The current issue and full text archive of this journal is available at
www.emeraldinsight.com/1742-7371.htm

Data confidentiality using fragmentation in cloud computing

Aleksandar Hudic
SBA Research gGmbH, Vienna, Austria
Shareeful Islam
School of Architecture, Computing and Engineering (ACE), University of East London, London, UK, and
Peter Kieseberg, Sylvi Rennert and Edgar R. Weippl
SBA Research gGmbH, Vienna, Austria

Abstract

Purpose – The aim of this research is to secure the sensitive outsourced data with minimum encryption within the cloud provider. Unfaithful solutions for providing privacy and security along with performance issues by encryption usage of outsourced data are the main motivation points of this research.

Design/methodology/approach – This paper presents a method for secure and confidential storage of data in the cloud environment based on fragmentation. The method supports minimal encryption to minimize the computations overhead due to encryption. The proposed method uses normalization of relational databases, tables are categorized based on user requirements relating to performance, availability and serviceability, and exported to XML as fragments. After defining the fragments and assigning the appropriate confidentiality levels, the lowest number of Cloud Service Providers (CSPs) is used required to store all fragments that must remain unlinkable in separate locations.

Findings – Particularly in the cloud databases are sometimes de-normalised (their normal form is decreased to lower level) to increase the performance.

Originality/value – The paper proposes a methodology to minimize the need for encryption and instead focus on making data entities unlinkable so that even in the case of a security breach for one set of data, the privacy impact on the whole is limited. The paper would be relevant to those people whose main concern is to preserve data privacy in distributed systems.

Keywords Cloud computing, Data confidentiality, Privacy-preserving, Data fragmentation, Data outsourcing, Data management, Computing

Paper type Research paper

1. Introduction

Cloud computing is one of the most popular and innovative approaches to outsourcing data storage and processing (Buyya *et al.*, 2009; Nishchal and Mathur, 2010; Vaquero *et al.*, 2008; Peter and Grance, 2011; Abdalla *et al.*, 2008). There are many benefits of cloud computing including virtualization, utility computing, scalability, pay-per-use services, the ability to outsource data and processes, and nearly unlimited computing resources. The success of this model has led to the emergence of a number of cloud service providers (CSPs), both large and small. In this new and competitive market, providers such as Amazon (2006), Google (2008), IBM (2011), Microsoft (2011) and Oracle (2010) are constantly improving and expanding their cloud computing services (Greenberg *et al.*, 2008). The wide range of reliable and cost-effective services

International Journal of Pervasive Computing and Communications
Vol. 9 No. 1, 2013
pp. 37-51

1742-7371
DOI 10.1108/17427371311315743

is attractive to many businesses, which have chosen to move their business data or parts of it into the cloud.

While cloud computing offers many benefits, there are a number of risks in particular relating to offsite storage of customer data. By storing data or applications in cloud, businesses give up a measure of control (Islam *et al.*, 2012). Therefore, it is very important for the customer to be able to trust their CSP to treat their data as confidential and respect the customers' privacy. The best security technology would be useless if a provider decides to start selling customer data to third parties. Even if the customer trusts the CSP, there can be little protection against a malicious internal/curious employee gaining access to unencrypted customer files. In addition to the possibility of internal misuse of data, there is the risk of external attacks (Kaufman, 2009). Over the past few years, there have been some spectacular cases of data theft, including credit card numbers, customer records, and other personal data, which shows that not even large organizations are immune to security breaches (Widman, 2011).

There are numerous issues that pose serious potential challenges for data security in cloud computing, including locality restrictions and domain-specific restrictions, legal or technical limitations, different security metrics among providers, (im)proper implementation of adequate security mechanisms, efficiency in data manipulation, the aggregation of data, and service-level security issues. Commonly cloud providers fragment the data and store it on multiple dynamic virtual servers. The data are encrypted to protect from unauthorized access but a purely cryptographic approach requires large computational resources. As a consequence this can cause issues in situations where clients need real-time access to their data, but delayed due to computational processing. Encryption also entails the issue of key management, which can be problematic both for the CSP and the user. Therefore, it is necessary to develop an approach that would allow users to access and manage their data efficiently while ensuring its confidentiality and privacy. To date, no adequate technical solution has been developed to deal with the issue of data fragmentation and encryption in the cloud (Curran *et al.*, 2011). Currently available fragmentation methods strive to facilitate and improve data manipulation, distribution and transportation, increase flexibility, and optimize storage while decreasing the processing time and distributing processing costs (Rangan and Vin, 1993; Randell, 1969). However, security is usually not a key feature considered by the approaches. To ensure the confidentiality of the fragmented data, most state-of-the-art security approaches rely on encryption (Ciriani *et al.*, 2010). In contrast, our approach aims to minimize the need for encryption and instead focuses on making data entities unlinkable so that even in the case of a security breach for one set of data, the privacy impact on the whole dataset is limited. To this end, we fragment normalized relational databases and then distribute the fragments – individual tables – to different cloud storage providers. These providers have to be non-colluding, which can be ensured, e.g. by service level agreements (SLAs) or legislation, though legal regulations usually only stipulate confidentiality requirements but do not specify countermeasures. While costs are an important factor in cloud computing, reducing them is not the primary objective of this paper. For our method, datasets have to be analysed initially, as proposed in Morali and Wieringa (2010), so that any additional confidentiality constraints needed for a certain subject domain can be included in order to develop SLAs that contain well-defined and user-specific confidentiality requirements. This paper discusses current confidentiality concerns and privacy issues for data in cloud storage in

Section 2, followed by a description of both the fragmentation and the distribution processes in our proposed model for secure data outsourcing in Section 3. Section 4 describes a real-world scenario for our model, Section 5 describes related work, and the final section outlines the conclusions and outlook of this paper.

2. Background

The core function of cloud computing is without doubt the ability to store data. While this convenient way of outsourcing storage for large volumes of data is used by many companies, placing sensitive business data and personal data of customers or employees into the shared environment of the cloud requires trust in the external CSPs. Confidentiality and privacy become the most valued commodities in the cloud. It is important to identify any vulnerabilities and attack vectors that could potentially harm customers and providers themselves. However, vulnerabilities and the extent of damage they can cause depend heavily on the data domain in question. CSPs that store mission-critical business functions, patient data, or other highly sensitive information would be much more severely affected than those that only host customer information of minor importance. In either case, however, when vulnerability is detected, users lose access to their data until the vulnerability can be brought under control. When data and applications are stored at CSPs, there is always a risk of someone gaining unauthorized access or processing the data or application (Chen *et al.*, 2010), which can lead to the data owner using control of their critical assets completely.

Despite the widespread use of cloud storage, users are to some extent aware of the risks it entails, as shown in a survey from Proofpoint (2010) where 40.5 percent of respondents said that companies could seriously increase their risk of data leaks by using software as a service and cloud computing. Data leaks can seriously affect customers' privacy. Privacy is a legal and moral right of every individual and is defined as the ability to decide how, when, and to what extent our information is shared with others. There are numerous issues related to the goals of privacy protection that must be addressed, including data subject rights, owner consent, access to data, and anonymity, according to the principles of fair information practice (Electronic Privacy Information Center, 1972).

The situation becomes even more critical when CSPs reserve the right to change their terms of service (Electronic Privacy Information Center, 1972). It is important that the CSPs provide transparent information on how they handle data and, in particular, on the security measures they use to protect the data, including provisions and mitigation plans for the event of data being compromised. In a stand-alone system, data confidentiality can easily be ensured. In the cloud, however, the data of many users are stored in multiple databases, which makes it very difficult to ensure paradigms of security and integrity such as transaction durability or consistency. This is even more complex when shared resources are used with multiple databases. It is of great importance that the data is unlinkable and that only the required data is used for data processing (Islam *et al.*, 2011; Mouratidis *et al.*, 2012). In this virtualized environment, the privacy and security of transactions cannot be ensured solely with standard methods such as HTTP. The application programming interface (API) is used instead (Subashini and Kavitha). However, APIs introduce new complexities that can lead to security vulnerabilities either in the API stack or in the technology that handles the API calls. It is therefore necessary to ensure the security of channels and the reliability of the API handling transactions to guarantee data integrity.

Another critical issue in ensuring confidentiality is where and how the data is stored. This includes the security of the data storage application, operating system, and any other applications that may be running on the same system as well as the physical storage location. The physical location must be adequately secured, with safeguards and regular backups for the event of system or hardware failure. For customers it is important to know the physical storage site and how the backups are protected. In addition to insufficient security provisions or an inadequately secured physical storage location, inconsistent use of encryption software keys can also pose a considerable threat to data confidentiality, as can operational, authentication and authorization failures. This can lead to an unauthorized individual deleting or modifying data, which can severely affect customers' privacy. If important intellectual property is lost, this may also have severe financial implications and damage the company's competitiveness. Such confidentiality breaches also have a negative impact on the service providers, who will lose the confidence of customers and may also experience other problems such as the loss of employee morale or business partners' willingness to work with them.

This paper presents a new method of protecting data in cloud storage from unauthorized access by external attackers or curious CSPs (which can be termed "semi-trusted" CSPs due to their or their employees' honest but possibly curious nature). We assume the customer perspective, i.e. examine how data owners can make their data less expressive and therefore less likely to be read by curious service providers. Our approach fragments the data and distributes individual fragments to different CSPs to ensure that they are unlinkable as long as the providers do not collude. This approach is cheaper and less complex then using searchable encryption schemes (Abdalla *et al.*, 2008), where the data owners have to maintain the secret keys.

3. Data fragmentation and distribution model

This section discusses the technical and architectural details of our model and explains both how the fragments are defined and created in order to comply with privacy requirements, and how they are distributed.

3.1 Data fragmentation model

Our model uses fragmentation methods to ensure data privacy and confidentiality. In fragmentation, we use two types of relational database tables, which contain different types of data. Payload tables contain the actual data that is to be stored in the cloud and forms the fundamental structure of the relational database. These payload tables are linked together by tables that define the relationships between them and contain only foreign or primary keys. While these relationship tables can potentially be linked to each other, only very little data can be gained through them alone, so they can all be stored safely at a single cloud storage provider. As the manipulation of large amounts of data requires considerable computational resources and is, therefore, expensive, data in relational databases is split up into smaller fragments (Widman, 2011). In our data fragmentation model, the fragments must satisfy three essential requirements:

(1) normalization of the database to third normal form before fragmentation;

(2) setting a confidentiality level for the data in each table/fragment; and

(3) conforming to any user requirements concerning the distribution of the fragments.

Database normalization (Umanath and Scamell, 2007) removes redundancies and reduces data anomalies and is, therefore, essential for ensuring that the tables are independent of each other and cannot be linked. The stages of normalization are referred to as normal forms: in the first normal form (1NF), duplicates are removed and separate tables are created for related data. The second normal form (2NF) also separates data subsets by placing the subsets in separate state tables and creating relational dependencies between them. The third normal form (3NF) meets both of these requirements and removes any columns that are not directly dependent on the primary keys. There are theoretical higher orders of normal forms, but they are beyond the scope of this paper. The third normal form is used as an essential requirement in this method because it ensures that the relations in each table are well formed with the following characteristics:

- each table can be an independent subject;
- no redundant data is stored;
- non-primary attributes depend only on the primary key; and
- the integrity and consistency of the data is ensured.

We use three levels of confidentiality for the tables depending on the data they contain: high, medium, and low. High confidentiality tables (Table II) contain highly sensitive data such as health information, social security numbers, credit card numbers, and other data that must be protected appropriately. It can be fully encrypted, even including the name of the entity it belongs to, in order to hide its domain and make correlation attacks even more complicated. A user could store different types of data in encrypted form and mark them all as highly confidential, so the attacker would not know whether a given dataset included credit card numbers or less sensitive and less interesting information. Encryption keys must be stored locally and be accessible to authorized users only. The drawback of encryption is that it increases computational overhead, as data has to be decrypted each time before a query is processed. In our approach, we attempt to minimize the use of encryption; data can either be stored unencrypted in the trusted local domain, or stored partially encrypted in the cloud (Table I).

Medium confidentiality tables (Table III) do not contain very confidential data, but may have dependencies with others tables and must therefore be treated and distributed with care. Low confidentiality tables (Table V) need an even lower level of security, so priority is usually given to storage that allows their efficient manipulation. Table IV is an example of a fragment that represents the relational table that stores entity connections in a relational database. These fragments must be distributed to several servers to ensure that they cannot be linked to each other (Tables II-V).

EmpID	Credit card number
RKT-12	acb195087d0d5424c17724a425c07ba1
RKT-09	d78c3b972ad9cf53ddf8ac7144aea80c
RKT-77	ddd97a3e5dbf2908fc1f58307e63f894
KNL-47	c8e562a275af91a90accffd7180999c6
FSI-03	2d9abf9143f741b41dab9c9a38b6c318

Table I.
Encrypted data from highly confidential tables prepared for distribution

Users can define additional requirements concerning the confidentiality and distribution of the fragments, such as increasing or decreasing the level of confidentiality, partial encryption of certain columns, storing some of the fragments in the local domain, or splitting the data into additional fragments.

3.2 Data distribution model

The distribution model is based on storing different data fragments in a number of different storage locations. The model distinguishes between two domains, one of which is the trusted local domain from where the data originates and where highly confidential data can be stored without encryption. The second domain is the semi-trusted public domain, i.e. cloud storage providers, who have large computational and storage resources

Table II.
High confidentiality table

EmpID	Credit card number
RKT-12	4111 1111 1111 1111
RKT-09	5500 6469 0000 0004
RKT-77	3400 6546 0000 009
KNL-47	3000 9183 0000 04
FSI-03	3000 9828 0000 04

Table III.
Medium confidentiality table

ProjectNum	ProjectBudget
AG-727	230,000,000.00
AG-121	500,848,500,000.00
AG-564	345,457,000.00
HX-124	19,279,025,000.00
HX-232	182,222.00

Table IV.
Example of a relational table

ProjectNum	EmpID
AG-328	RKT-12
AG-727	RKT-09
AG-727	RKT-77
RT-452	KNL-47
HX-232	FSI-03

Table V.
Low confidentiality table

ZIP	City
11000	Washington, DC
2080032	Tokyo
107207	Moscow
20000	Dubrovnik
10000	New York

and are therefore used to store the bulk of the data fragments in encrypted form. The fragments, which must comply with the three essential requirements listed above, are distributed to a number of different cloud storage providers (Figure 1) to ensure that the confidentiality of the data can be preserved even in the cloud. As the public domain is not fully trusted, the distributed data fragments require appropriate protection during the distribution process, both at the distribution point and while they are being used. This is accomplished by using a virtual private network (VPN) to transport the data between the customer's local domain and the public domain of the CSPs. SLAs that guarantee the appropriate level of service and confidentiality, include the three essential requirements listed above, and meet the additional performance, availability, and serviceability requirements defined by the user must be established with the CSPs.

The fragmentation algorithm below illustrates the steps of the distribution process. Step one of the fragmentation process treats each table (t) of the database scheme S as an individual fragment that is exported to an XML file to ensure efficient transport and compatibility with different types of relational databases. In step two of the fragmentation algorithm, the user defines a set of requirements such as availability, serviceability, and performance, depending on the use case scenario and the data contained in the fragments. Finally, the respective requirements and confidentiality levels are assigned to the fragments in step three.

Fragmentation algorithm:

(1) *Fragmentation process.* In this step, the tables in 3NF are exported as fragments, e.g. into XML files. For the rest of the algorithm, we let $T = \{t_i, i = 1,\ldots n\}$ be the set of all fragments:

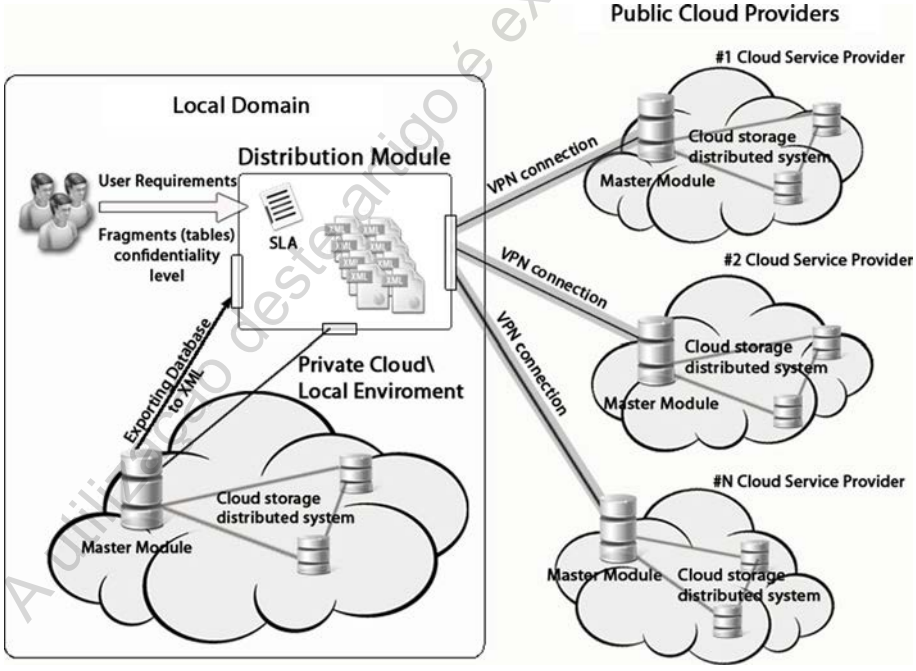

Figure 1.
Architecture of the proposed model

for i: = 1 **to** n **do**
Deploy fragment t_i as independent fragment.
end

(2) *Requirements definition.* Set of additional, user-defined distribution requirements. In this step, the user may choose to deploy certain fragments to different locations. The user requirements are given in the form (t_i, t_j), where $i \neq j$, denoting that fragments t_i and t_j should not be deployed together. Let $U' = \{(t_i, t_j)\}$ be the set of all user-set requirements, then we define U to be the set of all user requirements with $U = U' \cup \{(t, t')| (t', t) \in U'\}$.

(3) *Independent fragment deployment-fragmentation process*:

for i: = 1 **to** n **do**
Assign confidentiality level to fragment ti,
i.e. set $CL(t_i)$.
end

(4) *Selection of CSPs.* This step calculates the approximate number l of CSPs required for the distribution of the fragments and the l CSPs $P_1, \ldots, P_l$ are selected.

(5) *Assigning fragments to CSPs.* The following algorithm assigns different distribution destinations to the fragments following the user requirements. Tables that share columns and where at least one of them possesses a classification level higher than 1 may never be deployed to the same CSP:

P: = $(P_1, \ldots, P_l)$
T_c: = $\{t \in T|CF(t) > 1\}$
T_n: = $T \backslash T_c$
for $t \in T_c$ **do**
  for j: = 1 **to** l **step** 1 **do**
    if $(((t \cap t_s = \emptyset) \wedge ((t, t_s) \in U)), \forall t_s \in P_j)$
      P_j: = $P_j \cup t$
      break
    end
  end
end
for $t \in T_n$ **do**
  for j: = 1 **to** l **step** 1 **do**
    if $((t \cap t_s = \emptyset, \forall t_s \in Pj: CL(t_s) > 1) \wedge ((t, t_s) \in U, \forall t_s \in P_j))$
      Pj: = $P_j \cup t$
      break
    end

end
end

(6) *Distribution process.* This algorithm deploys the fragments to their assigned CSPs. In this step, fragments with classification level 3 are encrypted:

for $P_i \in P$ **do**
for $t \in P_i$ **do**
if CL(t) $= 3$ **then** encrypt (t) $\rightarrow$ ProviderP_i
else $t \rightarrow$ ProviderP_i
end
end

Step four calculates an upper bound for the number of CSPs required for data distribution using the relationship between primary and foreign keys (FKs). The primary key defines each row uniquely and establishes the relationship with other tables, where it is defined as a FK. The FK shows the connectivity of one table to another by mapping itself to the appropriate primary key:

$$Ns = \sum_{i=1}^{n} FK_i + 1, \quad FK_i \in \{0,1\}, \quad i \in \{0,1,2,3,\ldots n\} \tag{1}$$

where the FK_i represent foreign keys, N_s represents the maximum necessary number of cloud storage providers and n number of FK_i mapped on PK_i.

The formula in equation (1) determines the number of foreign keys that map themselves to primary keys, and, as a result, determines directly related tables as well. We add one additional unit representing the table that contains the primary key. The table with the largest number of mappings (N_s) shows how many CSPs are needed in the worst-case scenario. Once the maximum number of CSPs is known, we reduce it to the number of actually required CSPs by again evaluating the table with the most primary key-FK connections, this time together with the tables connected to it, in order to decrease the number of CSPs: any tables that cannot be correlated with each other directly by primary or FK can be distributed to the same CSP. This process gives us the minimum number of CSPs required.

We apply the fragmentation process (Figure 2) to each table, or fragment, taking into account the confidentiality level, any user-defined requirements, and the ER model, which helps visualize the relationships between tables and facilitates the fragmentation process. Each table is assigned to a distribution group for a CSP, as described in step five of the fragmentation algorithm. Step six of the fragmentation algorithm, the actual distribution process, transfers the data securely from the customer to the providers using VPN connections.

4. Real-world scenario

We have chosen a real-world scenario to demonstrate how our data fragmentation and distribution model works and to allow a more precise evaluation of our approach. Our scenario uses research centers (RCs) with a worldwide customer base that provide services for different research projects. Data from customers' works in progress is stored at the RC and is by its nature highly confidential. The RC is responsible for

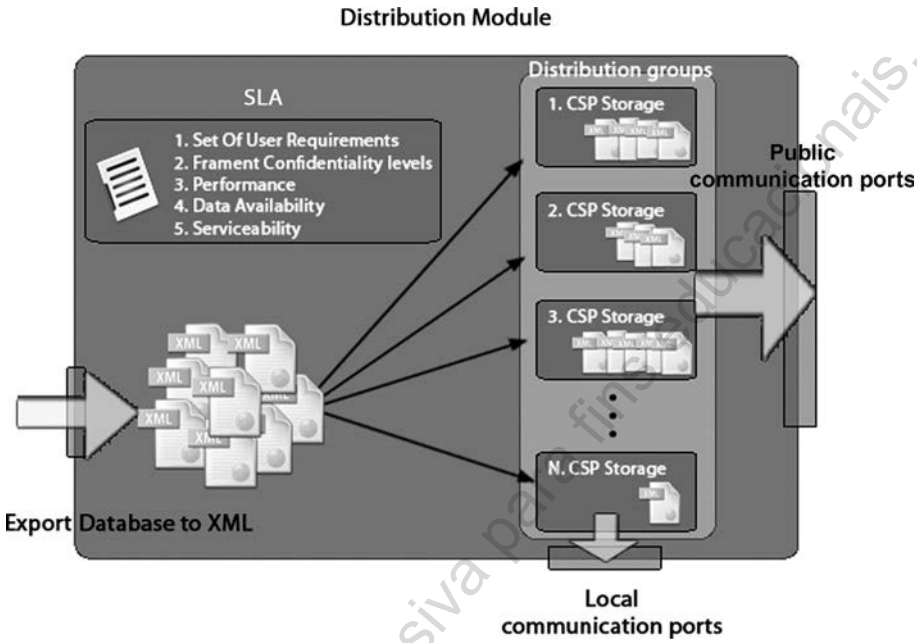

Figure 2.
Distribution module

ensuring that none of this data is leaked before publication. Storing the data securely on site, however, generates high processing and storage costs for the RC.

In our example, our RC decides to outsource its data storage to the cloud to reduce costs and internal management overhead while increasing flexibility. They must be able to guarantee that the data from confidential projects that are still in progress is stored in a way in which it cannot be compromised even in the event of a leak on the CSP side. As a solution, we propose our fragmented distribution model, which allows the data to be outsourced to a number of different cloud providers in a secure and efficient way. The ER diagram in Figure 3 shows the structure of our RC. An RC can have several offices around the world and will usually be working on several projects for different customers at the same time. Each project includes a colluding set of services and researchers. Sensitive data such as documentation of the work in progress or personal data concerning the researchers is marked to show that they must be protected from exposure.

Before beginning with the fragmentation algorithm, the database structure must be in the third normal form. Once this is ensured, the fragmentation algorithm is applied step-by-step, beginning with the generation of independent fragments deployed as XML files. Then the staff can define any necessary user requirements, which are added in the third step along with the confidentiality levels (Table VI).

In our example, the following user requirements were defined:

- low confidentiality fragments can be distributed together, regardless of cohesion; and
- highly sensitive data (bank account or credit card numbers) must be stored locally.

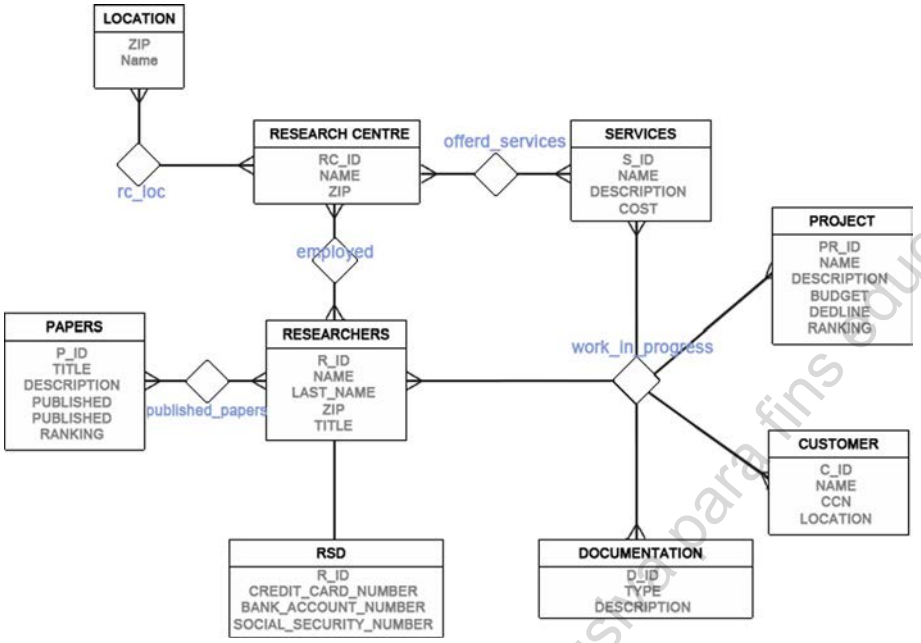

Figure 3.
Entity relationship model

Confidentiality level of fragments		
Low	Medium	High
Location	Employed_researchers	Customers
rc_loc	WorkInProgress	ResearchersSensitiveData
ResearchCenter	Project	Documentation
offerd_services	Researchers	
Services		
Papers		
Published_papers		

Table VI.
Assigned confidentiality levels

We use formula (1) to determine the maximum number of CSPs needed (in our case, $N_s = 5$). The analysis of user requirements together with the fact that several tables can safely be stored together in our approach shows that we can decrease the number of the CSPs to three. Then, each fragment is rigorously analysed and assigned a storage distribution group (step five, Table VII). Due to data privacy legislation, highly sensitive personal data such as credit card numbers, staff records, or customer records cannot be sent to the code storage providers but must instead be encrypted and kept in the local domain. Finally, the data fragments are distributed to their corresponding storage locations.

5. Related works

Research has been carried out previously on problems of outsourced storage of confidential data and some approaches have been proposed. They shall be described briefly in this section.

Chow *et al.* (2009) analyse issues related to outsourcing data to cloud storage, though they point out that many of these issues predate the concept of cloud computing. They address the problem of third-party data control, availability issues (single point of failure, uptime, ensuring computational integrity), and issues specific to the cloud (VM-level attacks, authentication and authorization, and vulnerabilities of CSPs). The authors also describe upcoming issues such as the growing pressure to provide cheap data analysis, increased authentication demands, or mash-up authorization. They advocate the use of cryptographic techniques and trusted computing.

Grobauer *et al.* (2011) identify a number of vulnerabilities that can affect data stored in cloud environments and may result in the customers' identification, authentication, or authorization being compromised or in denial of service through being locked out of an account. The issues include faulty or insufficient authorization checks and inadequate logging and monitoring options. Chen *et al.* (2010) identify data processing and the risk of unauthorized access to data as the main challenges of outsourced data storage and suggest that storage providers implement adequate technical and organizational measures to protect their users' data and ensure a certain level of trust. Takabi *et al.* (2010) illustrate how difficult it is to prevent a malicious storage provider from exploiting data that has been outsourced to its servers. They propose strong identity management with heavily separated customer identities and authentication information as well as dynamic and flexible access control services.

Ciriani *et al.* (2010) propose a model that combines fragmentation with encryption to improve data security of relational databases. In their model, fragmentation is used to break the relationships between sensitive data, such as links between attributes that an attacker should not be able to link, and is followed by whole-tuple encryption for additional security. They also attach confidential unencrypted attributes to improve the efficiency of selecting operations without searchable encryption techniques. The drawback, however, is that different attribute combinations are needed for different queries, which results in redundancies.

Damiani *et al.* (2003) suggest a model designed to balance efficiency and confidentiality. They encrypt data using (deterministic) encryption and hashing techniques to mask data elements, which they use to index the outsourced encrypted data to facilitate efficient querying. They employ hashing with reduced co-domains (intended collisions) to map values in a domain range to the same hashing output. This is a protection against statistical attacks, but the mapping must of course be kept secret and therefore stored in the local domain. However, such bucket-based approaches do not support aggregation queries such as SUMs. Bin and Yuxing (2010) propose a method that makes the management of the workload in cloud computing more efficient. The data

Table VII.
Designated distribution groups

Confidentiality level of fragments		
1. CSP	2. CSP	Local
WorkInProgress	Services	Customers
Published_papers	Project	ResearchersSensitiveData
Employed_researchers	Researchers	Documentation
rc_loc	Papers	
offerd_services	ResearchCentre	
	Location	

management scheme includes scalable, high-performance meta-data services based on parent directory path IDs, mimicking hierarchical directory structures, and cooperative double layer cache mechanisms.

6. Conclusion and outlook

Cloud computing has become a popular data storage solution that allows companies and individuals to store their data offsite to save on hardware and processing costs. However, the security of cloud storage usually only goes as far as the CSP can be trusted. While SLAs can ensure some level of trust, they cannot guarantee that the data is completely safe from the curious eyes of a CSP employee without additional measures. This paper presents a method for secure and confidential storage of data in the cloud environment based on fragmentation and only minimal encryption in order not to compromise on efficiency.

Our proposed method uses normalization of relational databases, whose tables are categorized by user requirements such as performance, availability and serviceability, and exported to XML as fragments. After defining the fragments and assigning the appropriate confidentiality levels, the lowest number of CSPs required to store all fragments that must remain unlinkable in separate locations is calculated and the fragments are uploaded to their designated CSPs using a VPN.

We aim to evaluate our approach thoroughly with long-term experiments and simulations of performance concerning parameters such as availability, costs, scalability, and others. Due to the flexibility of CSPs, data distribution tests will be easy to perform. However, the more complex tests to examine the scalability on several parallel CSPs with data distribution between them will require considerable time and effort. The results of the evaluation shall therefore be discussed in a future publication, where we will also analyse network limitations and describe how an appropriate test environment can be set up.

References

Abdalla, M., Bellare, M., Catalano, D., Kiltz, E., Kohno, T., Lange, T., Malone-Lee, J., Neven, G., Paillier, P. and Shi, H. (2008), "Searchable encryption revisited: consistency properties, relation to anonymous IBE, and extensions", *J. Cryptol.*, Vol. 21 No. 3, pp. 350-91.

Amazon (2006), *Amazon Elastic Compute Cloud* (*amazon ec2*), available at: http://aws.amazon.com/ec2/ (accessed May 1, 2012).

Bin, H. and Yuxing, P. (2010), "A novel metadata management scheme in cloud computing", *International Conference on Software Technology and Engineering* (*ICSTE*), pp. V1-433-8.

Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J. and Brandic, I. (2009), "Cloud computing and emerging it platforms: vision, hype, and reality for delivering computing as the 5th utility", *Future Gener. Comput. Syst.*, Vol. 25 No. 6, pp. 599-616.

Chen, Y., Paxson, V. and Katz, R.H. (2010), "Whats new about cloud computing security?", Tech. Report UCB/EECS-2010-5, EECS Department, University of California, Berkeley, CA, January.

Chow, R., Golle, P., Jakobsson, M., Shi, E., Staddon, J., Masuoka, R. and Molina, J. (2009), "Controlling data in the cloud: outsourcing computation without outsourcing control", *Proceedings of the 2009 ACM Workshop on Cloud Computing Security*, ACM, New York, NY, pp. 85-90.

Ciriani, V., De Capitani Di Vimercati, S., Foresti, S., Jajodia, S., Paraboschi, S. and Samarati, P. (2010), "Combining fragmentation and encryption to protect privacy in data storage", *ACM Trans. Inf. Syst. Secur.*, Vol. 13 No. 3, pp. 22:1-22:33 .

Curran, K., Carlin, S. and Adams, M. (2011), "Security issues in cloud computing", *Elixir*, Vol. 38 No. 1, pp. 4069-72.

Damiani, E., De Capitani Vimercati, S., Jajodia, S., Paraboschi, S. and Samarati, P. (2003), "Balancing confidentiality and efficiency in untrusted relational DBMSs", *Proceedings of the 10th ACM Conference on Computer and Communications Security, CCS '03*, ACM, New York, NY, pp. 93-102.

Electronic Privacy Information Center (1972), *The Code of Fair Information Practices*, available at: www.epic.org/privacy/consumer (accessed March 8, 2012).

Google (2008), "Google app engine", available at: http://code.google.com/appengine/ (accessed July 1, 2011).

Greenberg, A., Hamilton, J., Maltz, D.A. and Patel, P. (2008), "The cost of a cloud: research problems in data center networks", *SIGCOMM Comput. Commun. Rev.*, Vol. 39 No. 1, pp. 68-73.

Grobauer, B., Walloschek, T. and Stocker, E. (2011), "Understanding cloud computing vulnerabilities", *IEEE Security and Privacy*, Vol. 9 No. 2, pp. 50-7.

IBM (2011), *IBM Smart Cloud*, available at: www.ibm.com/cloud-computing/us/en/ (accessed November 8, 2011).

Islam, S., Mouratidis, H. and Jürjens, J. (2011), "A framework to support alignment of secure software engineering with legal regulations", *Journal of Software and Systems Modeling (SoSyM), Theme Section on Non-functional System Properties in Domain-Specific Modeling Languages (NFPinDSML)*, Vol. 10 No. 3, pp. 369-94.

Islam, S., Mouratidis, H. and Weippl, E. (2012), *A Goal-Driven Risk Management Approach to Support Security and Privacy Analysis of Cloud-Based System, Security Engineering for Cloud Computing: Approaches and Tools*, IGI Publication, Hershey, PA.

Kaufman, L.M. (2009), "Data security in the world of cloud computing", *IEEE Security and Privacy*, Vol. 7 No. 4, pp. 61-4.

Microsoft (2011), "Microsoft cloud", available at: www.microsoft.com/en us/cloud/default.aspx?fbid=K0xFa7eU XmQ (accessed May 8, 2012).

Morali, A. and Wieringa, R. (2010), "Risk-based confidentiality requirements specification for outsourced it systems", *Proceedings of the 2010 18th IEEE International Requirements Engineering Conference, RE '10*, IEEE Computer Society, Washington, DC, pp. 199-208.

Mouratidis, H., Kalloniatis, C., Islam, S., Philippe Huget, M. and Gritzalis, S. (2012), "Aligning security and privacy to support the development of secure information systems, special issue security in information systems", *Journal of Universal Computer Science* (*JUCS*), Vol. 18 No. 12, pp. 1608-27.

Nishchal, N. and Mathur, P. (2010), "Cloud computing: new challenge to the entire computer industry", *Parallel Distributed and Grid Computing* (*PDGC*), *2010 1st International Conference*, pp. 223-8.

Oracle (2010), *Oracle Cloud Computing White Paper*, available at: www.oracle.com/us/technologies/cloud/oracle-cloud-computing-wp-076373.pdf (accessed October 14, 2012).

Peter, M. and Grance, T. (2011), "The NIST definition of cloud computing (draft)", NIST Special Publication 800-145, National Institute of Standards and Technology, Gaithersburg, MD.

Proofpoint (2010), "Outbound email and data loss prevention in today's enterprise", Tech. Report, Proofpoint.

Randell, B. (1969), "A note on storage fragmentation and program segmentation", *Communications of the ACM*, Vol. 12 No. 4, pp. 365-9.

Rangan, P.V. and Vin, H.M. (1993), "Efficient storage techniques for digital continuous multimedia", *IEEE Trans. Knowl. Data Eng.*, Vol. 5 No. 4, pp. 564-73.

Takabi, H., Joshi, J.B.D. and Ahn, G.-J. (2010), "Security and privacy challenges in cloud computing environments", *IEEE Security and Privacy*, Vol. 8 No. 6, pp. 24-31.

Umanath, N.S. and Scamell, R.W. (2007), *Data Modeling and Database Design*, Course Technology Press, Boston, MA.

Vaquero, L.M., Rodero-Merino, L., Caceres, J. and Lindner, M. (2008), "A break in the clouds: towards a cloud definition", *SIGCOMM Comput. Commun. Rev.*, Vol. 39 No. 1, pp. 50-5.

Widman, J. (2011), "10 massive security breaches", available at: www.informationweek.com/news/galleries/security/attacks/229300675 (accessed July 8, 2012).

Further reading

Mell, P. and Grance, T. (2009), "The NIST definition of cloud computing", *National Institute of Standards and Technology*, Vol. 53 No. 6, p. 50.

About the authors

Aleksandar Hudic received his BSc degree in Computer Engineering and MSc degree in Telecommunication and Informatics at the Faculty of Electrical Engineering and Computing, University of Zagreb. Before joining SBA he worked at MBU ERSTE Bank, mainly in the field of database management systems and network security. Aleksandar Hudic is the corresponding author and can be contacted at: ahudic@sba-research.org

Shareeful Islam is currently working at the School of ACE, University of East London, UK. He was awarded his PhD for a thesis on a Software Risk Management Model using a goal-driven approach by the Chair of Software and Systems Engineering (I4), Technische Universität München, Germany. He received MSc in Information Communication System Security from the Royal Institute of Technology (KTH), Sweden and MSc in CS and BSc (Hons) in APE from the University of Dhaka, Bangladesh. He is a Fellow of the British Higher Education Academy (HEA) and has published more than 30 referred papers in high-quality journals and international conferences. He participated in EU, industry, KTP projects. His research interests and fields of expertise are risk management, requirements engineering, security, privacy, and trust.

Peter Kieseberg studied Mathematics in Computer Science at the Vienna University of Technology, specializing on cryptography and numerics. Before joining Secure Business Austria, he had been working as a Consultant in the telecommunication sector, mainly in the fields of interconnection billing and data warehousing. Currently his research interests include database forensics, fingerprinting, and application of cloud technologies.

Sylvi Rennert is based at SBA Research gGmbH, Vienna, Austria.

Edgar R. Weippl (CISSP, CISA, CISM, CRISC, CSSLP, CMC) is Research Director of SBA Research and Associate Professor (Privatdozent) at the Vienna University of Technology. His research focuses on applied concepts of IT security and e-learning. He is member of the Editorial Board of Computers and Security (COSE) and organizes the ARES conference. After graduating with a PhD from the Vienna University of Technology, he worked for two years in a research startup. He then spent one year teaching as an Assistant Professor at Beloit College, Wisconsin. From 2002 to 2004, while with a software vendor, he worked as a Consultant in New York, New York and Albany, New York, and in Frankfurt, Germany. In 2004 he joined the Vienna University of Technology and founded the research center SBA Research together with A Min Tjoa and Markus Klemen.