# Data Protection in Cloud Scenarios

Sabrina De Capitani di Vimercati, Sara Foresti, and Pierangela Samarati

Computer Science Department
Università degli Studi di Milano – 26013 Crema, Italy
{sabrina.decapitani,sara.foresti,pierangela.samarati}@unimi.it

**Abstract.** We present a brief overview of the main challenges related to data protection that need to be addressed when data are stored, processed, or managed in the cloud. We also discuss emerging approaches and directions to address such challenges.

## Data security and privacy in the cloud

The 'cloud' has emerged as a successful paradigm enabling users and companies to have access to a virtually unlimited amount of resources to store, manage, and process data in a reliable and dependable infrastructure, even with economic advantages with respect to 'in-house' solutions. Together with considerable evident convenience, the cloud also introduces novel security and privacy issues. In fact, when storing or processing data in the cloud, data owners lose control over their data, leaving them potentially exposed to unauthorized parties, including the provider itself that might be not fully trusted. While typically cloud providers may be considered reliable for guaranteeing basic security protection (such as protection from unauthorized accesses to data and resources by third parties), they might not be considered trusted for the confidentiality (i.e., authorized to know the content) - or guaranteeing integrity - of the data they store or process.

Many are the challenges that need to be addressed to guarantee proper security and privacy in the cloud. In this paper, we focus in particular on the challenges specifically related to data management [18,19,21,34,35]. Of course, there are also other security and privacy issues that characterize a cloud scenario (e.g., multi-tenancy and virtualization, fault-tolerance management [26,27,28]) on which we do not elaborate.

*Protection of data at rest.* Protection of data at rest concerns the security and confidentiality of data in storage. Data stored at an external cloud provider need to be protected from unauthorized accesses by third parties as well as from the cloud provider itself, which might be not trusted for knowing the content of the data it stores or the accesses performed on them (*honest-but-curios* provider). The protection of the confidentiality of stored data typically relies on encryption. In cloud scenarios, protecting data from the providers' eyes requires keeping the encryption key within the client's trust boundary. In other words, encryption should work at the client side, encrypting data before moving them

to the cloud. Since encryption makes query evaluation and application execution more expensive or not always possible (see next challenge on 'fine-grained access to outsourced data'), alternative *fragmentation-based* solutions have been also proposed (e.g., [1,6,7,8,9,10]). Fragmentation allows departing from encryption whenever what is sensitive are not the data values singularly taken but their association (e.g., in a medical database, patients' names and illnesses might be considered public, while the specific association between the name of each patient and her illness might be considered sensitive). In this case, instead of encrypting the values, the sensitive association can be protected by storing values that are sensitive in association in different fragments so to break the association itself impeding its visibility to non authorized parties. For instance, with reference to our example, the patients' names can be stored in one fragment and illnesses in another one. To ensure that the sensitive associations protected by fragments cannot be reconstructed, fragments can be stored at independent (and non communicating) providers, or fragments must be guaranteed to be unlinkable. Fragmentation limits encryption to values that are sensitive by themselves, or even completely departs from encryption in cases (e.g., hybrid cloud) where the availability of a trusted party can be assumed for some storage/computation support. The advantage of using fragmentation is the availability of data in the clear, which enables evaluation of conditions on them and therefore better support for query processing by the cloud provider. In addition to data confidentiality, data integrity (i.e., authenticity and integrity of the stored data) and availability (i.e., the satisfaction by cloud providers of the data storage and access requirements that users may wish to enforce) are two further aspects that need to be addressed [3,29].

*Fine-grained access to outsourced data.* As noted above, cloud providers cannot have full access to the data they store, which might be either encrypted or fragmented. Also, when data are encrypted, the encryption key should remain within the client's trust boundary to ensure data remain confidential even with respect to the storing and processing provider itself. Providers cannot then decrypt data for query execution, making evaluation of conditions and query support difficult (if at all possible). The problem of providing support for fine-grained access (i.e., retrieval of data satisfying given conditions) over encrypted data has been under the attention of the research and development community in recent years and several investigations have been carried out. Among the analyzed techniques there are: cryptographic techniques supporting keyword-based searches (e.g., [4]), homomorphic encryption (e.g., [22]), the use of different layers of encryption each supporting specific operations [33], and metadata (indexes) attached to the data and used for fine-grained information retrieval and execution of specific kinds of queries (e.g., [5,25,39]). The major difficulty in such investigations is the tradeoff existing between providing support for query processing and ensuring that such support does not leak sensitive information otherwise protected by encryption (or fragmentation).

*Selective data access.* Data stored in the cloud may be subject to different access control policies, meaning that different users might need to enjoy different views on the outsourced data. Enforcing authorizations providing such selective access in cloud environments results particularly challenging since, if on one side clearly the data owner cannot provide such enforcement itself (as it would mean intercepting every access to data), on the other side, the cloud provider may not be fully trusted for such an enforcement. Also, the policy itself might be sensitive or leak information on the data content. There are two main lines of work investigating solutions for enforcing access control policies in the cloud. The first line of work, under the generic umbrella of *attribute-based encryption* (ABE) [24,41] is based on public key encryption and enforces authorizations by ensuring that encryption depends on the values of certain attributes (which characterize authorized users). This way, a user will be able to access data if her set of attributes matches conditions on the attributes associated with the encrypted data. ABE allows enforcement of authorizations that depend on different conditions, thus providing expressive authorization support. The main limitation of such approaches relate to the evaluation cost (given the use of public key encryption) and to the difficulty of enforcing revocation. The second line of work, called *selective encryption* [12,13], is based instead on the use of symmetric encryption and enforces authorizations by translating the authorization policy into an equivalent encryption policy so that data can be encrypted with different keys and keys are distributed to users in such a way that they can decrypt all (completeness) and only (correctness) the data they are authorized to access. A hierarchical organization of the encryption keys employed enables enforcing such authorizations (providing different views over data) while ensuring both a single copy of the data and the use of only one key per user. In fact, proper organization of keys in a hierarchy, with tokens enabling key derivation allows users to derive, from their own key all and only the keys enabling access to data they are authorized to access. Selective encryption provides efficient access control, as only symmetric encryption is used. Also, the use of public tokens enabling key derivation allows their storage in the cloud itself. Changes to the access policies (i.e., grant or revocation of authorizations) can be conveniently enforced by over-encryption, by which the data owner can enforce changes to authorizations with the cooperation of the cloud provider, that - when demanded - can wrap the data with a further level of encryption at the provider side (encrypting resources already encrypted by the owner). Over-encryption enforces authorization changes without the need for the data owner of retrieving, re-encrypting, and re-uploading data already stored in the cloud.

*Query privacy.* In addition to data themselves, several scenarios may also require confidentiality guarantees on accesses made on data. A classical example of these scenarios is a medical encyclopedia: while the encyclopedia itself is not sensitive and neither might be (with respect to the storing provider) the identity of users accessing it, the specific entries that a user looks for might be considered confidential as they may disclose her (or of a person close to her) health condition. Similar query confidentiality guarantees might also be requested when

stored data are encrypted (as knowledge of the access might even compromise the confidentiality of the stored data). The problem then arises of guaranteeing *access confidentiality,* that is, the fact that a given access aims at given data, as well as *pattern confidentiality*, that is, the fact that two accesses aim at the same data. We call these new confidentiality guarantees *query privacy* as the aim is to have them while also supporting efficient access to data for query support (e.g., index-search and evaluation of range conditions). Classical solutions providing access privacy, such as private information retrieval proposals, offer strong guarantees but limited access functionality and bear performance overhead that make them not applicable in real-life scenarios. Among more recent approaches, Path ORAM and the shuffle index, provide better performance and therefore applicability. Both these solutions are based on specific index structures and provide protection by either relying on a local stash, with dynamic re-mapping and delayed writing (Path ORAM) [37] or by relying on caching, cover searches, and shuffling with dynamic re-allocation of data at every access (shuffle index) [16,17]. Open issues are related to the need of decreasing the performance overhead, providing more support for queries, and ensuring strong confidentiality guarantees.

*Integrity of query results.* In addition to confidentiality, integrity of data can also be put at risk when the involved provider(s) may not be fully trustworthy. While for storage integrity classical techniques (e.g., chaining and signature) can be used, ensuring integrity of data dynamically retrieved, or resulting from computation, is challenging. Assessing integrity for query results or computations entails providing users with the ability to verify that the result returned by the cloud provider is complete (i.e., computed on the whole data collection), correct (i.e., computed on genuine data and correctly performing the computation), and fresh (i.e., computed on the most recent version of the data). Approaches for guaranteeing integrity of query results can be classified as *deterministic* (e.g., [30,31,32]) and *probabilistic* (e.g., [15,23,36,38,40,42]). Deterministic solutions use authenticated data structures (e.g., signature chains, Merkle hash trees, skip lists) or encryption-based solutions and can detect an integrity violation only for queries formulated on the attribute(s) on which they have been defined. Probabilistic solutions are based on the insertion of fictitious information or checks in a dataset whose absence in a query result signals an integrity violation. Probabilistic approaches can detect an integrity violation for any query but only with probabilistic guarantees, meaning that they are subject to false negative results. The problem of assessing integrity of query results becomes even more complex in emerging scenarios for distributed computation, where different providers or workers may be involved (e.g., [11]).

*Collaborative computation with selective sharing.* In several scenario computation or query execution in the cloud might involve data under the control of different authorities (data owners) and stored at different providers, which may impose different access and sharing restrictions on their data. Some approaches have addressed the problem of performing collaborative computations in contexts

where no sharing is possible between the involved parties and only the query result can be known to them (e.g., secure multi-party computation or *sovereign joins* [2]). These solutions are based on the use of encryption together with the possible involvement of a trusted computing base. Other approaches have considered scenarios where data can be selectively shared with other parties and different data owners and/or cloud providers need to collaborate, and selectively share information with others, for query execution. The problem addressed is then the distributed query execution (which necessarily entails exchange of data among the involved parties) in such a way that data are made visible only to authorized parties and no information is improperly shared or leaked [14,43]. In this context, ongoing work is investigating novel techniques for expressing and enforcing sharing policies, regulating information flows in query execution, and efficiently computing a query execution plan ensuring that no information is improperly released or leaked. Other approaches address the orthogonal problem of protecting the objectives of queries from the providers involved in their evaluation (e.g., [20]).

**Acknowledgment.** This work was supported in part by: the EC within the 7FP under grant agreement 312797 (ABC4EU) and within the H2020 under grant agreement 644579 (ESCUDO-CLOUD); the Italian Ministry of Research within PRIN project "GenData 2020" (2010RTFWBH).

## References

1. Aggarwal, G., Bawa, M., Ganesan, P., Garcia-Molina, H., Kenthapadi, K., Motwani, R., Srivastava, U., Thomas, D., Xu, Y.: Two can keep a secret: A distributed architecture for secure database services. In: Proc. of the 2nd Biennial Conference on Innovative Data Systems Research (CIDR 2005). Asilomar, CA, USA (January 2005)
2. Agrawal, R., Asonov, D., Kantarcioglu, M., Li, Y.: Sovereign joins. In: Proc. of the 22nd International Conference on Data Engineering (ICDE 2006). Atlanta, GA, USA (April 2006)
3. Ateniese, G., Burns, R., Curtmola, R., Herring, J., Khan, O., Kissner, L., Peterson, Z., Song, D.: Remote data checking using provable data possession. ACM Transactions on Information and System Security (TISSEC) 14(1), 12:1–12:34 (May 2011)
4. Cao, N., Wang, C., Li, M., Ren, K., Lou, W.: Privacy-preserving multikeyword ranked search over encrypted cloud data. In: Proc. of the 30th IEEE International Conference on Computer Communications (INFOCOM 2011). Shanghai, China (April 2011)
5. Ceselli, A., Damiani, E., De Capitani di Vimercati, S., Jajodia, S., Paraboschi, S., Samarati, P.: Modeling and assessing inference exposure in encrypted databases. ACM Transactions on Information and System Security (TISSEC) 8(1), 119–152 (February 2005)
6. Ciriani, V., De Capitani di Vimercati, S., Foresti, S., Jajodia, S., Paraboschi, S., Samarati, P.: Fragmentation and encryption to enforce privacy in data storage. In: Proc. of the 12th European Symposium On Research In Computer Security (ESORICS 2007). Dresden, Germany (September 2007)

7. Ciriani, V., De Capitani di Vimercati, S., Foresti, S., Jajodia, S., Paraboschi, S., Samarati, P.: Enforcing confidentiality constraints on sensitive databases with lightweight trusted clients. In: Proc. of the 23rd Annual IFIP WG 11.3 Working Conference on Data and Applications Security (DBSec 2009). Montreal, Canada (July 2009)
8. Ciriani, V., De Capitani di Vimercati, S., Foresti, S., Jajodia, S., Paraboschi, S., Samarati, P.: Keep a few: Outsourcing data while maintaining confidentiality. In: Proc. of the 14th European Symposium On Research In Computer Security (ESORICS 2009). Saint Malo, France (September 2009)
9. Ciriani, V., De Capitani di Vimercati, S., Foresti, S., Jajodia, S., Paraboschi, S., Samarati, P.: Combining fragmentation and encryption to protect privacy in data storage. ACM Transactions on Information and System Security (TISSEC) 13(3), 22:1–22:33 (July 2010)
10. De Capitani di Vimercati, S., Foresti, S., Jajodia, S., Livraga, G., Paraboschi, S., Samarati, P.: Fragmentation in presence of data dependencies. IEEE Transactions on Dependable and Secure Computing (TDSC) 11(6), 510–523 (November/December 2014)
11. De Capitani di Vimercati, S., Foresti, S., Jajodia, S., Livraga, G., Paraboschi, S., Samarati, P.: Integrity for distributed queries. In: Proc. of the 2nd IEEE Conference on Communications and Network Security (CNS 2014). San Francisco, CA, USA (October 2014)
12. De Capitani di Vimercati, S., Foresti, S., Jajodia, S., Paraboschi, S., Samarati, P.: Over-encryption: Management of access control evolution on outsourced data. In: Proc. of the 33rd International Conference on Very Large Data Bases (VLDB 2007). Vienna, Austria (September 2007)
13. De Capitani di Vimercati, S., Foresti, S., Jajodia, S., Paraboschi, S., Samarati, P.: Encryption policies for regulating access to outsourced data. ACM Transactions on Database Systems (TODS) 35(2), 12:1–12:46 (April 2010)
14. De Capitani di Vimercati, S., Foresti, S., Jajodia, S., Paraboschi, S., Samarati, P.: Authorization enforcement in distributed query evaluation. Journal of Computer Security (JCS) 19(4), 751–794 (2011)
15. De Capitani di Vimercati, S., Foresti, S., Jajodia, S., Paraboschi, S., Samarati, P.: Integrity for join queries in the cloud. IEEE Transactions on Cloud Computing (TCC) 1(2), 187–200 (July-December 2013)
16. De Capitani di Vimercati, S., Foresti, S., Paraboschi, S., Pelosi, G., Samarati, P.: Efficient and private access to outsourced data. In: Proc. of the 31st International Conference on Distributed Computing Systems (ICDCS 2011). Minneapolis, MN, USA (June 2011)
17. De Capitani di Vimercati, S., Foresti, S., Paraboschi, S., Pelosi, G., Samarati, P.: Shuffle index: Efficient and private access to outsourced data. ACM Transactions on Storage (TOS) 11(4), 19:1–19:55 (October 2015), article 19
18. De Capitani di Vimercati, S., Foresti, S., Samarati, P.: Managing and accessing data in the cloud: Privacy risks and approaches. In: Proc. of the 7th International Conference on Risks and Security of Internet and Systems (CRiSIS 2012). Cork, Ireland (October 2012)
19. De Capitani di Vimercati, S., Foresti, S., Samarati, P.: Data security issues in cloud scenarios. In: Proc. of the 11th International Conference on Information Systems Security (ICISS 2015). Kolkata, India (December 2015)
20. Farnan, N., Lee, A., Chrysanthis, P., Yu, T.: PAQO: Preference-aware query optimization for decentralized database systems. In: Proc. of the 30th IEEE Interna-

tional Conference on Data Engineering (ICDE 2014). Chicago, IL, USA (March-April 2014)
21. Foresti, S.: Preserving Privacy in Data Outsourcing. Springer (2011)
22. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Proc. of the 41st ACM Symposium on Theory of Computing (STOC 2009). Bethesda, MD, USA (May-June 2009)
23. Ghazizadeh, P., Mukkamala, R., Olariu, S.: Data integrity evaluation in cloud database-as-a-service. In: Proc. of the 9th IEEE World Congress on Services (SERVICES 2013). Santa Clara, CA, USA (June 2013)
24. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: Proc. of the 13th ACM Conference on Computer and Communications Security (CCS 2006). Alexandria, VA, USA (October-November 2006)
25. Hacigümüş, H., Iyer, B., Li, C., Mehrotra, S.: Executing SQL over encrypted data in the database-service-provider model. In: Proc. of the 21th ACM SIGMOD International Conference on Management of Data (SIGMOD 2002). Madison, WI, USA (June 2002)
26. Jhawar, R., Piuri, V.: Fault tolerance management in IaaS clouds. In: Proc. of the IEEE Conference in Europe about Space and Satellite Telecommunications (ESTEL 2012). Rome, Italy (October 2012)
27. Jhawar, R., Piuri, V., Samarati, P.: Supporting security requirements for resource management in cloud computing. In: Proc. of the 15th IEEE International Conference on Computational Science and Engineering (CSE 2012). Paphos, Cyprus (December 2012)
28. Jhawar, R., Piuri, V., Santambrogio, M.: Fault tolerance management in cloud computing: A system-level perspective. IEEE Systems Journal 7(2), 288–297 (June 2013)
29. Juels, A., Kaliski, B.: PORs: Proofs of retrievability for large files. In: Proc. of the 14th ACM Conference on Computer and Communications Security (CCS 2007). Alexandria, VA, USA (October-November 2007)
30. Li, F., Hadjieleftheriou, M., Kollios, G., Reyzin, L.: Authenticated index structures for aggregation queries. ACM Transactions on Information and System Security (TISSEC) 13(4), 32:1–32:35 (December 2010)
31. Mykletun, E., Narasimha, M., Tsudik, G.: Authentication and integrity in outsourced databases. ACM Transcations on Storage (TOS) 2(2), 107–138 (May 2006)
32. Pang, H., Jain, A., Ramamritham, K., Tan, K.: Verifying completeness of relational query results in data publishing. In: Proc. of the 24th ACM SIGMOD International Conference on Management of Data (SIGMOD 2005). Baltimore, MD, USA (June 2005)
33. Popa, R., Redfield, C., Zeldovich, N., Balakrishnan, H.: CryptDB: Protecting confidentiality with encrypted query processin. In: Proc. of the 23rd ACM Symposium on Operating Systems Principles (SOSP 2011). Cascais, Portugal (October 2011)
34. Samarati, P.: Data security and privacy in the cloud. In: Proc. of 10th International Conference on Information Security Practice and Experience (ISPEC 2014). Fuzhou, China (May 2014)
35. Samarati, P., De Capitani di Vimercati, S.: Cloud security: Issues and concerns. In: Murugesan, S., Bojanova, I. (eds.) Encyclopedia on Cloud Computing. Wiley (2016)
36. Sheng, G., Wen, T., Guo, Q., Yin, Y.: Verifying correctness of inner product of vectors in cloud computing. In: Proc. of the 2013 International Workshop on Security in Cloud Computing. Hangzhou, China (May 2013)

37. Stefanov, E., van Dijk, M., Shi, E., Fletcher, C., Ren, L., Yu, X., Devadas, S.: Path ORAM: An extremely simple Oblivious RAM protocol. In: Proc. of the 20th ACM Conference on Computer and Communications Security (CCS 2013). Berlin, Germany (November 2013)
38. Umadevi, G., Saxena, A.: Correctness verification in outsourced databases: More reliable fake tuples approach. In: Proc. of the 7th International Conference on Information Systems Security (ICISS 2011). Kolkata, India (December 2013)
39. Wang, H., Lakshmanan, L.: Efficient secure query evaluation over encrypted XML databases. In: Proc. of the 32nd International Conference on Very Large Data Bases (VLDB 2006). Seoul, Korea (September 2006)
40. Wang, H., Yin, J., Perng, C., Yu, P.: Dual encryption for query integrity assurance. In: Proc. of the 17th Conference on Information and Knowledge Management (CIKM 2008). Napa Valley, CA, USA (October 2008)
41. Waters, B.: Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In: Proc. of the 14th IACR International Conference on Practice and Theory of Public Key Cryptography (PKI 2011). Taormina, Italy (March 2011)
42. Xie, M., Wang, H., Yin, J., Meng, X.: Integrity auditing of outsourced data. In: Proc. of the 33rd International Conference on Very Large Data Bases (VLDB 2007). Vienna, Austria (September 2007)
43. Zeng, Q., Zhao, M., Liu, P., Yadav, P., Calo, S., Lobo, J.: Enforcement of autonomous authorizations in collaborative distributed query evaluation. IEEE Transactions on Knowledge and Data Engineering (TKDE) 27(4), 979–992 (April 2015)