Client-side encryption for privacy-sensitive applications on the cloud

Stefano M P C Souzaa,*, Ricardo S Puttiniα

*Universidade de Brasília, Brasília-DF 70910-900, Brazil

Abstract

There are important concerns when trusting sensitive information to the cloud. Health and financial records, for instance, suffer strict legal restrictions to data escrow. Organizations holding such information need to assure end-users and authorities that a third party will never access restricted data. Client-side encryption is a common solution in literature. Most works fail, however, to reason the impact of security solutions on performance and usability. Homomorphic and order preserving encryption systems can mitigate such negative impacts, as they allow the computation of regular searches over encrypted records on the cloud, while preserving information confidentiality and the privacy of end-users.

Keywords: Cloud computing; security; privacy; client-side encryption; homomorphic encryption

1. Introduction

Cloud computing is a business model by which pooled computational resources are provisioned, on demand and with rapid elasticity, through a broadband network, in the form of a metered service. The main economic appeal of such services is that consumers are able to turn huge capital costs into a smaller and more flexible operational defrayal, pushing concerns with ownership and maintenance of the underlying infrastructure supporting their computing and communication systems to the service provider.

To reason the security of assets on the cloud, one must select a stakeholders’ perspective or needs. Providers need to efficiently charge for any access to cloud resources. Consumers need assurances about the control over their assets, stability of services and predictability in business conditions. End-users need privacy; they need accurate information and a good sense of control on who has access to their private information. No realistic security solution can be designed to respond to all needs at once.

Also, there are different service models and contract options that render different levels of control and responsibility over systems security to cloud consumers and providers. Infrastructure-as-a-Service consumers control VMs (virtual machines), operational systems, libraries and network configuration, and, thus, are able to implement sophisticated

* Corresponding author. Tel.: +55-61-982623011.
E-mail address: stefano@stm.gov.br
security techniques, such as VM nesting, memory and processes taint analysis. The same is not true to Platform-as-a-Service or Software-as-a-Service consumers.

There is an important trade-off here: the higher the complexity of cloud services, the lesser the flexibility and effective control over the assets for the consumer and, therefore, the greater the importance of the provider in information security. In most cases, the provider, as a larger organization, with access to cutting edge technology and highly qualified personnel, will be in a better position to deal with security. Consumers, therefore, are better off pushing security concerns and costs to the provider. In some cases, however, consumers are bound by law to guarantee themselves the integrity, confidentiality and privacy of information they hold.

Health care providers, for instance, are held responsible for the security of health records and their patient’s privacy. That is, records under their guard cannot be disclosed to any third parties, and, in the case that they are, cannot be used to identify a person. Such restrictions do not only apply to those records comprising health information, but also those that reveal the payment for the provision of any kind of health services. A hospital manager cannot simply respond to government officials saying that the cloud provider is HIPAA compliant and, therefore, he could share the escrow of the Electronic Health Records under his responsibility. The credit card operator, in the same way, cannot share a card holder’s payment history on the grounds that the provider is PCI compliant.

The provider will always have privileged access to every part of his service infrastructure, and a curious provider could misuse the virtualization and provisioning basic software stack to capitalize on eventual access to consumer data. The ‘2015 Information Security Breaches Survey’, a comprehensive study from the UK’s government, shows that 75% of large corporations in the country had security events related to internal personnel in that year. Despite the solid security and audit processes usually in place, large cloud providers are still vulnerable to the ‘insider threat’, thus cloud consumers are always at risk of leakage of data transported, processed or stored on the cloud.

Hence, with little control over environment configuration and the resulting security of processes, and little or no access to credible auditing tools, the average consumer can only control what is delivered to the cloud. That makes client-side encryption an important topic. The intuition is that, if every piece of data is encrypted at client-side in a way that even an attacker with enormous computing power cannot break information confidentiality, or end-user’s privacy, then the use of a cloud service does not impact information escrow policies.

For simple reading and writing to the cloud, the best fit is a fast, well tested and standardized symmetric encryption system. Along with the system used for storage, homomorphic and OPE (order-preserving encryption) systems can be used to provide search indexes and anonymous aggregate information – such as modes, means and other measures of central tendency. A careful combination of different schemes, each enabling a specific computation or feature, can be used to perform regular searches and the most commonly needed operations over the encrypted data on the cloud, revealing no relevant information to an eventual eavesdropper.

2. Related work

There is an intense work, both in industry and academia, to formulate practical solutions with reasonable security for cloud applications. Some works present ways to enhance the auditability of the cloud service software. The strongest trend, however, is to assume the cloud is unsafe and propose a form of secure delegation of computation. Secure multi-party computation protocols have been around for over thirty years now: some with provable security in the semi-honest setting, which fits the cloud scenario. Nevertheless, there are still very few efficient and broadly applicable implementations.

Client-side encryption, using homomorphic cryptography, may represent a simpler alternative. The non-interactive nature of homomorphic encryption solutions results in more efficient systems, not bound by network latency nor by the client’s thin hardware. Also, there are already many practical applications in literature. The system of Chase and Lauter, for instance, uses an anonymous credentials system, in the form of cryptographic tokens, to allow the exchange of data or endorsement messages between different agents in the health care industry, disclosing no information other than the strictly necessary for that single interaction.

---

a Health Insurance Portability and Accountability Act, a US law enforcing standards for electronic health record transactions
b Payment Card Industry Data Security Standard, a information security standard for credit cards and payment accounts
Bos et al use homomorphic encryption schemes for the execution of machine learning algorithms over encrypted records on the cloud. Their system uses a previously built classification model to process new encrypted observations sent to the cloud server. The classifier yields an encrypted result that can only be decrypted at the client that generated the query. Lauter and co-workers showed many examples of statistical and learning algorithms running over encrypted data on the cloud.

Those are very specialized applications, running complex algorithms and using complex homomorphic encryption schemes and protocols. The average consumer, however, needs simpler tools, leveraging industry level cryptographic libraries or, at most, implementing simpler cryptosystems. This orientation, towards solutions for common web application features, is the intuition behind works such as CryptDB and Mylar.

CryptDB is an extension to the MySQL client connector that allows the transformation of regular SQL queries so that they can run on encrypted databases. While it is a very innovative and practical solution, it still requires the consumer to run the web application on a trusted server, outsourcing only the database to the cloud. Mylar is an extension to Meteor (a web and mobile applications framework) that brings encryption functions to the client-side. It features a multi-key searchable encryption scheme that enables private communication and keyword search for end-users, while cloud servers receive only encrypted data.

Searchable or pattern matching encryption very well suits an array of small to mid-sized applications: messaging, chats, task lists, among others. Most corporate systems, nonetheless, deal with structured and numerical data. In such cases, range queries, modes, means, averages and simple arithmetic operations are constantly computed over large portions of the database – at the server side.

3. Homomorphic encryption at client-side

There are many partially or somewhat homomorphic encryption systems that enable such computations. The Pailler system and Benaloh’s DPE (Dense Probabilistic Encryption) schemes provide additively-homomorphic operations over the ciphertexts. ElGamal and unpadded RSA schemes allow multiplication. The somewhat homomorphic system of Boneh, Goh and Nissim allows one multiplication and many additions. The system of Naehrig, Lauter and Vaikuntanathan, based on a LWE (Learning With Error) primitive, allows a limited number of both additions and multiplications.

OPE schemes provide indexes for range queries and comparisons among encrypted values. These schemes are designed to leak some information on the encrypted values. The proportion of bits revealed usually varies with the cost of encryption. The system of Boldyreva at al, for instance, is very fast and has many open-source implementations available, but is known to leak even the order of magnitude of encrypted values – depending on the number of plaintexts encrypted with the same key, their range and dispersion. The OPE protocol of Popa et al leaks no information besides the order of queries to the database, but, for every new encryption, requires a number of accesses to the server logarithmic in the number of previously encrypted values – with each access followed by one AES decryption.

There is a clear use case for homomorphic and order-preserving systems in Electronic Health Records: an OPE scheme can be used to provide indexes to the types of observations (e.g. body measurements, medicine prescriptions, exams, diseases) in each record and to provide range queries over the magnitudes of such observations. That allows a doctor to use the computing power of the cloud to select records of interest in the daily use of the EHR system. It also allows a researcher to safely compute modes and a few statistical tests over the encrypted records on the cloud. An additively homomorphic system can be used to compute means over the encrypted values, while preserving patients privacy and the confidentiality requirements of health data. The Pailler system, for example, has been used as a building block of a protocol proven to be efficient in the computation of ridge regression over hundreds of millions of records on the cloud.

The right choice for any cloud application is really a matter of matching functional requirements and the operations provided by each cryptosystem. Now, when designing a cloud application, the consumer must evaluate the nature of data that will be stored and processed on the cloud. The consumer needs a clear discernment of the relationship between the assets to be protected and technically feasible and economically relevant threats in order to evaluate what kind of impacts a security breach may bring to end-users and the business.
If client-side encryption emerges from this process as a business requirement, then the engineers designing the application must assess the minimum set of computations required to be performed on the cloud, and select cryptographic primitives that will enable such computations under acceptable security parameters. One may argue that this kind of understanding and familiarity with different cryptographic schemes is not a common skill among average professionals. But the evolution of cloud computing security, both in terms of academic knowledge and industry-ready products diversification and maturation, will eventually bring about this culture. The same happened to the encryption and digital signature primitives wildly adopted in industry nowadays.

4. Conclusion

Client-side encryption with homomorphic systems does not constitute a ‘one-size-fits-all’ type of solution. Nevertheless, given the inherited risks of cloud services and the existence of special use cases, as the ones previously mentioned, where the only way to outsource computing power is to have all data encrypted beforehand, it is possible to state that the way ahead in cloud security certainly incorporates related techniques. Therefore, there remains a clear need for the development, consistent trying and wide application of simple and efficient homomorphic cryptographic primitives, as well as the need for the development of open and free software projects with carefully curated libraries of cryptographic functions derived from such primitives.

There are many examples of successful and well established projects, such as OpenSSL, GnuCrypto, BouncyCastle and LibTomCrypt – to name a few – that had a great impact on the dissemination and use of cryptographic primitives on day-to-day computing. Libraries, API’s, and frameworks, such as Mylar, can bridge the gap, bringing strong cryptographic functions to common programmers and improving the overall quality of software projects. That would greatly contribute to the advancement of security and privacy in the cloud computing scenario.

References

1. The NIST Definition of Cloud Computing; 2011. URL: http://dx.doi.org/10.6028/NIST.SP.800-145.
2. Payne, B.D.. Reducing the cost of security in the cloud. In: Proceedings of the 6th Edition of the ACM Workshop on Cloud Computing Security; CCS’14. New York, NY, USA: ACM. ISBN 978-1-4503-3239-2; 2014, p. 5–6.
3. Jansen, W., Grance, T.. Guidelines on Security and Privacy in Public Cloud Computing; 2011. URL: http://dx.doi.org/10.6028/NIST.SP.800-144.
4. Liu, C., Mao, Y.. Inception: Towards a nested cloud architecture. In: Proceedings of the 5th USENIX Workshop on Hot Topics in Cloud Computing · HotCloud’13. USENIX; 2013.
5. Andrew Miller, R.H., Potter, C.. 2015 information security breaches survey. Tech. Rep.; 2015.
6. Kerschbaum, F.. Client-controlled cloud encryption. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security; CCS’14. New York, NY, USA: ACM. ISBN 978-1-4503-2957-6; 2014, p. 1542–1543.
7. Bleikeritz, S., Schunter, M., Probst, C.W., Pendarakis, D., Eriksson, K.. Security audits of multi-tier virtual infrastructures in public infrastructure clouds. In: CCSW’10. 2010.
8. Kerschbaum, F.. Privacy-Preserving Computation. Berlin, Heidelberg: Springer Berlin Heidelberg; 2014, p. 41–54.
9. Chase, M., Lauter, K.. An anonymous health care system. In: USENIX Security Symposium. 2010.
10. Bos, J.W., Lauter, K., Naehrig, M.. Private predictive analysis on encrypted medical data. Cryptology ePrint Archive, Report 2014/336; 2014. URL: http://eprint.iacr.org/2014/336.
11. Lauter, K., Lopez-Alt, A., Naehrig, M.. Private computation on encrypted genomic data. Cryptology ePrint Archive, Report 2015/133; 2015. http://eprint.iacr.org/2015/133.
12. Popa, R.A., Redfield, C.M.S., Zeldovich, N., Balakrishnan, H.. Cryptdb: Protecting confidentiality with encrypted query processing. In: Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles; SOSP ’11. New York, NY, USA: ACM. ISBN 978-1-4503-0977-6; 2011, p. 85–100. URL: http://doi.acm.org/10.1145/2043566.2043566. doi:10.1145/2043566.2043566.
13. Popa, R.A., Stark, E., Helfer, J., Valdez, S., Zeldovich, N., Kaashoek, M.F., et al. Building web applications on top of encrypted data using mylar. 11th USENIX Symposium on Networked Systems Design and Implementation 2014.
14. Naveed, M., Prabhakaran, M., Gunter, C.. Dynamic searchable encryption via blind storage. In: Security and Privacy (SP), 2014 IEEE Symposium on. 2014, p. 639–654. doi:10.1109/SP.2014.47.
15. Paillier, P.. Public-key cryptosystems based on composite degree residuosity classes. In: IN ADVANCES IN CRYPTOLOGY - EUROCRYPT 1999. Springer-Verlag; 1999, p. 223–238.
16. Fousse, L., Lafourcade, P., Alnouaim, M.. Benaloh’s dense probabilistic encryption revisited. CoRR 2010;abs/1008.2991. URL: http://arxiv.org/abs/1008.2991.
17. Boneh, D., Goh, E.J., Nissim, K.. Evaluating 2-dnf formulas on ciphertexts. In: Proceedings of the Second International Conference on Theory of Cryptography; TCC’05. Berlin, Heidelberg: Springer-Verlag. ISBN 3-540-24573-1, 978-3-540-24573-5; 2005, p. 325–341.
18. Naehrig, M., Lauter, K., Vaikuntanathan, V.. Can homomorphic encryption be practical? In: Proceedings of the 3rd ACM Workshop on Cloud Computing Security Workshop; CCSW ’11. New York, NY, USA: ACM. ISBN 978-1-4503-1004-8; 2011, p. 113–124.
19. Boldyreva, A., Chenette, N., Lee, Y., O’Neill, A. Order-preserving symmetric encryption. In: *Proceedings of the 28th Annual International Conference on Advances in Cryptology: The Theory and Applications of Cryptographic Techniques*; EUROCRYPT ’09. Berlin, Heidelberg: Springer-Verlag. ISBN 978-3-642-01000-2; 2009, p. 224–241.

20. Popa, R.A., Li, F.H., Zeldovich, N.. An ideal-security protocol for order-preserving encoding. Cryptology ePrint Archive, Report 2013/129; 2013. [http://eprint.iacr.org/2013/129](http://eprint.iacr.org/2013/129).

21. Souza, S.M.P.C.. *Safe-Record: segurança e privacidade para Registros Eletrônicos em Saúde na nuvem*. Master’s thesis; Universidade de Brasília; 2016.

22. Nikolaenko, V., Weinsberg, U., Ioannidis, S., Joye, M., Boneh, D., Taft, N.. Privacy-preserving ridge regression on hundreds of millions of records. In: *Security and Privacy (SP), 2013 IEEE Symposium on*. 2013, p. 334–348. doi:10.1109/SP.2013.30.