

An Architecture for Privacy-Preserving Telemetry Scheme

Kenneth Odoh^[0000–0003–4892–4301]

kenneth.odoh@gmail.com
<https://kenluck2001.github.io>

Abstract. We present a privacy-preserving telemetry aggregation scheme. Our underlying frequency estimation routine works within the framework of differential privacy. The design philosophy follows a client-server architecture. Furthermore, the system uses a local differential privacy scheme where data gets randomized on the client before submitting the request to the resource server. This scheme allows for data analysis on de-identified data by carefully adding noise to prevent re-identification attacks, thereby facilitating public data release without compromising the identifiability of the individual record. This work further enhances privacy guarantees by leveraging Oblivious HTTP (OHTTP) to achieve increased privacy protection for data in transit that addresses pre-existing privacy vulnerabilities in raw HTTP. We provide an implementation that focuses on frequency estimation with a histogram of a known dictionary. Our resulting formulation based on OHTTP has provided stricter privacy safeguards when compared to trusting an organization to manually delete identifying information from the client’s request in the ingestor as deployed in reference work [3].

Source code available at <https://github.com/kenluck2001/miscellaneous/tree/master/src/Privacy-Preserving-Telemetry>

Keywords: Cryptography, Privacy, Security, Telemetry, Data mining

1 Introduction

Understanding the usage patterns of deployed devices can provide insight into improving customer experience from an organizational perspective. De facto attempts to obtain user data can increase privacy risks. Coincidentally, there is a market for trading customer data to facilitate precise advertisement targeting. Acxiom¹ is one of the world’s largest data brokers that harvests data from nearly a billion users worldwide. A privacy attack can result from the actions of malicious actors who act surreptitiously and pursue goals that are inconsistent with those of the users. Increasing financial motives for abusing users’ data have motivated our research into novel privacy-enhancing mechanisms to provide privacy by design.

¹ <https://www.acxiom.com/>

Internet standards ² led by a consortium of technology firms, university researchers, hobbyists, and others have a higher chance of building higher quality internet protocols due to the increased scrutiny of multiple industrial partners. Similarly, these setups are analogous to peer reviews in that they help researchers improve their engineering thinking through the critical evaluations of their work. In the same way, it is prudent to build industrial systems utilizing the privacy guarantees afforded by the Oblivious HTTP protocol rather than blindly trusting an Apple internal aggregator service. Given how recent news has demonstrated the prevalence of blatant privacy abuse of user data in the industry. Hence, we have taken this approach in this manuscript.

Differential privacy (DP) is a structured mathematical framework that supports principled reasoning about privacy loss in a database. Randomization happens by adding calibrated noise to the original data to prevent reverse-engineering the original value of the randomized data, thereby providing privacy protection. DP protects sensitive data while maintaining a trade-off between added noise and expected utility. As a result, DP has increased the adoption of privacy-preserving data mining tasks that facilitate public data release without compromising individual privacy. The rigorous nature of the DP mechanism makes it ideal for satisfying evolving privacy regulations.

Gathering telemetry is a necessary prerequisite for several data analytics tasks. Our work has adopted differential privacy as a standard for guaranteeing privacy protection. This work extends the privacy guarantee of the system built by Apple [3]. Their work [3] requires trust that the ingestor will not abuse client-identifying data. This expectation of trust is unrealistic as monetary benefits arise from potential trading on customer data. The unanswered question is how to improve privacy on this system without resorting to onion routing [6]? TOR utilizes flooding in its operation and incurs unacceptable overhead that can impact scalability. In contrast, systems such as Prio [8], DPrio [12], and Prio+ [1] may provide higher privacy guarantees due to their incorporation of multiparty-based secret sharing. On the contrary, several applications require reasonable privacy guarantees with minimal setup costs, which is the premise of our manuscript.

Our thesis focuses on enhancing the privacy of our telemetry scheme based on Oblivious HTTP [22] with significant simplification. We seek to understand common user patterns across devices by generating snapshot readings for summary device health or other information. Hence, we have developed a privacy-preserving telemetry system that uses local differential privacy, where data gets randomized on the client before submitting the request to the resource server. Therefore, it delivers a higher degree of privacy guarantee when compared to the central differential private scheme. The paper is structured as follows: a summary of contributions in Section 2, a literature review of previous works in Section 3, a brief explanation of differential privacy in Section 4, Oblivious HTTP in Section 5, an overview of our base implementation in Section 6, a discussion of the merits of our solution in Section 8. We have demonstrated the usefulness of our

² <https://www.ietf.org/standards/>

architecture with a case study and several experiments in Section 7. Finally, we present limitations, future work, and conclusions in Section 9 and Section 10.

2 Contributions

Our contributions are summarized as follows:

- We provide an implementation focused on frequency estimation with a histogram of known dictionary words. However, the reference work [3] has a known limitation where we trust the ingestor will not cooperate with bad actors that may abuse the customer’s privacy.
- HTTP does not provide privacy by design. The quest for rigorous privacy protection has motivated us to build our privacy scheme based on Oblivious HTTP, which fixes many privacy vulnerabilities in HTTP.
- We demonstrate a conceptual framework for enhancing privacy protection using a standardized internet protocol. Therefore, we no longer need to trust the ingestor will not abuse client identifiers (such as IP addresses or session data). This setup [3] results in a weaker notion as it is difficult to audit whether the required deletion happened.

3 Related Work

Private telemetry is very interesting to all service providers, as seen in all major browsers and operating systems. Procho [5] introduced the Encode, Shuffle, Analyze(ESA) framework widely used in telemetry, error reporting, and continuous monitoring. STAR [9] is a data aggregation system that enforces k-anonymity based on well-known cryptographic primitives. Privacy leaks from this scheme can impact users’ confidence. Similarly, several deployed telemetry systems exist in the industry, but most lack privacy-preserving characteristics. For example, Facebook created a system named PCAT [25] to continuously monitor production assets and offer support for change detection, alerting, monitoring, and diagnostics. As a result, when this telemetry scheme gets deployed beyond the sandboxed production environment to real-user devices on the edge. Privacy leaks from this scheme can impact users’ confidence. Subsequently, privacy-preserving data mining methods have evolved from theoretical abstractions to solving real-world applications.

Differential privacy [10] (DP) is a robust method for quantifying how privacy degrades under frequent adversarial evaluation of database records. DP can work in local or central settings where a local DP scheme has higher privacy guarantees. There are examples of public-facing industrial DP deployment in local settings such as RAPPOR [11] and central settings in PINQ [17]. Furthermore, alternative notions of privacy-enhancing technology include Verifiable Distributed Aggregation Functions (VDAFs) [19] and TOR [6] can provide more privacy guarantees at a higher cost than Oblivious HTTP [22].

Several lines of work utilize sketch-based algorithms for network monitoring because of their efficient approximate count estimation as follows: OctoSketch [24], TrustSketch [7], and HeteroSketch [2]. Hence, we have adopted sketch-based estimation as seen in (Algorithms 1, 3, 5, and 7) of our reference paper [3]. Several privacy-preserving analytics processing engines exist to support downstream data analysis. One such scheme is PRIVAPPROX [20] utilizing a zero-knowledge proof construction to provide higher privacy guarantees than differential privacy. POPSTAR [15] uses oblivious PRF and polynomial commitment for privacy-preserving aggregation schemes.

One such case is the privacy-aware deployment at Apple [3] to capture insights into crashes (and other events) from a collection of phones using well-known security policies and differential privacy. Through our work, we propose privacy-preserving frequency estimation without trusting that the ingestor will delete client-identifying information without a persistent audit. We have eliminated the trust by providing a simplified implementation with extended privacy guarantees by adopting Oblivious HTTP [22] to increase clients' privacy assurance.

4 Differential Privacy

Differential Privacy (DP) is a privacy-enhancing technology that allows for data analysis on de-identified data by carefully adding noise to prevent re-identification attacks, thereby facilitating public release without impacting the privacy of the individual record.

Definition 1: (Differential Privacy) Following Definition 7 of [10], for each pair of the data record D and D' , noise, ϵ , and a randomizer, \mathcal{M} satisfies $P(\mathcal{M}(D) \in \mathcal{O}) \leq e^\epsilon P(\mathcal{M}(D') \in \mathcal{O})$

When $\epsilon \approx 0$, we attain higher privacy guarantees with more similarities across the data set. Note, when $\epsilon = 0$, at that point, perfect secrecy is achieved by limiting the ability to perform statistical analysis. When $\epsilon = \infty$, we have a blatantly non-private mechanism. Therefore, we aim to achieve reasonable privacy within an appropriate budget.

5 Oblivious HTTP

Oblivious HTTP [22] (OHTTP) is an encapsulated abstraction built on top of HTTP to address inherent privacy risks within communicating peers. There are some industrial deployments of privacy-enhancing protocols such as iCloud private relay [13] and Flo period tracker app [23].

This OHTTP protocol allows exchanging encrypted messages where the server cannot link the request to the client. This setup eliminates the risk of leaking client information when communicating. For example, exposing an IP address can uncover an individual, as it is an identifier linked to a physical node (client), or reveal information about an IoT device in a house. We can mitigate privacy leaks between communicating parties (client-server architecture) using HTTP.

OHTTP operates by using a proxy (relay server) to send the request between the client and server by adopting this level of indirection to prevent request linkability.

We have added a simplification where we removed the gateway and instead used a 3-party system (client, relay server, resource server) instead of the 4-party system (client, relay server, gateway server, resource server) as defined in the standard [22].

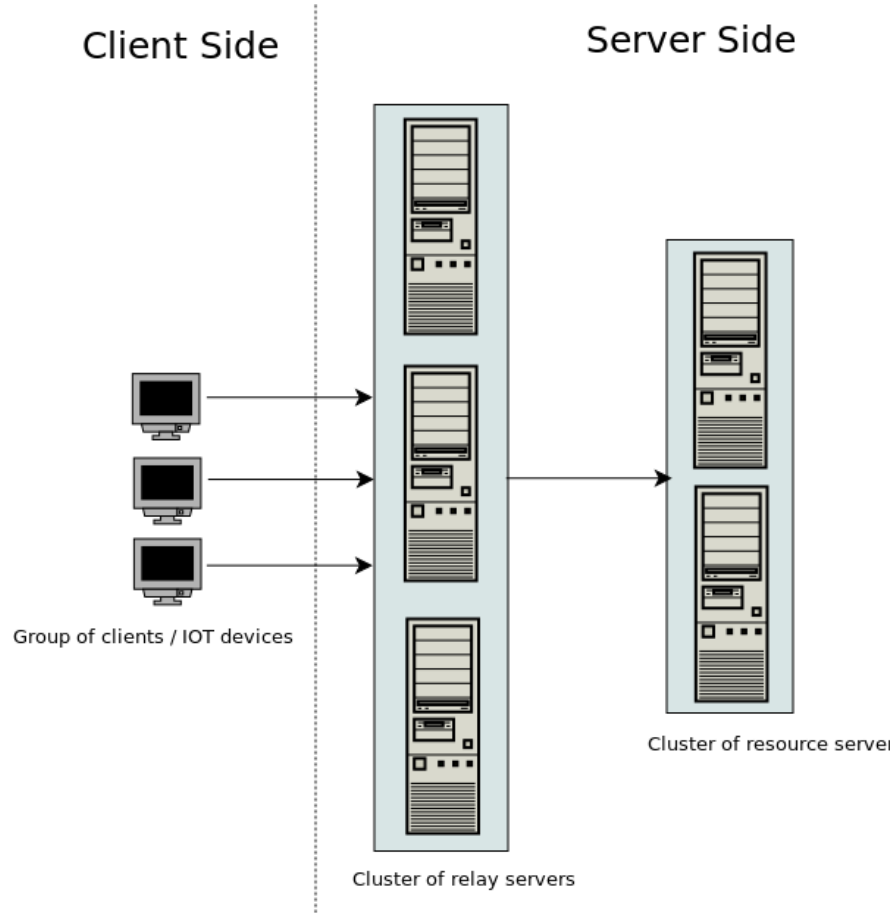


Fig. 1. Simplified Oblivious HTTP

The description of our simplified Oblivious HTTP protocol considering only the request flow (response flow got omitted because our requirement is unidirectional) is as follows, as shown in Figure 1:

- The Client creates an encrypted message using the (public key of the resource server) and forwards it to the relay server.
- The relay server forwards the encrypted message to the resource server. It is a requirement that the relay server cannot read the message, as it does not have the required key to decrypt the message on the relay server. The relay transfers information without knowing the message content.
- The resource server can decrypt the message using its private key.

6 System Overview

This work demonstrates privacy-aware frequency aggregation of event telemetry. Our implementation focuses on frequency estimation of events where we compute a histogram from a known word distribution. This formulation allows counting the frequency of a term from a known dictionary of terms. Furthermore, we have provided an aggregate of known terms as event identifiers. Apple deployment follows an equivalent naming convention with substitute names as described in paper [4] as shown in Figure 2.

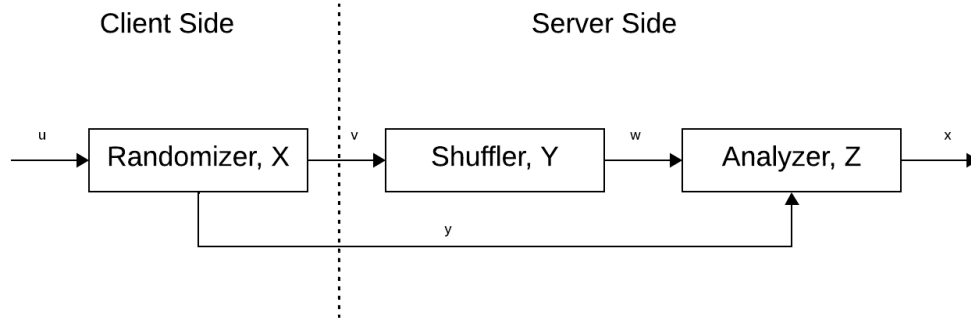


Fig. 2. DP Architecture [5]

Where u, v, w, x, y are variables depicting the life cycle of the data as it transits different stages through the pipeline as given the input data, u and resulting system output data, y , and transformation functions X, Y, Z are randomizer, shuffler, and analyzer respectively. The shuffler is optional based on the use case. This data processing pipeline has the following phases: randomizer (privatization), shuffler (ingestor), and analyzer (aggregator) as shown in Figure 3. From the perspective of a single user, they randomize the data in their device and send it to the ingestors, where identifying information is removed, and the resulting data gets forwarded to the analyzer, where aggregate statistics get computed.

We have implemented the following algorithms from the paper [3] that include $A_{client-HCMS}$ in Algorithm 5 of [3], Hadamard count mean sketch HCMs

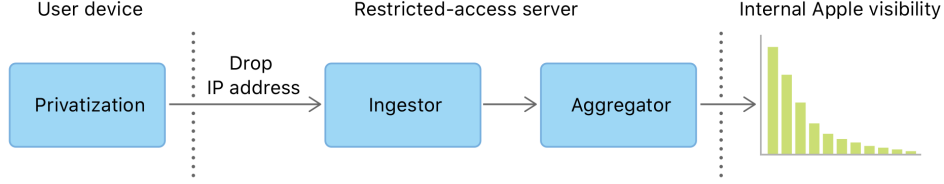


Fig. 3. Apple DP system [3]

in Algorithm 6 of [3], A_{server} in Algorithm 4 of [3], and computing the Sketch matrix known as Sketch-HCMS in Algorithm 7 of [3]. We extend previous work [3] by utilizing a proven privacy mechanism in Oblivious HTTP [22]. We can replace the ingestor shown in Figure 3 with a relay server in the setup of Oblivious HTTP. The relay server does not know anything about the requests forwarded through it. Hence, our scheme based on OHTTP has provided a stricter privacy safeguard when compared to trusting an organization to manually delete identifying information from the client's request in the ingestor.

The DP algorithm has a server and client mechanism shown in Figure 1, where the client-side algorithm is a locally differentially private scheme where the client randomized each data instance before sending transformed results to the server. Consequently, the server provides a near-precise count of events where aggregated results can handle customers' information-seeking needs. Data transfer between the client and the server can impact communication costs. Calibrated noise added to the client during the privatization phase can dictate the amount of privacy afforded and achievable accuracy at the analyzer stage. Therefore, we can achieve a trade-off between privacy, communication cost, and computation accuracy. The server-side algorithm averages the count for m number of hash functions. Similarly, the hash function should be a set of m instances of 3-wise independent hash functions where m is the number of hash functions.

k -wise independent hash function³: k -wise independent is satisfied for a set of discrete random variables X_1, \dots, X_n given that for any set $I \subseteq \{1, \dots, n\}$ with $|I| \leq k$ and any values x_i as shown in Equation 1.

$$\Pr [\wedge_{i \in I} X_i = x_i] = \prod_{i \in I} \Pr [X_i = x_i]. \quad (1)$$

Following Equation 1, k -wise independence is satisfied if we can choose a function from a hash family with a guarantee that any k keys are independent random variables. One example of a k -wise independent hash function follows the polynomial structure as shown in Equation 2.

$$H(x) = a_0 + a_1x + a_2x^2 + \dots + a_{k-1}x^{k-1} \quad (2)$$

Where $H(x)$ is a polynomial of degree $\leq k$

³ <https://www.cs.purdue.edu/homes/hmaji/teaching/Fall%202017/lectures/12.pdf>

7 Case Study

Sports tracking apps are vulnerable to adversary eavesdropping and capturing user activity to target individuals as recent news has shown ⁴. Following the prevalence of privacy abuse, we present a case study on one of our users named "Jane" who is a sports enthusiast and uses our fictitious generic sports tracking app ⁵, who wears a tracking device. We have demonstrated our privacy-preserving telemetry architecture with sport tracking. Our scheme supports a client-server architecture. The client side is the device tracker worn on the person, while the server component aggregates the data.

Our implemented scheme protects users' privacy as data gets randomized on the clients before transferring the same data to the server. As a result, we can achieve local differential privacy in our implementation. By utilizing this architecture, there is less chance of privacy risk, as an attacker can intercept the scrambled data on transmission to the server, and the adversary becomes incapable of reverse-engineering to uncover the original data from the randomized data. The device tracker uses sensors such as a GPS locator, gyroscope, accelerometer, and others to categorize the activities of users into telemetry events that include: "walking", "running", and "sleeping".

As part of our evaluation, we have arbitrarily randomly sampled these events ("walking", "running", and "sleeping") using the given probabilities $(\frac{3}{5}, \frac{3}{10}, \frac{1}{10})$. The sampled data is a list of snapshots of original data (telemetry events) on the client device and gets transformed using the privatization algorithm. The de-identified scrambled data is then transferred to the aggregator (relay server) and then to the analyzer, where aggregate statistics are estimated. We can claim that our telemetry scheme works as expected if the distribution of the events after the analyzer stage matches the data distribution of the original sampled events before randomization.

Jane always wears a device tracker to monitor her activities for health reasons. Let us define two concepts used in our discussion.

- Original data proportion: This measure is the ratio of the occurrence of telemetry events captured in the data before randomization on the clients. For example, if the data has the following events as follows: 10 "walking", 5 "running", and 5 "sleeping", then the resulting probabilities are $(\frac{1}{2}, \frac{1}{4}, \frac{1}{4})$ for ("walking", "running", and "sleeping") respectively.
- Randomized data proportion: This measure is the ratio of the occurrence of telemetry events captured in the data (after randomization). For example, if the data has the following events as follows: 10 "walking", 5 "running", and 5 "sleeping", then the resulting probabilities are $(\frac{1}{2}, \frac{1}{4}, \frac{1}{4})$ for ("walking", "running", and "sleeping").

⁴ <https://www.cnn.com/2023/07/11/europe/russian-submarine-commander-killed-krasnador-intl/index.html>

⁵ <https://www.bbc.com/news/av/technology-24379432>

We have provided two experiments to demonstrate the usefulness of our implementation as shown in SubSections 7.1, 7.2, and 7.3. The y-axis is the count of telemetry events after randomization in Figures 4 and 5.

7.1 Distributional mismatch between analyzer output and original data (input)

We have designed an experiment to understand how the randomized data proportion of telemetry events varies as the data size increases. The original data probability is kept constant for random sampling as $(\frac{3}{5}, \frac{3}{10}, \frac{1}{10})$ for ("walking", "running", and "sleeping") respectively as shown in Figure 4 with the noise, $\epsilon = 4$, and the data get increased to observe the influence on the randomized data proportions. The x-axis is the original telemetry count before privatization in Figure 4 and the combined telemetry count (y-axis) due to the approximate nature of the sketch-based frequency algorithm.

Similarly, we can see from Figure 4 that the randomized data proportion of events (after privatization) does not significantly change. This phenomenon implies that the privatization algorithm does not impact its utility at the set noise level. Our implementation shows that the frequency counting estimate is robust and preserves the distribution of the original data (before randomization).

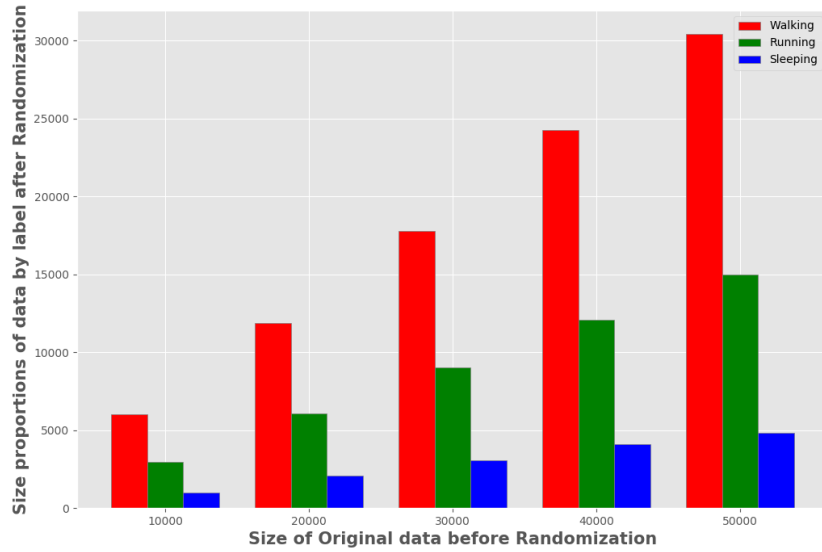


Fig. 4. Impact of original data size (before randomization) on proportion of randomized data

7.2 Noise level impact on randomized data distribution

The experiment demonstrates how the proportions of randomized data change with increasing noise levels shown in Figure 5. The x-axis is the telemetry count before privatization in Figure 5.

Furthermore, we can observe from Figure 5 how increasing the noise, ϵ , during randomization changes the proportion of randomized data.

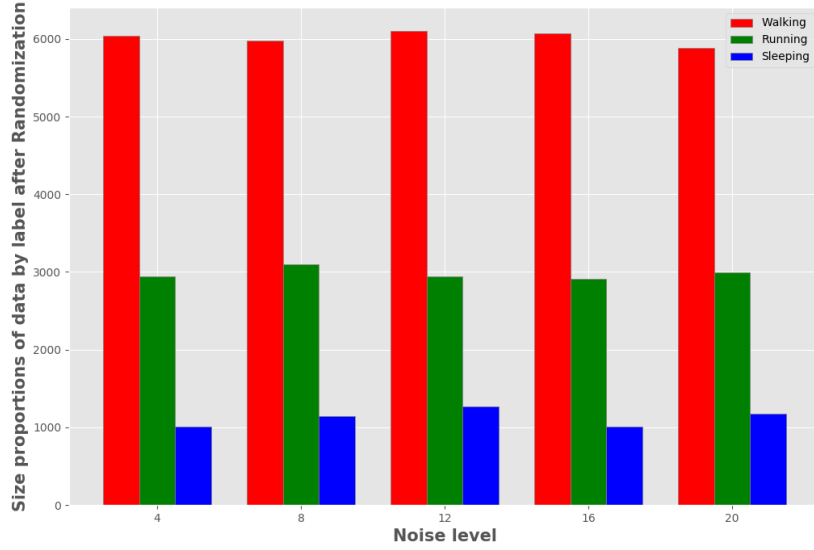


Fig. 5. Impact of noise levels on proportions of randomized data

7.3 Interpreting experimental results

We have derived insights from the experimental results visualized in Figures 4 and 5. The results show that the accuracy of the total count of original telemetry events can vary with the estimate obtained from the analyzer after applying the sketch count algorithm for frequency estimation on the telemetry event. The estimated count is an approximation of the actual telemetry event count. However, the total count of a class of telemetry elements from the sketch-based frequency algorithm in this work ensures that the data proportions are relatively fixed even if the data size keeps increasing. We also observe that the proportion of randomized events is relatively constant in the face of increasing noise levels, so the predefined noise in our sketch-based frequency estimation within reasonable

bounds does not impact the counting process. This finding implies that we obtain an approximate count of telemetry events. It captures the trend in the data as the original data distribution gets relatively unchanged from the output distribution of the analyzer phase within reasonable bounds.

8 Discussion

The system architecture demonstrated in this manuscript supports a variety of use cases. First, we can provide a platform for analyzing the sports activities of a group of users while restricting the identifiability of a single user without hindering the applicability of understanding the collective actions of individuals under observation. Second, we can organize each user's sporting events into groups and relax our privacy definition, where each group is linkable to the individual, and the randomized events in the group result in uncertainty in each event. For example, after randomization, a user "sleeping" may be confused with the same user "running" at a given time. As a result, individuals can analyze their sporting activities over an extended time horizon with a reasonable data size. The user gleans information about aggregated sporting activities. Furthermore, the successful deployment of differential privacy-based systems requires a principled way to determine the noise, ϵ , with minimal influence on the system's utility. As a result, several lines of work [14], [18] have focused on estimating the optimal noise magnitude, ϵ .

Setting up OHTTP requires a set of precautions to prevent privacy violations. OHTTP mandates that each request be stateless to avoid correlations between requests that can impact privacy and uncover the identity of the connecting client. OHTTP provides privacy, given that the relay and the resource server do not form a collusion ring. Our approach favors forwarding over flooding. As a result, we favor the forwarding scheme in OHTTP instead of TOR [16]. Relaying (forwarding by proxy) can likely impact latency. However, OHTTP fits nicely within the pre-existing internet infrastructure. We built our infrastructure on the foundation of distributed computing principles, including replication and failover, to provide high availability for the relay server. Hence, the existing fault-tolerant setup is sufficient for our unidirectional scheme.

We have used two layers of security, where the channel is secured using public key cryptography as part of the OHTTP protocol, and the data itself gets transformed utilizing differential privacy as part of the telemetry scheme based on the paper [3]. Denial-of-service and replay attacks are other security challenges when using OHTTP. This protocol can prevent denial-of-service attacks by being rate-limited. An attacker can stage a replay attack by positioning a rogue server to monitor packet traffic. Subsequently, OHTTP has a built-in method for mitigating replay attacks ⁶.

Let us consider the implementation intricacies of our DP implementation. Our approach to creating a family of hash functions, $H(x)$ shown in Equation 2

⁶ <https://www.rfc-editor.org/rfc/rfc8446#section-8>

was to generate a random matrix and extract the parameter for each hash function using vectors obtained row-by-row or column-by-column based on matrix dimensions. This setup provides an advantage to having a set of shared hash family functions for sampling the hash functions for both the client and the server. Another approach is to create a family of hash functions on the server and send them to the client. This scenario may be undesirable if we consider communication costs. A compromise solution may bias our frequency count estimates by utilizing different hash function families for clients and servers with similar distributions. As a result, we avoid sending huge matrices over the network. Eventually, a better optimization is to adopt an identical random seed with the same Pseudorandom Generator ⁷ (PRG) on the client and server, thereby enabling local sampling from the same hash family distribution without communication costs. Hadamard transforms help to reduce the variance of estimates at the analyzer (aggregator) stage as shown in Figures 2 and 3. The Hadamard count sketch algorithm is an optimized variant utilizing a dense vector instead of a sparse matrix. We observed that the quality of the solution depends on the choice of a hash function. Therefore, we created a custom hash function with appropriate statistical properties for ASCII ⁸ strings.

9 Limitations and Future Work

Our work has a limitation due to using a family of hash functions that support only ASCII strings, thereby restricting our ability to handle events in wide-character languages requiring more than 8 bits to represent a character. Furthermore, we can improve our work by categorizing client devices by utilizing a set of gateway servers in our Oblivious HTTP flow to support the logical grouping of requests. Also, the relay server can strengthen privacy protection by using anycast [21] address on a cluster of relays by purposefully increasing uncertainties linking a client's request to a particular relay server. It is imperative to ensure that the relay servers are not under any big tech firm to prevent compromise.

10 Conclusion

We have extended the privacy-preserving frequency of event telemetry [3] with oblivious HTTP. Furthermore, we have provided a working implementation of the privacy-preserving architecture with several significant improvements. Our work would facilitate a privacy-aware telemetry system for obtaining internet measurements (or other measures) while providing privacy protection.

⁷ https://en.wikipedia.org/wiki/Pseudorandom_generator

⁸ <https://en.wikipedia.org/wiki/ASCII>

References

1. Addanki, S., Garbe, K., Jaffe, E., Ostrovsky, R., Polychroniadou, A.: Prio+: Privacy Preserving Aggregate Statistics via Boolean Shares. In: Proceedings of the Security and Cryptography for Networks Conference. pp. 516–539 (2022)
2. Agarwal, A., Liu, Z., Seshan, S.: HeteroSketch: Coordinating Network-wide Monitoring in Heterogeneous and Dynamic Networks. In: Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (2022)
3. Apple Differential Privacy Team: Learning with Privacy at Scale. <https://docs-assets.developer.apple.com/ml-research/papers/learning-with-privacy-at-scale.pdf> (2017), Date accessed: February 1, 2024
4. Balcer, Victor and Cheu, Albert : Separating Local & Shuffled Differential Privacy via Histograms. CoRR **abs/1911.06879** (2019), <http://arxiv.org/abs/1911.06879>
5. Bittau, A., Erlingsson, U., Maniatis, P., Mironov, I., Raghunathan, A., Lie, D., Rudominer, M., Kode, U., Tinnes, J., Seefeld, B.: Prochlo: Strong Privacy for Analytics in the Crowd. In: Proceedings of the Symposium on Operating Systems Principles (2017)
6. Burleigh, Scott C. and Farrell, Stephen and Ramadas, Manikantan : The Onion Router. <https://www.torproject.org/> (2022), Date accessed: February 1, 2024
7. Cheng, Z., Apostolaki, M., Liu, Z., Sekar, V.: TrustSketch: Trustworthy Sketch-based Telemetry on Cloud Hosts. In: Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (2022)
8. Corrigan-Gibbs, H., Boneh, D.: Prio: Private, Robust, and Scalable Computation of Aggregate Statistics. In: Proceedings of the Symposium on Networked Systems Design and Implementation. pp. 259–282 (2017)
9. Davidson, A., Snyder, P., Quirk, E.B., Genereux, J., Livshits, B., Haddadi, H.: STAR: Secret Sharing for Private Threshold Aggregation Reporting. In: Proceedings of the SIGSAC Conference on Computer and Communications Security. pp. 697–710 (2022)
10. Dwork, C., Smith, A.D., Steinke, T., Ullman, J.: Exposed! A Survey of Attacks on Private Data. Annual Review of Statistics and Its Application **4**, 61–84 (2017)
11. Fanti, G., Pihur, V., Erlingsson, Ú.: Building a RAPPOR with the Unknown: Privacy-Preserving Learning of Associations and Data Dictionaries. Proceedings of Privacy Enhancing Technologies Symposium **2016**(3), 41–61 (2016)
12. Keeler, D., Komlo, C., Lepert, E., Veitch, S., He, X.: DPrio: Efficient Differential Privacy with High Utility for Prio. Proceedings of the Privacy Enhancing Technology **2023**(3), 375–390 (2023)
13. Lalkaka, Rustam: iCloud Private Relay: information for Cloudflare customers. <https://blog.cloudflare.com/icloud-private-relay/> (2022), Date accessed: February 1, 2024
14. Laud, P., Pankova, A.: Interpreting Epsilon of Differential Privacy in Terms of Advantage in Guessing or Approximating Sensitive Attributes. CoRR **abs/1911.12777** (2019), <http://arxiv.org/abs/1911.12777>
15. Li, H., Navot, S., Tessaro, S.: POPSTAR: Lightweight Threshold Reporting with Reduced Leakage. Cryptology ePrint Archive, Paper 2024/320 (2024), <https://eprint.iacr.org/2024/320>
16. Mani, Akshaya and Wilson-Brown, T. and Jansen, Rob and Johnson, Aaron and Sherr, Micah: Understanding Tor Usage with Privacy-Preserving Measurement. In: Proceedings of the Internet Measurement Conference. pp. 175–187 (2018)

17. McSherry, Frank D.: Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In: Proceedings of the SIGMOD International Conference on Management of Data. pp. 19–30 (2009)
18. Odoh, K.: Group-wise K-anonymity meets (ϵ, δ) Differentially Privacy Scheme. In: Proceedings of the International World Wide Web Conference (2024)
19. Patton, Christopher and Schoppmann, Phillipp and Barnes, Richard: Verifiable Distributed Aggregation Functions. <https://www.ietf.org/archive/id/draft-patton-cfrg-vdaf-01.html> (2021), Date accessed: February 1, 2024
20. Quoc, D.L., Beck, M., Bhatotia, P., Chen, R., Fetzer, C., Strufe, T.: PrivApprox: Privacy-Preserving stream analytics. In: Proceedings of the USENIX Annual Technical Conference. pp. 659–672 (2017)
21. Sommesse, Raffaele and Bertholdo, Leandro and Akiwate, Gautam and Jonker, Mattijs and van Rijswijk-Deij, Roland and Dainotti, Alberto and Claffy, KC and Sperotto, Anna: MAnycast2: Using Anycast to Measure Anycast. In: Proceedings of the Internet Measurement Conference. pp. 456–463 (2020)
22. Thomson, Martin and Wood, Christopher: Oblivious HTTP. <https://www.ietf.org/archive/id/draft-thomson-http-oblivious-01.html> (2021), Date accessed: February 1, 2024
23. Wetsman, Nicole and Faife, Corin: Flo period tracker launches 'Anonymous Mode' to fight abortion privacy concerns. <https://www.theverge.com/2022/9/14/23351957/flo-period-tracker-privacy-anonymous-mode> (2022), Date accessed: February 1, 2024
24. Zhang, Y., Chen, P., Liu, Z.: OctoSketch: Enabling Real-Time, continuous network monitoring over multiple cores. In: Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (2024)
25. Zhou, Y., Zhang, Y., Yu, M., Wang, G., Cao, D., Sung, E., Wong, S.: Evolvable Network Telemetry at Facebook. In: Proceedings of the USENIX Symposium on Networked Systems Design and Implementation. pp. 961–975 (2022)