A NEW FRAMEWORK FOR SECURING PERSONAL DATA USING THE MULTI-CLOUD

Hassan Saad Alqahtani[1], Paul Sant[1] and Ghita Kouadri-Mostefaoui[2]

[1]IRAC, UCMK, Milton Keynes, UK
[2]Department of Computer Science, UCL, London, UK

ABSTRACT

Relaying On A Single Cloud As A Storage Service Is Not A Proper Solution For A Number Of Reasons; For Instance, The Data Could Be Captured While Uploaded To The Cloud, And The Data Could Be Stolen From The Cloud Using A Stolen Id. In This Paper, We Propose A Solution That Aims At Offering A Secure Data Storage For Mobile Cloud Computing Based On The Multi-Clouds Scheme. The Proposed Solution Will Take The Advantages Of Multi-Clouds, Data Cryptography, And Data Compression To Secure The Distributed Data; By Splitting The Data Into Segments, Encrypting The Segments, Compressing The Segments, Distributing The Segments Via Multi-Clouds While Keeping One Segment On The Mobile Device Memory; Which Will Prevent Extracting The Data If The Distributed Segments Have Been Intercepted.

KEYWORDS

Multi-cloud security, mobile cloud computing, cloud security, secure storage, untrusted environment

1. INTRODUCTION

Similar to other ICT solutions, cloud computing services have some open challenges and issues to meet in terms of security and privacy. One of the most recent security breaches related to the cloud is an iCloud privacy breach that exposed some celebrities' personal pictures to the public [1]. This incident drew attention to 1) the authentication techniques that were applied and 2) the efficiency of the applied encryption method [1]. Multiple-cloud models have emerged as a potential solution that could be used to overcome single cloud limitations and obstacles. The National Institute of Standards and Technology (NIST) define a multiple-cloud as a set of geographically distributed clouds that could be used in a serial or simultaneous way [2]. The multiple-cloud computing paradigms help to: i) promote independence; ii) enhance security; iii) increase redundancy; iv) optimise operational costs and v) improve the quality of delivered services. The multiple-cloud paradigms allow two or more distributed cloud services to collaborate and work together serially or simultaneously, regardless of whether they are dependant or not [2]. [3] identifies the purpose of developing multiple-cloud models for overcoming the limitations of the single cloud by distributing dependency, trust, privacy, accessibility, and security among multiple-clouds. This study concludes that using a single cloud as a storage medium, is totally insecure [4-6]. In this paper, this statement is confirmed by a systemic literature review, which identifies the existing security issues/threats and highlights the most critical ones. This research proposes an approach that provides a secure data storage mechanism using a multi-cloud computing model. The research will be validated by an android app prototype. The structure of the paper is as follows: the motivation for this research is presented in Section 2. The literature review is presented in Section 3. Details of the proposed approach are shown in Section 4. Finally, the conclusion and the future directions are shown in Section 5.

2. MOTIVATION

This study proposes a new approach to secure data in the cloud, using a multiple-cloud scheme. It develops a new mechanism/algorithm for providing a secure personal data storage system using smart devices. The proposed mechanism relies on a number of encryption methods in order to guarantee the integrity of stored data. Data confidentiality and integrity are preserved by splitting and distributing the original data file on the multi-cloud. Before distributing the data, the file's segments are encrypted, which will enhance the security level. In order to guarantee the privacy and the confidentiality of the targeted file, one file segment will remain stored in the mobile device to allow future data retrieval and reconstruction.

3. RELATED WORK

We studied nine core approaches that have been developed in order to enhance the security and availability of the delivered service. Table 1 shows a comparison between the studied approaches based on the identified criteria, while Table 2 defines the used criteria. It is very important to highlight that some approaches/algorithms have not been included in this study, for instance, NubiSave [7], Cleversafe [8], POTSHARDS [9], Cryptographic Cloud Storage System (CS2) [10], for the following reasons:

- Some approaches are too similar, lack novelty/contribution, or have been developed based on the same core algorithm. It is obvious that a lot of algorithms/ approaches have been developed based on the Byzantine Fault Tolerance algorithm (BFT), for instance, [11-13].
- There is a lack of efficiency in the approaches developed for cloud computing. It is clear that this is the case with approaches that have been developed for a distributed computing environment and have been proposed as potential solutions for the cloud.
- The lack of documentation, testing, or development such as in the following two commercial products - **Cumulus4j** (http://cumulus4j.org) and **TrustedSafe** (http://www.trustedsafe.de/).

Actually, the multiple-cloud concept has been inspired by the information dispersal concept. This is not new, nevertheless, it is very efficient and very popular [14, 15]. This concept aims to enhance the system's confidentiality, availability and reliability by distributing the information through more than one location [16, 17].

The Byzantine Fault Tolerance (BFT) approach has been proposed as a protocol to be used over distributed systems. Technically, the BFT aims to manage the communication process between the end user and the replicated data/system [14, 18, 19]. There are a number of approaches that have been inspired by BFT, one of which is TCLOUD [6, 16], which has been supported by the European Union (EU). This project focuses on the critical application in terms of availability and security. TCloud is presented to the end user as a platform that is capable of improving the quality of the delivered service in terms of security and availability. The researchers argue that relying on the application layer in order to guarantee the availability and security of delivered services is not an efficient approach, and the proposed solutions need to cover the platform and infrastructure as well.

Real-time cloud services, [20] developed an approach that claims to guarantee the reliability of the infrastructure that is used. This approach will maintain the reliability of each virtual machine after each computational cycle, as a result of a continuous change of resources over time. However, it consumes the provided resources because of the need to collect, analyse and compute

the resources' status. This approach can improve reliability by carrying out the required tasks over the most reliable resources. It uses a proactive mechanism in order to prevent unreliable resources from performing any task. The reliability level is determined by analysing the targeted resource's performance. DepSKY is a system that has been developed across distributed cloud storage [5]. This system has two versions: DepSKY-Availability and DepSKY-Confidentiality and Availability. This system shows a degree of inefficiency for some applications because its needs a large number of communications between all the involved data writers which could be very expensive for some applications.

The Redundant Array of Cloud Storage (RACS) has been developed as a proxy for cloud storage [21]. RACS aims to enhance the availability of stored data, and prevent vendor lock-in issues, through dispersing the targeted file across a number of cloud storages. InterCloud Storage (IC storage) [22] is similar to RACS. However, IC Storage has been developed due to the need to guarantee the dependability of the clouds used, and to overcome asynchrony issues by utilising the toleration of clients' protocol failures that has been developed by [23] to be used in terms of distributed storage.

IRIS is an approach that has been developed by [24], and aims to enhance the authentication level. Technically, IRIS relies on Proofs of Retrievability (POR) which have been developed in order to prove to the client that the stored data is recoverable. IRIS supports multi-writers, but has deficiencies in terms of access management and granted privileges. Also, it is costly in terms of consumed communications. Tahoe - The Least-Authority File system (Tahoe-LAFS) has been developed by [25] as a solution for securing distributed storage systems in order to increase the availability of stored data. In order to apply this solution across cloud storage, the cloud itself must be capable of executing code, because Tahoe-LAFS needs to perform some computations in the used clouds.

In [26], the authors propose an approach for cloud storage which is entitled CloudProof. This approach aims to guarantee the security of stored data by focusing on the auditability aspects. Also, this approach needs a code execution as part of the cloud storage. Offloading a lot of tasks to the used clouds means increasing the cost in terms of required resources, and the consumed communication. [27] developed Robust Data Sharing with Key-Value Stores (RDSKVS), which will help to develop a trusted distributed cloud storage system that supports multi-writers and multi-readers with regard to the stored data. RDSKVS is concerned with the availability and freshness of the shared data, which means that if the users need additional security features, they have to have additional tools to deliver these features. This approach will save communication costs because it does not require any communication between the involved users. However, this approach is not efficient in term of cost and functionality.

High Availability and Integrity Layer (HAIL) has been developed by [28] as an updated version of Redundant Arrays of Inexpensive Disks (RAID). One of HAIL's core disadvantages is the lack of capability to provide a live data version, which means limitations in terms of static files only. Besides this, it requires code execution across the provided clouds, in order to prevent any kind of latency that might be caused by executing the code across other platforms. Also, the segments are distributed across the clouds in the same order for each file, which could be considered to be a vulnerable feature. This is because the lack of randomisation will facilitate tracing the data segments. The client will not check the integrity for all the segments, but the integrity validation process will be implemented over specific segments. This could allow a creeping-corruption attack over time. Additionally, the HAIL does not support the infrastructure's heterogeneity. In addition, it is supposed to be run over federated, or intra-cloud paradigms. Technically, HAIL will be consumed as an integrated layer through the provided infrastructure.

Table 1. Multiple Cloud computing security approaches properties comparison

	BFT	RACS	HAIL	IC Store	DepSky	RDSKVS	IRIS	Tahoe-LAFS	CloudProof
Environment	Distributed	Cloud	Cloud	Cloud	Cloud	Cloud	Cloud	Distributed	Cloud
Cloud Module	---	Federated	Federated	Intra Cloud	Federated	Intra Cloud	Federated	Federated	Federated
Customers	Enterprise	Enterprise	Enterprise	Enterprise	Enterprise	Enterprise	Enterprise	Enterprise	Enterprise
Platform	--	Python	C++	C#	Java	Java	C#	Python	C#
Computation	√	X	√	X	X	X	X	X	√
Storage	√	√	√	√	√	√	√	√	√
Multi-users	-	X	X	X	√	√	√	X	X
Consistency	-	X	X	-	√	√	√	X	√
Confidentiality	√	X	X	√	√	X	X	√	√
Integrity	√	X	√	√	√	X	√	√	√
Retrievability	X	X	√	X	X	X	√	X	X
Authenticity	X	X	X	X	X	X	√	√	√
Auditing	X	X	X	X	X	X	√	X	√
Access control	X	X	X	X	X	X	X	√	√

Table 2. Comparison Criteria Explanation

Criteria	Explanation
Environment	The targeted computing environment/architecture (Distributed or Clouding).
Cloud Module	The targeted multiple cloud paradigms (Single, Hybrid, inter-cloud, federated, or multi-clouds).
Customers	The approach expected end-user (Enterprise, or individual customer).
Platform	The programming language that has been used for developing the approach.
Computation	The approach needs for executing code through off-load mode (through the cloud),
Storage	If the approach could be consumed as a cloud storage, computation platform, or both.

Multi-users	The approach capability to support more than one user.
Consistency	The ability of cloud service provider to deliver a fresh version of the file to the end users to consume it in parallel way.
Confidentiality	The ability of the approach to protect and maintain the Confidentiality of the involved files.
Integrity	The capability of implementing the required measurements in order to verify the integrity of the file/block.
Retrievability	The ability of the service provider to prove possessing of the targeted file.
Authenticity	Log records for verifying if person; who create, modify, or delete, is authorised to implement such tasks.
Auditing	Monitoring due to prove any kind of occurred breach, or suspicious behaviour.
Access control	The approach capability to Manage and control accessing to the files (privileges management).

The majority of developed approaches are relatively expensive in terms of configuration, development, and operational costs. In addition, all the studied approaches are based on intra-cloud or federated-cloud models, which are costly even for small enterprises. Both of these models (intra-cloud, and federated-cloud) are suitable for large and medium enterprises due to the characteristics of these models (prior-agreement and the coupled level). In other words, no solution has been developed that can be implemented through multiple-cloud models and can be used by individuals. Additionally, all the developed approaches vary in terms of interconnection protocols, internal layers or general algorithms. Besides that, all the existing approaches increase the complexity of the system used, and require a very high level of expertise to be configured and implemented. Also, all the studied approaches lack portability.

In addition, there is a lack of products that could be provided to customers and could be used immediately in order to enhance the level of delivered security, integrity, authenticity, and redundancy. Nor are there products which can be used by users with a minimum required level of experience, regardless of whether the customer was an enterprise or a single user. Additionally, the developed approaches cannot deliver an optimum solution that is able to solve/consider the majority of security aspects. This means that the used approach needs to be combined with other approaches to deliver the required security level. In other words, the total operational cost will be too high due to the need to configure, manage, maintain, and consume additional approaches in conjunction with the main one. Besides that, some approaches required a code execution across the provided clouds, in order to prevent any kind of latency that might be caused by executing the code across other platforms. In addition, executing code over the used clouds might introduce the possibility of attacks if it does not follow good practice.

To sum up, the performed literature review clearly shows the need to develop an approach capable of maintaining the security, integrity, privacy, and availability of the stored data for individuals. Besides that, the required experience level must be reduced, as well as the operational costs. Our approach aims to cover that gap and achieve the study aims.

4. PROPOSED APPROACH

Developing a cloud system that is stable and capable of delivering a very high level of security and availability cannot be achieved by relying on a higher layer of the delivered system (the software). Instead, the lower layer (the infrastructure) must be involved. We will develop an approach that combines security and availability, and deliver it to the end user for them to use it. We believe that the end user has to define the settings in terms of the required security and privacy, instead of relying on the service provider. In addition, the data encryption process must be implemented on the end user's mobile device instead of the cloud. This will guarantee the privacy of the stored data. Additionally, exchanging encrypted data is much safer, and the end user will guarantee that the providers cannot access or alter the data, as long as they do not have the key.

Keeping one segment in the end user's machine will prevent any attempt to recover the distributed data, because the attacker must have all the segments in order to recover that data, together with the key. Also, by keeping the last segment in the end user's device, the attacker will not be able to identify the first and last segments of the targeted data, because all the exchange segments have a constant size.

This approach relies on firstly, dividing, encrypting and storing (distributing) data on several protected multi-cloud storage facilities, and secondly, storing one segment only on the smart device. The proposed approach will achieve the study aim through the following:

- Using a chaotic map encryption technique, which is a strong and novel mathematical algorithm used to encrypt the targeted data.
- Storing one segment locally on the mobile device in order to prevent reconstructing the whole chunk of data.
- Avoiding un-authenticated access by distributing the data into a multi-cloud environment, and imposing two authentication levels.

4.1. Multiple-Cloud Computing Model

There are various existing models and forms of multiple clouding; consequently, it is very important to compare these models in order to select the model that is best fit and capable of delivering the expected service level. The multi-cloud libraries-based has been selected as the most suitable paradigm. In the multi-cloud computing model, there is more than one independent cloud that will be used to execute the requested tasks through consuming the provided resources/capabilities; here the customers will take responsibility for resource/capabilities management, task scheduling, and load balancing [29]. In the multi-cloud libraries-based model, the users develop their own service broker through a unified Application Programming Interface (API) [30]. The developed broker will be in the form of libraries and be embedded in the users' access point (host) [31]. This model offers an availability level higher than the previous model, and that is caused by associating the multi-cloud library with the user's machine. However, the users must be qualified to develop their multi-cloud libraries. In addition, this model does not require any prior agreement between the involved clouds.

4.2. Framework Design

This section discusses the overall architecture of the proposed framework. It illustrates the modules of the developed application and their functions/tasks. Figure 2 illustrates the approach modules and their internal units.

4.2.1. Preference Module

This module is responsible for cloud selection procedures, based on the end-users' requirements (data size, number of segments, available storage). Also, it is responsible for the involved cloud providers. In addition, the user could customise the security requirements via this module. Additionally, the option of backing-up the secured data could be performed in this module after calling the data logs from the data distribution module.

4.2.2. Data Management Module

This module is responsible for data fragmentation, recovery tasks, and the erasing of temporary data. Additionally, it takes responsibility for segment re-naming. The re-naming process will help to hide the identity of the exchanger data against phishing attacks, and will facilitate the archiving of the secured data.

4.2.3. Data Security Module

This module is responsible for two key procedures: data encryption/decryption and data compression/de-compression. The preferred encryption techniques will be performed through this module, while the executed technique relies on the user's selection from the preference module.

4.2.4. Data Distribution Module

This module is responsible for the distribution procedure and for segment requests (for recovery). This module relies on the cloud storage APIs to provide an abstraction layer in order to provide stable and reliable connections. The distribution process will based on two main factors: the user security requirements and the status of the involved cloud. The status of the involved cloud contains:

- Having access (correct user name and password).
- The service status (offline, online).
- The available storage.
- The size of the segment being sent (some service providers have conditions regarding the maximum file size).

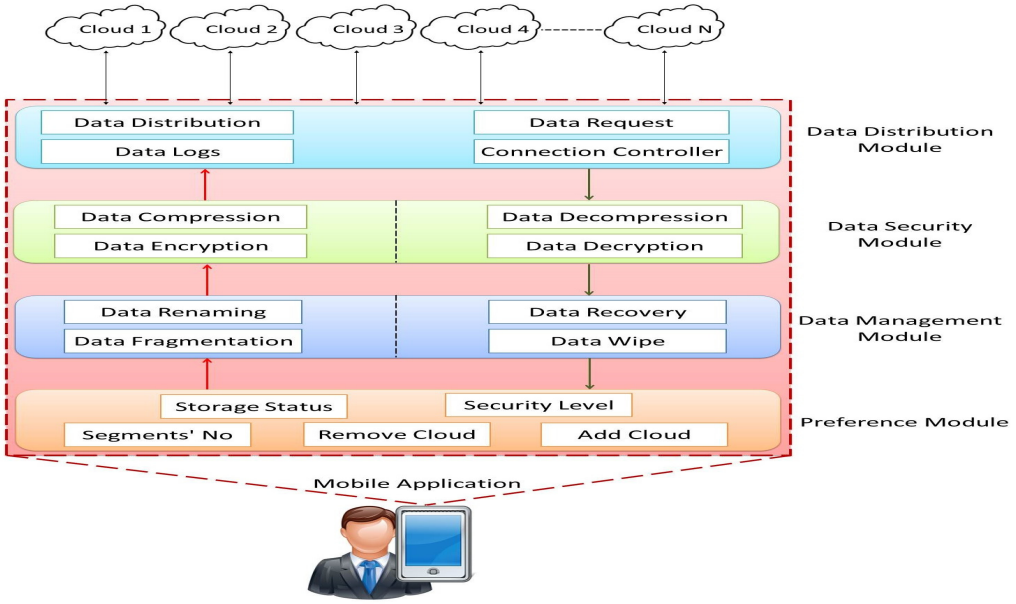

Figure 2. Proposed approach architecture design

4.3. Chaotic Logistic Map

Traditional data encryption algorithms such as Data Encryption Standard (DES), Triple DES (3DES), Advanced Encryption Standard (AES), International Data Encryption Algorithm (IDEA), and linear feedback shift register (LFSR), consider plaintext as either block cipher or data stream and are not suitable for fast encryption of large data (for example, colour images). Their implementation, when they are realized by software, of traditional algorithms for image encryption is even more complicated because of high correlation between image pixels; in other words, these mechanisms are not appropriate for encrypting digital pictures, especially in the case of smart devices [32]. These techniques are expensive in terms of consumed time, computational resources, and power consumption. Additionally, applying these techniques to large pictures is not as efficient as it is to small pictures. For that, chaos techniques are better for picture encryption, and can provide security, speed, power and computational resource savings. The structure of the core chaos-based image encryption algorithm is shown in Figure 3 [33]. This algorithm represents the core encryption technique used in our approach. It is worth mentioning that the applied encryption techniques will not be limited to the chaotic logistic map. Rather, other techniques will be included as part of this approach.

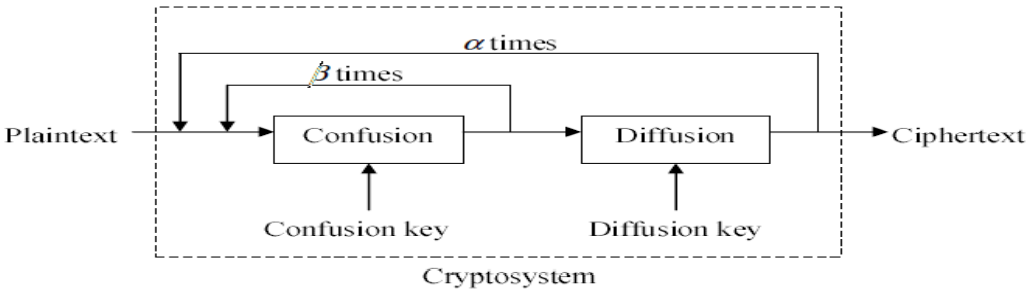

Figure 3. General Chaos Algorithm Architecture [33]

4.4. Approach Workflow

Figure 4 shows the key operational tasks associated with the distribution and recovery procedure respectively. For the distribution procedure, the distributed segments will be recorded with the associated cloud, in order to facilitate the recovery procedure. On the other hand, the segments will be wiped from the clouds after recovering, and the logs will be updated. The consumed time for these two procedures will vary and this is caused by the variation into the cloud providers for each segments and their status.

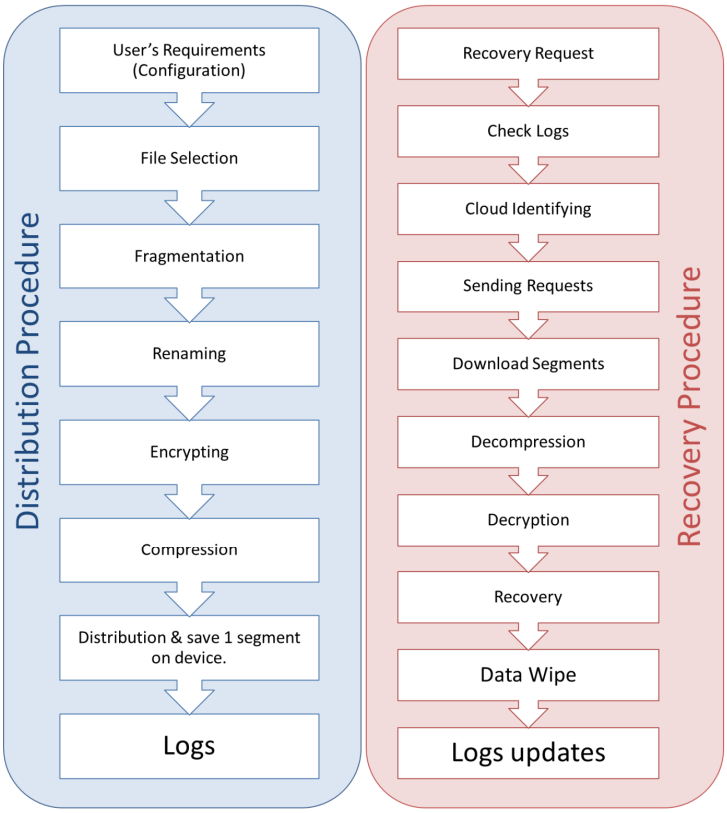

Figure 4. Distribution & Recovery procedures

Figure 5 illustrates the flow of operations. It also shows how the mobile application can interact and update the multiple-cloud status, and at which stage the data on-site storing and data archiving need to be completed. In order to enhance the delivered security level, the customer can keep one of the segments on the device itself. This means that the data can never be decrypted/read/modified outside the smart phone.

Splitting the targeted file into segments is very important for many reasons:

- Distributing over the multi-clouds.
- Avoiding cloud maximum-size file problems.

In addition, splitting the file will enhance the utilization of the provided bandwidth and facilitate load balancing. Also, it will facilitate the encryption/decryption processes by reducing the size of the computed files. Technically, the data splitting improves the processing throughout, by

allowing pipeline and parallel process, which will reduce the operational cost and the consumed time [34]. Compressing the distributed data could help reduce the segment's size before uploading to the cloud, which will reduce the consumed time and bandwidth and save storage in the cloud. Technically, the compression efficiently depends on the type of compressed data (text, image and video). For instance, the text files can be compressed, although compressing pdf files cannot efficiently reduce the size because these types of files are already compressed. For that, compressing all the files will not reduce the size and save time and operational cost, it might consume the system resources without benefits. With the limitation of mobile devices' resources, and in order to avoid the compression problem, the technique that has been developed by [35] will be used for choosing the best algorithm that offers the highest compression, and ignores the incompressible files.

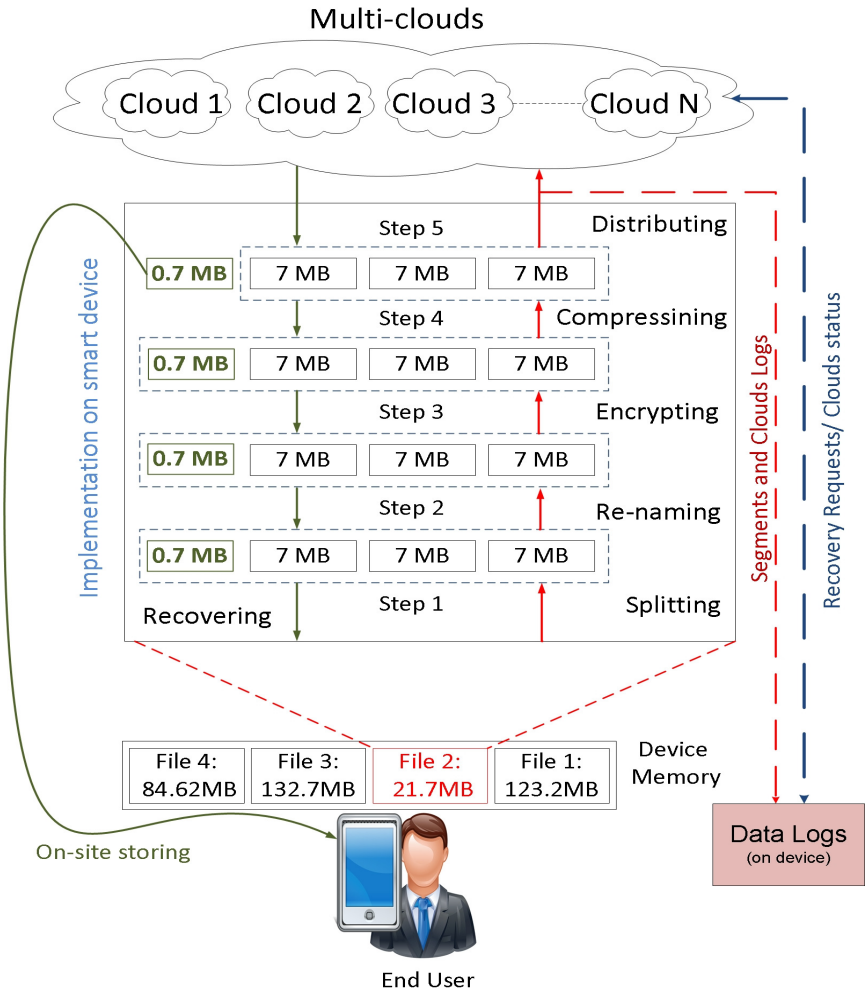

Figure 5. The proposed framework's workflow

5. CONCLUSION

To sum up, we have proposed an approach that offers secure data storage over multi-clouds for mobile computing users to ensure the security of the stored data. The study shows the importance of cloud computing services and the features that encourage users to consume the cloud service instead of the traditional IT solutions. Then, the limitations of cloud computing services' security

approaches, covered by previous studies, were discussed in order to identify the current security weaknesses that we need to overcome. The proposed approach overcomes the single cloud storage limitations by shifting to multi-cloud schemes and omits the phishing attack efficiently by keeping one segment of the distributed data on the mobile device. As for future work, the proposed approach can be enhanced by adding more techniques and features to help improve the delivered security levels. It will offer a diversity of security levels for users to select from. Finally, we aim to extend the developed mobile application to cover the case where the end users use their own cloud storage (private).

REFERENCES

[1] Kelion, L., 2014. Apple toughens iCloud security after celebrity breach, Available at: http://goo.gl/vyxS3S [Last accessed on October 20, 2015].

[2] Hogan, M., Liu, F., Sokol, A. & Tong, J. 2011, "Nist cloud computing standards roadmap", NIST Special Publication, vol. 35.

[3] Vukolic, M. 2010, "The Byzantine Empire in the Intercloud", SIGACT News, vol. 41, no. 3, pp. 105-111.

[4] AlZain, M.A., Soh, B. & Pardede, E. 2013, "A Byzantine Fault Tolerance Model for a Multi-cloud Computing", Computational Science and Engineering (CSE), 2013 IEEE 16th International Conference on 2013, IEEE, pp. 130-137.

[5] Bessani, A., Correia, M., Quaresma, B., André, F. & Sousa, P. 2013, "DepSky: dependable and secure storage in a cloud-of-clouds", ACM Transactions on Storage (TOS), vol. 9, no. 4, pp. 12.

[6] Verissimo, P., Bessani, A. & Pasin, M. 2012, "The TClouds architecture: Open and resilient cloud-of-clouds computing", Dependable Systems and Networks Workshops (DSN-W), 2012 IEEE/IFIP 42nd International Conference on 2012, IEEE, pp. 1-6.

[7] Spillner, J., Bombach, G., Matthischke, S., Muller, J., Tzschichholz, R. & Schill, A. 2011, "Information Dispersion over Redundant Arrays of Optimal Cloud Storage for Desktop Users", Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on 2011, pp. 1-8.

[8] Resch, J.K. & Plank, J.S. 2011, "AONT-RS: Blending Security and Performance in Dispersed Storage Systems", Proceedings of the 9th USENIX Conference on File and Stroage Technologies USENIX Association, Berkeley, CA, USA, pp. 14.

[9] Storer, M.W., Greenan, K.M., Miller, E.L. & Voruganti, K. 2009, " POTSHARDS—a secure, recoverable, long-term archival storage system ", ACM Transactions on Storage (TOS), 5(2), pp. 5.

[10] Kamara, S., Papamanthou, C. & Roeder, T. "Cs2: A searchable cryptographic cloud storage system".

[11] Tchana, A., Broto, L. & Hagimont, D. 2012, "Approaches to cloud computing fault tolerance", Computer, Information and Telecommunication Systems (CITS), 2012 International Conference on 2012, pp. 1-6.

[12] Zhao, W., Melliar-Smith, P.M. & Moser, L.E. 2010, "Fault Tolerance Middleware for Cloud Computing", Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on 2010, pp. 67-74.

[13] Wu, L., Liu, B. & Lin, W. 2013, "A Dynamic Data Fault-Tolerance Mechanism for Cloud Storage", Emerging Intelligent Data and Web Technologies (EIDWT), 2013 Fourth International Conference on 2013, pp. 95-99.

[14] Rabin, M.O. 1989, "Efficient Dispersal of Information for Security, Load Balancing, and Fault Tolerance", J.ACM, vol. 36, no. 2, pp. 335-348.

[15] Spillner, J. & Schill, A. 2014, "Towards Dispersed Cloud Computing", Communications and Networking (BlackSeaCom), 2014 IEEE International Black Sea Conference on 2014, pp. 170-174.

[16] Bessani, A., Cutillo, L.A., Ramunno, G., Schirmer, N. & Smiraglia, P. 2013, "The TClouds platform: concept, architecture and instantiations", Proceedings of the 2nd International Workshop on Dependability Issues in Cloud Computing 2013, ACM, pp. 1.

[17] Spillner, J. & Muller, J. 2014, "Tutorial on Distributed Data Storage: From Dispersed Files to Stealth Databases", Utility and Cloud Computing (UCC), 2014 IEEE/ACM 7th International Conference on 2014, pp. 535-536.

[18] Correia, M., Costa, P., Pasin, M., Bessani, A.N., Ramos, F.M. & Verissimo, P. 2012, "On the Feasibility of Byzantine Fault-Tolerant MapReduce in Clouds-of-Clouds.", SRDS 2012, pp. 448-453.

[19] Garraghan, P., Townend, P. & Xu, J. 2011, "Byzantine fault-tolerance in federated cloud computing", Service Oriented System Engineering (SOSE), 2011 IEEE 6th International Symposium on 2011, IEEE, pp. 280-285.
[20] Malik, S. & Huet, F. 2011, "Adaptive Fault Tolerance in Real Time Cloud Computing", Services (SERVICES), 2011 IEEE World Congress on 2011, pp. 280-287.
[21] Abu-Libdeh, H., Princehouse, L. & Weatherspoon, H. 2010, "RACS: a case for cloud storage diversity", Proceedings of the 1st ACM symposium on Cloud computing 2010, ACM, pp. 229-240.
[22] Cachin, C., Haas, R. & Vukolic, M. 2010, Dependable storage in the Intercloud .
[23] Chockler, G., Guerraoui, R., Keidar, I. & Vukolic, M. 2008, Reliable distributed storage.
[24] Stefanov, E., van Dijk, M., Juels, A. & Oprea, A. 2012, "Iris: A scalable cloud file system with efficient integrity checks", Proceedings of the 28th Annual Computer Security Applications Conference 2012, ACM, pp. 229-238.
[25] Wilcox-O'Hearn, Z. & Warner, B. 2008, "Tahoe: the least-authority filesystem", Proceedings of the 4th ACM international workshop on Storage security and survivability 2008, ACM, pp. 21-26.
[26] Popa, R.A., Lorch, J.R., Molnar, D., Wang, H.J. & Zhuang, L. 2011, "Enabling Security in Cloud Storage SLAs with CloudProof.", USENIX Annual Technical Conference.
[27] Basescu, C., Cachin, C., Eyal, I., Haas, R., Sorniotti, A., Vukolic, M. & Zachevsky, I. 2012, "Robust data sharing with key-value stores", Dependable Systems and Networks (DSN), 2012 42nd Annual IEEE/IFIP International Conference on 2012, IEEE, pp. 1-12.
[28] Bowers, K.D., Juels, A. & Oprea, A. 2009, "HAIL: a high-availability and integrity layer for cloud storage", Proceedings of the 16th ACM conference on Computer and communications security 2009, ACM, pp. 187-198.
[29] Ferrer, A. J., Hernández, F., Tordsson, J., Elmroth, E., Ali-Eldin, A., Zsigri, C., et al. 2012, “OPTIMIS: A holistic approach to cloud service provisioning”, Future Generation Computer Systems, 28(1), 66-77.
[30] Kecskemeti, G., Kertesz, A., Marosi, A., & Kacsuk, P. 2012, “Interoperable resource management for establishing federated clouds”, IGI Global Theory and Practice, Hershey, pp.18-35.
[31] Petcu, D., Crăciun, C., Neagul, M., Panica, S., Di Martino, B., Venticinque, S., et al. 2011, “Architecturing a sky computing platform. Towards a Service-Based Internet”, ServiceWave 2010 Workshops, pp. 1-13.
[32] Ismail, I.A., Amin, M. & Diab, H. 2010, "A Digital Image Encryption Algorithm Based A Composition of Two Chaotic Logistic Maps.", IJ Network Security, vol. 11, no. 1, pp. 1-10.
[33] Lian, S., Sun, J. & Wang, Z. 2005, "Security analysis of a chaos-based image encryption algorithm", Physica A: Statistical Mechanics and its Applications, vol. 351, no. 2–4, pp. 645-661.
[34] Ashokkumar, S., Karuppasamy, K., Srinivasan, B. & Balasubramanian V. 2010, "Parallel Key Encryption for CBC and Interleaved CBC," International Journal of Computer Applications, vol. 2, pp. 21-25, 2010.
[35] Harnik, D, Kat, R, Margalit, O, Sotnikov, D, Traeger, A. 2013, “To zip or not to zip: effective resource usage for real-time compression”. In: Proceedings of the 11th USENIX conference on file and storage technologies.

Authors

Dr Paul Sant joined the department of Computer Science and Technology (UoB) in September 2005 as a lecturer and he became a Senior Lecturer in September 2006. He was promoted to Principal Lecturer in August 2011. Dr. Paul completed his PhD from King's College, London in 2003 with a thesis entitled "Algorithmics of edge-colouring pairs of 3-regular trees" and prior to this, a BSc. in Computer Science from the University of Liverpool (1999). He is an active member of the British Computer Society and a Chartered Information Technology Professional (CITP) as well as being a fellow of the Higher Education Academy.

Dr Ghita kouadri Mostefaoui is a member of the Advanced Teaching Group, Department of Computer Science, University College London. Ghita has been awarded her PhD in adaptive security, jointly from the University of Fribourg and University Paris VI. Her research interests include cloud computing, automatic extraction of software models and computer science education. Ghita is a fellow of the Higher Education Academy.

Hassan Saad Alqahtani started his PhD March-2014 at university of Bedfordshire. His research interest includes cloud computing, mobile cloud computing, cyber security, and encryption. He received his Master degree from Teesside University in 2012, and his Postgraduate certificate from Essex University in the Telecommunication and Information System.