The Complexity of Secure RAMs

Giuseppe Persiano[1,2(✉)]

[1] Università di Salerno, Fisciano, Italy
[2] Google, Mountain View, US
giuper@gmail.com

Abstract. In this invited lecture, I survey the recent results on the complexity of Oblivious RAMs and of related cryptographic data structures and highlight the proof techniques employed.

In recent years, there has been significant work in studying data structures that provide privacy for the operations that are executed. These primitives aim to guarantee that observable access patterns to physical memory do not reveal substantial information about the queries and updates executed on the data structure.

The concept of an Oblivious RAMs (ORAMs) has been introduced by Goldreich and Ostrovsky [6]. An ORAM can be viewed an a secure implementation of the simplest data structure: an *array* (or a RAM) whose entries can be read and over-written. The typical setting is that of a client that has limited memory and outsources the storage of the array to a remote server and accesses the data stored in the array over a network. Clearly, to protect the confidentiality of the data, each entry can be encrypted before the upload and decrypted once it is downloaded. Still, the server sees the access pattern and from this deduce the type of algorithm that is being executed which in turn can reveal the interest of the client. An ORAM is a protocol between the client and server that hides the access pattern. The *obliviousness* guarantee of an Oblivious RAM requires that no adversary that picks two *challenge* sequences of operations of the same length and observes the access pattern incurred by the execution of one of the sequences still cannot determine which of the two sequences gave rise to the access pattern observed.

In recent years, ORAMs have been studied extensively to try and determine the optimal overhead (see [3,6,7,9,11] and references therein) that was reduced from $O(\log^3 n)$ to $O(\log n)$, for a RAM with n entries. Indeed, for b-bit entries on a server with memory cell (word) size of $\omega = \Theta(b)$ bits, the best known construction obtains logarithmic overhead $O(b/\omega \cdot \log n)$ [1] and requires only constant client memory.

Is this the best we can do?

The first logarithmic lower bounds were proven by Goldreich and Ostrovsky [6] of the form $\Omega((b/\omega) \cdot (\log n / \log c))$ where the client has storage of c bits. Boyle and Naor [2] pointed out that these lower bounds assumed the so-called *balls-and-bins* model with a non-encoding assumption on the underlying blocks.

M. Mavronicolas (Ed.): CIAC 2023, LNCS 13898, pp. 17–20, 2023.
https://doi.org/10.1007/978-3-031-30448-4_2

Larsen and Nielsen [10] were the first to prove lower bounds for the general case; i.e., without making any encoding assumption. They proved that a RAM of n entries each of b bits implemented by a server with a memory consisting of ω-bit words and a client with c bits of local memory requires $\Omega((b/\omega) \cdot \log(nb/c))$. This bound becomes increasingly weak as ω grows and Komargodski and Lin [8] proved a lower bound of $\Omega(\log(nb/c)/\log(\omega/b))$ for the case $\omega > b$.

In the hope of obtaining faster RAM that would still offer an adequate level of security, researchers have looked at weaker but still meaningful notions of security. In this talk we will overview three attempts and show that indeed any meaningful notion of security for RAMs seems to be as hard as Obliviousness.

Differentially Private RAMs. In various practical applications, including the field of privacy-preserving data analysis, the notion of *Differential Privacy* [5] is considered to offer an adequate level of protection. Differentially Private RAMs (DPRAMs) aim to provide privacy for individual operations, but may reveal information about a sequence consisting of many operations. In more detail, if an adversary receives two candidate equal-length operational sequences that differ in one operation and the access pattern incurred by the execution of one of the two sequences, the adversary should not be able to guess the identity of the executed sequence with too high probability. Unfortunately, DPRAMs incur in the same overhead as ORAM. Specifically, the $\Omega(b/\omega \cdot \log nb/c)$ lower bound for DPRAMs by Persiano and Yeo [15] showed that this is impossible when $b = \Omega(\omega)$ and, recently, this has been extended to $\Omega(\log(nb/c)/\log(\omega/b))$ which is significant for the case $\omega > b$ by [16].

Leaky RAMs. A second approach allows the RAM to leak some partial information about the sequence of operations. Currently, all known leaky RAMs with constant overhead reveal if two operations are performed on the same key or not. We denote this as *global key−equality pattern*. The result of [12] gives strong evidence that the leakage of the global key-equality pattern is inherent for any leaky RAM construction with $O(1)$ efficiency. In particular, they consider the slightly smaller leakage of *decoupled key-equality pattern* where leakage of key-equality between update and query operations is decoupled and the adversary only learns whether two operations of the *same type* are performed on the same key or not. They show that any leaky RAM with at most decoupled key-equality pattern leakage incurs $\Omega(b/w \cdot \log n)$ overhead.

Snapshot Adversaries. In some applications the server executing the access is not trusted but it could be temporarily compromised by an external adversary. Very recently, Du, Genkin and Grubbs [4] presented an ORAM construction with $O(\log \ell)$ overhead protecting against a *snapshot* adversary that observes the transcript of ℓ consecutive operations from a single breach. For small values of ℓ, this outperforms standard ORAMs. However, if one allows to have 3 breaches, it has been recently proved [14] that we go back to $\Omega(b/w \cdot \log(nb/c))$ overhead.

Open Problem. The following question is thus still open. Is there a *meaningful* notion of security for which RAMs require a sub-logarithmic, or maybe even constant, overhead?

Also, it would be interesting to look at different data structures. The research reported in [16] has a general framework to prove lower bounds for more sophisticated data structures.

Acknowledgments. Most of the work discussed in this invited lecture is co-authored with Sarvar Patel and Kevin Yeo.

References

1. Asharov, G., Komargodski, I., Lin, W.-K., Nayak, K., Peserico, E., Shi, E.: OptORAMa: optimal oblivious RAM. In: Canteaut, A., Ishai, Y. (eds.) EUROCRYPT 2020, Part II. LNCS, vol. 12106, pp. 403–432. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-45724-2_14
2. Boyle, E., Naor, M.: Is there an oblivious RAM lower bound?. In: Sudan, M., (ed.) ITCS 2016, pp. 357–368. ACM (2016)
3. Devadas, S., van Dijk, M., Fletcher, C.W., Ren, L., Shi, E., Wichs, D.: Onion ORAM: a constant bandwidth blowup oblivious RAM. In: Kushilevitz, E., Malkin, T. (eds.) TCC 2016, Part II. LNCS, vol. 9563, pp. 145–174. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49099-0_6
4. Du, Y., Genkin, D., Grubbs, P.: Snapshot-oblivious RAMs: sub-logarithmic efficiency for short transcripts. In: Dodis, Y., Shrimpton, T. (eds.) CRYPTO 2022. Lecture Notes in Computer Science, vol. 13510, pp. 152–181. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-15985-5_6
5. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 265–284. Springer, Heidelberg (2006). https://doi.org/10.1007/11681878_14
6. Goldreich, O., Ostrovsky, R.: Software protection and simulation on oblivious RAMs. J. ACM (JACM) **43**(3), 431–473 (1996)
7. Goodrich, M.T., Mitzenmacher, M., Ohrimenko, O., Tamassia, R.: Privacy-preserving group data access via stateless oblivious RAM simulation. In: Rabani, Y. (ed.), 23rd SODA, pp. 157–167. ACM-SIAM (2012)
8. Komargodski, I., Lin, W.-K.: A logarithmic lower bound for oblivious RAM (for all parameters). In: Malkin, T., Peikert, C. (eds.) CRYPTO 2021, Part IV. LNCS, vol. 12828, pp. 579–609. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-84259-8_20
9. Kushilevitz, E., Lu, S., Ostrovsky, R.: On the (in)security of hash-based oblivious RAM and a new balancing scheme. In: Rabani, Y. (ed.), 23rd SODA, pp. 143–156. ACM-SIAM (2012)
10. Larsen, K.G., Nielsen, J.B.: Yes, there is an oblivious RAM lower bound! In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part II. LNCS, vol. 10992, pp. 523–542. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-96881-0_18
11. Patel, S., Persiano, G., Raykova, M., Yeo, K.: PanORAMa: oblivious RAM with logarithmic overhead. In: Thorup, M. (ed.) 59th FOCS, pp. 871–882. IEEE Computer Society Press (2018)
12. Patel, S., Persiano, G., Yeo, K.: Lower bounds for encrypted multi-maps and searchable encryption in the leakage cell probe model. In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020, Part I. LNCS, vol. 12170, pp. 433–463. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-56784-2_15

13. Persiano, G., Yeo, K.: Limits of preprocessing for single-server PIR. In: SODA 2022, SIAM, pp. 2522–2548 (2022)
14. Persiano, G., Yeo, K.: Limits of breach-resistant and snapshot-oblivious RAMs. Unpublished manuscript (2023)
15. Persiano, G., Yeo, K.: Lower bounds for differentially private RAMs. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019, Part I. LNCS, vol. 11476, pp. 404–434. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-17653-2_14
16. Persiano, G., Yeo, K.: Lower bound framework for differentially private and oblivious data structures. EUROCRYPT (2023)