Comparative Analysis of Data Protection Mechanisms in Public Clouds

Nicolas Bolouri
Department Electrical and Computer Engineering
McGill University
Montreal, Canada
nicolas.bolouri@mail.mcgill.ca

Abstract —Cloud computing is one of the fastest-growing technologies in computing, and public cloud servers now hold vast amounts of people's data. The nature of cloud computing is rooted in virtualization leading to massive cybersecurity risks, such as rootkit attacks and API threats. This paper compares three mechanisms to mitigate these data protection issues: data splitting, anonymization, and cryptography. Each technique is analyzed based on its level of protection, ability to preserve cloud computing functionalities, and local proxy overhead. The investigation concludes that cryptography provides the best level of data protection and induces the least amount of overhead on the local proxy, making it the most viable alternative despite its unsatisfactory preservation of cloud functionalities. Further developments in lowering data anonymization's local proxy overhead and increasing data splitting's protection level have the potential to make these techniques as, if not more, effective than cryptography.

***Keywords* —cloud computing, data protection, machine virtualization, data splitting, data anonymization, cryptography**

I. Introduction

Cloud computing (CP) is one of the most exciting and revolutionary technologies in computing. Its ability to reduce costs, eliminate the need for hardware, and increase computational speed and efficiency is unmatched by traditional computing. However, the nature of cloud computing's cutting-edge technology compromises server security, thereby increasing vulnerability to cyber-attacks and data leaks. While many modern methods successfully secure user data, difficulties arise when combining data protection with a low local proxy overhead and cloud functionalities such as storage, update, computation, and interoperability. As these characteristics would prompt the complete transition to cloud computing, generating data protection mechanisms that conform to the CP paradigm is of paramount importance to the development of the field. As the amount of user data stored and processed in the cloud quickly increases, finding an efficacious strategy to protect private data is imperative to ensure a seamless shift to cloud computing.

To innovate and produce contemporary solutions to this problem, computer scientists have devised various techniques to protect user data while maintaining cloud functionalities. Namely, these include data splitting, data anonymization, and cryptography. In order to determine which of these methods accomplishes this mandate best, this paper compares each solution based on its level of protection, ability to preserve cloud computing functionalities, and local proxy overhead. Ultimately, the high degree of protection and low strain on the local proxy give cryptography a clear advantage in providing security to private data outsourced to the cloud.

II. Background

A. Cybersecurity Threats in Cloud Computing

The core concepts underpinning cloud computing make data susceptible to cyberattacks and leakage. The architecture comprises three central concepts rooted in virtualization: virtual machines (VM), hypervisors (VMM), and management interfaces (MI). Virtual machines, the software that allows cloud servers to run without hardware, are vulnerable to VM-based rootkit attacks, which hide files, registry keys, and other OS objects from antivirus software [1]. In addition, hypervisors, which allow multiple operating systems to run synchronously, complicate multi-layer server security. Malicious insider guest systems can thus harvest confidential data or gain control over the cloud with little risk of detection [1]. Finally, management interfaces, which allow multiple users to access the cloud from different sources, are vulnerable to many API threats, such as reusable tokens, inflexible access controls, improper authorizations, and logging capabilities on the overall cloud system [1]. While these three components may seem to cause only complications, researchers have yet to replicate an adequate CP model without them. Therefore, they have diverted their energy to finding solutions to these cybersecurity threats.

B. Data Splitting

The most recent attempt at securing cloud data is data splitting. When the proxy receives user data to be outsourced, an algorithm, shown below through pseudo-code in Algorithm 1, assesses the risks based on user privacy requirements and states whether attributes are identifiers or

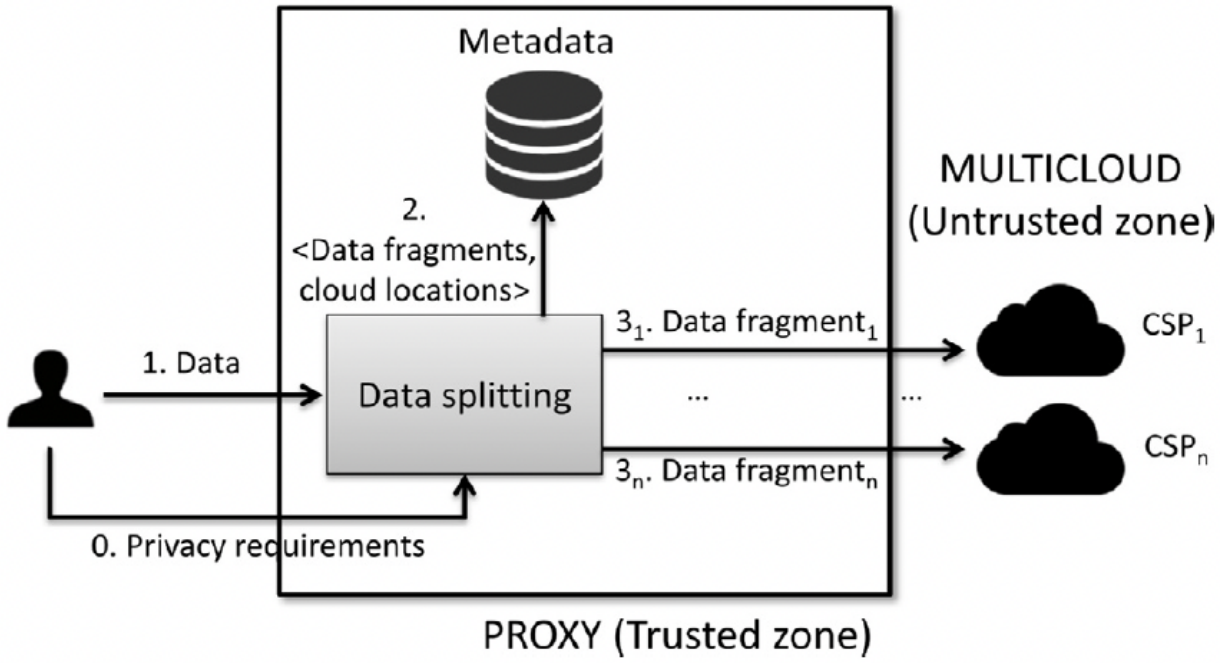

Fig. 1. Data Splitting Workflow [2]

quasi-identifiers. Based on this computation, the proxy decides how data is split and how many storage locations are necessary. It then stores the splitting criterion via metadata so the system can process future queries on split data and aggregate partial results. Finally, a complementary algorithm forwards each data fragment to separate cloud service providers (CSP), making the process lossless and privacy-preserving [2]. The data splitting method is summarized in Figure 1 above.

C. Data Anonymization

In contrast with data splitting, data anonymization irreversibly masks data. Data to be outsourced is stored in a table, *T*, where each record has several attributes, split into three categories: key attributes, quasi-identifiers (comprised of sub-attributes $\{Q_1, \ldots, Q_n\}$), and sensitive attributes [3]. Five steps are taken when the enclave receives the data to be anonymized. First, the system removes or obscures key attributes. Second, certain specific values are replaced with general values, and third, certain values are not released. Fourth, the frequency list of *T*, Q_i, is computed. If not all $Q_i \geq$ *k-anonymity* (anonymity parameter), then anonymous records are added to *T*, and the frequency list is re-calculated until the k-condition is satisfied [4]. The optimization algorithm which executes this computation is shown through pseudo-code in Algorithm 2. Fifth, the frequency list of distinct sensitive attributes associated with each Q_i, SA_i, is computed. Similar to step four, if not all $SA_i \geq$ *l-diversity* (diversity parameter), then anonymous records are added to *T*, and the frequency list is re-calculated until the l-condition is satisfied [3]. Once these five steps have been completed, the coarsened data is secured and outsourced to the cloud.

Algorithm 1. Detection of Risky Items

```
Input:  D     //the input document
        C     //the ordered set of entities to be protected
        g(C)  //the ordered set of generalizations corresponding to C (optional)
Output: RT    //the set of risky terms
01  for (n=1;n<=MAX;n++) do //cardinality of the combination of terms to evaluate
//create all combinations of terms in D of cardinality n sorted by informativeness
02  Comb_n=createCombinations(D,n);
03  while (not(empty(Comb_n))) do //evaluate each combination
04  T=getCombination(Comb_n); //obtain the next (set of) term(s)
//evaluate if PMI(c_k;T)>IC(g(c_k)) ∀ c_k ∈ C until true or c_k=null
05      if (checkDisclosure(C,g(C),T)) then
06        add(T,RT); //add the risky term(s) T to RT
07        remove(T,Comb_n,D); //remove T in the combinations and in D
08      end if
09    end while
10  end for
11  return RT;
```

Algorithm 2. Data Anonymization k-condition

```
K-OPTIMIZE(k, head set H, tail set T, best cost c)
    ;; This function returns the lowest cost of any
    ;; anonymization within the sub-tree rooted at
    ;; H that has a cost less than c (if one exists).
    ;; Otherwise, it returns c.
    T ← PRUNE-USELESS-VALUES(H, T)
    c ← MIN(c, COMPUTE-COST(H))
    T ← PRUNE(H, T, c)
    T ← REORDER-TAIL(H, T)
    while T is non-empty do
        v ← the first value in the ordered set T
        H_new ← H ∪ {v}
        T ← T − {v}        ;; preserve ordering
        c ← K-OPTIMIZE(k, H_new, T, c)
        T ← PRUNE(H, T, c)
    return c
```

D. Cryptography

As opposed to the more recent data splitting and anonymization techniques, cryptography has existed since the Turing era of computing and has continuously evolved. As numerous cryptographic techniques exist, this paper will focus on the Li-Gai-Qiu-Zhao model because it is designed specifically for cloud computing [5]. In their model, standard data is assigned to a single CSP, while sensitive data is assigned to two different CSPs. The two algorithms responsible for this process are *Alternative Data Distribution* (AD2) and *Secure Efficient Data Distributions (SED2)*, shown through pseudo-code in Algorithm 3 and Algorithm 4 below [5]. The sensitive data is then encrypted by the *Efficient Data Conflation* (EDCon) algorithm, which converts the data into an unreadable format called cyphertext tied to specific encryption keys [5]. The EDCon algorithm is shown through pseudo-code in Algorithm 5. Finally, the encrypted data is sent to the cloud through standard means.

Algorithm 3. Alternative Data Distribution (AD2) Algorithm [5]

Require: *NDP*, *PNL*
Ensure: D_{xor}, α, β
Input *NDP*, *PNL*
for ∀*NDP* **do**
  for each data packet **do**
    if ∃ a $L_i \in PNL$ **then**
      Execute SED2 Algorithm /* Algorithm 2 */
      Generate α and β
    else
      Do XOR operation to the data packet
      /*Do XOR operation before the data packet is sent out*/
      Generate D_{xor}
    end if
  end for
  Obtain the values of *D*
end for
Output D_{xor}

Algorithm 4. Secure Efficient Data Distribution (SED2) Algorithm [5]

Require: *D*, *C*
Ensure: α, β
Input *D*, *C*
Initialize $R \leftarrow 0$, $\alpha \leftarrow 0$, $\beta \leftarrow 0$
/* C is a random binary that is shorter than D */
Randomly generate a key *K*
for ∀ input data packets **do**
  if $D \neq C$ && $C \neq 0$ **then**
    DO $R \leftarrow D - C$
    $\alpha \leftarrow C \oplus K$
    $\beta \leftarrow R \oplus K$
  end if
end for
Output α, β

Algorithm 5. Efficient Data Conflation (EDCon) Algorithm [5]

Require: α, β, K
Ensure: *D*
Input α, β, K
Initialize $\gamma \leftarrow 0$, $\gamma' \leftarrow 0$, $D \leftarrow 0$
/* User receives α, β from separate cloud servers*/
$\gamma \leftarrow \alpha \oplus K$
$\gamma' \leftarrow \beta \oplus K$
$D \leftarrow \gamma + \gamma'$
Output D

E. Analysis Criteria

In order to determine which of these mechanisms offers the most secure data protection while preserving CSP functionalities, each solution will be compared against three criteria. First, the level of protection delivered by each method will be analyzed to determine the degree to which the data is guarded against cyberattacks. Following this, the best method for preserving cloud computing functionalities will be determined. This criterion can be further divided into four sub-criteria: storage – the ability to store outsourced data in the cloud; update – the ability to modify any individual piece or set of outsourced data without re-uploading it to the cloud; computation – the ability to perform operations on outsourced data; and interoperability – the ability for many people to work on and access a data set from different sources. Finally, each method's local proxy overhead will be compared to establish the amount of work necessitated from the user's local proxy. This analysis will yield a comprehensive critique of each method's ability to secure data while preserving the benefits of CP.

III. Analysis

A. Level of Protection

The level of protection offered by each method is the most important of the three criteria. Data splitting and anonymization offer less protection than cryptography because the data remains clear in the formers and encrypted in the latter. Indeed, splitting and anonymization generate new data sets based on the original in a manner that is skewed (splitting) or coarsened (anonymization) [6]. In both cases, the protection is ineffective if the characterization of which attributes are identifiers and quasi-identifiers is incomplete [2]. Furthermore, when splitting data, if the metadata stored at the proxy is compromised, the protection may be broken [2]. While data anonymization is immune to data leakages at the proxy, CSPs could infer the original data by comparing successive anonymized versions of the data set [6]. While the encrypted data produced by the Li-Gai-Qiu-Zhao cryptography model is inherently more secure than clear data, the protection can be undone by leakage of compromised encryption keys stored in the proxy or brute-force attacks [2]. Nonetheless, such cyberattacks are significantly less common than the ones targeting splitting and anonymization models, making cryptography the most protective of the three solutions.

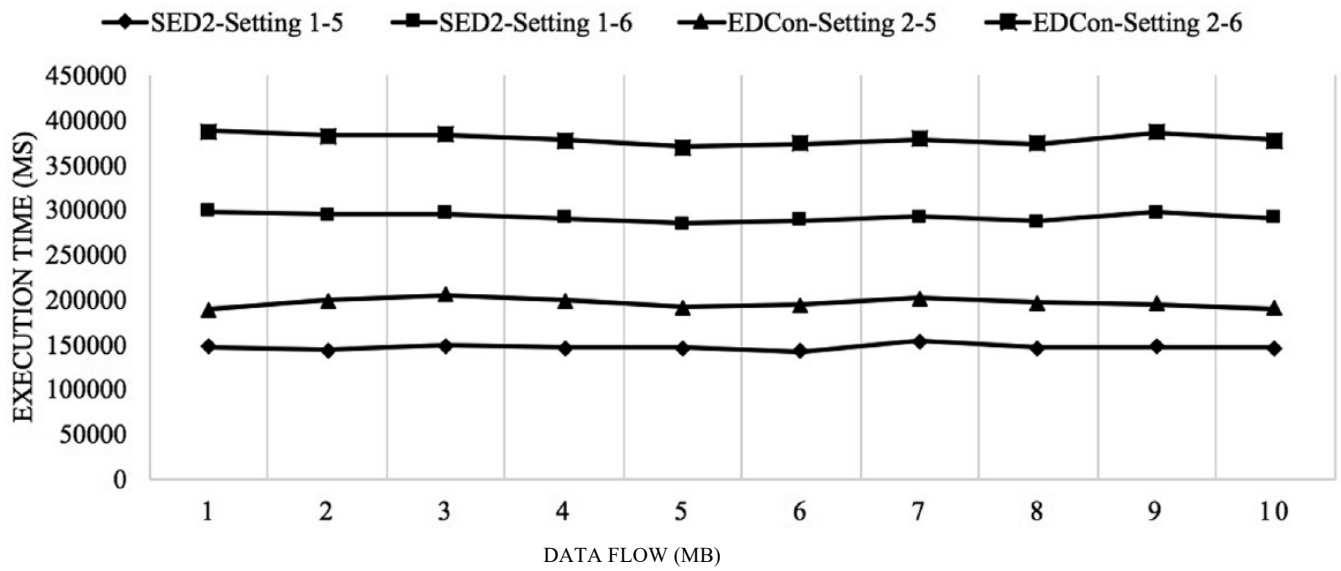

Fig. 2. Storage Execution Time for Different Encryption Settings [6]

B. *Ability to Preserve Cloud Computing Functionalities*

1) *Storage*

While all three methods allow for the storage of vast amounts of data in the cloud, the difference lies in the processing time of each method. In data splitting, data fragments are stored in separate CSPs according to the semantics and sensitiveness of the document contents and the privacy requirements of the user [7]. The amount of data and desired degree of protection influence storage time because of the higher number of fragments. Similarly, in the Li-Gai-Qiu-Zhao cryptography model, storage time increases based on the level of encryption required [6]. Figure 2 shows how the storage processing time increases by changing the SED2 and EDCon settings, i.e., the degree of encryption. Conversely, in data anonymization, the data remains clear but distorted, meaning data can be stored effectively with little to no impact on the processing time [6]. Thus, the nature of data anonymization makes it indisputably more effective than splitting or cryptography in storing data.

2) *Update*

The differentiation in update abilities manifests itself in whether the data must be re-uploaded to complete the update. Accordingly, data splitting is the most effective method for updating outsourced to the cloud. Splitting allows immediate updates on each data fragment individually [6]. This capability is the main advantage of data splitting, as anonymization and cryptography require offloading and re-uploading data, which strains CSPs [6]. Indeed, data anonymization requires re-anonymizing the original data and re-uploading the entire re-anonymized set because the new data must satisfy the *k-anonymity* and *l-diversity* parameters [3]. Similarly, cryptography requires re-encrypting the original data and re-uploading the entire re-encrypted set with unique keys generated by the EDCon algorithm [6]. Hence, data splitting's nimbleness in updating fragments makes it superior to anonymization and cryptography in updating.

3) *Computation*

Computations on outsourced data differentiate and increase in complexity when adding layers of protection. In data splitting, CSPs can run any calculation on single fragments. However, issues arise when operating on multiple inter-related fragmented data sets, as results tend to disagree [2]. Likewise, cryptography is a weak computation alternative as it permits only a small number of operations aside from adding and multiplying ciphertexts [6]. Contrariwise, with data anonymization, CSPs can conduct any calculation on anonymized data as they would on regular data [2]. In fact, anonymization and deanonymization performed by a secure enclave save computational power [3]. For this reason, data anonymization is unequivocally the best computation alternative of the three solutions.

4) *Interoperability*

Interoperability differs in each mechanism's requirements from interoperating users. For instance, data splitting requires the metadata indicating the location of each data fragment to be shared among interoperating proxies [6]. Similarly, cryptography requires the encryption keys to be shared among interoperating proxies, which research shows can lead to security weaknesses: "in terms of security, interoperability involves the exchange of the encryption keys between interoperable proxies. This is a simple limitation since if the key is corrupted during delivery, it can no longer protect the outsourced data" [6, p. 127]. Furthermore, complications may arise while sharing metadata and encryption keys, making splitting and cryptography systems inferior to anonymization in interoperability [6]. Indeed, interoperating proxies need not be aware of the

Table 1. Summary of Cloud Computing Functionality Preservation

Cloud Computing Functionality	Data Protection Method		
	Data Splitting	***Data Anonymization***	***Cryptography***
Storage	Data fragments are stored in separate CSPs	Data remains clear but distorted, having little effect on processing time	Storage time increases based on level of encryption
Update	Allows for immediate update on each data fragment individually	Requires re-anonymizing the original data and re-uploading the entire re-anonymized set	Requires re-encrypting the original data and re-uploading the entire re-encrypted set
Computation	CSPs can run any calculation on single fragments, but not on multiple inter-related fragmented data sets	CSPs can conduct any calculation on anonymized data as they would on regular data	Permits only a small number of operations aside from the addition and multiplication of ciphertexts
Interoperability	Requires the metadata indicating the location of each data fragment to be shared among interoperating proxies	Interoperating proxies need not be aware of the anonymization methods applied to the data	Requires the encryption keys to be shared among interoperating proxies

anonymization methods applied to the data [2]. In sum, data anonymization unquestionably trumps both splitting and cryptography in terms of interoperability.

5) Summary of Ability to Preserve Cloud Computing Functionalities

Preserving cloud computing functionalities is a critical criterion for analyzing the best solution. While data splitting offers a more effective way of updating outsourced data, data anonymization allows for more effective storage, computation, and interoperability than splitting and cryptography. Therefore, data anonymization is the best alternative for preserving cloud computing functionalities. Table 1 summarizes each mechanism's performance in preserving cloud computing functionalities based on the four sub-criteria.

C. Local Proxy Overhead

The amplitude of a data protection mechanism's local proxy overhead is directedly correlated to increased server latency, which considerably impedes the CSP's efficacy. Consequently, the best solution is the one that least encumbers the local proxy. As a rule, data anonymization requires quasi-linear computation from the proxy to generate the anonymized data set, necessitating substantially more computational power than linear computations [3]. While data splitting does not necessitate quasi-linear computation, research shows it

> […] takes constant work to split the original data set into fragments; after that, within-fragment operations by CSPs require no help from the proxy, but operations involving several fragments need constant computation by the local proxy (typically, the proxy participates at the end of the ad hoc protocol run among the CSPs to decrypt the computation result or remove the noise from it). [2, p. 52]

In contrast, the Li-Gai-Qiu-Zhao cryptography model bears little stress on the local proxy as it requires only linear computation to generate cyphertext and decrypt the results of operations [2]. Thus, cryptography appears to be the most appropriate of the three proposed techniques for minimizing local proxy overhead.

IV. Conclusion

Cloud computing is transforming the computing landscape. However, the very nature of CP components rooted in virtualization makes CSPs susceptible to cyberattacks. Not only is this a crucial barrier to expanding cloud computing, but as the volume of data outsourced to the cloud continues to increase, hundreds of millions of people's data are at risk. By assessing the degree of protection, ability to preserve cloud computing functionalities, and local proxy overhead of three proposed solutions, this paper evaluated which mechanism is best suited to protect private user data while preserving the advantages of CP. In conclusion, current research indicates that cryptography looks to be the forerunner in CP data protection techniques. The Li-Gai-Qiu-Zhao cryptography model provides the best level of data protection and induces the least amount of overhead on the local proxy compared to data splitting and anonymization. However, it is critical to note that cryptography unsatisfactorily preserves cloud functionalities compared to data splitting and anonymization. Still, data anonymization's high local proxy overhead and data splitting's low-level protection make them less practical candidates. That being said, if computer scientists can successfully address these aforementioned splitting and anonymization issues, perhaps by coalescing the two techniques, they will form a more complete and compelling data protection method than cryptography. Nevertheless, in the meantime, cryptography remains the best cloud data protection alternative.

References

[1] M. Jouini and L. Ben Arfa Rabai, "Surveying and Analyzing Security Problems in Cloud Computing Environments," *2014 Tenth International Conference on Computational Intelligence and Security*, 2014.

[2] J. Domingo-Ferrer, O. Farràs, J. Ribes-González, and D. Sánchez, "Privacy-Preserving Cloud Computing on Sensitive Data: A Survey

of Methods, Products and Challenges," Computer Communications, vol. 140-141, pp. 38–60, 2019.
[3] R. S. George and S. Sabitha, "Data Anonymization and Integrity Checking in Cloud Computing," *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, 2013.
[4 R. J. Bayardo and R. Agrawal, "Data Privacy through Pptimal K-Anonymization," 21st International Conference on Data Engineering (ICDE'05).
[5] Y. Li, K. Gai, L. Qiu, M. Qiu, and H. Zhao, "Intelligent Cryptography Approach for Secure Distributed Big Data Storage in Cloud Computing," *Information Sciences*, vol. 387, pp. 103–115, 2017.
[6] V. E. and D. N. Umadevi, "A Brief Survey on Various Technologies Involved in Cloud Computing Security," *Asian Journal of Applied Science and Technology*, vol. 04, no. 03, pp. 119–128, 2020.
[7] D. Sánchez and M. Batet, "Privacy-Preserving Data Outsourcing in the Cloud via Semantic Data Splitting," *Computer Communications*, vol. 110, pp. 187–201, 2017.