Sharing-based Privacy and Availability of Cloud Data Warehouses

Varunya Attasena, Nouria Harbi, Jérôme Darmont

Sharing-based Privacy and Availability of Cloud Data Warehouses

Varunya Attasena, Nouria Harbi and Jérôme Darmont

Université de Lyon (ERIC Lyon 2)
varunya.attasena@eric.univ-lyon2.fr
nouria.harbi@univ-lyon2.fr
jerome.darmont@univ-lyon2.fr

Abstract. Cloud computing can help reduce costs, increase business agility and deploy applications with a high return on investment such as data warehouses. However, storing and managing data in the cloud may not be fully trustworthy. In this article, we focus on both data security (data privacy, availability and integrity) and data analysis in the cloud. To solve the data security issue, we propose a new (m, n, t) multi secret sharing scheme based on block cryptography, secret sharing and hash functions. Moreover, we apply this solution onto a cloud data warehouse such that data security and data analysis are addressed. An extensive security and performance analysis shows that the proposed schemes can prevent most attacks, guarantee data availability and integrity, and allow analyzing data at low costs (data storage, data transfer and time computation) in the pay-as-you-go economic model in the cloud.

1 Introduction

Business intelligence (BI) and data analytics have been an ever-growing trend in many business (e.g., finance, telecoms, insurance, logistics...) and non-business (e.g., agriculture, medicine, health and environment...) domains for more than twenty years. The more recent advent of cloud computing now theoretically allows to deploy data analytics more easily. Data that would have earlier been too costly to process in time, money or human resources can be analyzed efficiently and at lower costs. Building a traditional BI system indeed typically necessitates an important initial investment. By contrast, with the cloud pay-as-you-go model, users punctually devote small amounts of resources in return for a one-time advantage. This trend is currently supported by numerous "BI as a service" offerings by both cloud start-ups and major BI industry vendors, with high economic stakes.

Moreover, the elasticity characteristic of cloud computing, i.e., the dynamic on-demand provisioning of resources, does not only help scale performance up or down, but also enables to dynamically bring in new data sources to meet emerging needs for new analyses. Thus, data analytics is likely to be increasingly demanded by independent actors grouping together to achieve a temporary common goal through a collaborative community effort. For instance, open data, which are easily accessed from the Web, are in high demand. They could be integrated to private data and cross-analyzed with intelligible on-line tools featuring advanced

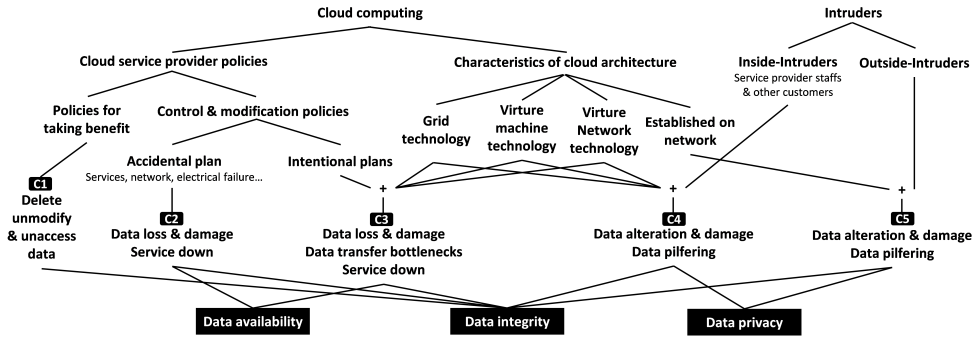


FIG. 1 – *Cloud data security issues.*

collaborative capabilities that enable new users to share and reuse BI concepts and analyses on a large scale, and to share the results with a group of interest of even worldwide. Such cloud BI users could be companies, cooperatives, NGOs or even citizens. Thus, cloud analytics is not only a current technological and research challenge, but also an important societal stake.

Although cloud computing is currently booming, data security remains one of the top concerns for cloud users and would-be users. Some security issues in the cloud are inherited from classical distributed architectures (e.g., authentication, network attacks, vulnerability exploitation...), but some directly relate to the new framework of the cloud (e.g., cloud service provider or subcontractor espionage, cost-effective defense of availability, uncontrolled mashups...) (Chow et al., 2009). In the particular context of cloud BI, privacy is of course of critical importance. Up to now, security issues have been handled by cloud service providers. But with the multiplication of cloud service providers and subcontractors in many different countries, intricate legal issues also arise, as well as another fundamental issue: trust. Telling whether trust should be placed in cloud service providers eventually falls back onto end-users, with the implied costs.

Many security issues are raised by data storage in a public cloud, including data privacy, data availability, data integrity, data backup and recovery, and data transfer safety. Moreover, security risks may come from both cloud service providers and intruders (Figure 1). While cloud data warehouses should support both data analysis and security (Chow et al., 2009), solving both issues is a great challenge that involves a trade-off between the level of security, and storage and computation costs.

In this paper, we address the data privacy, availability and integrity issues as well as allowing data analysis at the lowest possible cost. To this aim, we propose a new (m, n, t) multi secret sharing scheme called Scheme-I. It is then extended to apply onto a cloud data warehouse in Scheme-II.

The rest of this paper is organized as follows. Section 2 introduces related works. Section 3 presents the principle of our two schemes. Sections 4 and 5 present Scheme-I and Scheme-II, respectively. Section 6 gives the security analysis and performance evaluation. Finally, Section 7 concludes this paper and provides future research perspective.

2 Related works

Data security may be enforced through several means. First, encryption turns original data into an unreadable cipher-text. Modern encryption schemes, such as homomorphic encryption (Melchor et al., 2008; Gentry, 2009) and incremental encryption (Bellare et al., 1994, 1995), help perform computation and modification, respectively, on cipher-texts without decrypting them first. However, they are currently not efficient enough to process data on a very large scale. An older, well-known encryption strategy is secret sharing (Shamir, 1979; Blakley, 1979; Asmuth and Bloom, 1983), which distributes data amongst a group of participants. Each participant cannot learn anything and cannot reconstruct the secret from his share. The secret can be reconstructed only when a sufficient number of shares are combined together. Although aimed at enforcing privacy, a nice side effect of secret sharing is data availability, since if some participants disappear, the secret can still be reconstructed. The drawback of this solution is the multiplication of the initial data volume by the number of participants/shares. Modern secret sharing schemes, such as multi secret sharing (Yang et al., 2004; Chan and Chang, 2005; Parakh and Kak, 2009; Hu et al., 2012; Wu et al., 2012) and verifiable secret sharing (Hwanga and Chang, 1998; Changa et al., 2005; Shao and Cao, 2005; Zhao et al., 2007; Dehkordi and Mashhadi, 2008a,b), help reduce the volume of shares and verify the honesty of each participant, respectively.

Data anonymization (Cormode and Srivastava, 2009; Kenneally and Claffy, 2010; Sweeney, 2002; Machanavajjhala et al., 2007) can also be used to enforce privacy, by preventing the identification of key information. For example, k-anonymized (Sweeney, 2002) and l-diversity (Machanavajjhala et al., 2007) models make k indistinguishable records and l different sensitive values in each key identification combination, respectively. While cheap when performing analyses, anonymization is not strong enough to protect against some attacks such as homogeneity and background knowledge attacks.

Data replication (Padmanabhan et al., 2008) is the process of copying some or all parts of data from one location to one or several other locations. Its main purposes are to improve availability, fault-tolerance, and/or accessibility, but it does not handle privacy issues. Eventually, data verification (Shacham and Waters, 2008; Wang et al., 2009; Juels and Kaliski, 2007; Bowers et al., 2009) is the process of checking data integrity, by verifying data corruption caused by either accident or intruder attack, with the help of signatures (digital signature, message authentication, fingerprint...). However, since signature creation methods typically involve random or hash functions, they cannot guarantee 100% data correctness. Finally, note that so-called outer code verifying methods (Juels and Kaliski, 2007) allow checking encrypted data without decrypting them first.

In conclusion, though there are many approaches that address data privacy, availability, integrity and analysis, none can solve all issues at once.

3 Principle of our schemes

The solution we propose in this paper does enforce data privacy, availability, integrity and analysisy. This based on trusting neither cloud service providers nor network data transfers. By proposing a new (m, n, t) multi secret sharing scheme (Scheme-I) based on block cryptography, secret sharing (Shamir, 1979) and hash functions, we transform data to blocks and

share them at several providers'. Each provider only stores part of the data blocks (called shares), which are not exploitable (i.e., transformed by a mathematical function), neither by the provider nor any intruder. Performing cheap computations on shares is possible (data need not be decrypted), though, but it yields meaningless results. It is only when all results are mathematically transformed back at the user's that they can be reconstructed into global, meaningful information. Since individual shares and computed results have no value, network transfers to and from providers are safe. Hence, privacy is achieved at any point outside of the user's (network, providers).

Scheme-I operates with respect to two parameters: n, the total number of providers; and t, the number of providers required to reconstitute the data ($t \leq n$). A nice side effect of Scheme-I is that if up to $n - t$ providers fail or disappear, the data can still be reconstructed, thus enforcing availability. Moreover, to verify the honesty of providers and the correctness of shares, we propose two types of signatures created by hash functions. The first one is an inner signature created from all data in each block. It helps verify data correctness in case some providers are not honest. The second one is an outer signature created from each piece of encrypted data. It helps verify incorrect or erroneous (lost, damaged, alternative...) data before decryption and prevents useless data transfers.

Scheme-II applies onto a cloud data warehouse so that each attribute value in each record is encrypted independently. This scheme first transforms each attribute value to at least one block, and then encrypts each data block with Scheme-I. Thus, Scheme-II guarantees data privacy, availability and integrity; and allows analyzing data over shares without decrypting them first.

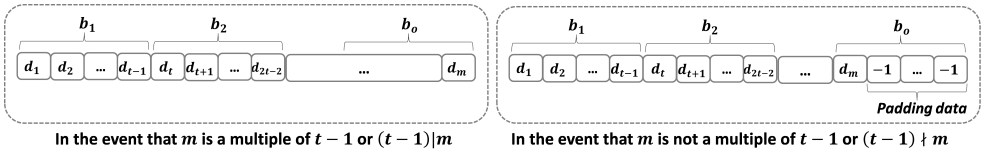


FIG. 2 – *Organization of data in blocks*

4 Scheme-I: A new (m, n, t) multi secret sharing scheme

In this scheme, m pieces of data are encrypted and shared among n cloud service providers (CSPs). t out of n shares are sufficient for reconstructing the original data. To reduce both of computation and space costs, data are organized into blocks. Each block is encrypted and decrypted all at once. The priorities of blocks and data in the block are important. All pieces of data in the block are encrypted by mapping them and their signature to coefficients of a polynomial equation of degree $t - 1$ ($y = f(x)$). y is an encrypted piece of data. x is a CSP identifier number. There are two types of signatures in this scheme. The first signature is created from all pieces of data in the block. It helps verify the honesty of CSPs. The second signature is created from each piece of encrypted data. It helps reduce the cost of data transfer in the reconstructing process, because no erroneous encrypted data is transfered.

Parameters	Definitions
n	Number of CSPs
CSP_k	CSP number k
m	Number of pieces of data
o	Number of data blocks
t	Number of shares necessary for reconstructing original data
P	A big prime number
D	Original data such that $D = \{d_1, \ldots, d_m\}$ and $D = \{b_1, \ldots, b_o\}$
d_i	The i^{th} piece of D in integer format such that $0 < d_i < P - 2$
b_j	The j^{th} block of D such that $b_j = \{d_{(j-1)(t-1)}, \ldots, d_{(j)(t-1)}\}$
ID_k	Identifier number of CSP_k such that $ID_k > 0$
$e_{j,k}$	Encrypted data of b_j stored at CSP_k
s_in_j	Signature of original data in b_j such that $0 < s\_in_j < P - 2$
$s\_out_{j,k}$	Signature of encrypted data of b_j stored at CSP_k

TAB. 1 – *Scheme-I parameters*

Parameters of Scheme-I are listed in Table 1. $ID_{i=1..m}$ are randomly selected from distinct integers and are stored at the user. D is split to o blocks with $o = \left\lceil \frac{m}{t-1} \right\rceil$. If m is not a multiple of $t-1$, the last block is padded by the integer values -1 (Figure 2).

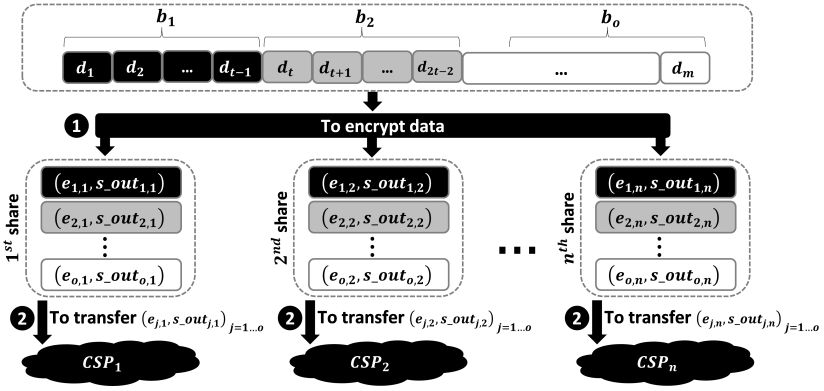


FIG. 3 – *Data sharing process in Scheme-I*

4.1 Data sharing process

In Scheme-I (Figure 3), each data block is encrypted independently. Pieces of data in block b_j are encrypted as follows.

1. Compute s_in_j from b_j with a hash function: $s\_in_j = H_1(b_j)$.

2. Create a polynomial equation of degree $t-1$ (Equation 1).

$$f_j(x) = \left(\sum_{l=1}^{t-1} \left(2 + d_{(j-1)(t-1)+l}\right) x^l + s\_in_j \right) \bmod P \tag{1}$$

3. Compute $\{e_{j,k}\}_{k=1\ldots n}$ as $e_{j,k} = f_j\left(ID_k\right)$.
4. Compute $\{s\_out_{j,k}\}_{k=1\ldots n}$ by another hash function: $s\_out_{j,k} = H_2\left(e_{j,k}\right)$.
5. Distribute each $e_{j,k}$ and $s\_out_{j,k}$ to CSP_k. Thus, encrypted data and their signature are shared among n providers. CSP_k stores o pairs of encrypted pieces of data and signatures ($\left(e_{j,k}, s\_out_{j,k}\right)_{j=1\ldots o}$).

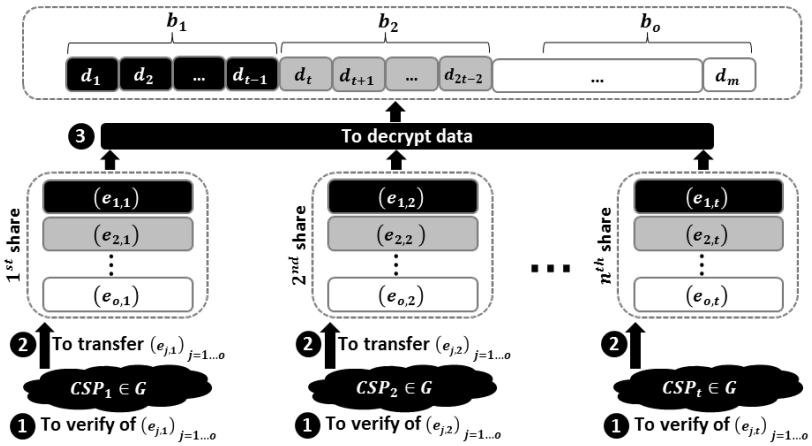


FIG. 4 – *Data reconstruction process in Scheme-I*

4.2 Data reconstruction process

In Scheme-I (Figure 4), D is reconstructed from $\left(e_{j,k}, s\_out_{j,k}\right)_{j=1\ldots o}$ at all $CSP_k \in G$ (group of t CSPs). Their are two phases to reconstruct original data: the initialization phase and the reconstruction phase.

4.2.1 Initialization phase

In this phase, the correctness of encrypted data is verified and a matrix C that is used in the reconstruction phase is created as follows.

1. Verify information at all $CSP_k \in G$. At each CSP's, only pieces of data to be decrypted are verified for correctness. $e_{j,k}$ is correct if $s\_out_{j,k} = H_2\left(e_{j,k}\right)$. In case of error at CSP_k, then another CSP is selected and its information is verified again.
2. In the user, matrix A is created from ID_k of $CSP_k \in G$ as $A = [a_{x,y}]_{t\times t}$, where $a_{x,y} = (ID_x)^{y-1} \bmod P$. Then, C is computed as $C = A^{-1} \bmod P$. Let $c_{x,y}$ be an entry in the x^{th} row and the y^{th} column of matrix C.

4.2.2 Reconstruction phase

In this phase, original data are reconstructed. To decrypt b_j, $e_{j,k}$ of $CSP_k \in G$ are transfered to the user and decrypted as follows.

1. Compute b_j and s_in_j as in Equations 2 and 3, respectively.

$$d_{(j-1)(t-1)+l} - 2 \equiv \sum_{h=1}^{t} c_{l+1,h} \times e_{j,h} \,(\mathrm{mod} P)\,; \forall l\,[1, t-1] \tag{2}$$

$$s\_in_j = \sum_{h=1}^{t} c_{1,h} \times e_{j,h} \,(\mathrm{mod} P) \tag{3}$$

2. Verify data. If $s\_in_j = H_1\,(b_j)$, then all data in b_i are correct. In case of errors, the user can reconstruct data again from encrypted data from a new G.

5 Scheme-II: Sharing a data warehouse in the cloud

In this section, we propose a solution to share data from a data warehouse among CSPs with regard to Scheme-I and provide a raw framework for performing data analysis over shares. However some interesting topics including complex predicates (conjunction, negation, disjunction...) and data aggregation over multiple encrypted tables are moved to future research. Database attribute values are encrypted and shared on relational databases at CSPs'. Figure 5 shows an example of customer table that is shared among three CSPs. To handle and utilize data in this data warehouse, we propose solutions to encrypt data from various data types (Section 5.1), to analyze data (Section 5.2) and to perform some fundamental data management processes (Section 5.3).

id	name	salary	sex
124	Bob	75€	M
125	Anna	80€	F

(a) Original data

id	name	salary	sex
124	(0,0),(10,3),(11,4)	(3,3)	(9,2)
125	(6,6),(10,3),(10,3),(0,0)	(0,0)	(10,3)

(b) Encrypted data at CSP_1

id	name	salary	sex
124	(6,6),(6,6),(2,2)	(5,5)	(9,2)
125	(2,2),(5,5),(5,5),(0,0)	(3,3)	(7,0)

(c) Encrypted data at CSP_2

id	name	salary	sex
124	(2,2),(0,0),(0,0)	(10,3)	(1,1)
125	(12,5),(11,4),(11,4),(0,0)	(11,4)	(7,0)

(d) Encrypted data at CSP_3

FIG. 5 – *Example of original and encrypted data at different CSPs'*

5.1 Data types

To handle the usual data types featured in databases (i.e., integers, dates, timestamps, reals, characters and texts), we propose that each piece of data can be encrypted and handled independently. In this solution, pieces of data in each data type are first transformed with different methods to integers, and then split into a data block encrypted with Scheme-I.

1. **Integers, dates and timestamps.** For sharing an integer D, it is split into $t-1$ pieces $d_{i=1\ldots t}$ such that $d_i = \left\lfloor \frac{D}{p^{i-1}} \right\rfloor \bmod p$ where p is a prime number and $\|p\| > \frac{\|maxint\|}{t-1}$, where $\|maxint\|$ is the size of maximum integer value. Then, d_i can be encrypted with Scheme-I.
2. **Reals.** For sharing a real D, it is transformed to an integer D' by multiplication bysome value. For example, let D be stored in numeric(p,s), where p is a precision value and s a scale value. Then, D is transformed as $D' = D \times 10^s$. After being transformed, D' is encrypted as an integer.
3. **Characters.** For sharing a character D, it is transformed into an integer D' by using its ASCII code. For example, let D be 'A'. D is transformed to $D' = 65$. After being transformed, D' is encrypted as an integer.
4. **Texts.** For sharing a text D, it is transformed into integers D' by using the ASCII code of each character of D. For example, let D be 'ABC'. Then, D is transformed to $D' = \{65, 66, 67\}$. After being transformed, each character of D' is encrypted independently as an integer.

An example of sharing and reconstruction processes of an integer (p) data is shown as follows.

Example: Sharing an integer

1. Suppose parameters are assigned as follows: $n = 4$, $t = 3$, $p = 11$, $P = 13$, $ID_1 = 3$, $ID_2 = 4$, $ID_3 = 5$ and $ID_4 = 6$.
2. Suppose hash functions are $H_1(b_j) = \prod_{d_i \in b_i} d_i \,(\mathrm{mod} P)$ and $H_2(e_{j,k}) = e_{j,k} \bmod 7$, respectively.
3. Suppose the integer to share is 75 (Bob's salary on Figure 5(a)).
4. We compute $d_{i=1,2}$ as follows: $d_1 = \left\lfloor \frac{75}{11^{1-1}} \right\rfloor \bmod 11 = 9$ and $d_2 = \left\lfloor \frac{75}{11^{2-1}} \right\rfloor \bmod 11 = 6$.
5. We compute s_in_1: $s\_in_1 = H_1(b_1) = 9 \times 6 \bmod 13 = 2$.
6. We create a polynomial equation: $f_1(ID_i) = \left((9+2) \times ID_i + (6+2) \times ID_i^2 + 2\right) \bmod 13 = \left(11 \times ID_i + 8 \times ID_i^2 + 2\right) \bmod 13$.
7. We compute $e_{1,k=1\ldots4}$ such as $e_{1,1} = \left(11 \times 3 + 8 \times 3^2 + 2\right) \bmod 13 = 3$. Once computed, $e_{1,1} = 3$, $e_{1,2} = 5$, $e_{1,3} = 10$ and $e_{1,4} = 5$.
8. We compute $s\_out_{1,k=1\ldots4}$ such as $s\_out_{1,1} = H_2(3) = 3 \bmod 7 = 3$. Once computed, $s\_out_{1,1} = 3$, $s\_out_{1,2} = 5$, $s\_out_{1,3} = 3$ and $s\_out_{1,4} = 5$.
9. We distribute each $(e_{1.k}, s\_out_{1,k})$ to CSP_k.

Example: Reconstructing an integer

1. Suppose CSP_1, CSP_2 and CSP_4 are selected into G.
2. We verify $s\_out_{1,j=1,2,4}$ such as $s\_out'_{1,1} = H_2(3) = 3 \bmod 7 = 3 = s\_out_{1,1}$. Then $e_{1,1}$ is correct. Once the three shares are verified, all of $\{e_{1,1}, e_{1,2}, e_{1,4}\}$ are correct.
3. We create the matrix A from $ID_{i=1,2,4}$:
$A = \begin{bmatrix} 3^{1-1} & 3^{2-1} & 3^{3-1} \\ 4^{1-1} & 4^{2-1} & 4^{3-1} \\ 6^{1-1} & 6^{2-1} & 6^{3-1} \end{bmatrix} \bmod 13 = \begin{bmatrix} 1 & 3 & 9 \\ 1 & 4 & 3 \\ 1 & 6 & 10 \end{bmatrix}$.
4. We compute the matrix B: $B = A^{-1} \bmod P = \frac{\begin{bmatrix} 9 & 11 & 12 \\ 6 & 1 & 6 \\ 2 & 5 & 1 \end{bmatrix}}{6} (\mathrm{mod} 13)$.
5. We compute $d_{i=1,2}$ as follows:
   (a) $d_1 + 2 \equiv (6 \times 3 + 1 \times 5 + 6 \times 5)/6 (\mathrm{mod} 13)$. Then $d_1 = 9$.
   (b) $d_2 + 2 \equiv (2 \times 3 + 5 \times 5 + 1 \times 5)/6 (\mathrm{mod} 13)$. Then $d_2 = 6$.
6. We compute s_in_1: $s\_in_1 \equiv (9 \times 3 + 11 \times 5 + 12 \times 5)/6 (\mathrm{mod} 13) = 2$.
7. We verify the original data. The result is correct because $s\_in'_1 = H_1(b_1) = 9 \times 6 \bmod 13 = 2 = s\_in_1$.

5.2 Data analysis over shares

Scheme-II can analyze data (search and aggregation operations) over shares while not decrypting all data first. For a search operation, Scheme-II searches for the records that match with an encrypted keyword from t shares. Once they are transfered to the user, some are deleted if less than t shares are transfered. Only remainder records are decrypted. An example of search operation follows.

1. Let the search query be "select name from customer where sex='M'" and consider the customer table from Figure 5.
2. Keyword D='M' is encrypted with Scheme-II. Then $(e_{1.k}, s\_out_{1,k})_{k=1...3}$ are $(9, 2)$, $(9, 2)$ and $(1, 1)$, respectively.
3. We create a query for each share, for example "select id, name from customer where sex=(9,2)" is computed at CSP_1.
4. We compute each query at each CSP_k and transfer the matching record to the user. Only the first record of each share is transfered.
5. In this case, no record is deleted.
6. We reconstruct data with Scheme-II.

We currently handle two groups of aggregation operations. Average operations can apply on shares directly. For example, the query is "select avg(salary) from customer". Query is computed for each share and only the result of them is transfered to the user for decryption. Summation, maximum and minimum operations are computed as a search operation.

5.3 Load, backup and recovery processes

Scheme-II supports three fundamental processes for managing data warehouse: load, backup and recovery processes. For loading data into a data warehouse, each piece of data is encrypted and loaded independently. New data are loaded by without decrypting previous data first (Figure 6; data from Figure 5 are previous data and the last record is new). A backup process is actually unnecessary because each share is a backup share of other shares. In the case that some shares are erroneous, they are recovered from other t shares.

id	name	salary	sex
124	Bob	75€	M
125	Anna	80€	F
126	Kris	95€	F

(a) Original data

id	name	salary	sex
124	(0,0),(10,3),(11,4)	(3,3)	(9,2)
125	(6,6),(10,3),(10,3),(0,0)	(0,0)	(10,3)
126	(3,3),(10,3),(8,1),(10,3)	(4,4)	(10,3)

(b) Encrypted data at CSP_1

id	name	salary	sex
124	(6,6),(6,6),(2,2)	(5,5)	(9,2)
125	(2,2),(5,5),(5,5),(0,0)	(3,3)	(7,0)
126	(5,5),(9,2),(2,2),(10,3)	(5,5)	(7,0)

(c) Encrypted data at CSP_2

id	name	salary	sex
124	(2,2),(0,0),(0,0)	(10,3)	(1,1)
125	(12,5),(11,4),(11,4),(0,0)	(11,4)	(7,0)
126	(10,3),(6,6),(5,5),(8,1)	(0,0)	(7,0)

(d) Encrypted data at CSP_3

FIG. 6 – *Example of original and encrypted data after insertion*

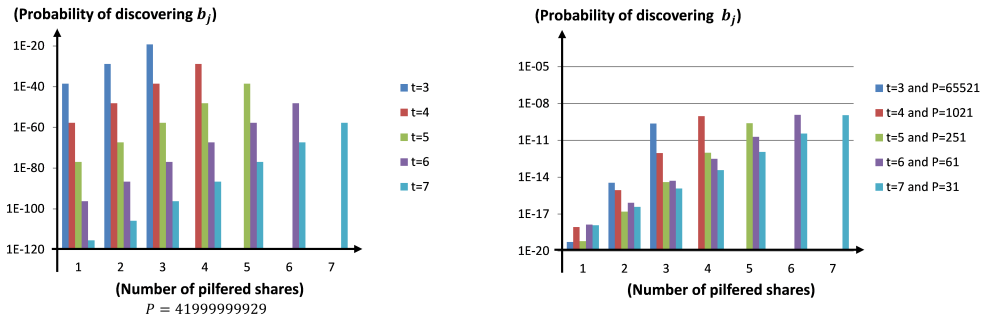
6 Security analysis and performance evaluation

6.1 Security analysis

Our security analysis focuses on data pilfering both from CSPs and intruders. The data encrypted by our schemes is not easy to decode because they are shared among many CSPs. Neither the CSP nor any intruder can decode the original data from only one share, and it is very difficult to retrieve shares from all CSPs' by attacking them simultaneously.

In the case that an intruder can steal shares from x CSPs such that $x \leq t$, the probability of discovering b_j is very low. The probabilities in Scheme-I and Scheme-II are indeed $\frac{1}{P^{2t-x-1}}$ and $\frac{1}{p^{2t-x-1}}$, respectively (Figure 7). The probability of discovering b_j (the original data in j^{th} block) depends on the following.

1. The size of data controlled by a value of P in Scheme-I and p in Scheme-II. In Scheme-I, the probability is very low because P is a big prime number. In Scheme-II, the probability ranges between 10^{-20} and 10^{-8} because p depends on t.
2. The value of t defined by the user. The higher t, the lower the probability of breaking the secret (other parameters are fixed).
3. The number of pilfered shares. The probability increases with the number of shares an intruder can pilfer.



(a) Scheme-I (b) Scheme-II

FIG. 7 – *Probability of discovering an original data block from some or all shares*

However, some data can be decrypted. But in Scheme-II, if an intruder attempts to generate all possible cases from t shares by a brute-force algorithm and finds the data block pattern from them, it would take more than 38 years (2 weeks per block) if the sufficient number of blocks is 1000, t is equal to 3, with a computer similar to that used in Section 6.3. In addition, parameter assignment affects the security of our schemes. Notably, big values should be assigned to P and t in Scheme-I and p and t in Scheme-II.

6.2 Reliability analysis

Our reliability analysis focuses on data availability, data integrity and data recovery. Our schemes guarantee the user can reconstruct D if k or more CSPs are honest and their shares are accessible. Moreover, our schemes can verify both the honesty of CSPs and the correctness of CSPs' shares. The verification performance depends on two hash functions defined by the user. As plotted in Figure 8, in case p is a big prime number and signature size, the probability of incorrect data not being detected is nearly zero. If some shares are erroneous (lost, damaged, alternative...), they are reconstructed from t other shares. Thus, backups are unnecessary when using our schemes.

6.3 Cost analysis

The most important advantage of cloud computing is that the users pay only what they consume. Cost mainly depends on disk space used and processing time. Our cost analysis focuses on three factors: time complexity, stored data volume and transfered data volume.

We implement Scheme-II under 100 test cases such that the volume of each case is 1 GB. The input of each test case are random 32 bits unsigned integers. Our experiment is conducted on a PC with an Intel(R) Core(TM) i5 processor running 2.76 GHz, 3 GB of RAM.

1. **Time complexity:** The time complexity of the data sharing process (Section 4.1) and the data reconstruction process (Section 4.2) in both schemes are $O(otn)$ and $O\left(ot^2\right)$,

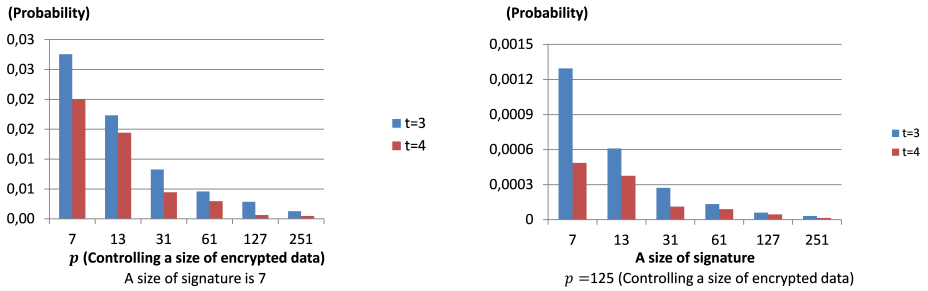
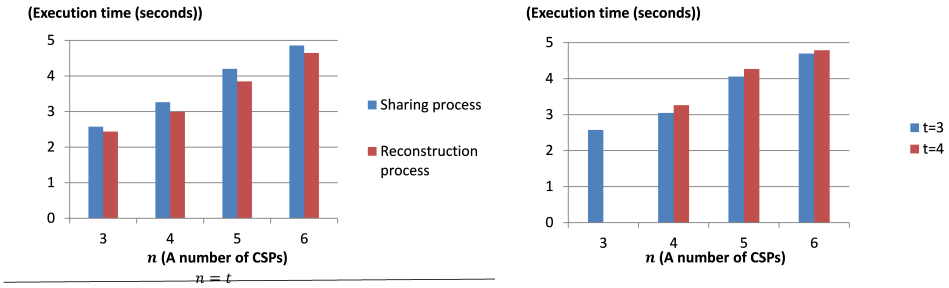


FIG. 8 – *Probability of incorrect data not being detected (false negative)*



An execution time of sharing process is slower than an execution time of reconstruction process. If data is shared more CSPs, it is slower.

FIG. 9 – *Execution time of Scheme-II*

respectively. The execution time of Scheme-II is shown in Figure 9. In the data reconstruction process, the execution time is about 3:04 seconds, and throughput is 336 MB per second when $n = 4$ and $t = 3$. This is fast enough to support ad-hoc querying on data warehouses.

2. **Stored data volume:** One advantage of our schemes is that the volume of shares is not much greater than that of original data when $n = t$ and t is a big value. It is indeed lower than $on\ \|P\|$ in Scheme-I and $on\ \|p\|$ in Scheme-II, where $\|P\|$ and $\|p\|$ are sizes of P and p, respectively.

   (a) By implementation of Scheme-II, share volume is shown in Figure 10. The volume of all shares is greater than the volume of D but less than $D \times 2$. The volume of each share is lower than the volume of D.

   (b) For example, with Scheme-II, 32 bits unsigned integers are shared among 6 CSPs and 5 CSPs are sufficient for reconstruct them. Let $\|p\|$ 9 bits. Then, the volume of all shares is lower than $1 \times 6 \times 9 = 54$ bits. The volume of each share is lower than $1 \times 9 = 9$ bits.

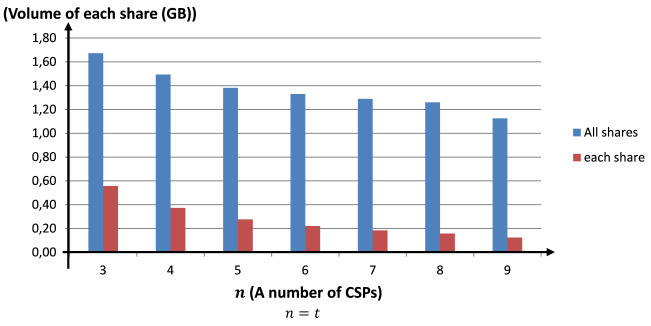


FIG. 10 – *Volume of shares with Scheme-II*

3. **Transferred data volume:** A nice side effect of Scheme-II is the low cost of transfered data volume in the data analysis and loading processes (Sections 5.2 and 5.3), because it can analyze and load data while not decrypting data first.

Here, the user should assign to a P value nearing the maximum value of D and assign to a t value nearing n in order to reduce costs.

7 Conclusion

In this paper, we propose two distributed schemes. Scheme-I helps protect data and guarantee data availability based on block cryptography and secret sharing. Moreover, it ensures data correctness by utilizing hash functions. Scheme-II extends from Scheme-I to allow analyses over cloud data warehouses. It allows analyzing data over shares without decrypting them first. Moreover, data transfer cost during the data analysis process is low. Our detailed security and performance analysis shows that our schemes are robust and low-cost when storing and querying data.

Future research shall run along two axes. First, we plan to further assess the cost of our solution in the cloud pay-as-you-go paradigm. Sharing data indeed implies increasing the initial data volume, and thus storage cost. However, it also guarantees data availability. Hence, we must run monetary cost evaluations against classical data replication schemes. It would also be very interesting to balance the cost of our solution against the cost of risking data loss or theft. Second, although we provide in this paper a raw framework for performing data analysis over shares, move research is required to achieve the sophisticated aggregations and complex predicates (conjunction, negation, disjunction...) that are required in OLAP analyses.

References

Asmuth, C. and J. Bloom (1983). A modular approach to key safeguarding. *IEEE Trans. Information Theory 29(2)*, 208–210.

Bellare, M., O. Goldreich, and S. Goldwasser (1994). Incremental cryptography: The case of hashing and signing. In *14th Annual International Cryptology Conference (CRYPTO 1994)*, Santa Barbara, USA, pp. 216–233.

Bellare, M., O. Goldreich, and S. Goldwasser (1995). Incremental cryptography and application to virus protection. In *27th annual ACM symposium on Theory of computing (STOC 1995)*, New York, USA, pp. 45–56.

Blakley, G. R. (1979). Safeguarding cryptographic keys. In *AFIPS National Computer Conference*, pp. 313–317.

Bowers, K., A. Juels, and A. Oprea (2009). Proofs of retrievability: theory and implementation. In *2009 ACM workshop on Cloud computing security (CCSW 2009)*, New York, USA, pp. 43–54.

Chan, C. and C. Chang (2005). A scheme for threshold multi-secret sharing. *Applied Mathematics and Computation 166(1)*, 1–14.

Changa, T., M. Hwangb, and W. Yang (2005). An improvement on the lin-wu (t,n) threshold verifiable multi-secret sharing scheme. *Applied Mathematics and Computation 163(1)*, 169–178.

Chow, R., P. Golle, M. Jakobsson, E. Shi, J. Staddon, R. Masuoka, and J. Molina (2009). Controlling data in the cloud: Outsourcing computation without outsourcing control. In *First ACM Cloud Computing Security Workshop (CCSW 2009)*, pp. 85–90.

Cormode, G. and D. Srivastava (2009). Anonymized data: generation, models, usage. *2009 ACM SIGMOD International Conference on Management of Data (SIGMOD 2009)*, 1015–1018.

Dehkordi, M. and S. Mashhadi (2008a). An efficient threshold verifiable multi-secret sharing. *Computer Standards and Interfaces 30(3)*, 187–190.

Dehkordi, M. and S. Mashhadi (2008b). New efficient and practical verifiable multi-secret sharing schemes. *Information Sciences 178(9)*, 2262–2274.

Gentry, C. (2009). Fully homomorphic encryption using ideal lattices. In *41st annual ACM symposium on Theory of computing (STOC 2009)*, New York, USA, pp. 169–178.

Hu, C., X. Liao, and X. Cheng (2012). Verifiable multi-secret sharing based on lfsr sequences. *Theoretical Computer Science 445*, 52–62.

Hwanga, R. and C. Chang (1998). An on-line secret sharing scheme for multi-secrets. *Computer Communications 21(13)*, 1170–1176.

Juels, A. and B. Kaliski (2007). Pors: Proofs of retrievability for large files. In *14th ACM conference on Computer and Communications Security (CCS 2007)*, Alexandria, USA, pp. 584–597.

Kenneally, E. and K. Claffy (2010). Dialing privacy and utility: a proposed data-sharing framework to advance internet research. *IEEE Security and Privacy 8(4)*, 31–39.

Machanavajjhala, A., D. Kifer, J. Gehrke, and M. Venkitasubramaniam (2007). L-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data 1(1)*.

Melchor, C., G. Castagnos, and P. Gaborit (2008). Lattice-based homomorphic encryption of vector spaces. In *IEEE International ISIT Conference*, pp. 1858–1862.

Padmanabhan, P., L. Gruenwald, A. Vallur, and M. Atiquzzaman (2008). A survey of data replication techniques for mobile ad hoc network databases. *34th International Journal on Very Large Data Bases (VLDB 2008) 17(5)*, 1143–1164.

Parakh, A. and S. Kak (2009). Space efficient secret sharing: A recursive approach. *IACR Cryptology ePrint Archive 2009*.

Shacham, H. and B. Waters (2008). Compact proofs of retrievability. In *4th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT 2008)*, Melbourne, Australia, pp. 90–107.

Shamir, A. (1979). How to share a secret. *Communications of the ACM 22(11)*, 612–613.

Shao, J. and Z. Cao (2005). A new efficient (t,n) verifiable multi-secret sharing (vmss) based on ych scheme. *Applied Mathematics and Computation 168(1)*, 135–140.

Sweeney, L. (2002). K-anonymity: a model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems 10(5)*, 557–570.

Wang, Q., C. Wang, J. Li, K. Ren, and W. Lou (2009). Enabling public verifiability and data dynamics for storage security in cloud computing. In *14th European conference on Research in Computer Security (ESORICS 2009)*, Saint Malo, France, pp. 355–370.

Wu, W., G. Hu, H. Lai, , and S. Jiang (2012). Novel space efficient secret sharing for implicit data security. In *Information Science and Digital Content Technology (ICIDT)*, pp. 283 – 286.

Yang, C., T. Chang, , and M. Hwang (2004). A (t,n) multi-secret sharing scheme. *Applied Mathematics and Computation 151(2)*, 483–490.

Zhao, J., J. Zhang, and R. Zhao (2007). A practical verifiable multi-secret sharing scheme. *Computer Standards and Interfaces 29(1)*, 138–141.

Résumé

L'informatique dans le nuage peut contribuer à réduire les coûts et à augmenter la flexibilité en permettant aux entreprises de déployer leurs applications et leurs entrepôts de données. Cependant, le stockage et la gestion des données dans le nuage posent des problèmes de sécurité. Dans cet article, nous nous intéressons à la sécurité des données stockées dans le nuage (confidentialité, disponibilité et intégrité). Pour sécuriser ces données, nous proposons une nouvelle méthode de partage multiple de clés secrètes. L'application de cette solution aux données entreposées dans le nuage permet de résoudre à la fois les problèmes de sécurité et d'analyse des données. L'analyse des performances de notre proposition montre qu'elle peut prévenir les intrusions, garantir la disponibilité des données et leur intégrité, pour un coût réduit (stockage, transfert de données et temps de calcul) dans le paradigme économique de paiement à la demande des nuages informatiques.