I. Agudo, et al., "Cryptography Goes to the Cloud", *1st International Workshop on Security and Trust for Applications in Virtualised Environments (STAVE 2011)*, Communications in Computer and Information Science vol. 187, pp. 190-197, 2011.
http://doi.org/10.1007/978-3-642-22365-5_23
NICS Lab. Publications: https://www.nics.uma.es/publications

Cryptography goes to the Cloud

Isaac Agudo [1], David Nuñez [1], Gabriele Giammatteo [2], Panagiotis Rizomiliotis [3], Costas Lambrinoudakis [4]

[1] Department of Computer Science, E.T.S. Ingeniería Informática,
University of Málaga, E-29071 Málaga, Spain
[2] Research and Development Laboratory, Engineering Ingegneria Informatica S.p.A.
Via Riccardo Morandi, 32 00148 Roma - Italy
[3] Laboratory of Information and Communication Systems Security,
Department of Information and Communication Systems Engineering,
University of the Aegean Samos, GR-83200, Greece
[4] Department of Digital Systems, University of Piraeus,
Piraeus, Greece
{isaac,dnunez}@lcc.uma.es, gabriele.giammatteo@eng.it, prizomil@aegean.gr, clam@unipi.gr

Abstract. In this paper we identify some areas where cryptography can help a rapid adoption of cloud computing. Although secure storage has already captured the attention of many cloud providers, offering a higher level of protection for their customer's data, we think that more advanced techniques such as searchable encryption and secure outsourced computation will become popular in the near future, opening the doors of the Cloud to customers with higher security requirements.

Keywords: Cloud Computing, Searchable Encryption, Secure Storage.

1 Introduction

The wide adoption of cloud computing is raising several concerns about treatment of data in the cloud. Advantages of cloud storage are enormous: I) ubiquitous access: anywhere, anyhow, anytime access to your data, ii) high reliability, iii) resilience and iv) scalability, v) cost efficiency. But, unfortunately, to date, also several security and legal risks should be considered: i) unauthorized access, ii) sensitive data disclosure, iii) IPR protection, iv) communication threats and loads in transferring data, v) data integrity. As stated in [17][18], cloud data security is the most worrying issue of cloud technology and before enterprises or public authorities will fully outsource their data management to cloud vendors replacing their internal facilities, security has to improve to acceptable levels. This is also the opinion of the European Commission [16].

Cryptography could help increasing adoption of Cloud Computing by skeptic or more security concerned companies. The first level of security where cryptography can help Cloud computing is secure storage and this is the focus of Section 2. There

are already some cloud providers that have started providing secure storage services but offering different levels of protection.

The major handicap of secure storage is that we cannot outsource the processing of this data without decrypting it before or without revealing the keys used for encryption. If we want companies to use the full potential of Cloud Computing we cannot restrict our effort to secure storage but also to secure processing or computation. The first steps in this direction are directed to enable search on encrypted data, see Section 2.2. This allows companies to store confidential information in the cloud whilst still being able to access it partially without having to decrypt it. In this model many companies could share a dataset, update and query it without leaking any information to the cloud provider. By using these techniques, governments could store their citizen databases in the Cloud and access individual records without worrying about disclosure of personal information.

The last measure we discuss in this paper is Secure Computation (Section 3). By secure computation we understand not only that the Cloud Provider cannot get any confidential data used in the computation but also that the output of the computation is verifiable.

2 Secure Storage

The main concern around data storage is the protection of information from unauthorized access. In several usage scenarios the risk of data being disclosed, lost, corrupted or stolen is unacceptable. Until data is stored on resources owned, controlled and maintained by the data owner, the possibility of unauthorised access is reduced by any physical countermeasure or trust in authentication/authorisation mechanism put in place by him/her self (e.g. physically located in room at the establishment, behind closed doors and installing network firewalls and ACLs at software level).

Things radically change when moving from resources fully controlled by the data owner to resources administrated by third party entities like public clouds. Resources that sit outside the user's domain are resources not owned and not controlled by the user and even trusting the resources' provider, the risk that someone (e.g. an employee of the resources' vendor) can access and disclose/corrupt data is considerable. In the literature this risk is usually known as insider abuse or insider threat [18][19][20]. This is the major risk that, presently, is preventing the large adoption of cloud-based solutions by enterprises. Before companies move their data to the cloud, benefitting from the cloud storage advantages, all issues deriving from storing data on un-owned and un-trusted resources must be addressed, including the inconsistencies with this new model, put there by the legal frameworks.

The secure storage approach aims to avoid insider threats using encryption techniques to protect data from unauthorised access. The core concept of secure storage is the encryption of data in the trusted environment before sending it outside to the un-trusted cloud storage resource. There are a wide range of encryption algorithms at the cutting edge which have been proved to be secure that can be used

to perform encryption/decryption operations (e.g. AES, Serpent, Blowfish). Theoretically both symmetric and asymmetric algorithms can be used, but, since the latter are much slower than the former, for performance reasons symmetric algorithms are preferred. The usage of encryption as a technique to secure data guarantees the confidentiality of data and helps to detect any corruption in data.

The main issue in the secure storage approach is the management of encryption keys. In fact, once data is encrypted, keys become the true bits to protect! If keys were stored in the un-trusted environment along with data, an attacker could have at his/her disposal both data and the keys to decrypt the data, with disastrous consequences. Keys are stored on a *keystore* that can be implemented either on a i) portable device (e.g., an usb pen-drive) owned by the user who can plug it anywhere (within the trusted environment) or ii) in a specialised server which sits somewhere in the trusted environment. The last solution is the most flexible schema which would allow the user to access data from any location within the trusted environment and would also enable scenarios in which keys could be shared by multiple users (e.g. see Adi Shamir's *How to share a secret* [21] work on the sharing of keys).

A basic architecture of a secure storage system is presented in fig. 1: when the user wants to store data on an un-trusted resource: i) encrypts data at an encryption point (it may be either the user's machine or a service offered within the trusted environment), ii) encrypted data is sent to the storage service and iii) keys are stored on the keystore. Inverse procedure is applied when data must be accessed: it is transferred from the un-trusted storage to the encryption point, here keys are retrieved, the decryption takes place and original data is sent back to the user.

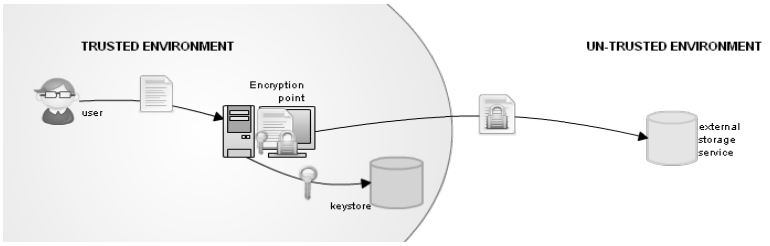

Fig. 1. Basic architecture of a secure storage system

Starting from this basic architecture a more complex one can be developed, for instance to replicate data on multiple storage services (even coming from different vendors) to augment data integrity and reliability in case of failure or corruption of one of the storages.

2.1 Cloud storage offerings

All commercial cloud providers offer to their customers at least one cloud storage service: Amazon' S3 and SimpleDB, Microsoft Windows Azure Storage services, Google App Engine Datastore, just to cite the most popular. All these products are very powerful for their scalability and storage capacity, but their security mechanisms

are based on authentication of users and the ACLs attached to the data stored. That said, these security mechanisms are too weak in a distributed and un-trusted environment since the risk of data disclosure and corruption still exists. In order to fill the gap between common storage services offered by cloud vendors and security requirements, a secure storage system can be built on top of bare cloud storage. Implementing such a system, all data stored on the cloud is encrypted and illegible, addressing, in this way, security requirements while still keeping the advantages of cloud storage.

Two good examples of commercial secure storage services which encrypt data client-side before transmitting it outside the user's machine (reasonably considered trusted) are: i) *Spideroak*, that is a cloud based backup and file synchronisation service which uses client-side encryption and implements a so called zero knowledge system where keys (neither the master passwords nor encryption keys) are never transmitted to the service's provider and ii) *GoldKey – Cloud Storage Security Key* with the peculiarity of storing keys on the GoldKey Token: a portable device very similar to a usb pen-drive instead of the user's machine. In contrast, an example of an ineffective secure storage service is *Dropbox*. It offers its users an online space to store data using Amazon's S3 as a back-end storage service. Users' data is encrypted before being stored on S3, but it has to be considered that Dropbox encrypts data in its servers keeping the keys in its servers. This, in fact, nullifies the added value of encryption since, unless for particular use cases, users consider Dropbox' servers as trusted as S3 servers.

Looking at the open source panorama, a promising project released under GNU License named Tahoe-LAFS aims to provide components to build a distributed secure storage system. Besides the encryption mechanism, it also implements an algorithm [22] that splits data in n chunks storing them on different nodes. At retrieval time only $m<n$ nodes are needed to rebuild the original data. This algorithm makes the system more resistant to server failures and/or attacks, by increasing data integrity assurance. A similar approach is used by EncryptMe: a proof of concept secure storage system realised for grid infrastructures [23].

2.2 Searchable Encryption

Searchable encryption is a broad concept that deals with searches in encrypted data. The goal is to outsource encrypted data and be able to conditionally retrieve or query data without having to decrypt all the data.

An interesting problem related with searchable encryption is Private Information Retrieval (PIR) [8][9]. The problem here is for the user to retrieve information from a database without revealing any information about the requested data to the server. Searchable encryption goes one step further by allowing the user not only to retrieve information privately but also to search it.

The first searchable encryption scheme was defined in 2000 [10]. It makes use of symmetric encryption and provides:

- query isolation for searches

- controlled searching
- hidden queries

Those three properties guarantee that the server is not able to learn anything more about the plaintext than the search result, it needs the user's authorisation to search for an arbitrary word and the user need not reveal the word they are searching for. There are some other symmetric schemes for searchable encryption that enhance security and efficiency [11].

Another approach for searchable encryption is to use asymmetric encryption. The first scheme for searchable encryption that makes use of public key cryptography is the Public-Key Encryption with keyword Search (PEKS) scheme, proposed by Boneh et al. [12]. By using public key cryptography it is possible for multiple persons to encrypt data but only the owner of the private key will be able to search for a keyword. This scheme has been enhanced in [13][14]

More advanced solutions also allow searching with wildcards [15]. Most of those schemes use *hidden vector encryption* (HVE). We can see HVE as an identity-based encryption where both the encryption and the decryption key are derived from a vector.

3 Secure outsourcing of computation

Computation outsourcing is applied when an entity has to perform a task that requires computation power and resources that the entity cannot dispose and the delegation of this task to an external provider is the only option. Secure outsourcing refers to such an outsourcing in which security requirements, and sometimes privacy requirements can be met. The main challenge of a secure outsourcing is to delegate a computation to a set of service providers that are either distrusted or partially trusted without exposing and revealing either the input data or the computed output. Moreover, the client wants to be able to verify the correctness of the result in order to trust the new data that they possess.

The concept of computation outsourcing can be seen as the core of the cloud computing model. However, such a model of outsourcing, i.e. a distribution of computations under rather loose restrictions cannot be applied in all cases. Indeed, there is a plethora of applications for which there are strong security and privacy requirements and there are companies and research institutes that avoid at any cost a Cloud infrastructure due to security scepticism. Hospitals and health service providers, clinical research institutes, stockbrokers and capital investment companies are just a few examples. It is clear that the design of secure outsourcing computation schemes is a very challenging research area.

3.1 Technical Solutions

Several secure computation outsourcing techniques have been proposed and there are classic results for the problems of secure two-party and multi-party computation. In most cases, the main idea is that the client performs some pre-processing over the

input data before sending it to the entity in charge of the computation. This pre-processing adds some structured randomness. After the computation some post-processing is required to remove the added randomness and reveal the final computation output. These technical solutions can be divided into two main categories. The first that a third trusted party (TTP) plays a vital role in the security and solutions so the presence of the TTP is not necessary. We will concentrate on the second category. The computing can be performed either by using special encrypted functions, called garbled circuits, or by using encrypted input data, with a special form of encryption, called homomorphic encryption.

The notion of garbled circuits was introduced by Yao [1]. Initially, the function to be computed is first encrypted by an entity, called the constructor, with symmetric cryptography. Then, another party, called the evaluator, decrypts the function using the keys that correspond to the input data. The use of symmetric encryption algorithms endows efficiency in terms of implementation to the garbled circuits. However, this procedure is one-time-pad like. That means that the garbled circuits can be used only once and their size is proportional to the size of the function to be computed. Several hardware implementations have been proposed to accelerate the procedure.

Homomorphic encryption has several applications including e-voting systems. It allows the computation of encrypted data with requiring any additional information. The first homomorphic encryption schemes were designed to perform specific operations (e.g., multiplications for RSA, additions for Paillier, or additions and one multiplication [4]), allowing the outsourcing of encryption or digital signing. These schemes are very useful for another critical cryptography related secure cloud challenge, the secure storage (see next section). However, over the last few years, fully homomorphic encryption schemes that have been proposed allow for arbitrary computations on encrypted data [5, 6]. However, these schemes are not yet practical, as shown in fully homomorphic encryption which is not yet efficient enough to be used in practical applications.

Fig. 1. The Trusted/Commodity Cloud Architecture. [7]

Several of the existing solutions have been adapted to the cloud model. The majority of these solutions are lacking either in generality or practicality and scalability [3]. As a general architecture, a multi-cloud approach has been proposed, i.e. two or more clouds that can be used for securing the outsource computation. More precisely, in one of proposals, a trusted cloud is responsible for the security critical operations, i.e. the pre-processing and post-processing encryption and decryption (see

Fig. 2). Then, the main operations can be performed by any untrusted cloud. In this approach, any existing MPC solution can be used, but there is no guarantee concerning the correctness of the result.

The second architecture aims to fill this gap. It is based on the cheap computational power that cloud offers, and it uses the core of error correction codes theory, i.e. redundancy. Instead of using only one cloud provider to perform a given computation, two or more different clouds are used. If at least one of them is honest, then the user can either verify the correctness of the result or identify an incorrect output.

As a direct generalisation of the problem of secure outsourcing one can look at the case where a group of clients, that trust each other, want to use a cloud based computation service that they do not fully trust. In this scenario, the proposed fully homomorphic encryption schemes do not apply [2] and additional assumptions are required.

4 Conclusions

We have reviewed in this paper the recent advances in crypto that we foresee will add a new layer of security to Cloud Computing and boost its adoption. As we have shown in the paper, most cryptographic primitives are ready to be used. We only need to convince Cloud Providers to implement them or produce efficient implementations that could ease its inclusion in open source Cloud Computing platforms.

We think that Europe is in a good position to lead this activity. There are many EC funding projects working on Cloud Computing and we expect that collaboration between them can produce the drive needed for this development.

5 Acknowledgements

The work in this paper was partly sponsored by the EC Framework Programme as part of the ICT PASSIVE project (http://ict-passive.eu/) and by the Ministry of Science and Innovation through the ARES (CSD2007-00004) project.

6 References

[1] A. C.-C. Yao. How to generate and exchange secrets. In Foundations of Computer Science (FOCS'86), pages 162-167. 1986.

[2] M. v. Dijk and A. Juels. On the impossibility of cryptography alone for privacy-preserving cloud computing. In Hot topics in Security (HotSec'10), pages 1{8. USENIX Association, 2010.

[3] C. Gentry and S. Halevi. Implementing Gentry's fully-homomorphic encryption scheme. Cryptology ePrint Archive, Report 2010/520, 2010. http://eprint.iacr.org/2010/520.

[4] D. Boneh, E.-J. Goh, and K. Nissim. Evaluating 2-DNF formulas on ciphertexts. In Theory of Cryptography Conference (TCC'05), vol. 3378 LNCS, pages 325-341. Springer, 2005

[5] N. Smart and F. Vercauteren. Fully homomorphic encryption with relatively small key and ciphertext sizes. In Public Key Cryptography (PKC'10), vol. 6056 LNCS, pages 420-443. Springer, 2010.

[6] M. v. Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan. Fully homomorphic encryption over the integers. In Advances in Cryptology - EUROCRYPT'10, vol. 6110, LNCS, pages 24-43. Springer, 2010.

[7] S. Bugiel, St. Nurnberger, A. R. Sadeghi, Th. Schneide.Twin Clouds: An Architecture for Secure Cloud Computing. Workshop on Cryptography and Security in Clouds, March 15-16, 2011, Zurich.

[8] Benny Chor, Eyal Kushilevitz, Oded Goldreich, and Madhu Sudan. 1998. Private information retrieval. J. ACM 45, 6 (November 1998), 965-981

[9] Kushilevitz, E.; Ostrovsky, R.; , "Replication is not needed: single database, computationally-private information retrieval," Foundations of Computer Science, 1997. Proceedings., 38th Annual Symposium on , vol., no., pp.364-373,

[10] Song, D.X., Wagner, D., Perrig, A.: Practical techniques for searches on encrypted data. In: 21st Symp. on Security and Privacy (S&P), Berkeley, California, May 2000, pp. 44–55. IEEE Computer Society, Los Alamitos, 2000.

[11] Curtmola, R., Garay, J.A., Kamara, S., Ostrovsky, R.: Searchable symmetric encryption: improved definitions and efficient constructions. 13th ACM Conference on Computer and Communications Security (CCS '06) , pp. 79–88, 2006.

[12] Boneh, D., Di Crescenzo, G., Ostrovsky, R., Persiano, G.: Public key encryption with keyword search. In EUROCRYPT 2004. LNCS 3027, pp. 506–522. Springer, Heidelberg, 2004.

[13] Giovanni Di Crescenzo and Vishal Saraswat. Public key encryption with searchable keywords based on Jacobi symbols. In Proceedings of the 8th International Conference on Progress in Cryptology (INDOCRYPT'07), Springer-Verlag, Berlin, Heidelberg, 282-296, 2007.

[14] Rhee, H.S., Park, J.H., Susilo, W., Lee, D.H.: Improved searchable public key encryption with designated tester. In ASIACCS, pp. 376–379. ACM, New York, 2009.

[15] Boneh, D., Waters, B.: Conjunctive, subset, and range queries on encrypted data. In TCC 2007. LNCS 4392, pp. 535–554. Springer, Heidelberg, 2007.

[16] Keith Jeffery, Burkhard Neidecker-Lutz, The Future Of Cloud Computing, Opportunities For European Cloud Computing Beyond 2010. http://cordis.europa.eu/fp7/ict/ssai/docs/cloud-report-final.pdf

[17] Yanpei Chen, Vern Paxson and Randy H. Katz, “What’s New About Cloud Computing Security?” Technical Report No. UCB/EECS-2010-5, http://www.eecs.berkeley.edu/Pubs/TechRpts/2010/EECS-2010-5.html, Jan. 20, 2010.

[18] RSA, The Role of Security in Trustworthy Cloud Computing

[19] Ebenezer A. Oladimeji, Security threat Modeling and Analysis: A goal-oriented approach, 2006

[20] Ristenpart, Thomas and Tromer, Eran and Shacham, Hovav and Savage, Stefan, Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds, 2009

[21] Shamir, Adi, How to share a secret, Commun. ACM, 1979, 612—613

[22] J. S. Plank and J. Luo and C. D. Schuman and L. Xu and Z. Wilcox-O'Hearn, A Performance Evaluation and Examination of Open-Source Erasure Coding Libraries For Storage, 2009

[23] G. Galiero, G. Giammatteo, Trusting third-party storage providers for holding personal information. A context-based approach to protect identity-related data in untrusted domains, Identity in the Information Society 2 (2):99-114