Reliable the Resources of Mobile Devices in Cloud Computing

Diaa salama Abd-Elminaam [1] , Farah Mohammed Alenezi[2] ,Prof.khalid M Hosny[3]

Information Systems Department, Faculty of computers & Informatics, Benha university, Egypt[1,2]
Information Technology Department, Faculty of computers & Informatics, Zagazig university, Egypt[3]

ABSTRACT
Every day, an enormous amount of terabyte authentication data is generated. These data are generally stored, analyzed and retrieved according to different computational methods. The need of stored, analyzed and retrieved these data efficiently become a must. The aim of the proposed system is to build an authentication system that determine the identity of the cloud users based on their fingerprints. This system uses image processing techniques to analyze and retrieve the user data. The system shows that applying all processing of registering new user, converting and checking fingerprints and saving fingerprints on the mobile side results that the average time consumed to process the data is 16 second. The system shows that applying all processing of registering new user, converting and checking fingerprints and saving fingerprints on the server side with keeping the interface on the mobile side results that the average time consumed to process the data is 4.11 second. The system will help cloud users to finish the authentication process efficiently.

Keywords
Big data, Cloud Computing, Image Processing, Authentication, User Identity, Fingerprints.

1. INTRODUCTION

Therefore, the emergence of scientific computing, especially large-scale data-intensive computing for science discovery, is a growing field of research for helping people analyze how to predict or to explore more possibilities for data analysis. Since tremendous amounts of data are collected, parallel computing options, such as the cloud pleasingly model, have recently been adopted due to its easy-to-use and easy-to-scale features. However, this model is not yet universally accepted. Software developers are always worried about cost, computational models, and comparisons to previous computational models. Typically, the migration from original sequential model to parallel model is the main challenge [1].

Cloud computing is a general term for anything that involves delivering hosted services over the Internet. The name cloud computing was inspired by the cloud symbol that's often used to represent the Internet in flowcharts and diagrams. Services that the cloud computing systems provide divided in to three categories: Infrastructure-as-a-Service (IaaS) which represents the capability provided to the consumer is to deploy onto the cloud infrastructure consumer created or acquired applications created using programming languages and tools supported by the provider, Platform-as-a-Service (PaaS) which represents the capability provided to the consumer is to deploy onto the cloud infrastructure consumer created or acquired applications created using programming languages and tools supported by the provider and lastly Software-as-a-Service (SaaS) which represents the capability provided to the consumer is to use the provider's applications running on a cloud infrastructure [2]. Cloud computing is changing the way we interact with devices, software, data and processes. But some things never change, and one thing that remains true across the old and new computing paradigms is the importance of authentication to confirm the identity of the user and/or system with which we're communicating [3].

Security as a service (SECaaS) is a business model in which a large service provider integrates their security services into a corporate infrastructure on a subscription basis more cost effectively than most individuals or corporations can provide on their own, when total cost of ownership is considered. In this scenario, security is delivered as a service from the cloud, without requiring on-premises hardware avoiding substantial capital outlays. These security services often

Manuscript received November 5, 2025
Manuscript revised November 20, 2025
https://doi.org/10.22937/IJCSNS.2025.25.11.22

include authentication, anti-virus, anti-malware/spyware, intrusion detection, and security event management, among others [4].

Identity management and authentication form the basis for security whether in the cloud or on the local network. Managing identities has been enough of a challenge within the corporate network, and became more so as businesses formed federations for the purpose of sharing resources across organizational lines. Private, public and hybrid clouds are adding yet another layer of complexity [5].

The proposed system aims to build an authentication system that determine the identity of the cloud users based on their fingerprints. The proposed system register the fingerprints of the should user, and minimize the time taken to verify this fingerprints in the user's login on cloud.

The remainder of the paper is organized as follows: In section 2, an overview for the previous works related to our subject is presented. In section 3, the materials and methods of the proposed system is described. In section 4, results and discussions are produced, before drawing conclusions and future work in section 5 and 6.

2. RELATED WORK

Kumar et al. [6] enlightened the current trend in Mobile Cloud Computing, its security issues/ challenges and associated research challenges in which our research would be focusing on. They offered a more reliable security in Mobile cloud computing. They mentioned that the need of security is very much essential on various concerned areas. They conclude that their future research would be focusing on enhancement of existing security frameworks & multi-factor authentication technologies for highly secured data using cryptography.

Chavan et al. [7] described how the images will be browsed from local directory, and how they will be stored in a blob storage on cloud. They proposed a cloud based CBIR SaaS architecture due to which the services of CBIR will be dynamically made available throughout the desired systems resulting in increase in applications scalability, flexibility and availability. The outcome of the similarity measure is expected to be higher than the threshold value. The cloud services provided by cloud architecture will handle all the unexpected traffic, and it will simultaneously benefit with minimized cost. CBIR will no longer behave as a product and hence will be available to the intended users dynamically.

Bonkra et al. [8] focused on very sophisticated technology defined as mobile offloading and mash up. They also described the computational offloading scheme for mobile devices to perk up the energy and time consumption for any kind of application. They also represented how this computational offloading is works with the client and server distributed system. For this offloading, in this paper we also defined the optimal partition algorithm for find out the optimal solution for our problem. At last there are some evaluation shows significant improvements of energy consumption and time for the android mobile device with the help of offloading. They conclude that this technique will be applicable for more complex instance and tasks. Also they will take care about the hardware operations and solved the calculation delay and memory usage.

Sundaram et al. [9] proposed a novel approach to enhance the QoS of multimedia streaming in the context of smart environments. In the proposed scheme Mobile Edge Computing plays a vital role in enhancing QoS by a new method called the Track the Edge Approach. They considered the applications of mobile cloud for developing the smart environments. They explored the container-based virtualization techniques to provide active Mobile Edge Computing (MEC) environments. In particular, Their scheme guarantees fast response time, by proactively increasing service replication. They have created a test bed for conducting experiments and the results prove that the significant improvements in the performance of the proposed strategy over traditional approaches in terms of quick migration handover and less latency.

Besides the original MapReduce implementation by Google [11], several other MapReduce implementations have been realized within other systems, including Hadoop [12], GridGain [10], Skynet [11], MapSharp [12], and Disco [13]. Another system sharing most of the design principles of MapReduce is Sector/Sphere [14], which has been designed to support distributed data storage and processing over large Cloud

systems. Sector is a high-performance distributed file system, and Sphere is a parallel data processing engine used to process Sector data files. In [15], a distributed data mining application developed using such system has been described.

Several applications of the MapReduce paradigm have been demonstrated. Other examples of interesting applications that can be expressed as MapReduce computations, including: performing a distributed grep, counting URL access frequency, building a reverse Web-link graph, building a term-vector per host, and building inverted indices, performing a distributed sort. In [12], many significant types of applications that have been (or are being) implemented by exploiting the MapReduce model, including machine learning and data mining, log file analysis, financial analysis, scientific simulation, image retrieval and processing, blog crawling, machine translation, language modeling, and bioinformatics have been mentioned.

3. The Proposed Mobile Cloud Computing Model (PMCC)

The proposed system developed to minimize the time taken to check the user authentication on mobile. The aim of the system is to efficiently store and retrieve the data to/from the cloud. The system was developed based on two phases which are: applying all processing of registering new user, converting and checking fingerprints and saving fingerprints on the mobile side; applying all processing of registering new user, converting and checking fingerprints and saving fingerprints on the server side with keeping the interface on the mobile side. Figure 1 Illustrate the first phase which is applying processing on mobile side and Figure 2 illustrate the second phase which is applying processing on server side.

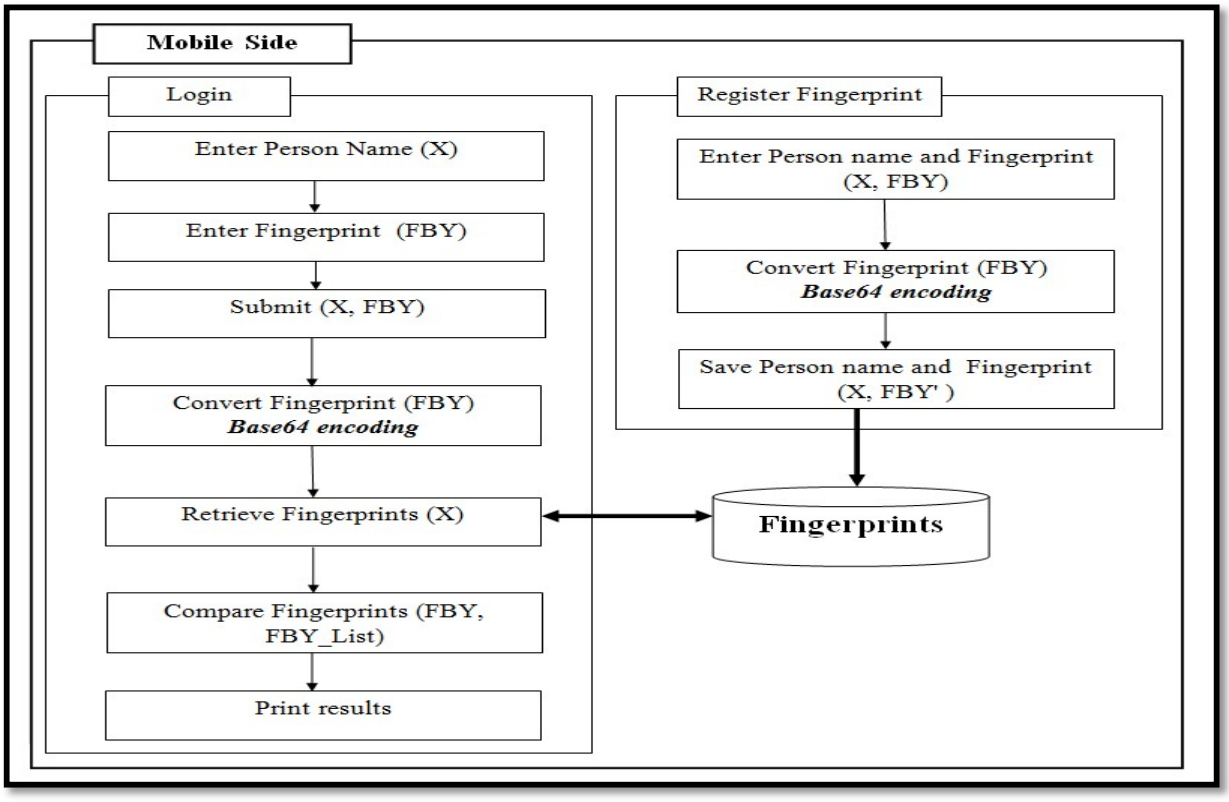

Figure 1 - Applying processing on mobile side

3.1. The algorithm of phase one

❖ **Preprocessing:**

To register new person several steps have to be done:

Step 1: Person will enter his name (X) and his fingerprint (FBY).
Step 2: The system will convert fingerprint (FBY) to encoded fingerprint (FBY') using base64 encoding algorithm.
Step 3: The system will save the person name and encoded fingerprint (FBY') in the database "Fingerprints".

Hint: Person (X) can register more than one fingerprints.

To login several steps have to be done:

Step1: Person will enter his name (X) and his fingerprint (FBY).
Step2: The system will submit person name (X) and his fingerprint (FBY) to the web service GetData (X, FBY).
Step3: The system will convert fingerprint (FBY) to encoded fingerprint (FBY') using base64 encoding algorithm.
Step 4: The system will retrieves all fingerprints (FBY_List) belongs to person (X) form the database "Fingerprints".
Step5: The system will compares all fingerprints (FBY_List) belongs to person (X) form the database "Fingerprints" with the encoded fingerprint (FBY').
Step6: The system will return the results to the web service GetData (X, FBY).
Step7: The system will return the results to the user (Login successfully - error).

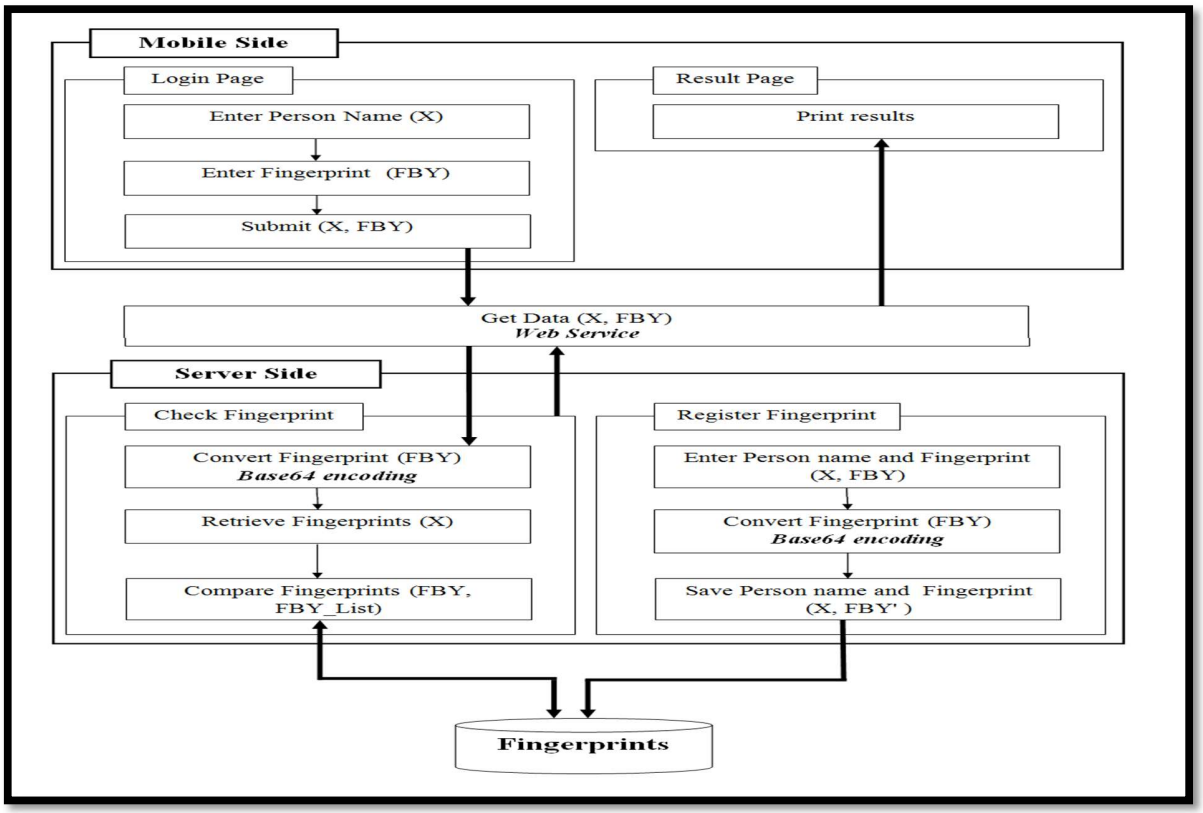

Figure 2 - Applying processing on server side

1.2. The algorithm of phase two

❖ **Preprocessing:**

At server side, to register new person several steps have to be done:

Step1: Person will enter his name (X) and his fingerprint (FBY).
Step2: The system will convert fingerprint (FBY) to encoded fingerprint (FBY') using base64 encoding algorithm.
Step3: The system will save the person name and encoded fingerprint (FBY') in the database "Fingerprints".

Hint: Person (X) can register more than one fingerprints.

At mobile side, to login several steps have to be done:

Person will enter his name (X) and his fingerprint (FBY).

At server side, to check person name (X) and his fingerprint (FBY) several steps have to be done:

Step1: The system will submit person name (X) and his fingerprint (FBY) to the web service GetData (X, FBY).
Step2: The system will convert fingerprint (FBY) to encoded fingerprint (FBY') using base64 encoding algorithm.
Step3: The system will retrieves all fingerprints (FBY_List) belongs to person (X) form the database "Fingerprints".
Step4: The system will compares all fingerprints (FBY_List) belongs to person (X) form the database "Fingerprints" with the encoded fingerprint (FBY').
Step5: The system will return the results to the web service GetData (X, FBY).

❖ **Print result:**

At mobile side, to print the results several steps have to be done:

The system will return the results to the user (Login successfully - error).

4. RESULTS AND DISCUSSION

Extensive experiments are preformed to study the efficiency of the implemented system. The system was tested on different datasets sizes. The size of the images that contain the fingerprints ranging from 10 kb to 480 kb. The two phases were tested and Table 1 and Figure 3 show the time consumed by applying all processing of registering new user, converting and checking fingerprints and saving fingerprints on the mobile side. The system runs 50 times for each datasets size to calculate the average time. The result show that the average time consumed to process the data is 16 second.

Table 1 - Time consumed in mobile side

Database size (records)	Number of tests	Time (seconds)
250	50	4.82
500	50	6.03
750	50	8.37
1000	50	9.23
1250	50	10.96
1500	50	12.17
1750	50	14.65
2000	50	16

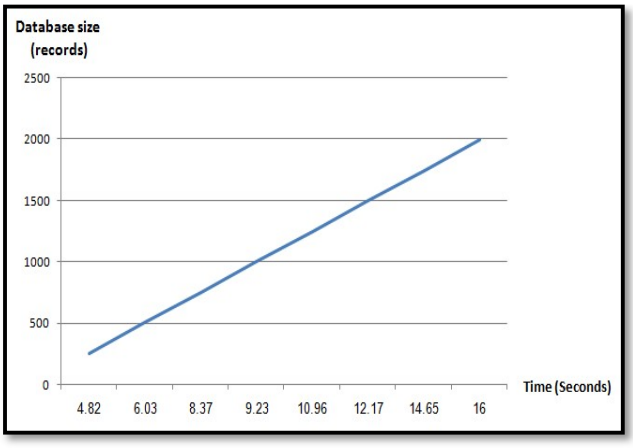

Figure 3 - Time consumed in mobile side

Also Table 2 and figure 4 show the time consumed by applying all processing of registering new user, converting and checking fingerprints and saving

fingerprints on the server side with keeping the interface on the mobile side. The average time consumed to process the data is 4.11 second.

Table 2 - Time consumed in PMCC model

Database size (records)	Number of tests	Time (seconds)
250	50	1.23
500	50	1.76
750	50	2.03
1000	50	2.72
1250	50	3.17
1500	50	3.56
1750	50	3.87
2000	50	4.11

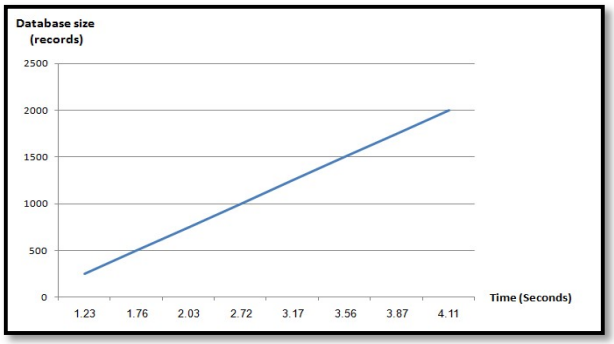

Figure 2 - Time consumed in server side

We apply a time comparison between phase one and phase two illustrated in Table 3 and Figure 3 before making a comparison between the two phase based on security level, memory space and Centralization in Table 4.

Table 3 - A time comparison between the two phases

Database size (records)	Number of tests	Time (seconds) Mobile side	Time (seconds) PMCC model
250	50	4.82	1.23
500	50	6.03	1.76
750	50	8.37	2.03
1000	50	9.23	2.72
1250	50	10.96	3.17
1500	50	12.17	3.56
1750	50	14.65	3.87
2000	50	16	4.11

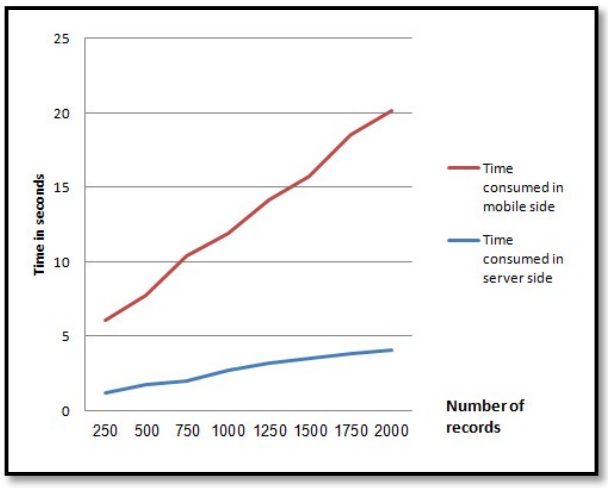

Figure 3 - Time comparison between phase one and phase two

5. CONCLUSTION AND FUTURE WORK

In this paper we present a mobile computing model that efficiently stores, analyzes and retrieves data between mobile side and cloud side. The proposed model improves the performance of the mobile applications by decreasing the time consumed on the run time. The system is separated into two parts, the first part (a small part) run on the mobile side and the second part (a large part) run on the cloud side. The proposed system helps in the authenticating the cloud by determining the identity of the cloud users based on their fingerprints. This system uses image processing techniques to analyze and retrieve the user data. The proposed system proves that applying all processing of registering new user, converting and checking fingerprints and saving fingerprints on the cloud side with keeping the interface on the mobile side is more efficient than applying all processing of registering new user, converting and checking fingerprints and saving fingerprints on the server side with keeping the interface on the mobile side. By calculating the time consumed by applying all processing of registering new user, converting and checking fingerprints and saving fingerprints on the mobile side, the result show that the average time consumed to process the data is 16 second. By calculating the time consumed by applying all processing of registering new user, converting and checking fingerprints and saving fingerprints on the cloud side

with keeping the interface on the mobile side, the result show that the average time consumed to process the data is 4.11 second. This is mean that applying the large part of processes on the cloud side saves the time of the system. The result shows that applying the large part of processes on the cloud side consumed only 25% of the time consumed to run all processes on the mobile side.

Table 4 - A comparison between the mobile side and server side

Factor	Mobile side	Server side
Time	Time increases dramatically by the increasing on the size of the database	Time does not increase dramatically by the increasing on the size of the database
Space	Large space is needed to store the database and affect the time	the database size does not affect the time
Security	-	The disadvantage of security issues
Centralization	The disadvantage of decentralization	The advantage of centralization

6. REFERENCES

[1]. Marjani, M., Nasaruddin, F., Gani, A., Karim, A., Hashem, I. A. T., Siddiqa, A., & Yaqoob, I. (2017). Big IoT Data Analytics: Architecture, Opportunities, and Open Research Challenges. IEEE Access, 5, 5247-5261.

[2]. Tilwani Mashook1, Harshvardhan Shani and Nidhi Barot, "A Survey on Cloud Computing." (2017).National Conference on Latest Trends in Networking and Cyber Security, March 2017.

[3]. Sindhu, S., & Sindhu, D. (2017). Cloud Computing Models and Security Challenges. International Journal of Engineering Science, 10934.

[4]. Kaur, S., & Singh, A. (2017). Security issues in Cloud Computing. International Education and Research Journal, 3(5).

[5]. Singh, A., & Boparai, R. S. (2017). A Review on Big Data and Cloud Computing Security. International Journal of Engineering Science, 11365.

[6]. Kumar, G. Kishore, and M. Gobi. "Survey on Mobile Cloud Computing [MCC], its Security & Future Research Challenges." International Research Journal of Engineering and Technology (IRJET), Volume: 04 Issue: 06 (2017).

[7]. Chavan, C., Dixit, M., & Scholar, P. G. (2017). A Survey of Content Based Image Retrieval in Cloud Computing. International Journal of Engineering Science, 12718.

[8]. Bonkra, A., & Student, U. G. (2017). Survey on Computational Offloading In Mobile Cloud Computing Environment. International Journal of Engineering Science, 11893.

[9]. Sundaram, C. M., & Shanavas, A. M. (2017). Improving the QoS of Mobile Cloud Computing Applications in Smart Environments. International Journal of Applied Engineering Research, 12(11), 2754-2760.

[10]. Hadoop Credits Page (2012) http://hadoop.apache.org/core/credits.html. (Visited: September 2012)

[11]. Google's Map Reduce (2012). http://labs.google.com/papers/mapreduce.html [Accessed: September 2016]

[12]. Hadoop (2012) http://hadoop.apache.org [Accessed: September 2016]

[13]. Gridgain (2012) http://www.gridgain.com [Accessed: September 2016]

[14]. Skynet (2012) http://skynet.rubyforge.org [Accessed: September 2016]

[15]. MapSharp (2012) http://mapsharp.codeplex.com [Accessed: September 2016]