Oblivious Group Storage

Nelly Fazio*‡, Antonio R. Nicolosi†, and Irippuge Milinda Perera‡

*The City College of CUNY
fazio@cs.ccny.cuny.edu
†Stevens Institute of Technology
nicolosi@cs.stevens.edu
‡The Graduate Center of CUNY
{nfazio,iperera}@gradcenter.cuny.edu

***Abstract*—We propose Oblivious Group Storage (OGS), an extension of the oblivious storage (OS) model [1] to the setting of group data sharing. In addition to providing server-side data-secrecy and access-pattern obliviousness guarantees as in traditional OS protocols, an OGS protocol allows arbitrary subsets of clients to share read/write access to cloud-stored data while preserving data-secrecy and access-pattern obliviousness guarantees with respect to unauthorized clients. To show the feasibility of OGS, we describe a provably secure generic construction based on the notions of *outsider-anonymous broadcast encryption* [2] and *multi-user oblivious RAM* [3].**

***Index Terms*—Cloud storage, data sharing, oblivious storage.**

I. Introduction

Recent developments in cloud computing has given rise to the convenience of remote storage services, a.k.a. cloud storage. Taking advantage of economies of scale, third party companies are now able to provide cloud storage with high availability and low latency for pay-per-use cost structures that amount to a couple of pennies per year per gigabyte stored [4]–[6]. Instead of enduring the cost and the hassle of managing in-house storage infrastructure, more and more individuals and companies alike are relying on cloud storage to manage large amounts of data they generate.

One of the biggest concerns with outsourcing data storage to a cloud is the confidentiality of the outsourced data. Most security-conscious clients encrypt their data before exporting to a cloud. Although it preserves the secrecy of the data, encryption alone cannot hide the access patterns between the clients and the storage providers. Indeed, prior work has shown that in certain situations, these access patterns can reveal a considerable amount of sensitive information regarding the outsourced data [7]–[9]. For example, Islam et al. [9] demonstrated that in the case of an encrypted email repository, statistical attacks on access patterns can recover about 80% of the search queries. Thus, to protect outsourced data effectively, one ought also to obfuscate the access patterns between the clients and their cloud storage providers. These requirements have been formalized in the cryptographic literature under the term Oblivious Storage (OS) [1], [8], [10]–[29].

An attractive benefit of having the data outsourced to a cloud is the ability to easily share that data with other clients. In fact, most of the popular cloud storage providers nowadays provide some mechanism, such as temporary links [30], to facilitate data sharing by their clients. However, enabling group access to outsourced data while also preserving the oblivious storage guarantees is still a problem not well understood, let alone solved. There have been several proposals, however, to provide oblivious group data access to cloud storage [3], [13], [18], [24]. All these proposals only support all-or-nothing sharing of the outsourced content, as they all require (either explicitly [13], [18], [24] or implicitly [3]) the authorized clients to share the secret keys to the cloud storage. Real-world clients, however, may have more nuanced sharing needs, e.g., sending party pictures to recent acquaintances without having to share an entire album.

To illustrate the need for restricting access to shared data to subsets of clients, consider a hospital that wishes to outsource the storage of electronic medical records (EMRs) to a third-party cloud storage provider. Since encryption alone is not enough to protect EMRs from the prying eyes of a motivated attacker who might have gained control of the cloud storage server [7]–[9], the hospital may decide to employ an OS protocol to obfuscate accesses to the cloud storage. However, with any of the existing OS protocols that support some form of cloud storage sharing, every member of the hospital staff requiring access to the cloud storage (e.g., a nurse who needs to update the chart of a patient in his care) will be able to pull the EMR of *any* patient of their choosing. This would result in a serious violation of federal statutes at protection of patient privacy [31], which mandate that a patient's medical record only be accessible to the medical staff involved in the patient's care or in strictly related matters (e.g., billing). A second, equally alarming concern with this approach is that compromise of the access credentials of a single member of the hospital staff could lead to the unauthorized disclosure of the EMRs of all the patients of the hospital. Clearly, the security guarantees of existing OS systems are inadequate for the requirements of this setting, which call for the ability to control access to the outsourced medical data at the record level.

II. Our contributions

We propose Oblivious Group Storage (OGS), an extension of the OS model to support data sharing by multiple clients at

TABLE I
COMPARISON OF OBLIVIOUS STORAGE PROTOCOLS IN TERMS OF THE EXTENT TO WHICH THEY SUPPORT SHARING OF STORAGE BY MULTIPLE CLIENTS.

	Stateful ORAM	Stateless ORAM	M-ORAM	OGS
Supports cloud storage sharing by mutually trusting clients.	✗	✓	✓	✓
Hides a client's access patterns from an attacker controlling *either* the server *or* unauthorized clients.	✗	✗	✓	✓
Hides a client's access patterns from an attacker controlling *both* the server *and* unauthorized clients.	✗	✗	✓	✓
Hides a client's data from unauthorized clients.	✗	✗	✗	✓

the granularity of individual data items. In addition to providing server-side data-secrecy and access-pattern obliviousness guarantees as in traditional OS protocols, an OGS protocol allows arbitrary subsets of clients to share read/write access to cloud-stored data while preserving data-secrecy and access-pattern obliviousness guarantees with respect to unauthorized clients. Table I shows how OGS compares to other types of oblivious storage protocols in terms of support for data sharing by multiple clients. To the best of our knowledge, OGS is the first proposal of an OS-like system capable of affording clients fine-grained control over who can access their data.

At a technical level, our first contribution is the formalization of a security model for the setting, so that one can argue about oblivious shared data access without ambiguity. A crucial aspect of our OGS model is the inclusion of a sequence of proxies to mediate access by the clients to the cloud storage provider—an idea originally proposed by Jinsheng et al. [3] in the context of multi-user oblivious RAM (M-ORAM). Second, we provide a generic construction of OGS based on M-ORAM and outsider-anonymous broadcast encryption (oABE) [2], along with a formal analysis of its security with respect to semi-honest adversaries.

REFERENCES

[1] P. Williams and R. Sion, "Usable PIR," in *Network and Distributed System Security Symposium—NDSS*, 2008.

[2] N. Fazio and I. M. Perera, "Outsider-anonymous broadcast encryption with sublinear ciphertexts," in *Public Key Cryptography—PKC*, 2012, pp. 225–242.

[3] Z. Jinsheng, Z. Wensheng, and D. Qiao, "A multi-user oblivious ram for outsourced data," Manuscript, 2014.

[4] Google Inc., "Google Cloud Platform," https://cloud.google.com/.

[5] Amazon.com Inc., "Amazon S3," http://aws.amazon.com/s3/.

[6] Microsoft Corp., "Microsoft Azure," http://azure.microsoft.com/en-us/.

[7] S. Chen, R. Wang, X. Wang, and K. Zhang, "Side-channel leaks in web applications: A reality today, a challenge tomorrow," in *IEEE Symposium on Security and Privacy—IEEE S&P*, 2010, pp. 191–206.

[8] B. Pinkas and T. Reinman, "Oblivious RAM revisited," in *Advances in Cryptology—CRYPTO*, 2010, pp. 502–519.

[9] M. S. Islam, M. Kuzu, and M. Kantarcioglu, "Access pattern disclosure on searchable encryption: Ramification, attack and mitigation," in *Network and Distributed System Security Symposium—NDSS*, 2012.

[10] M. T. Goodrich and M. Mitzenmacher, "Privacy-preserving access of outsourced data via oblivious RAM simulation," in *Automata, Languages and Programming—ICALP*, 2011, pp. 576–587.

[11] E. Shi, T.-H. Chan, E. Stefanov, and M. Li, "Oblivious RAM with $o((logn)^3)$ worst-case cost," in *Advances in Cryptology—ASIACRYPT*, 2011, pp. 197–214.

[12] D. Boneh, D. Mazieres, and R. A. Popa, "Remote oblivious storage: Making oblivious RAM practical," Manuscript, 2011, http://dspace.mit.edu/handle/1721.1/62006.

[13] M. T. Goodrich, M. Mitzenmacher, O. Ohrimenko, and R. Tamassia, "Privacy-preserving group data access via stateless oblivious ram simulation," in *ACM-SIAM Symposium on Discrete Algorithms—SODA*, 2012, pp. 157–167.

[14] E. Stefanov, E. Shi, and D. X. Song, "Towards practical oblivious RAM," in *Network and Distributed System Security Symposium—NDSS*, 2012.

[15] E. Stefanov and E. Shi, "Multi-cloud oblivious storage," in *ACM Conference on Computer and Communications Security—CCS*, 2013, pp. 247–258.

[16] ——, "Oblivistore: High performance oblivious cloud storage," in *IEEE Symposium on Security and Privacy—IEEE S&P*, 2013, pp. 253–267.

[17] J. Zhang, W. Zhang, and D. Qiao, "S-oram: A segmentation-based oblivious ram," in *ACM Symposium on Information, Computer and Communications Security—ASIACCS*, 2014, pp. 147–158.

[18] E. Boyle, K. Chung, and R. Pass, "Oblivious parallel RAM," Cryptology ePrint Archive, Report 2014/594, 2014.

[19] M. T. Goodrich, M. Mitzenmacher, O. Ohrimenko, and R. Tamassia, "Oblivious RAM simulation with efficient worst-case access overhead," in *ACM Workshop on Cloud Computing Security Workshop—CCSW*, 2011, pp. 95–100.

[20] ——, "Practical oblivious storage," in *ACM Conference on Data and Application Security and Privacy—CODASPY*, 2012, pp. 13–24.

[21] E. Kushilevitz, S. Lu, and R. Ostrovsky, "On the (in)security of hash-based oblivious RAM and a new balancing scheme," in *ACM-SIAM Symposium on Discrete Algorithms—SODA*, 2012, pp. 143–156.

[22] P. Williams and R. Sion, "Single round access privacy on outsourced storage," in *ACM Conference on Computer and Communications Security—CCS*, 2012, pp. 293–304.

[23] P. Williams, R. Sion, and B. Carbunar, "Building castles out of mud: Practical access pattern privacy and correctness on untrusted storage," in *ACM Conference on Computer and Communications Security—CCS*, 2008, pp. 139–148.

[24] P. Williams, R. Sion, and A. Tomescu, "Privatefs: A parallel oblivious file system," in *ACM Conference on Computer and Communications Security—CCS*, 2012, pp. 977–988.

[25] S. Lu and R. Ostrovsky, "Distributed oblivious RAM for secure two-party computation," in *Theory of Cryptography—TCC*, 2013, pp. 377–396.

[26] M. Ajtai, "Oblivious RAMs without cryptographic assumptions," in *ACM Symposium on Theory of Computing—STOC*, 2010, pp. 181–190.

[27] I. Damgård, S. Meldgaard, and J. Nielsen, "Perfectly secure oblivious RAM without random oracles," in *Theory of Cryptography—TCC*, 2011, pp. 144–163.

[28] K.-M. Chung, Z. Liu, and R. Pass, "Statistically-secure ORAM with $\tilde{O}(\log^2 n)$ overhead," in *Advances in Cryptology—ASIACRYPT*, 2014, pp. 62–81.

[29] 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 *ACM Conference on Computer and Communications Security—CCS*, 2013, pp. 299–310.

[30] Dropbox Inc., "Sharing files and folders," https://www.dropbox.com/help/category/Sharing.

[31] Centers for Medicare & Medicaid Services, "The Health Insurance Portability and Accountability Act of 1996 (HIPAA)," 1996, http://www.cms.hhs.gov/hipaa/.