On Attribute-Based Encryption for Access Control to Multidimensional Data Structures

Anna Epishkina and Sergey Zapechnikov
National Research Nuclear University (Moscow Engineering Physics Institute), Moscow, Russia
AVEpishkina@mephi.ru, SVZapechnikov@mephi.ru

Abstract
Multidimensional data structures are widely used in modern information technologies. They sometimes contain private or other sensitive information. We argue that attribute-based encryption is a handy tool for access control to multidimensional data structures, discussing the advantages and disadvantages of ciphertext-policy and key-policy attribute-based encryption for this task. We propose a scheme of attribute management for multidimensional data structures.

Keywords: **multidimensional data structures, OLAP, access control, attribute-based encryption**

1 Introduction

Currently, it is hard to overestimate role of information in the life of modern society. Information technologies became essential part in every area of human life. Special attention is paid to data storage and processing technologies. Millions of databases across the world provide storage for large amount of data. OLAP (On-Line Analytical Processing) systems are extremely popular in the field of data processing. Such systems are used in different areas such as sales, financial transactions, telecommunications, healthcare etc. Over the last few years, this approach was widely used in vital areas of human life. Because of that, security had a profound impact on OLAP systems development. Privacy of information, stored in data warehouses, have an important role to play in these technologies. Confidentiality of data stored in cloud services is particular acute nowadays.

Recent trends show a transition from companies' proprietary data storage to cloud storage. Besides economic benefits, the main motive power for outsourcing data storage is the versatility of such decisions. However, at the same the security aspect become more and more important, which implies the necessity of using encryption. In this scenario Cloud Storage Service provider always has access to the information, stored in its system. The progress of cloud technologies makes possible efficient and secure data storage. However, access control is very important to make cloud storage secure. Existing solutions provide a versatility access control system, but they are not fully secure because in most cases cloud provider can access to plaintext data. So, in most cases users prefer not to send confidential information to the cloud at all. As an example, such information may concern organization's partners or clients, and in accordance with the agreement on the inadmissibility it must not be disclosed to a third party.

Currently, one of the most prominent techniques for access control to databases and data warehouses is attribute-based access control (ABAC). In this paper, we consider the problem of access control to multidimensional datasets and databases. It is shown in our paper, that ciphertext-policy attribute-based encryption schemes can be effective tools for access control to multidimensional datasets. We offer attribute and key management schemes to support such technique. All data in such system is encrypted by means of special encryption schemes before being sent into the cloud.

The rest of the paper is organized as follows. In section 2 we consider some introductory material and related works. In section 3 we discuss the possibilities of using attribute-based encryption for access control to multidimensional datasets. An attribute management scheme is proposed in section 4. We give the conclusion and future research directions in section 5.

2 Preliminaries and related works

Formal model of multidimensional databases was introduced in (Pedersen, 2001). This technology can be used to effectively store, interactively process and analyze multidimensional data from multiple perspectives. In particular, it became a basis of OLAP (Jensen, 2010). For example, one can use this technology to store and analyze healthcare data, predict diagnoses and create specific therapy for specific patients.

Main advantage of this approach against simple relational databases is speed of large queries processing. Complex large multi-table queries are often used during data analytic processing. In case of relational databases, this leads to problems with performance of query processing. OLAP technology was developed with these ideas in mind to negate performance issues for complex queries.

Main object of this technology is multidimensional cube also called hypercube that represents multidimensional coordinate system. Dimensions are used as axes of this system. Dimension is the hierarchical system, which consists of attributes of the process being analyzed. For example, sales business process can have such dimensions as Data, Goods, and City and so on. At the same time, Data dimension can have such members as Quarter-Month-Week-Day, organized in hierarchy.

Measure is one of the quantitative properties of the process being analyzed in multidimensional system. Measures are stored in the cell of the coordinate system represented by cube. In the above example, sale amounts can be used as measures for quantitative description of sales process. Each cell store one value of any type, but all values of one measure have the same type. If a cube contains some measures, then the set of all measures is considered to be one more dimension.

Another essential part of multidimensional dataset technology is aggregates. Aggregate is the pre-calculated analytical query, stored in data warehouse and based on the source data, aggregated accordingly to the received request. Thus, combination of every possible aggregate and source data can be used to process any query. However, only small amount of determined aggregates are calculated during initialization of the system, while rest of the aggregates are being calculated on demand. The aggregation of many cells at some level of hierarchy gives the value of one cell at the next, higher level of hierarchy at the same dimension.

There are three main approaches to store data and aggregates in OLAP systems (Thomsen, 2002):

- MOLAP (Multidimensional OLAP) – data and aggregates are fully stored in multidimensional database;
- ROLAP (Relational OLAP) – data and aggregates are stored in relational database of special form and are virtually interpreted as multidimensional structure;

- HOLAP (Hybrid OLAP) – data stored in relational database, but aggregates are stored in multidimensional database.

In this work, we will keep in mind MOLAP as main approach. Even though this approach has large data overhead, it can achieve good performance.

3 Using Attribute-Based Encryption for Multidimensional Datasets

Other crucial technique is attribute-based encryption (ABE) proposed by Sahai and Waters (Sahai, 2005), It is intended for "one to many" encryption, so that ciphertexts are created for users who meet certain requirements. There are two types of such schemes: ciphertext-policy attribute-based encryption (CP-ABE) and key-policy attribute-based encryption (KP-ABE). CP-ABE implies that ciphertexts are associated with the access policy, and the corresponding attributes are included in the private key, and KP-ABE vice versa. In both cases, ciphertext can be decrypted if and only if the attributes correspond to specified access policy (Zhenfu, 2012).

Thus, CP-ABE is suitable for implementation of attribute-based access control for data storage into untrusted environment (Fig.1). According to (Sukhodolskiy, 2017), the following main components should be identified:

- ABAC Servers, including server resource attributes, server entity attributes, server environment attributes, and server actions attributes;
- Certification Authority;
- server working as a gateway for interaction with cloud storage;
- access control server;
- client’s application.

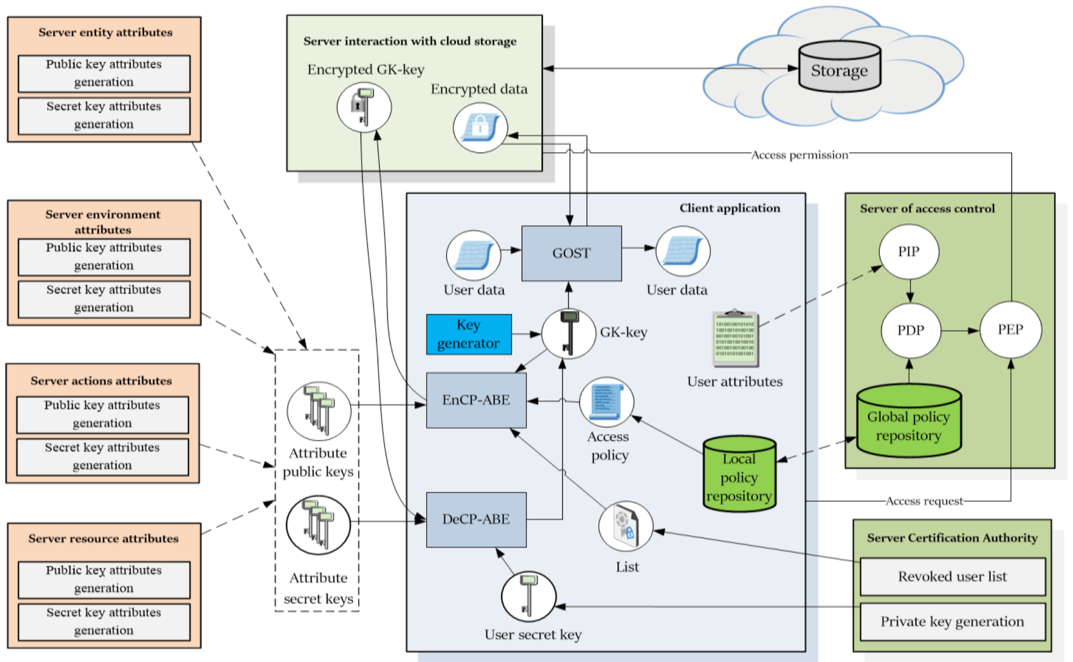
Server entity attributes
Public key attributes generation
Secret key attributes generation
Server environment attributes
Public key attributes generation
Secret key attributes generation
Server actions attributes
Public key attributes generation
Secret key attributes generation
Server resource attributes
Public key attributes generation
Secret key attributes generation
Attribute public keys
Attribute secret keys
Server interaction with cloud storage
Encrypted GK-key
Encrypted data
Storage
Access permission
Client application
User data
GOST
User data
Key generator
GK-key
User attributes
EnCP-ABE
Access policy
Local policy repository
DeCP-ABE
List
User secret key
Server of access control
PIP
PDP
PEP
Global policy repository
Access request
Server Certification Authority
Revoked user list
Private key generation

Figure 1: Using ABE scheme to access control in cloud storage

Access control policies are directly connected with attributes. Data encryption and access are determined by a variety of attributes, some of which are required for CP-ABE (principal attributes). These principal attributes are responsible for encryption and decryption of all stored files. Other attributes, which we called non-principal, are checked only by access control server and are not associated with the encrypted data by means of CP-ABE scheme.

Dividing attributes into two categories is made manually by administrator and/or data owner. The decision which attributes will be principal and which is not, should be made taking into account existing access control system and company's business model.

The client application consists of a key generation algorithm, a symmetric block cipher algorithm, an, of course, an encryption (EnCP-ABE) and a decryption (DeCP-ABE) algorithms for CP-ABE. It also provides information about the possible access policies and the values of user attributes. Attributes can be divided into those that are appointed by system's security administrator and those that are generated automatically (time, IP-address, etc.).

To encrypt a file, at first user selects an access policy. Then it starts the key generation algorithm, which generates a key for the symmetric algorithm (symmetric key, for short). User's data may be encrypted using, for example, GOST R 34.12-2015 symmetric cipher (GOST, 2015) or some other block cypher. Then encrypted file is delivered to a cloud gateway server. It is worth emphasizing, that the encrypted data must be associated with a particular access policy.

After that, the application receives a revoked user list from CA, which is required for the EnCP-ABE algorithm as well as public key attributes, access policy and symmetric key. The algorithm produces symmetric key encrypted by means of CP-ABE scheme, which is sent to the cloud gateway with the encrypted file.

To decrypt data, the application sends the user attributes to the access control server. After checking for compliance with the policy of the user attributes are sent to secret keys and private key. Having the keys of CA and application ABAC server it is possible to decrypt symmetric key. By means of symmetric key algorithm, the application decrypts the file. The computational complexity of decryption strongly depends on the number of attributes, so to improve performance it is undesirable to associate all of the attributes with encryption policy, some of them can be simply checked by the access control server without ABE. If the user meets all the rules of the access policy for the data it receives unencrypted file.

ABAC Servers manage key attributes for ABE. Each of these keys holds implicitly a set of user's attributes.

Certification Authority provides support for current users of the system and generates a list of application user's secret keys.

Gateway for interaction with the cloud provides a lot of functions for search and delivery of encrypted files; also it indexes the encrypted data and keys before sending to the cloud.

4 The Scheme of Attribute Management

Now let us consider what kind of attribute-based encryption and how can be used to implement the above-mentioned scheme. We will show CP-ABE schemes allow assigning a set of attributes for each cell of dataset for fine-grained access control. More precisely, each cell must be associated with a set of attributes as follows:

- *"Cube_C"*;
- *"Dimension_i_1"; "Dimension_i_1.Level_j_1"; "Dimension_i_1.Level_j_1.Cell_k_1";*
  ...
- *"Dimension_i_d"; "Dimension_i_d.Level_j_d"; "Dimension_i_d.Level_j_d.Cell_k_d",*

where *"Cube_C"* is the attribute that indicates that the cell belongs to a particular cube *C* and its content may be read by any user who has no restrictions on the access to the hypercube *C*, *"Dimension_i"* is the attribute that indicates that the cell may be read by any user who has no restrictions on reading hierarchies on *i*-th hypercube's dimension, *"Dimension_i.Level_j"* is the the attribute that indicates that the cell may be read by any user who has access for reading cell having *j*-th level of hierarchy at *i*-th dimension, and finally *"Dimension_i.Level_j.Cell_k"* is the attribute that indicates that the cell may be read by any user who has access for reading *k*-th cell at *j*-th level of hierarchy at *i*-th dimension.

Let hypercube *C* has *d* dimensions. Then every cell must have 3*d*+1 attributes, because each cell has exactly *d* coordinates on each of hypercube's dimensions, and each coordinate is a set of three values: a dimension name, a level of hierarchy name for the given dimension, and an index at the given level of hierarchy.

However, our ABE is ciphertext-based, so we finally must construct an access policy, such that only users having a set of necessary and sufficient attributes but nobody else will have access to the specified cell. Such access policy can be simply constructed as the following predicate:

$$Policy := Cube\_C \vee \Big(\big(Dimension\_i_1 \vee Dimension\_i_1.Level\_j_i \vee Dimension\_i_1.Level\_j_i.Cell\_k_1 \big) \wedge \\ \wedge \ldots \wedge \big(Dimension\_i_d \vee Dimension\_i_d.Level\_j_d \vee Dimension\_i_d.Level\_j_d.Cell\_k_d \big) \Big)$$

This predicate means that the access to the cell is granted either to the users who are entitled to access to hypercube without restrictions, or those who have a right of access to such subsets of cells for each dimension which include this cell. The right of access to subsets of cells for each dimension is given for such users who can access all cells along the dimension without restrictions or such users who can access all the hierarchical level to which the cell belongs or such users who can access this particular cell. It is important that every cell must have the above-mentioned set of attributes and access policy in the specified form associated with it.

To implement the access policy, each user must have a set of attributes assigned by ABAC servers according to the following set of rules:

- If user should have right to access all cells of hypercube *C*, it is assigned the attribute *"Cube_C"*;
- If user should have right to access all cells along the dimension *i*, it is assigned the attribute *"Dimension_i"* (however, she can have no rights to access every cell along one or more other dimensions). Simultaneously it is must not be assigned the attribute *"Cube_C"* because it makes no sense to assign attribute *"Dimension_i"*;
- If user should have right to access all cells at the certain hierarchical level *j* of the specified dimension *i*, it is assigned the attribute "*Dimension_i.Level_j"* (however, she can have no rights to access one or more other hierarchical level). Simultaneously it is must not be assigned the attribute *Dimension_i"*, because it makes no sense to assign attribute "*Dimension_i.Level_j"*;
- If user should have right to access certain cell with index *k* at certain hierarchical level *j* of the specified dimension *i*, it is assigned the attribute *"Dimension_i.Level_j.Cell_k"* (however, she can have no rights to access one or more other cells at the given hierarchical level). Simultaneously it is must not be assigned the attribute "*Dimension_i.Level_j"*, because it makes no sense to assign attribute *"Dimension_i.Level_j.Cell_k"*.

Access control is essential mechanism for privacy-preserving OLAP over encrypted data (Gorlatykh, 2017).

5 Conclusion

We have analyzed the problem of using attribute-based encryption to implement access control to multidimensional datasets.

The main result of this research is a model of attribute-based access control to multidimensional datasets. It is based on a set of rules for assigning access rules to users and a predicate for access control policy which must be applied to each cell of multidimensional dataset.

We will continue to develop our solution in order to implement attribute-based data authentication in multidimensional datasets. It seems that the most convenient tool for this task is attribute-based signature scheme.

Acknowledgement

This work was supported by Competitiveness Growth Program of the Federal Autonomous Educational Institution of Higher Education National Research Nuclear University MEPhI (Moscow Engineering Physics Institute).

References

Gorlatykh, A., Zapechnikov, S. (2017). *Challenges of privacy-preserving OLAP techniques*. Proc. of IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering. Moscow, Saint-Petersburg, 01 – 03 February 2017. (in printing)

GOST R 34.12-2015 (2015). Information *technology. Cryptographic protection of information. Block cyphers.* Russian state standard. Moscow, Standartinform, 2015. 25 p. (In Russian)

Jensen, C., Pedersen, T., Thomson, C. (2010). *Multidimensional databases and data warehousing.* Morgan and Claypool, 2010. 111 pp.

Pedersen, T. (2001). *Multidimensional database technology*. Journal Computer. Vol.34, Issue 12, Dec. 2001. Pp. 40-46.

Sahai, A., Waters, B. (2005). *Fuzzy identity-based encryption.* In: Cramer R. (eds) Advances in Cryptology – EUROCRYPT 2005. Lecture Notes in Computer Science, vol 3494. Springer, Berlin, Heidelberg. pp. 457-473.

Sukhodolskiy, I., Zapechnikov S. (2017). *An access control model for cloud storage using attribute-based encryption*. Proc. of IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering. Moscow, Saint-Petersburg, 01 – 03 February 2017. (in printing)

Thomsen, E. (2002). *OLAP solutions: building multidimensional information systems*. 2nd ed. Wiley, 2002. 696 pp.

Zhenfu, C. (2012). *New directions of modern cryptography*. CRC Press. Boca Raton, USA. 2012. 400 pp.