Enhancing Methods and Preventing Tracking of Users

Moni Naor[1]
Department of Computer Science and Applied Mathematics,
Weizmann Institute of Science, Rehovot, 76100, Israel
http://www.wisdom.weizmann.ac.il/~naor
moni.naor@weizmann.ac.il

Neil Vexler
Department of Computer Science and Applied Mathematics,
Weizmann Institute of Science, Rehovot, 76100, Israel
neil.vexler@weizmann.ac.il

Abstract

We present a new concern when collecting data from individuals that arises from the attempt to mitigate privacy leakage in multiple reporting: tracking of users participating in the data collection via the mechanisms added to provide privacy. We present several definitions for untrackable mechanisms, inspired by the differential privacy framework.

Specifically, we define the trackable parameter as the log of the maximum ratio between the probability that a set of reports originated from a single user and the probability that the same set of reports originated from two users (with the same private value). We explore the implications of this new definition. We show how differentially private and untrackable mechanisms can be combined to achieve a bound for the problem of detecting when a certain user changed their private value.

Examining Google's deployed solution for everlasting privacy, we show that RAPPOR (Erlingsson et al. ACM CCS, 2014) is trackable in our framework for the parameters presented in their paper.

We analyze a variant of randomized response for collecting statistics of single bits, Bitwise Everlasting Privacy, that achieves good accuracy and everlasting privacy, while only being reasonably untrackable, specifically grows linearly in the number of reports. For collecting statistics about data from larger domains (for histograms and heavy hitters) we present a mechanism that prevents tracking for a limited number of responses.

We also present the concept of Mechanism Chaining, using the output of one mechanism as the input of another, in the scope of Differential Privacy, and show that the chaining of an ε_1-LDP mechanism with an ε_2-LDP mechanism is $\ln \frac{e^{\varepsilon_1+\varepsilon_2}+1}{e^{\varepsilon_1}+e^{\varepsilon_2}}$-LDP and that this bound is tight.

2012 ACM Subject Classification Security and privacy → Privacy-preserving protocols

Keywords and phrases Differential Privacy, Surveillance

Digital Object Identifier 10.4230/LIPIcs.FORC.2020.4

Related Version A full version of the paper is available at [17], https://arxiv.org/abs/2004.03002.

Funding *Moni Naor*: Israel Science Foundation (no. 950/16)

Acknowledgements We thank Alexandra Korolova for many discussions and suggestions regarding the RAPPOR project as well as Guy Rothblum and Kobbi Nissim for much appreciated comments for their invaluable insights into various parts of this work. Part of this work was done while the second author was visiting the Simons Institute as part of the Data Privacy: Foundations and Applications program.

[1] Incumbent of the Judith Kleeman Professorial Chair.


© Moni Naor and Neil Vexler;
licensed under Creative Commons License CC-BY

1 Introduction

The cure should not be worse than the disease. In this paper we raise the issue that mechanisms for Differentially Private data collection enable the tracking of users. This wouldn't be the first time an innocent solution for an important problem is exploited for the purposes of tracking. Web cookies, designed to let users maintain a session between different web pages, is now the basis of many user tracking implementations. In the Differential Privacy world, we examine how various solutions meant to protect the privacy of users over long periods of time actually enable the tracking of participants.

To better understand this, consider the following scenario: A browser developer might wish to learn what which are the most common homepages, for caching purposes, or perhaps to identify suspiciously popular homepages that might be an evidence for the spreading of a new virus. They develop a mechanism for collecting the URLs of users' homepages. Being very privacy aware, they also make sure that the data sent back to them is Differentially Private. They want to ensure they can collect this data twice a day without allowing someone with access to the reports to figure out the homepage of any individual user.

If fresh randomness is used to generate each differentially private report, then the danger is that information about the users homepage would be revealed eventually to someone who follows the user's reports. We strive to what we call "Everlasting Privacy", the property of maintaining privacy no matter how many collections were made. In our example, the users achieve everlasting privacy by correlating the answers given at each collection time: e.g. a simple way is that each user fixes the randomness they use, and so sends the same report at each collection.

Now consider Alice, a user who reports from her work place during the day and from her home during the evening. At every collection, Alice always reports regarding the same homepage[2], and therefore (since the randomness was fixed) sends identical reports at home and at work. An eavesdropper examining a report from the work IP address and a report from Alice's home IP address would notice that they are the same, while if they examined a report generated by Alice and one generated by Bob (with the same homepage) they will very likely be different. This allows the adversary to find out where Alice lives.

To elaborate, correlation based solutions open the door to the new kind of issue, tracking users. The correlation between reports can be used as an instrument of identifying individuals, in particular it makes the decision problem of whether or not two sets of reports originated from the same user much easier. This concern has been suggested by the RAPPOR project [13] but without a formal definition, or analysis in the framework where their solution was provided.

The problem of tracking users is related to the problem of point change detection, i.e. identifying when a stream of samples switched from one distribution to another. While this problem has been researched in the past under the lens of privacy by Cummings et al. [5, 4], these works focused on private release of point change detection, i.e. how to enable researchers to detect changes in the sampled distribution while not being too reliant on any specific sample. Our goal is different. We wish to *prevent* change point detection as much as we can; as in our case, a change in distribution correlates to a change in private value. Detecting a change in private value jeopardizes the privacy of the user (think of a case where the gender is changed).

[2] The reader may be wondering why bother reporting about the same value if it does not change. For instance it may for purposes of aggregating information about the currently online population.

1.1 Our Contributions

The main conceptual contribution of this work is the definition of reports being untrackable, presented in Section 3. Roughly, the definition states that the distribution on outputs generated by a single user needs to be sufficiently close to that generated by two users. For the discussion on motivation and possible variants see Section 3.4

▶ **Definition 1** (informal). *A mechanism M is (γ, δ)-Untrackable for k reports, if for any k reports*

$$Pr\left[\text{Reports were generated by one user}\right] \leq e^{\gamma} Pr\left[\text{Reports were generated by two users}\right] + \delta.$$

We present a formal definition to Everlasting Privacy. Roughly speaking, a mechanism is (γ, δ)-Everlasting Privacy if executing it any number of times is (γ, δ)-DP. Our main goal is to simultaneously achieve both tracking prevention and everlasting privacy, while maintaining a reasonable accuracy for the global statistics. We explore the implications of this new definition, specifically how it composes and what a fixed state that is reported in a noisy manner can achieve.

We describe how our tracking definitions can be extended to the change point detection framework, namely to bound the probability that a change in the user's private value is ever detected. In that section we also discuss the necessity of correlating answers between data collections to ensure Differential Privacy, and define various general constructions for mechanisms that can achieve this Everlasting Differential Privacy.

As a tool for analyzing such constructions, in Section 4 we prove a theorem about running a Local Differential Privacy mechanism on the output of another such mechanism.

▶ **Theorem 2** (informal). *A mechanism that consists of running an ε_2-LDP mechanism on the result of an ε_1-LDP mechanism results in $\frac{1}{2}\varepsilon_1 \cdot \varepsilon_2$-LDP for small ε_1 and ε_2.*

Theorem 20 and Corollary 21 provide the formal statement and proof.

We then continue to analyze Google RAPPOR's [13] performance under the framework of tracking. We show the pure tracking bound RAPPOR achieves as well as estimate its "average" case performance. We cocnslude that according to our definition of untrackable, RAPPOR achieves poor protection guarantees. This is presented in Section 5.

As a warm up, in Section 6 we present a mechanism that deals with data collection of a single private bit from each participant. One can view it as the extension of randomized response in this setting. Each user generates a bit at random and remembers it. At each collection, the user generates an new bit and sends the XOR of the private bit, the remembered bit and the new bit. The remembered bit is generated by flipping one biased coin, parameterized by ε_1. The new bits are generated from fresh coin flips from another biased coin, parameterized by ε_2. The aggregator collects all the reports and outputs estimated frequencies for both 0 and 1. We prove that for a choice of privacy parameters $\varepsilon_1, \varepsilon_2 < 1$, and for n participating users, the mechanism has the properties:

- **(i)** It is ε_1-Everlasting Differentially Private.
- **(ii)** Accuracy: the frequency estimation of 0 and 1 is no further than $\tilde{O}\left(\frac{1}{\varepsilon_1 \cdot \varepsilon_2 \cdot \sqrt{n}}\right)$ from the actual values.
- **(iii)** It is $\lfloor \frac{k}{2} \rfloor \varepsilon_2$-untrackable for k reports.

In Section 7 we present a mechanism that allows the collection of statistics of users private values when their data is d bits. This mechanism is particularly relevant for the problems of heavy hitters estimations and histograms. The mechanism's state consists of

the results of the inner product of the private value with multiple vectors in a way that is Differentially Private, reporting one such vector and the private result of the inner product at each data collection. The aggregator collects all the reports and produces an estimate for the frequencies of all possible values, such that the sum of frequencies is 1. We prove that for a choice of privacy parameters $\varepsilon < 1$, setting the state to consist of L reports, and for n participating users, the mechanism has the properties:

(i) It is (ε, δ)-Approximate Everlasting Differentially Private.

(ii) The estimation of the frequency of all values is no further than $\tilde{O}\left(\frac{1}{\varepsilon'}\sqrt{\frac{d}{n}}\right)$ from the actual frequency, for $\varepsilon' = \frac{\varepsilon}{2\sqrt{2L\ln\left(\frac{1}{\delta}\right)}}$.

(iii) It is $\left(0, \frac{k^2}{L}\right)$-untrackable.

Concretely, to obtain (ε, δ)-Everlasting Privacy and α accuracy, then for k reports the guarantee on the mechanism is $\left(0, \widetilde{O}\left(\frac{k^2}{\alpha^2\varepsilon^2 n}\right)\right)$-Untrackable.

Coming up with better bounds or showing the inherent limitations is the main open direction we propose (see Section 8).

2 Preliminaries

2.1 Differential Privacy

For background on Differential Privacy see Dwork and Roth [10] or Vadhan [18].

Throughout most of this paper we consider a variant of Differential Privacy, called *Local Differential Privacy*. Local Differential Privacy regards mechanisms where each individual user runs on their own data to create a report, which is then sent to the server and aggregated there to produce a population level result. The setting we consider is one where the aggregator access the users' data only through a randomized mapping, a mechanism, that has the following property:

▶ **Definition 3** ([15]). *Let $\varepsilon, \delta > 0$. A mechanism $M : U \mapsto O$ is (ε, δ)-Local Differentially Private if for every two possible inputs, $u, u' \in U$, and $\forall S \subseteq O$, $Pr[M(u) \in S] \leq e^{\varepsilon} \cdot Pr[M(u') \in S] + \delta$.*

One of the significant properties of Differential Privacy is the way it composes. Composing two mechanisms that are $(\varepsilon_1, \delta_1)$ and $(\varepsilon_2, \delta_2)$-Differentially Private respectively is $(\varepsilon_1 + \varepsilon_2, \delta_1 + \delta_2)$-Differentially Private. A small deterioration in the δ parameter achieves a great improvement in the ε parameter of the composition.

▶ **Theorem 4** (Advanced composition for Differential Privacy [11]). *Let $\delta' > 0$. The k fold composition of (ε, δ)-Differentially Private mechanisms is $(\varepsilon', k\delta + \delta')$-Differentially Private for $\varepsilon' = \sqrt{2k\ln(1/\delta')}\varepsilon + k\varepsilon(e^{\varepsilon} - 1)$.*

Another useful property of Differential Privacy is that running any function on the output of an (ε, δ)-Differentially Private mechanism is (ε, δ)-Differentially Private. That is, Differential Privacy is *closed under post-processing*.

When using the same mechanism to collect reports multiple times, if not done carefully, the privacy guarantee might deteriorate as the number of collections periods grows. We define *Everlasting Differential Privacy* as an upper bound on the privacy parameter of a mechanism, no matter how many times it is executed, as long as the private data had not changed. Definition 10 formalizes this idea.

2.2 Background

The need for everlasting privacy became apparent since the early stages of the Differential Privacy research. As mentioned in Section 2, independent repetitive executions of Differential Privacy mechanisms inevitably deteriorate the privacy guarantee. While Theorem 4 teaches us that the privacy guarantee can grow as low as only the square root of the number of reports, practical implementations might require users to participate in as many as thousands of data collections (e.g. anything requiring daily reports).

This led researchers to suggest data collection mechanisms that allow numerous data collections, while maintaining individuals' privacy. Certain solutions, such as Google's RAPPOR [13] and Microsoft's dBitFLip [6], use the concept of statefulness, maintaining some data between executions. This enables them to correlate outputs between executions, which allows for a manageable upper bound of the privacy leakage that does not rely on the number of collections made. This effectively allows for a privacy guarantee that holds forever, namely Everlasting Privacy.

Heavy Hitter Mechanisms

Two problems that have been very interesting for data collectors are the histogram and heavy hitters problems. In the histogram problem the goal is to accurately estimate the frequencies of all possible values the population might hold. The heavy hitters problem is about identifying the most common values amongst the population. Both histograms and heavy hitters in the local model has been researched before by Bassily et al. [2, 1], who used Hadamard transformations on the users private data that allow users to send succinct reports to the curator while allowing the required statistics to be generated very efficiently. These works do not fit our framework, as they intrinsically allow for trackability. In their solution, each user is associated with a specific piece of some shared randomness. The aggregator must know to which piece of randomness a specific report belongs to, essentially forcing their solution to be highly trackable. The techniques used in their paper are similar to the ones used by Naor et al. [16]. In that work the authors use an inner product mechanism to identify and ban the most common passwords. This enables the increase in the effective time an adversary will need to invest in order to guess a user's password. Their mechanism maintains Differential Privacy to prevent the leakage of each individual's password, but it does not maintain Everlasting Privacy. They also mention a modification to their scheme achieves Everlasting Privacy, by reusing the same random vector for all future inner products, but such a solution is highly trackable. The inner product mechanisms used in [2, 1, 16] were the inspiration of our Noisy Inner Product mechanism presented in 7.

Continual Observation and Pan Privacy

Other models and solutions to long-lasting privacy have been developed as well, such as the Continual Observation model in [7, 3]. The goal is to maintains differential privacy for values that change over time, e.g. a counter that updates over time, or streams of data, like traffic conditions and so on. This solution is in the central, or streaming, model and not in the local model. Another model is that of Pan Privacy, where the goal is to maintain privacy even if the internal representation of the secret state is leaked from time to time (Dwork et al. [8]). In Erlingsson et al. [12] this idea was extended, transforming the mechanism in [7] to the local model, in order to solve the 1-bit histogram problem, and thus achieving privacy over extended periods of time. The transformation means that every user reports genuinely

only once throughout all data collections, thus resulting in accuracy that relies linearly in the number of times their value changed. This suggests that accuracy will drop as collection times increase.

Joseph et al. [14] suggested an approach where at the beginning, a global update occurs, where each individual participates in a private histogram estimation. At each subsequent potential collection time, each user compares their current contribution to the histogram compared to the last time a global update occurs. Depending on how different it is, they are more likely to suggest that another global update occurs. If enough users vote in favor, the curator initiates another round of global update, creating a more accurate histogram. This solution allows for collections to be made from users only when it is likely that the previously computed output is no longer accurate, greatly increasing the privacy guarantee of individuals. On the other hand, their accuracy analysis relies on the existence of a small number of user types, where all users of the same type behave identically.

3 Stateful Mechanisms and Tracking

Consider a mechanism for users to report their values to a center.Such mechanisms may be *stateless*, i.e. ones that receive an input and (probabilistically) produce an output, or *stateful* mechanisms, ones that receive in addition to the input a *state* and produce in addition to an output a state for the next execution. The power of stateful mechanisms is that they enable the correlation of outputs between different executions through the states passed from one execution to the next.

3.1 Definitions of Mechanisms and Report Stream Generators

Stateless mechanisms are randomized mappings for which each execution is independent of the others. Stateless mechanisms receive the user's data and publicly available information, namely auxiliary information, and output a report. The publicly available information can be anything known to all parties, like time of day, value of some publicly accessible counter, etc.

▶ **Definition 5** (Stateless Mechanism). *A stateless mechanism M is a randomized mapping from a user's data and auxiliary information to the domain of reports, $M : U \times A \times \{0,1\}^\star \mapsto R$. In our setting it is used to generate a stream of reports, $r_1, r_2, \ldots$, where each report is generated independently.*

Stateless mechanisms might provide very poor everlasting privacy, as each iteration reveals more information about the user's data.

Therefore, to achieve everlasting privacy one must *correlate* the reports sent by the user(s) (see for instance [9] where this is proved for counting queries). For this we define *Stateful Mechanisms*, where the mechanism maintains a state that is updated with each call to the mechanism.

▶ **Definition 6** (Fully Stateful Mechanism). *A fully stateful mechanism M is a randomized mapping from a user's data, current state and auxiliary information to the domain of reports and to a new state, $M : U \times S \times A \times \{0,1\}^\star \mapsto R \times S$. In our setting it is used to generate a stream of reports $r_1, r_2, \ldots$ and a stream of states $s_0 = \perp, s_1, \ldots$, such that each pair of state and report are generated by the previous state, auxiliary information and the user's data, $r_i, s_i = M(u, s_{i-1}, a_{i-1})$.*

Notice that the execution number and all previous outputs can be encoded into the state. A fully stateful mechanism can achieve everlasting privacy by correlating answers using the data stored in the state. For example, it can execute a DP mechanism on the user's data and remember the result, reporting the same result whenever queried.

One shortcoming of correlating the reports in such a manner is that it might be used as an identifier by an adversary, potentially allowing the adversary to identify that a group of reports all originated from the same user, thus allowing tracking other activities of the user (see Section 3.2).

We define *Permanent State Mechanisms* as mechanisms that maintain the same state once set, i.e. $s_1 = s_2 = s_3 ....$ As we shall see, such mechanisms are very convenient to work with and have good properties wrt composition.

Report stream generators (RSG) are mappings that use mechanisms to generate a stream of reports. The responsibility of the RSG is to get the user's data and iteratively call the mechanism.

▶ **Definition 7** (Stateless Report Stream Generator). *For a domain of user data U, a range of reports R, and a report stream size n, a Stateless Report Stream Generator using a stateless mechanism M is a mapping $G_n^M : U \mapsto R^n$, that acquires the auxiliary information required at each step and calls M to generate the reports $r_1, \ldots, r_n$.*

Similarly, stateful RGSs use fully stateful mechanisms to generate the stream of reports.

▶ **Definition 8** (Stateful Report Stream Generator). *For a domain of user data U, a range of reports R, and a report stream size n, a Stateful Report Stream Generator using a fully stateful mechanism M is a mapping $G_n^M : U \mapsto R^n$, that acquires the auxiliary information required at each step and calls M with the state of the current step to generate report r_i and the next step's state s_i.*

3.2 Everlasting Privacy and Tracking

The problem we focus on is the ability of an adversary to distinguish whether or not a set of reports originated from a single user or by two users (or more, see Section 3.4). For example, If an adversary had two sets of reports belonging to two different IP addresses, the adversary could learn if those IP addresses belong to the same user or not (potentially identifying the user's work place or home address). The definition of untrackable we propose is inspired by definition of Differential Privacy.

▶ **Definition 9.** *For a domain of user data U, a range of reports R and a report stream size k, a report stream generator G_k^M is (γ, δ)-**untrackable** if for all user data $u \in U$, for all subsets of indices $J \subseteq [k], J^{\complement} = [k] \setminus J$ and $\forall T \subseteq R^k$ we have:*

$$Pr\left[G_k^M(u) \in T\right] \leq e^{\gamma} \cdot Pr\left[G_{|J|}^M(u) \in T_J\right] \cdot Pr\left[G_{k-|J|}^M(u) \in T_{J^{\complement}}\right] + \delta$$

and

$$Pr\left[G_{|J|}^M(u) \in T_J\right] \cdot Pr\left[G_{k-|J|}^M(u) \in T_{J^{\complement}}\right] \leq e^{\gamma} \cdot Pr\left[G_k^M(u) \in T\right] + \delta.$$

For report stream generators that are (γ, δ)-untrackable, an adversary has only a small advantage in distinguishing between the following two cases: the reports originated from a single user or two users. A discussion for the idea behind this definition and its benefits can

be found in Section 3.4. If we want this property to hold for any possible output (i.e. always have the ambiguity), then we can demand that the mechanism be $(\gamma, 0)$-untrackable. We call such mechanisms γ-untrackable. We leverage the similarity to DP show composition theorems on untrackable mechanisms.

Everlasting Privacy is meant to limit the leakage of information users suffer, no matter how many executions a mechanism had. For the following definitions let T be a collection of report streams. For a set of indices J let T_J be the collection of partial report stream, where the reports taken are those in indices J.

▶ **Definition 10** (Everlasting Privacy). *For a domain of user data U, a range of reports R, a report stream generator G_k^M is (ε, δ)-**Everlasting Privacy** if for all user data $u, u' \in U$, for all report stream size k and for all sets of output streams $T \subseteq R^k$, $Pr\left[G_k^M(u) \in T\right] \leq e^{\varepsilon} Pr\left[G_k^M(u') \in T\right] + \delta$.*

If a mechanism is $(\varepsilon, 0)$-Everlasting Privacy we say it is ε-Everlasting Privacy.

These definitions are tightly related to the problem of change-point detection. We define undetectability similarly to untrackability, only we do not assume both report sets originated from the same private data:

▶ **Definition 11.** *For a domain of user data U, a range of reports R and a report stream size k, a report stream generator G_k^M is (γ, δ)-**undetectable** if for all pairs of user data $u, u' \in U$, for all subsets of indices $J \subseteq [k], J^{\complement} = [k] \setminus J$ and $\forall T \subseteq R^k$ we have:*

$$Pr\left[G_k^M(u) \in T\right] \leq e^{\gamma} \cdot Pr\left[G_{|J|}^M(u) \in T_J\right] \cdot Pr\left[G_{k-|J|}^M(u') \in T_{J^{\complement}}\right] + \delta$$

and

$$Pr\left[G_{|J|}^M(u) \in T_J\right] \cdot Pr\left[G_{k-|J|}^M(u') \in T_{J^{\complement}}\right] \leq e^{\gamma} \cdot Pr\left[G_k^M(u) \in T\right] + \delta.$$

We can now connect being untrackable and everlasting privacy with being undetectable.

▶ **Theorem 12.** *A mechanism that is (γ, δ)-untrackable and (ε, δ')-everlasting differentially private is also $(\gamma + \varepsilon, \delta_{\max})$-undetectable, for $\delta_{\max} = \max\left\{e^{\varepsilon}\delta + \delta', \delta + e^{\gamma}\delta'\right\}$.*

The proof for this theorem can be found in the full version of the paper.

3.3 Tracking Bounds, Composition Theorems and Generalizations

For the special case of Permanent State Mechanisms, we can show an upper bound on the untrackable parameter. If the mechanism is ε-Differentially Private in its state, i.e. the mechanism protects the privacy of the state, then the untrackable parameter grows linearly in ε:

▶ **Theorem 13.** *A Permanent state mechanism whose reports are generated by an ε-Differentially Private mechanism receiving the state as its input is $\left\lfloor \frac{k}{2} \right\rfloor \varepsilon$-untrackable for k reports.*

The proof for this theorem can be found in the full version of the paper.

An important question is how tracking composes, i.e. how does a user's participation in multiple Report Stream Generators affect his untrackable guarantees. The similarity between the definition of untrackability and differential privacy allows us to apply results

regarding the latter to obtain results on the former. We show an advanced composition for untrackable mechanisms that is analogous to advanced composition for differential privacy [11] and Theorem 4.

▶ **Theorem 14** (Advanced composition for untrackability). *Let m be a positive integer. Let $\{M_i\}_{i\in[m]}$ be m mechanisms that are (γ,δ)-untrackable for k_i reports respectively. The composition of these mechanisms, $\widehat{M}$, is $(\gamma', m\delta+\delta')$-untrackable for*

$$\gamma' = \sqrt{2m\ln(1/\delta')}\cdot\gamma + m\cdot\gamma\left(e^{\gamma}-1\right).$$

The proof for this theorem, as well as the formal definition of composition, can be found in A.1

Another important question is what can be said about the untrackable guarantees in the settings where the reports are split into more than two sets, i.e. when we want to answer the question whether some reports were generated by a single user or any number of users. For this we define untrackable for n users for k reports.

▶ **Definition 15** (Multiple User Untrackable). *For a domain of user data U, a range of reports R and a report stream size k, and n users, a report stream generator G_k^M is γ-multiple user untrackable if for all user data $u\in U$, all partitions $P=\{P_i\}_{i\in[n]}$ of $[k]$ into n parts, and all output stream sets $T\subseteq R^k$:*

$$e^{-\gamma} \le \frac{\prod_{j\in[n]} Pr\left[G^M_{|P_j|}(u)\in T_{P_j}\right]}{Pr\left[G_k^M(u)\in T\right]} \le e^{\gamma}.$$

We show two connections between Definitions 9 and 15: the first is a general bound, essentially saying that the untrackable parameter increases linearly in the number of users.

▶ **Theorem 16.** *A mechanism that is γ-untrackable for k reports, is $(n-1)\,\gamma$-multiple user untrackable for n users for k reports.*

We can significantly improve this bound for *permanent state mechanisms* by leveraging the fact that their untrackable parameter is linear in the number of reports used.

▶ **Theorem 17.** *A permanent state mechanism M, whose reports are generated by an ε-Differentially Private mechanism receiving the state as its input, is $\lceil\log n\rceil\left\lfloor\frac{k}{2}\right\rfloor\varepsilon$-multiple user untrackable for n users for k reports.*

The proofs of these theorems can be found in Section A.2 and A.3

3.4 Discussion

The way we defined untrackable is not the only one possible. The "typical" attack we wish to prevent is against an adversary that sees many sets of reports and tries to identify two that belong to the same user. However, making this the basis of a definition might result in weak guarantees, as it disregards any prior information that an adversary might have. The adversary might know that Alice only lives in one of two houses, and only tries to identify where she lives. Our definition is designed to protect against exactly this kind of attacker, who only tries to distinguish whether a stream of reports was generated by Alice, or partly by Alice and partly by Bob.

Another natural definition is to prevent distinguishing whether a stream of reports was generated by any combination of users vs. any other combination of users. Our definition, though appearing weaker than this one, actually implies it, with some deterioration to the parameter; Theorem 16 suggests that the parameter deteriorates linearly in the number of users, while Theorem 15 suggests that in some cases it can deteriorate logarithmically.

Our definition also implies that it would be hard to decide whether any two reports were both generated by Alice, or one by Alice and one by Bob. This property might seem tempting as a basis of an alternative untrackable definition, but it is too weak on its own. A mechanism that has this property might have very poor protection against adversaries with access to more than two reports.

Finally, Theorem 12 teaches us that our definition, when combined with everlasting privacy, naturally extends to the problem of change point (un)detection. That is, a mechanism that adheres both to the everlasting privacy requirement and our untrackable definition also protects the fact that a user changed their private value.

In conclusion, This definition is strong enough to protect users against reasonable adversaries, i.e. ones who have some prior knowledge about the locations of users. On the other hand, while seeming weaker than other definitions it actually implies them. Additionally, as can be seen in Sections 6 and 7, it is achievable while also allowing for reasonable everlasting privacy guarantees and accuracy.

4 Mechanism Chaining

In this section we generalize the idea presented in Theorem 13 of using a Differential Privacy mechanism on the output of another such mechanism. We first provide a formal definition for this mechanism chaining, and then state and prove two theorems about the Differential Privacy guarantee achieved by doing such chaining. The first weak, but intuitive, the second much more powerful and also optimal.

4.1 Definitions

We now present mechanism chaining in three different settings: In the first setting we simply define the chaining of two mechanisms as taking the output of the first and using it as the input of the second.

▶ **Definition 18** (2 Local Mechanism Chaining). *Given two mechanisms $A : U \to V$ and $B : V \to O$, the chaining of these two mechanisms $\mathcal{M}_{B\circ A} : U \to O$ is defined as $\mathcal{M}_{B\circ A}(u) = B(A(u))$.*

The second setting we examine is the chaining of k mechanism, and the third and final setting is the chaining of k families of mechanisms that are not necessarily local. They are not relevant for the rest of this paper, but for completeness we present them in the full version of the paper.

4.2 Differential Privacy Guarantees for Two Mechanism Chaining

We now present a tight bound on the Differential Privacy guarantee of the chaining of two mechanisms. We begin by presenting the "Basic Chaining Upper Bound", which is not tight, but is perhaps more intuitive. We then present a better upper bound called the "Advanced Chaining Upper Bound". Basic Chaining simply says that the resulting Differential Privacy is no worse than the Differential Privacy of either mechanisms.

▶ **Theorem 19** (Basic Chaining). *Given two mechanisms $A : U \to V$ and $B : V \to O$ that are ε_1-LDP and ε_2-LDP respectively, $\mathcal{M}_{B\circ A} : U \to O$ is $\min\{\varepsilon_1, \varepsilon_2\}$-LDP.*

The advanced chaining bound is always better:

▶ **Theorem 20** (Advanced Chaining). *Given two mechanisms $A : U \to V$ and $B : V \to O$ that are ε_1-LDP and ε_2-LDP respectively, $\mathcal{M}_{B \circ A} : U \to O$ is $\ln \frac{e^{\varepsilon_1+\varepsilon_2}+1}{e^{\varepsilon_1}+e^{\varepsilon_2}}$-LDP.*

The proof of these theorem can be found in the full version of the paper. The privacy parameter can be upper bounded by a more simple bound that is meaningful for small ε_1 and ε_2:

▶ **Corollary 21.** *Given two mechanisms $A : U \to V$ and $B : V \to O$ that are ε_1-LDP and ε_2-LDP respectively, $\mathcal{M}_{B \circ A} : U \to O$ is $\frac{1}{2}\varepsilon_1 \cdot \varepsilon_2$-LDP.*

When ε_1 or ε_2 are greater than 2 this upper bound is worse than the bound in Theorem 19, let alone the optimal one in Theorem 20, but otherwise this bound has little error compared to the optimal bound and is easier to work with.

5 (Un)Trackability in RAPPOR

Equipped with a new framework to analyze tracking, we first consider one of the most significant deployments of a differentially private mechanism, used in all Chrome copies, and analyze its trackability. Introduced in [13], RAPPOR is a DP mechanism designed to allow repeated collection of telemetry data from users in Chrome. This mechanism was the starting point of this work, since some of the goals stated in the original paper indicate the desirability of being untrackable.

Roughly speaking, RAPPOR reports a value (e.g. the homepage of a user) from a large set. It does so with the help of a Bloom filter that initially encodes a set that contains a single element, the desired value. A Bloom filter's output is an all 0 array that is set to 1 at locations corresponding to hashes of the value. The mechanism proceeds to randomly flip bits in the Bloom filter, generating what we call the Permanent Randomization. At each point in time when data is to be collected, the mechanism generates a report by taking a copy of the Permanent Randomization and, again, randomly flipping bits and reporting the resulting array. The details of the mechanism can be found in the original paper, but for completeness we also present them in the full version of the paper.

In the paper introducing RAPPOR, the authors mention that preventing tracking of users is an issue with their construction: *"RAPPOR responses can even affect client anonymity, when they are collected on immutable client values that are the same across all clients: if the responses contain too many bits (e.g. the Bloom filters are too large), this can facilitate tracking clients, since the bits of the Permanent randomized responses are correlated"*. On the other hand, when talking about the reason behind the second phase of the mechanism execution, generating a report from the permanent randomization, they mention that *"Instead of directly reporting B' [The Permanent Randomization] on every request, the client reports a randomized version of B'. This modification significantly increases the difficulty of tracking a client based on B', which could otherwise be viewed as a unique identifier in longitudinal reporting scenarios"*. We wish to show that in our framework, using the same parameters they used in the RAPPOR data collections, RAPPOR is more aligned with the first statement than with the second. We analyzed RAPPOR's untrackable parameter in the worst case setting, which can be found in the full version of the paper. We present an analysis of the "average case" behavior of RAPPOR.

Estimated Percentile of the Trackability Random Varaible

We estimate the statistics of the trackabiltiy random variable for RAPPOR. In essence, the trackability random variable is the distribution of trackability leaks that happen when participating in the mechanism. The pure version of the untrackable bound in Definition 9 is an upper bound on the possible values of the trackability random variable.

Formally, denote the RAPPOR mechanism by R. For k reports we define a vector of partitions $\vec{J} = \{J_i\}_{i\in[k]}$, where $J_i = [i]$. We also define two report vectors $\vec{T} = \{T_i\}_{i\in[k]}$ and $\vec{T'} = \{T'_i\}_{i\in[k]}$, where T_i is drawn from the product distribution $\left(G_i^R(u), G_{n-i}^R(u)\right)$ and all of the T'_i are drawn from $G_n^R(u)$. The trackability random variable for k reports is the value: $\tau \coloneqq \max\left\{\max_{i\in\left[\lfloor\frac{k}{2}\rfloor\right]} C_{T_i,J_i}, \max_{i\in\left[\lfloor\frac{k}{2}\rfloor\right]} C_{T'_i,J_i}\right\}$, where $C_{T,J}$ is:

$$C_{T,J} \coloneqq \left|\ln \frac{Pr\left[G_n^R(u) = T\right]}{Pr\left[G_{|J|}^R(u) = T_J\right] \cdot Pr\left[G_{n-|J|}^R(u) = T_{J^0}\right]}\right|.$$

The random variable τ is the maximum measured tracking for the $\left\lfloor\frac{k}{2}\right\rfloor$ cases where the reports are generated by two users and the $\left\lfloor\frac{k}{2}\right\rfloor$ cases where the reports are generated by one user. In our setting a mechanism should protect against both types of cases.

The measures of interest are percentiles of the trackability random variable distribution. We estimate the median and the 90^{th} percentile of the trackability random variable. The full version of the paper presents the details of the estimation process.

Figure 1, in Appendix B, shows the estimated median and 90^{th} percentile of the Trackability random variable for between 2 and 15 reports, and their respective 95% confidence interval. Our estimation shows that RAPPOR's trackability random variable's median is better than the worst case trackability, but reaches high values, around 5 after as few as 10 reports. The 90^{th} percentile is worse, reaching trackability of 5 after as little as 7 reports.

6 Bitwise Everlasting Privacy Mechanism

We present a mechanism for collecting statistics about the distribution of a single bit in the population, in such a way that everlasting privacy is maintained. Our mechanism is a permanent state one, using a state that consists of a noisy copy of the private bit. At each report, the user sends a noisy version of the state, effectively sending a doubly noisy version of their private bit. We show the mechanism achieves good accuracy, and reasonable everlasting privacy. Since this mechanism is a permanent state mechanism, we can use Theorem 13 to give a less than reasonable upper bound on the untrackable parameter of this mechanism. We show, however a lower bound of the untrackable parameter of this mechanism that is not far off from the upper bound in Theorem 13.

Consider the mechanism where each user holds one bit, b. First they generate a permanent randomization, $b' = b \oplus x$, where $x \sim \text{Ber}\left(\frac{1}{e^{\varepsilon_1}+1}\right)$. Then at each report they generate a report bit, $r = b' \oplus y$, where $y \sim \text{Ber}\left(\frac{1}{e^{\varepsilon_2}+1}\right)$. The aggregator receives these reports from all users and invokes the frequency oracle to output an estimate:

$$\tilde{p}_0 = \frac{e^{\varepsilon_1+\varepsilon_2} + 1 - (e^{\varepsilon_1}+1)(e^{\varepsilon_2}+1)\sum_{i\in[n]} r_i}{(e^{\varepsilon_1}-1)(e^{\varepsilon_2}-1)}$$

and $\tilde{p}_1 = 1 - \tilde{p}_0$. Let $\tilde{p}$ be the vector whose coordinates are $\tilde{p}_0$ and $\tilde{p}_1$. Let p be the vector of true frequencies.

Privacy, Accuracy and Trackability

Bitwise Everlasting Privacy is ε_1-EDP, outputs $\tilde{p}$ such that with probability $1-\beta$:

$$\left\|\tilde{p}-p\right\|_\infty \leq \frac{(\varepsilon_1+2)(\varepsilon_2+2)}{\varepsilon_1\cdot\varepsilon_2}\sqrt{\frac{32\ln(2/\beta)}{n}}$$

and is $\left\lfloor\frac{k}{2}\right\rfloor\varepsilon_2$-Untrackable, but no better than $\frac{k}{2}\varepsilon_2-\varepsilon_1-\ln 2$-Untrackable. The proof of these claims can be found in the full version of the paper.

7 Report Noisy Inner Product

In this section we present a method for collecting statistics about users' data when it is encoded in a vector of d bits. This mechanism allows us to solve the heavy hitters or histograms problems, while maintaining everlasting privacy. This solution achieves good accuracy with high probability and is effectively untrackable with high probability, but only for a "not so large" number of reports (where "not so large" is approximately the square root of the number of vectors in the state).

The "delta" part of the untrackable bound of this solution can be small, but most likely not *cryptographically* small. While in Differential Privacy one should make sure the "delta" part is cryptographically small, it is not clear whether or not the same requirement applies to the framework of tracking.

The construction of this mechanism follows a general transformation from a Locally Differential Privacy mechanism to an Everlasting Privacy mechanism with certain trackability parameters: memorize a fixed number (L) of executions of a local privacy preserving computation. At each collection the mechanism mimics one of these stored executions, choosing one of them at random. Everlasting Privacy is maintained by the finite access to a user's data: only L total different executions are ever available to the adversary. On the other hand, in terms of trackability, as long as no two different stored execution are played, there is no difference between one user and two users. No guarantees are given if the same stored execution is chosen twice.

In our instantiation of this idea, Report Noisy Inner Product is based on creating a state that contains random d-bit vectors as well as their noisy inner product with the user's private value.

In this setting there are n users. Let:

$$\varepsilon' := \frac{\varepsilon}{2\sqrt{2L\ln\left(\frac{1}{\delta}\right)}}.$$

At initialization, every user i, with private value $u_i\in\{0,1\}^d$ chooses L random vectors $\{v_{i,j}\}_{j\in[L]}$, $v_{i,j}\in\{0,1\}^d\setminus\{\vec{0}\}$, and L noisy bits $\{x_{i,j}\}_{j\in[L]}$ such that $x_{i,j}\sim \mathrm{Ber}\left(\frac{1}{e^{\varepsilon'}+1}\right)$ and calculates $b_{i,j}=\langle v_{i,j},\,u_i\rangle\oplus x_{i,j}$.

At each time of collection, every user j picks at random a vector from the state generated in the previous step. That is, they choose one of the v_i's generated before and the corresponding result of the inner product s_i. They then send it to the server. We refer to the report user i sends at a given collection time as (V_i,B_i) (i.e. the vector and the noisy inner product). The aggregator receives these reports from all users and invokes the frequency oracle to output an estimate:

$$\tilde{p}_u := \frac{2^d-1}{2^d n}\frac{e^{\varepsilon'}+1}{e^{\varepsilon'}-1}\sum_{i\in[n]}(-1)^{\langle V_i,\,u\rangle\oplus B_i}+\frac{1}{2^d}.$$

Since we never choose the vector $\vec{0}$ as one of the vectors of the state, we introduce a small bias to the probability that a report will agree with any other value than the one used to generate it. This bias is corrected by the multiplicative $\frac{2^d-1}{2^d}$ factor and the additive $\frac{1}{2^d}$ factor, resulting in an unbiased estimator.

Let p be the entire true frequency vector and $\tilde{p}$ as the entire estimated frequency vector.

7.1 Privacy, Accuracy and Trackability

The mechanism Report Noisy Inner Product (RNIP) maintains (ε, δ)-Approximate Everlasting Privacy, outputs $\tilde{p}$ such that with probability $1-\beta$:

$$\begin{aligned}\left\|\tilde{p}-p\right\|_\infty &\leq \frac{\varepsilon'+2}{\varepsilon'}\sqrt{\frac{8\ln(2^{d+1}/\beta)}{n}} \\ &= O\left(\sqrt{\frac{\ln(2^{d+1}/\beta)\ln(1/\delta)L}{n\varepsilon^2}}\right).\end{aligned}$$

And it is $\left(0, \frac{k^2}{L}+\frac{L^2}{2^d}\right)$-untrackable for k reports.

The proofs of these claims can be found in the full version of the paper and are similar to the analysis in [16].

7.2 Parameter Selection

When deploying this mechanism, the significant parameters considered are the everlasting privacy and desired accuracy. In our setting we have n users and our data consists of values that can be encoded into d bits. Assume we wish to have everlasting privacy (ε, δ) and accuracy α with probability $1-\beta$. By the results of the accuracy analysis, the required value of the Differential Privacy parameter of every report, which we denoted ε', needs to be at least

$$\frac{2\sqrt{2\ln(2^{d+1}/\beta)}}{\alpha\cdot\sqrt{n}-\sqrt{2\ln(2^{d+1}/\beta)}}.$$

For most interesting settings we can assume that $\alpha > 2\sqrt{\frac{2\ln(2^{d+1}/\beta)}{n}}$, which allows us to choose $\varepsilon' = \frac{4}{\alpha}\sqrt{\frac{2\ln(2^{d+1}/\beta)}{n}}$. Once we have ε' we can say that the mechanism needs to have a state of size at most $L = \left\lfloor \frac{\varepsilon^2}{8\varepsilon'^2\ln(1/\delta)} \right\rfloor$. This means that the mechanism is $\left(0, \frac{k^2}{L}\right)$-Untrackable for k reports.

To summarize, if we were to require (ε, δ)-Everlasting Privacy and α accuracy with probability at least $1-\beta$, then for k reports we can guarantee:

$$\left(0, \widetilde{O}\left(\frac{k^2}{\alpha^2\varepsilon^2 n}\right)\right)\text{-Untrackable.}$$

where the $\widetilde{O}$ hides the logarithmic factors in the relaxation parameter for differential privacy δ, the failure probability β and the size of the vectors d.

8 Conclusions and Open Problems

The issue of using differentially private mechanisms in order to track users is a newly formulated problem. While avoiding tracking is very natural, it has not been investigated before in a formal manner. The notion of Everlasting Privacy is very tempting, and indeed,

some companies implemented and deployed it. But Everlasting Privacy should be handled with caution; We have shown that one such deployment of Everlasting Privacy left much to be desired in terms of the untrackable parameter. The risks of tracking are real, and as such every mechanism deployed to a user base must try to prevent it as much as it can.

Many questions concerning tracking are open and the results presented here should be treated as a preliminary investigation. The most important one is how do you combine the constraints on accuracy and on everlasting differential privacy to produce a lower bound on the untrackable parameter. In particular, are the schemes of Sections 6 and 7 the best one can hope for, or are there better mechanisms? One downside of the scheme of Section 7 is the rapid deterioration in the untrackable parameter once k reaches $\sqrt{L}$. Is there a scheme with a more graceful degradation of the untrackable parameter?

The mechanisms we presented are permanent state mechanisms. Perhaps mechanisms which transform the state between executions can achieve better untrackable parameter bounds? Doing such a construction is delicate, since if not done correctly one of two things might happen:

1. The Differential Privacy guarantee will decline the more the state alters.
2. The accuracy will decline, as many different inputs might converge to the same states over time.

But perhaps a clever construction of a mechanism that transforms its state can achieve a much better untrackable parameter bound for given Differential Privacy and accuracy requirements.

Also, perhaps everlasting privacy is an unreasonable demand. A mechanism that achieves privacy for many executions, but not for infinite executions, can be very suitable for practical purposes as well. If so, how can we extend these results to these "long-lasting" privacy mechanisms?

References

1 Raef Bassily, Kobbi Nissim, Uri Stemmer, and Abhradeep Guha Thakurta. Practical locally private heavy hitters. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 2288–2296, 2017. URL: `http://papers.nips.cc/paper/6823-practical-locally-private-heavy-hitters`.

2 Raef Bassily and Adam D. Smith. Local, private, efficient protocols for succinct histograms. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 127–135. ACM, 2015. `doi:10.1145/2746539.2746632`.

3 T.-H. Hubert Chan, Elaine Shi, and Dawn Song. Private and continual release of statistics. *ACM Trans. Inf. Syst. Secur.*, 14(3):26:1–26:24, 2011. `doi:10.1145/2043621.2043626`.

4 Rachel Cummings, Sara Krehbiel, Yuliia Lut, and Wanrong Zhang. Privately detecting changes in unknown distributions. *CoRR*, abs/1910.01327, 2019. `arXiv:1910.01327`.

5 Rachel Cummings, Sara Krehbiel, Yajun Mei, Rui Tuo, and Wanrong Zhang. Differentially private change-point detection. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pages 10848–10857, 2018. URL: `http://papers.nips.cc/paper/8280-differentially-private-change-point-detection`.

6 Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. Collecting telemetry data privately. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 3574–3583, 2017. URL: http://papers.nips.cc/paper/6948-collecting-telemetry-data-privately.

7 Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N. Rothblum. Differential privacy under continual observation. In Leonard J. Schulman, editor, *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 715–724. ACM, 2010. doi:10.1145/1806689.1806787.

8 Cynthia Dwork, Moni Naor, Toniann Pitassi, Guy N. Rothblum, and Sergey Yekhanin. Pan-private streaming algorithms. In *Proceedings of Innovation in Computer Science, ICS 2010*, pages 66–80. Tsinghua University Press, 2010. URL: http://conference.iiis.tsinghua.edu.cn/ICS2010/content/papers/6.html.

9 Cynthia Dwork, Moni Naor, and Salil P. Vadhan. The privacy of the analyst and the power of the state. In *FOCS*, pages 400–409, 2012. doi:10.1109/FOCS.2012.87.

10 Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014. doi:10.1561/0400000042.

11 Cynthia Dwork, Guy N. Rothblum, and Salil P. Vadhan. Boosting and differential privacy. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 51–60. IEEE Computer Society, 2010. doi:10.1109/FOCS.2010.12.

12 Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Abhradeep Thakurta. Amplification by shuffling: From local to central differential privacy via anonymity. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 2468–2479. SIAM, 2019. doi:10.1137/1.9781611975482.151.

13 Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. RAPPOR: randomized aggregatable privacy-preserving ordinal response. In Gail-Joon Ahn, Moti Yung, and Ninghui Li, editors, *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, November 3-7, 2014*, pages 1054–1067. ACM, 2014. doi:10.1145/2660267.2660348.

14 Matthew Joseph, Aaron Roth, Jonathan Ullman, and Bo Waggoner. Local differential privacy for evolving data. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pages 2381–2390, 2018. URL: http://papers.nips.cc/paper/7505-local-differential-privacy-for-evolving-data.

15 Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam D. Smith. What can we learn privately? In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*, pages 531–540. IEEE Computer Society, 2008. doi:10.1109/FOCS.2008.27.

16 Moni Naor, Benny Pinkas, and Eyal Ronen. How to (not) share a password: Privacy preserving protocols for finding heavy hitters with adversarial behavior. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, 2019*, pages 1369–1386. ACM, 2019. doi:10.1145/3319535.3363204.

17 Moni Naor and Neil Vexler. Can two walk together: Privacy enhancing methods and preventing tracking of users, 2020. arXiv:2004.03002.

18 Salil Vadhan. *The Complexity of Differential Privacy*, pages 347–450. Springer International Publishing, Cham, 2017. doi:10.1007/978-3-319-57048-8_7.

A Proofs for Section 3

A.1 Proof of Theorem 14

First, we properly define the composition of m mechanisms, $\{M_i\}_{i\in[m]}$. In our setting a user generate m report streams using the m mechanisms. Namely, each M_i was used to generate k_i reports. Let the sum of all k_i's be k. Let $\widehat{M}$ be the composition of all the M_i's. Formally, $\widehat{M}$ will first generate k_1 reports from the report stream generator of M_1, it will then continue to generate k_2 reports from the report stream generator of M_2, and so on, until all k reports were generated. Let the indices of the reports generated by M_i be J_i, i.e. $J_1 = \{1, ..., k_1\}$, $J_2 = \{k_1 + 1, ..., k_1 + k_2\}$, and so on. Notice that the probability that $G_k^{\widehat{M}}$, on input u, will generate a report stream t of k reports is exactly:

$$Pr\left[G_k^{\widehat{M}}(u) = t\right] = \prod_{i\in[m]} Pr\left[G_{k_i}^{M_i}(u) = t_{J_i}\right]$$

since all mechanisms M_i are executed independently.

We are now ready to prove Theorem A.1.

Proof. In the definition of untrackable, we consider whether a set of reports were generated by a single user or two, according to any partition. To prove that the composition mechanism is $(\gamma', m\delta + \delta')$-untrackable we will prove that the bound in the definition holds for every partition possible.

Consider the partition $P = \{P_1, P_2\}$ of all the reports generated by $\widehat{M}$, where P_1 are the reports associated with the first user and P_2 are the reports associated with the second. We will split this partition into partitions for each mechanism separately. Namely, for every M_i we define a partition $P^i = \left\{P_1^i, P_2^i\right\}$, such that each $P_1^i = P_1 \cap J_i$ and similarly for P_2^i. The partition P^i is exactly the partition on reports generated by M_i induced by P. Notice that we allow P_1^i (or P_2^i) to be empty.

Consider new mechanisms $\{F_i\}_{i\in[m]}$. that each receives as input a bit b. For every F_i, If $b = 0$ the mechanism outputs a stream generated by one copy of M_i, and if $b = 1$ the mechanism outputs a stream generated by two independent copies of M_i according to partition P_i. If either P_1^i or P_2^i are empty, the output F_i will not depend on its input b. If the M_i's are (γ, δ)-untrackable then the F_i's are (γ, δ)-differentially private. This allows us to use Advanced Composition for differential privacy (Theorem 4) to say that the m-fold composition of all of the F_i's is $(\gamma', m\delta + \delta')$-differentially private for γ' as is in the theorem statement. Notice that conditioned on all mechanisms receiving input 1, the output product distribution over reports is identical to the case where all reports, for each mechanism, were generated by two users. Similarly if all inputs are 0, the output product distribution over reports is identical to the case where all reports, for each mechanism, were generated by one user. This implies that the composition of the original M_i's, $\widehat{M}$, is $(\gamma', m\delta + \delta')$-untrackable. ◀

A.2 Proof of Theorem 16

Proof. The proof for this theorem is very intuitive. Since the mechanism is γ-untrackable for k reports, it is also γ-untrackable for fewer reports. This teaches us that by paying no more than γ we can reduce the question of being untrackable for n users to the question of being untrackable for $n - 1$ users. Continuing this until we have 2 users costs us $(n - 2)\,\gamma$ to the untrackable parameter, resulting in a total of $(n - 1)\,\gamma$ untrackable.

Formally, we prove this by induction. Assume that a mechanism M is $(t-1)\,\gamma$-untrackable for t users for k reports. We wish to prove that the mechanism M is $t\gamma$-untrackable for $t+1$ users for k reports. The base case, $t=2$, follows directly from the fact that the mechanism is γ-untrackable for k reports.

If the mechanism M is γ-untrackable for k reports, then it is γ-untrackable for fewer reports as well. We denote $Pr[A] \coloneqq Pr\left[G^M_{|A|}(u) \in T_A\right]$. Notice that for all user data $u \in U$, all partitions $P = \{P_i\}_{i\in[t+1]}$ of $[k]$ into $t+1$ parts and all output stream sets $T \subseteq R^k$:

$$\begin{aligned}\prod_{j\in[t+1]} Pr[P_j] &= Pr[P_1]\cdot Pr[P_2]\cdot \prod_{j\in[t+1]\setminus\{1,2\}} Pr[P_j] \\ &\le e^{\gamma} Pr[P_1\cup P_2] \prod_{j\in[t+1]\setminus\{1,2\}} Pr\left[G^M_{|P_j|}(u)\in T_{P_j}\right] \\ &\le e^{t\gamma} Pr\left[G^M_k(u)\in T\right]\end{aligned}$$

Where the first inequality is due to the mechanism being γ-untrackable for $|P_1|+|P_2|$ reports and the second inequality is due to the induction hypothesis.

Similarly, in the other direction:

$$\begin{aligned}Pr\left[G^M_k(u)\in T\right] &\le e^{(t-1)\gamma} Pr[P_1\cup P_2] \prod_{j\in[t+1]\setminus\{1,2\}} Pr[P_j] \\ &\le e^{t\gamma} Pr[P_1]\cdot Pr[P_2]\cdot \prod_{j\in[t+1]\setminus\{1,2\}} Pr[P_j] \\ &= e^{t\gamma} \prod_{j\in[t+1]} Pr[P_j]\end{aligned}$$

Where the first inequality is due to the induction hypothesis and the second inequality is due to the mechanism being γ-untrackable for $|P_1|+|P_2|$ reports. ◀

A.3 Proof of Theorem 17

Proof. The proof for this is also rather intuitive. Theorem 13 teaches us that we can exchange the probability that two sets of reports, of size totaling k', originated from two users to the probability they originated from one user by paying no more than $\left\lfloor \frac{k'}{2} \right\rfloor \varepsilon$ in the untrackable parameter. By combining pairs of users, we can use this to reduce the question of being untrackable for n users to the question of untrackable for $\lceil \frac{n}{2} \rceil$ users, by paying no more than $\lfloor \frac{k}{2} \rfloor \varepsilon$. By repeating this process $\lceil \log n \rceil - 1$ times we can reduce the question of untrackable for n users to the question of untrackable for 2 users, by paying no more than $\lceil \log n \rceil \lfloor \frac{k}{2} \rfloor \varepsilon$.

Formally, we prove this by induction. Assume that the mechanism is $\lceil \log t \rceil \lfloor \frac{k}{2} \rfloor \gamma$-untrackable for t users for k reports. We wish to prove that the mechanism is $\lceil \log (t+1) \rceil \lfloor \frac{k}{2} \rfloor \gamma$-untrackable for $t+1$ users for k reports. The base case $t=2$ follows directly from the fact that the mechanism is γ-untrackable for k reports.

Assume t is odd. The proof is very similar when it is even, but for simplicity we will only show it for odd values of t. We denote $\ell \coloneqq \lceil \log (t+1) \rceil$ and use the same notation for $Pr[A]$ as before. Notice that for all user data $u \in U$, all partitions $P = \{P_i\}_{i\in[t+1]}$ of $[k]$ into $t+1$ parts and all output stream sets $T \subseteq R^k$:

$$\begin{aligned}\prod_{j\in[t+1]} Pr\left[P_j\right] &= \prod_{j\in\left[\frac{t+1}{2}\right]} Pr\left[P_{2j}\right]\cdot Pr\left[P_{2j+1}\right]\\ &\leq \prod_{j\in\left[\frac{t+1}{2}\right]} e^{\left\lfloor\frac{|P_{2j}\cup P_{2j+1}|}{2}\right\rfloor\gamma} Pr\left[P_{2j}\cup P_{2j+1}\right]\\ &\leq e^{\left\lfloor\frac{k}{2}\right\rfloor\gamma}\prod_{j\in\left[\frac{t+1}{2}\right]} Pr\left[p_{2j}\cup P_{2j+1}\right]\\ &\leq e^{\ell\left\lfloor\frac{k}{2}\right\rfloor\gamma} Pr\left[G_k^M\left(u\right)\in T\right]\end{aligned}$$

Where the first inequality is due to Theorem 13 and the third inequality is due to the induction hypothesis.

Similarly, for the other direction:

$$\begin{aligned}Pr\left[G_k^M\left(u\right)\in T\right] &\leq e^{(\ell-1)\left\lfloor\frac{k}{2}\right\rfloor\gamma}\prod_{j\in\left[\frac{t+1}{2}\right]} Pr\left[P_{2j}\cup P_{2j+1}\right]\\ &\leq e^{\ell\left\lfloor\frac{k}{2}\right\rfloor\gamma}\prod_{j\in\left[\frac{t+1}{2}\right]} Pr\left[P_{2j}\right]\cdot Pr\left[P_{2j+1}\right]\\ &= e^{\ell\left\lfloor\frac{k}{2}\right\rfloor\gamma}\prod_{j\in[t+1]} Pr\left[P_j\right]\end{aligned}$$

Where the first inequality is due to the induction hypothesis and the second inequality is due to Theorem 13. ◀

B Estimated Median and 90th Percentile Of RAPPOR Figure

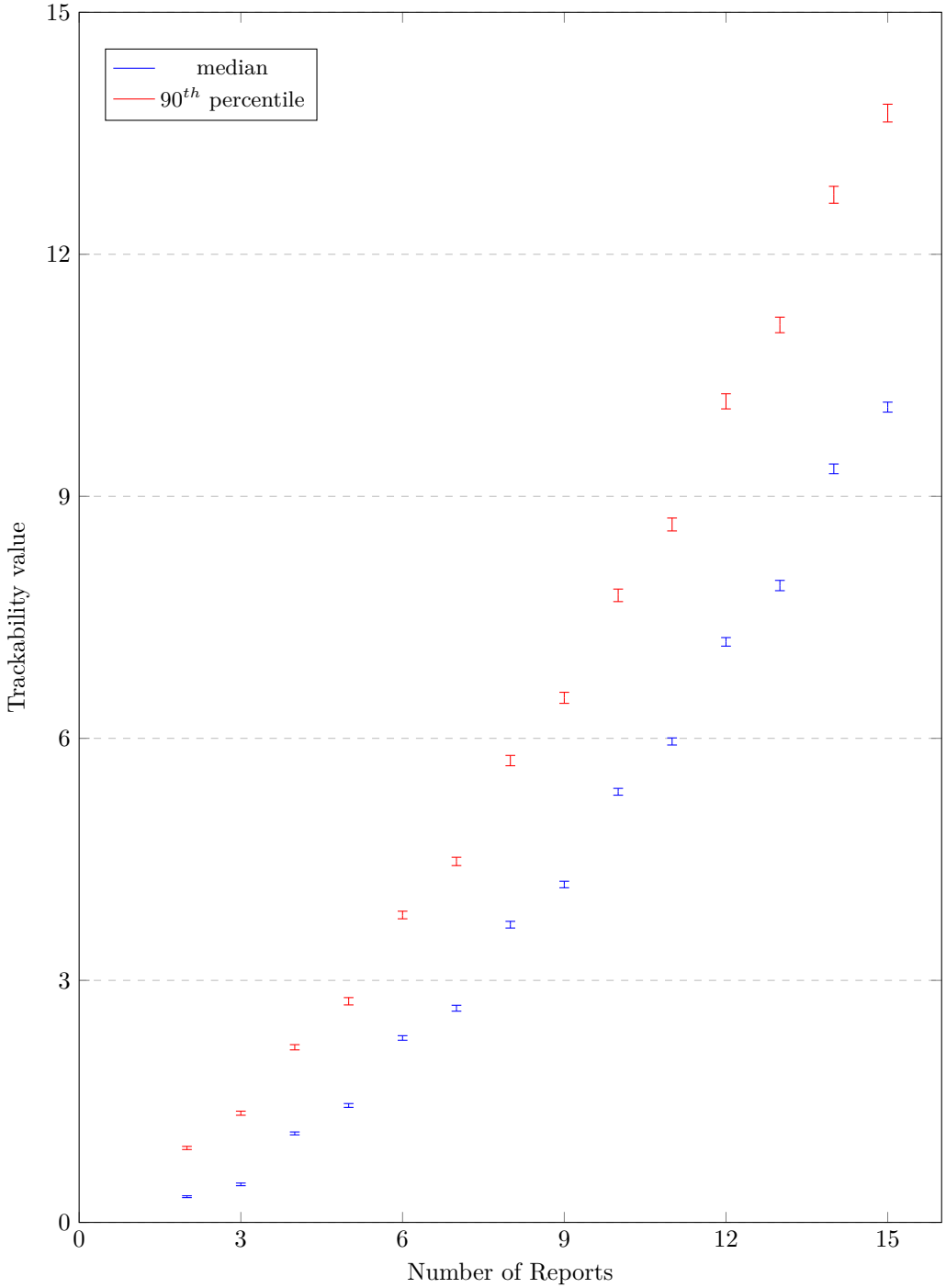

Figure 1 Growth of the estimated median and 90^{th} percentile of the trackability random variable of RAPPOR as a function of the number of reports.