Journal of Computer Sciences and Applications, 2017, Vol. 5, No. 1, 25-28
Available online at http://pubs.sciepub.com/jcsa/5/1/4

DOI:10.12691/jcsa-5-1-4

Data Offloading Security Framework in MCLOUD

Akomolafe Patrick Oladeji, Ajayi Olubunmi*

Department of Computer Science, University of Ibadan
*Corresponding author: ajayiolubunmipeter@gmail.com

Abstract With the popularity of smartphones and upsurge of mobile applications, mobile devices have become prevalent computing platform. Although MCLOUD paradigm solves the problem of limited resources constraint of mobile system and unavailability of internet, through several offloading technique. Nevertheless, mobile users are still reluctant to adopt this paradigm, due to security concerns of their data. This research provided security on the users task and still minimize the total computational time of the MCLOUD. Tasks were programmatically broken down into smaller tasks and encrypted using homomorphic encryption system and assigned to slave devices which were admitted into the MCLOUD during the resort point process through the WIFI Direct wireless network. Using a test bed of three (3) smartphone devices, several task sizes ranging from 2KB, 4KB up to 20KB were used to test the implemented security framework and the time taken to complete computation of each task size is recorded for both MCLOUD and standalone architecture, the total execution time was compared and findings shows that computation involving security on MCLOUD takes less time compared to computation on standalone devices, the following readings were recorded. For the 4KB task size, MCLOUD spent 48500microseconds while standalone spent 241000microseconds; for the 6KB task size, MCLOUD spent 99500microseconds while standalone spent 453000microseconds; whereas in the 8KB task size, MCLOUD spent 109500microseconds while standalone spent 553000microseconds,which is approximately five (5) times faster than standalone execution. MCLOUD framework was observed to have a lower computation time, decreasing computational time ratio, higher throughput per seconds, It was discovered that computation on distributed encrypted data (MCLOUD) using homomorphic encryption is safer and faster than standalone single device computation.

Keywords: *MCLOUD, mobile applications, security framework*

Cite This Article: Akomolafe Patrick Oladeji, and Ajayi Olubunmi, "Data Offloading Security Framework in MCLOUD." *Journal of Computer Sciences and Applications*, vol. 5, no. 1 (2017): 25-28. doi: 10.12691/jcsa-5-1-4.

1. Introduction

1.1. Background

MCLOUD is an offspring of mobile cloud computing and is a cloud of mobile devices connected together with or without the availability of internet for the purpose of sharing resources for various reasons like computation, storage and so on. MCLOUD as a photogene of Mobile Cloud Computing is also the mixture or hybrid of cloud computing technology, mobile computing technology [1], and wireless network to produce good computational resources to mobile users. The ultimate goal of MCLOUD is to enable offloading and execution of powerful mobile applications on a large amount of mobile devices, with a good and satisfying user experience, Thus saving the limited resources of the mobile phone users. According to [2], the term "cloud" refers to a hosted service of a configurable distributed resource pool of networks, servers or storage over the Internet, where a user can gain an application using a "pay as you go" manner. The "cloud" can be viewed as an unlimited resource pool. Mobile cloud computing can be delivered when a mobile device uses cloud-based services by means of a mobile apps installed locally in the mobile devices and secondly, when cloud-based applications are running inside the user's mobile devices. Mobile cloud technology basically focuses on the user's mobile devices to access cloud-based services through wireless network communications [3]. This is achieved when mobile applications are deployed (i.e., mobile offloading) to the cloud servers or cloud-based applications running on other user's mobile devices. According to [4] Computation offloading is an enriched way to improve the performance as well as reducing the battery power consumption of a smartphone application by executing some parts of the application through code offloading to a remote server. There are two basic approaches to carrying out Data offloading, first, offloading in a static environment and second, offloading in a dynamic environment [5,6]. In static offloading, the programmers pre-determine the application components and in dynamic offloading (also called context-aware offloading), the execution location of the components is not pre-determined.

According to [7], there are three major architecture under mobile cloud computing which are

- Device to cloud architecture
- Cloud to device architecture
- Device to device architecture.

This research work focused mainly on the third tier of the architecture of mobile computing and the security of it.

Cryptography in data security is the process of transforming data using an algorithm to make it unreadable to anyone except those possessing special knowledge, usually referred to as a key and vice versa, to ensure privacy, confidentiality and integrity [8].

2. Literature Review

Traditional cloud computing can be categorize into three (3) tiers namely infrastructure as a service (IaaS), software as a service (SaaS) and platform as a service (PaaS) according to *NIST* standard. research and work has been carried out in the field of mobile cloud computing over the decades, and according to [7] mobile cloud computing is categorise into three specific class of architectures according to their mode of use and the intent they deliver . They are;

- Mobile cloud "Device to Cloud" (D2C) architecture,
- Mobile cloud "Cloud to Device" (C2D) architecture and
- Mobile cloud "Device to Device" (D2D) architecture.

[9] in their research also classify mobile cloud computing into two(2) architecture namely

- Non cloudlet architecture and
- Cloudlet architecture.

According to our research, based on the existing and reviewed work and mode of processing, we categorize mobile cloud (MCLOUD) into two (2) main Architecture namely

- Mobile cloud "Device to Cloud" (D2C) architecture,
- Mobile cloud "Device to Device" (D2D) architecture.

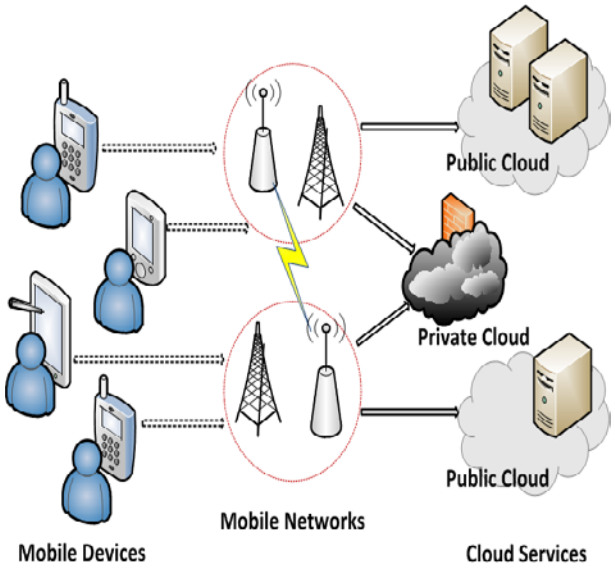

Figure 1. Mobile cloud computing device to cloud Architecture (Shantanu et.al, 2013)

2.1. Device to Cloud (D2C) Architecture

Under this category, users "mobile applications run in the cloud, where mobile devices are becoming part of a larger cloud environment, mobile devices connect to the remote infrastructure clouds through Internet connection as a medium Unlike the traditional cloud-based applications (e.g., via a desktop or server), the mode of using the devices are different in this case. Users connect their mobile devices to the remote cloud servers through mobile networks (e.g., wireless access points).

2.2. Device to Device (D2D) Architecture

In this category of architecture, mobile devices instantiates and share their own locally created network connections thus forming an environment for sharing information with other users within proximity [10]. In this approach, mobile devices are able to convey information directly to their nearby mobile devices without any help of the overlay network, as seen in Figure 2.

According to [10], they worked on content distribution in mobile cloud, focusing mainly on smartphone to Smartphone device platform of communication, but unlike our work they failed to address the security issue on the content data.

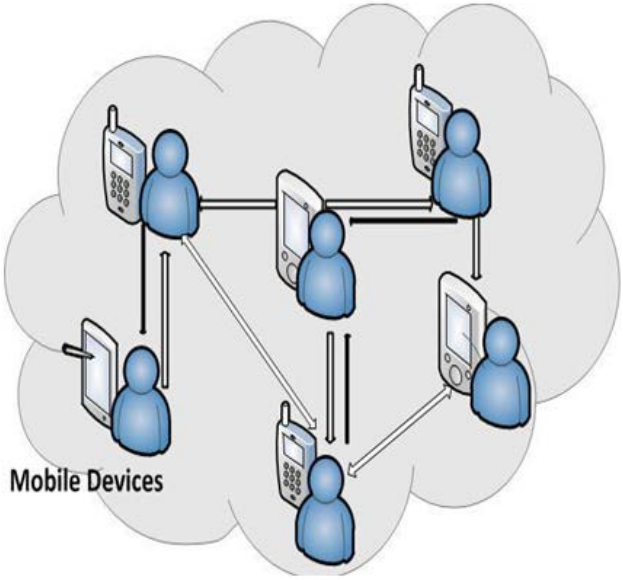

Figure 2. Mobile cloud computing "Device to Device" (D2D) Architecture. (Shantanu et.al, 2013)

3. Methodology

Traditional mobile cloud computing architecture utilize remote cloud environment for storage and computation operation, unlike this, MCLOUD paradigm utilize neighboring mobile devices for temporary storage and computation operation. We design this framework by using WI-FI direct to discover and create a cluster of smartphone devices, $D_1 \ldots D_n$, and we breakdown task T into n subtasks $t_1 \ldots t_n$, programmatically, using the list of smartphone admitted into the MCLOUD and then we apply homomorphic encryption on the subtasks before assigning them to the devices in the MCLOUD, using the resort point generated list. The cooperating devices received their apportioned chunks, which executes itself due to the pre-installed MCLOUD application on the cooperating devices and return result back to the master device in an encrypted state, which can only be decrypted by the master device.

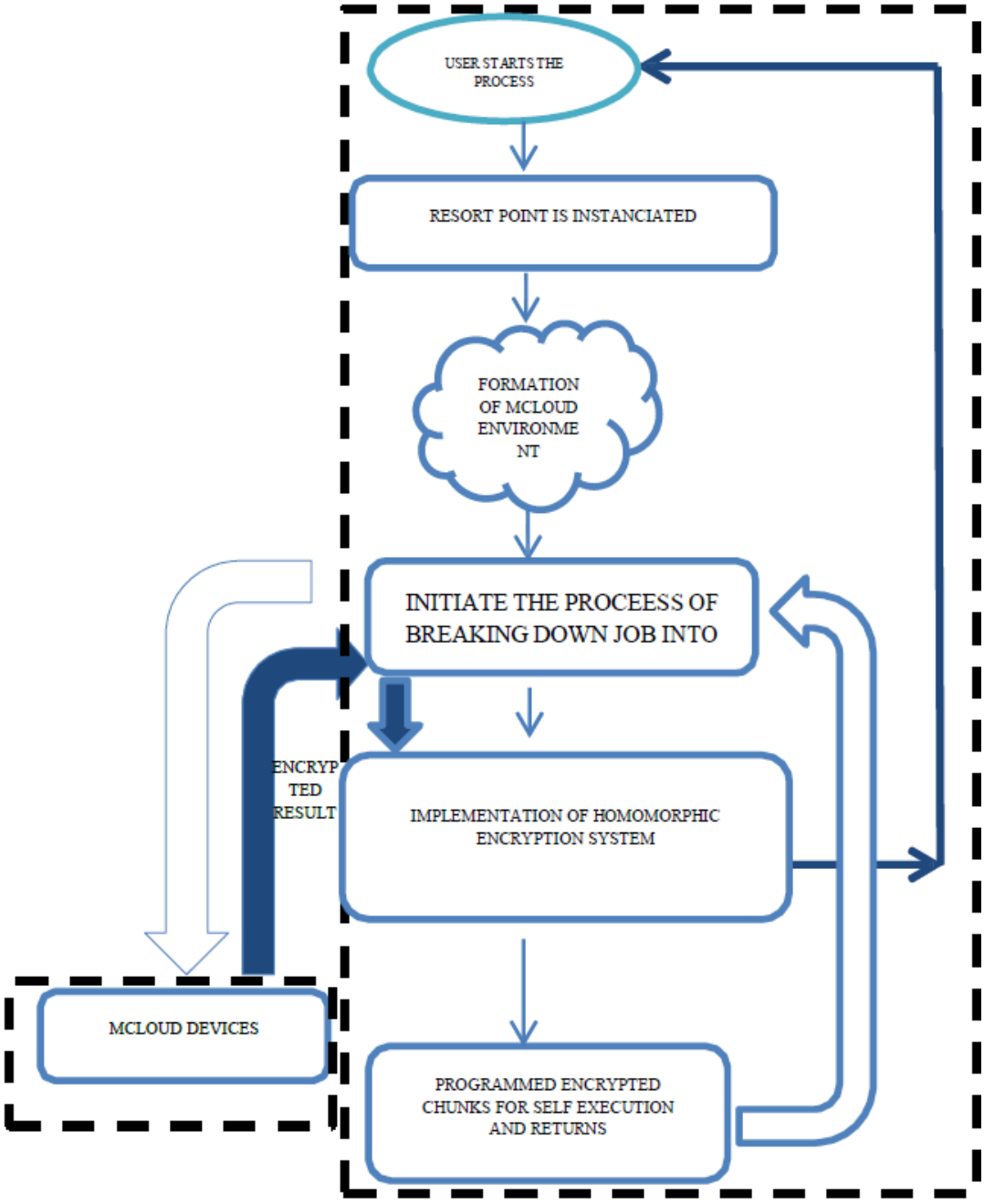

Figure 3. PROPOSED OFFLOADING OVERVIEW

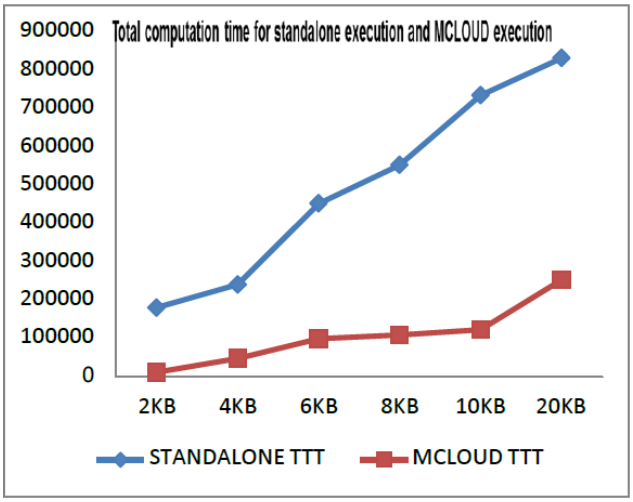

Figure 4. Computation Time of Standalone against MCLOUD

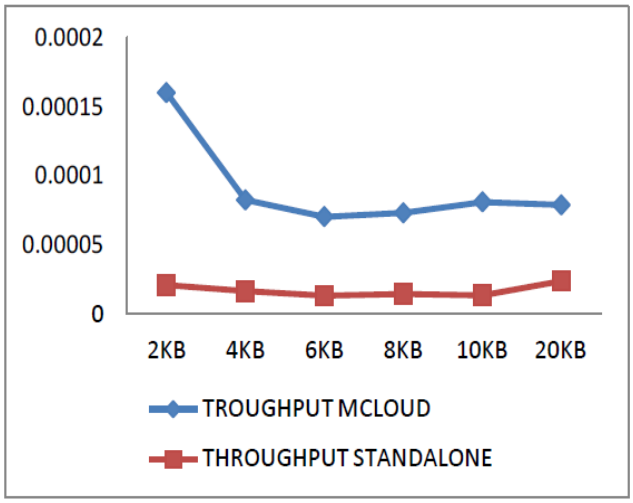

Figure 5. Throughput of Standalone against MCLOUD Execution

4. Result

Using a test bed of three (3) smartphone devices, several task sizes ranging from 2KB, 4KB up to 20KB were used to test the implemented security framework and the time taken to complete computation of each task size is recorded for both MCLOUD and standalone architecture, the total execution time was compared and findings shows that computation involving security on MCLOUD takes less time compared to computation on standalone devices, the following readings were recorded. For the 4KB task size, MCLOUD spent 48500microseconds while standalone spent 241000microseconds; for the 6KB task size, MCLOUD spent 99500microseconds while standalone spent 453000microseconds; whereas in the 8KB task size, MCLOUD spent 109500microseconds while standalone spent 553000microseconds, which is approximately five (5) times faster than standalone execution. MCLOUD framework was observed to have a lower computation time, decreasing computational time ratio, higher throughput per seconds.

5. Conclusion

This research presented a concept of security in mobile device computing (MCLOUD), considering device to device architecture of mobile cloud computing. This secure offloading framework for smartphone devices will surely affect subscriber's response positively towards (MCLOUD) mobile device computing and it will facilitate and encourage the adoption of MCLOUD by smartphone users, due to the security of their work and considerable time taken for the distributed computation rather than standalone computation since all smartphone are resource limited.

References

[1] Dinh, H. T., Lee, C., Niyato, D., & W ang, P. (2013). A survey of mobile cloud computing: architecture, applications, and approaches. *Wireless communications and mobile computing*, *13*(18), 1587-1611.

[2] Mell, P. M., & Grance, T. (2011). Sp 800-145. the nist definition of cloud computing.

[3] Qi, H., & Gani, A. (2012, May). Research on mobile cloud computing: Review, trend and perspectives. In *Digital Information and Communication Technology and it's Applications (DICTAP), 2012 Second International Conference on* (pp. 195-202). ieee.

[4] Li, X., Zhang, H., & Zhang, Y. (2009, December). Deploying mobile computation in cloud service. In *IEEE International Conference on Cloud Computing* (pp. 301-311). Springer Berlin Heidelberg.

[5] Kemp, R., Palmer, N., Kielmann, T., & Bal, H. (2010, October). Cuckoo: a computation offloading framework for smartphones. In *International Conference on Mobile Computing, Applications, and Services* (pp. 59-79). Springer Berlin Heidelberg.

[6] Nagar, N., & Suman, U. A Secure Mobile Cloud Storage Environment using Encryption Algorithm.

[7] Pal, S., & Henderson, T. (2013, September). MobOCloud: extending cloud computing with mobile opportunistic networks. In *Proceedings of the 8th ACM MobiCom workshop on Challenged networks* (pp. 57-62). ACM.

[8] Kester, Q. A. (2013). A Hybrid Cryptosystem based on Vigenere cipher and Columnar Transposition cipher. *arXiv preprint arXiv:1307.7786*.

[9] Li, X., Zhang, H., & Zhang, Y. (2009, December). Deploying mobile computation in cloud service. In *IEEE International Conference on Cloud Computing* (pp. 301-311). Springer Berlin Heidelberg.

[10] Pedersen, M. V., & Fitzek, F. H. (2012). Mobile clouds: The new content distribution platform. *Proceedings of the IEEE*, *100* (Special Centennial Issue), 1400-1403.