Reducing the communication rounds for oblivious datastores

Sujaya Maiyya
University of California Santa Barbara

I. Abstract

Data privacy is becoming one of the major challenges faced by the database community today. Industries such as Facebook [1] and Google [2] are fined millions to billions of dollars for violating user data privacy, creating an urgent need to build systems that provide data privacy. When we think of data privacy, *data encryption* is the first solution that comes to mind. But many works [12], [13], [9], [17] have shown the inefficiency of mere data encryption in preserving data privacy. These works show that just by observing the data access patterns, an adversary can infer non-trivial information about the data or the user. These type of inference attacks are termed *access pattern attacks*.

Solutions such as Oblivious RAM (ORAM) [7], [16], [5], [15], [14] and Private Information Retrieval (PIR) [6], [18], [10] provide mechanisms that hide access patterns. Most of these works assume trusted clients who host their data on untrusted servers, typically managed by third party cloud providers. ORAM solutions achieve access pattern obliviousness by shuffling the physical locations of the data stored on the untrusted server after every access. While some PIR schemes support writing data [5], [11], most PIR schemes focus on retrieving or reading a data item without revealing the identity of the retrieved item to the external server. More recently, Pancake [8] proposed *frequency smoothing* to obfuscate access patterns wherein the access frequencies to all entries in the database are smoothed so that an adversary cannot infer any non-trivial insights on the data. Albeit using a less stringent but realistic security model, Pancake's frequency smoothing is shown to be highly pragmatic, with significantly better performance than ORAM based solutions.

In general, the access pattern obliviousness in the above schemes consists of two aspects: (i) hiding the exact data item, or rather the exact physical location of the data item accessed by a client; (ii) hiding the *type* of access, i.e. a read vs a write, requested by a client. To our knowledge, most existing solutions for access pattern obliviousness focus on proposing novel ways to solve aspect (i); whereas for aspect (ii), the most commonly [16], [15], [8], [14] adapted solution is to always perform a read followed by a write, irrespective of the type of client's request. Always reading followed by writing to hide the type of access incurs *two sequential rounds* of accesses between the clients and the external server resulting in significant overhead; *eliminating this additional overhead is the focus of this proposed idea.*

Contrasting oblivious datastores with their trusted, non-privacy-preserving datastores, many of these real world databases such as MongoDB [3] and Redis [4] read and write (or get and put) data in a single round. Low performance of oblivious datastores is a major reason for its low adoption rate in the industry. Hence, by proposing a technique that allows oblivious datastores to read/write data in a single round trip, we aim to bridge some of the gaps prevalent in commercializing oblivious datastores.

Given this motivation, the goal of this proposed project is to provide a one round solution to read or write the data stored on an external server *without revealing the type of access*. This work does not focus on hiding what physical locations are accessed by the clients, which is orthogonal to hiding the type of access. By proposing a novel one-round-trip oblivious access protocol, we reduce the communication rounds by half. This reduction in one round of communication not only reduces the bandwidth cost but also plays a vital role in reducing end-to-end latency, especially in geo-distributed settings.

The proposed idea – to read or write in one round while hiding the type of operation – can be easily integrated with Pancake [8] to reduce Pancake's communication rounds by half. Further, the proposed idea can be leveraged to create a novel ORAM solution that both hides the type of access and the specific item accessed in a single round of communication between a trusted proxy and an untrusted storage server. These potential applications across various solutions that preserve obliviousness highlight the relevance and adaptability of the proposed idea.

References

[1] Facebook fined $5b over data privacy violation. "`https://www.ftc.gov/news-events/press-releases/2019/07/ftc-imposes-5-billion-penalty-sweeping-new-privacy-restrictions`". Accessed June 7, 2021.

[2] Google fined $57M over GDPR violation. "`https://digitalguardian.com/blog/google-fined-57m-data-protection-watchdog-over-gdpr-violations`". Accessed June 7, 2021.

[3] Mongodb. "`https://www.mongodb.com/`". Accessed March 14, 2022).

[4] Redis. "`https://redis.io/`". Accessed March 14, 2022).

[5] D. Boneh, E. Kushilevitz, R. Ostrovsky, and W. E. Skeith. Public key encryption that allows pir queries. In *Annual International Cryptology Conference*, pages 50–67. Springer, 2007.

[6] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan. Private information retrieval. In *Proceedings of IEEE 36th Annual Foundations of Computer Science*, pages 41–50. IEEE, 1995.

[7] O. Goldreich and R. Ostrovsky. Software protection and simulation on oblivious rams. *Journal of the ACM (JACM)*, 43(3):431–473, 1996.

[8] P. Grubbs, A. Khandelwal, M.-S. Lacharité, L. Brown, L. Li, R. Agarwal, and T. Ristenpart. Pancake: Frequency smoothing for encrypted data stores. In *29th {USENIX} Security Symposium ({USENIX} Security 20)*, pages 2451–2468, 2020.

[9] M. S. Islam, M. Kuzu, and M. Kantarcioglu. Access pattern disclosure on searchable encryption: ramification, attack and mitigation. In *Ndss*, volume 20, page 12. Citeseer, 2012.

[10] E. Kushilevitz and R. Ostrovsky. Replication is not needed: Single database, computationally-private information retrieval. In *Proceedings 38th annual symposium on foundations of computer science*, pages 364–373. IEEE, 1997.

[11] H. Lipmaa and B. Zhang. Two new efficient pir-writing protocols. In *International Conference on Applied Cryptography and Network Security*, pages 438–455. Springer, 2010.

[12] A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pages 111–125. IEEE, 2008.

[13] A. Narayanan and V. Shmatikov. Myths and fallacies of" personally identifiable information". *Communications of the ACM*, 53(6):24–26, 2010.

[14] L. Ren, C. Fletcher, A. Kwon, E. Stefanov, E. Shi, M. Van Dijk, and S. Devadas. Constants count: Practical improvements to oblivious {RAM}. In *24th USENIX Security Symposium ({USENIX} Security 15)*, pages 415–430, 2015.

[15] C. Sahin, V. Zakhary, A. El Abbadi, H. Lin, and S. Tessaro. Taostore: Overcoming asynchronicity in oblivious data storage. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 198–217. IEEE, 2016.

[16] E. Stefanov, M. Van Dijk, E. Shi, C. Fletcher, L. Ren, X. Yu, and S. Devadas. Path oram: an extremely simple oblivious ram protocol. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 299–310, 2013.

[17] F. Wang, C. Yun, S. Goldwasser, V. Vaikuntanathan, and M. Zaharia. Splinter: Practical private queries on public data. In *14th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 17)*, pages 299–313, 2017.

[18] S. Yekhanin. Private information retrieval. In *Locally Decodable Codes and Private Information Retrieval Schemes*, pages 61–74. Springer, 2010.