The Seybold Report
ISSN 1533-9211
DOI 10.5281/zenodo.6686469

AN EFFICIENT AND SECURED FRAMEWORK FOR MOBILE CLOUD COMPUTING

[1]Dr. S.D.PRABU RAGAVENDIREN, [2]MOHAMED FAHAD A, [3]DHANANJEYAN TG, [4]VENKATESH K and [5]SARAVANAN K

[1] Professor, Department of Computer Science and Engineering, RVS Technical Campus Coimbatore.
[2,3,4,5] Students, Department of Computer Science and Engineering, RVS Technical Campus Coimbatore.

Abstract

Our daily lives are dominated by the use of smart phones. These gadgets, however, have drawbacks such as short battery life, limited compute power, small memory sizes, and unreliable network access. As a result, a variety of offloading-based methods have been developed to alleviate these limits and improve battery life time. A novel paradigm for offloading intense processing activities from mobile devices to the cloud is proposed. The offloading choice is made dynamically using an optimization model based on four primary parameters: energy consumption, CPU utilisation, execution time, and memory usage. In addition, a new security layer has been added to secure the cloud-transferred data from any assault. The results of the experiments demonstrated that the framework can choose an appropriate offloading decision for various sorts of mobile app workloads while obtaining considerable performance gains. Furthermore, unlike earlier solutions, the framework can defend application data against any threat.

Keywords: Cloud Computing, Mobile Computing, Offloading Technique, Mobile Device, Energy Consumption, Battery Lifetime, Memory Size, Security Layer, Task Analysis, Computer Architecture, Execution Time, Optimization Model, Smart phones, CPU utilization, Transferred Data

1. INTRODUCTION

Face identification, augmented reality, picture and video processing, video gaming, and speech recognition are just a few of the applications available on smart phones. These are complicated applications, and the need for computing resources is growing. Despite developments in smart phone technology, one of the key hurdles in improving computing requirements through battery upgrades has remained the duration of battery life. Cloud computing provides unrestricted access to resources over the internet. Self-service provisioning, elasticity, extensive network access, resource pooling, low costs, and ease of use are just a few of the benefits of cloud computing. As a result, mobile cloud computing was developed to address the shortcomings of smart phone devices. Mobile cloud computing is a novel paradigm that combines cloud computing technologies with mobile devices to boost application speed and lengthen battery life. Recent research has suggested that all or portion of mobile applications be offloaded to the cloud for remote execution. In the offloading decision, these frameworks are meant to make a trade-off between one or more limitations, such as mobile device energy consumption, CPU utilisation, execution time, remaining battery life, and data transfer amount on the network. However, most of these models do not take memory utilisation into account when deciding whether or not to offload. Mobile applications use memory as one of their primary resources. Furthermore, no security

The Seybold Report
ISSN 1533-9211
DOI 10.5281/zenodo.6686469

procedures are used to safeguard offloaded data from attackers. As a result, the focus of this work is on developing a new model that incorporates the majority of the aforementioned limitations in order to increase the speed of mobile applications while also protecting the application data from attack. In our day-to-day lives, we frequently utilise mobile phone devices.

Nonetheless, these gadgets have drawbacks, such as a short battery life, limited computation control, a small memory gauge, and an unusual framework arrangement. As a result, many courses of action have been recommended to remove these impediments and extend the battery lifetime by utilising the offloading mechanism. In this paper, a new system is proposed for offloading real-time counting tasks from mobile phones to the cloud. This system employs a breakthrough model to make an offloading decision that is logically based on four key parameters: essentialness, CPU usage, execution time, and memory usage.

Another layer of protection is added to ensure that the data in the cloud is safe from assault. The preliminary results revealed that the structure may choose an appropriate offloading strategy for various types of adaptive application assignments while generating massive performance improvements. Furthermore, while not identical to previous approaches, the framework can protect application data from any danger.

2. LITERATURE REVIEW

Author: N. Vallina-Rodriguez and J. Crowcroft

Depiction: In today's PDAs, effectively managing energy is critical. Because of the wide range of remote interfaces and sensors available, as well as the growing popularity of power-hungry apps that take advantage of these features, the battery life of multifunctional handhelds can be reduced to just a few hours of use. The research area, as well as the working framework and equipment vendors, uncovered remarkable enhancements and approaches for extending the battery life of mobile phones. Nonetheless, the cutting-edge of lithium-ion batteries unambiguously demonstrates that energy productivity should be achieved at both the equipment and programming levels.

Author: G.Motta, N. Sfondrini, and D. Sacco

Depiction: We provide a Systematic Literature Review (SLR) on Cloud Computing, which included 39 papers selected from first-level diaries and meetings between 2008 and 2012[12].The methodology captures the relevant viewpoints on Cloud Computing, particularly the overall worldview on conveyance and transmitting.

Following that, we will discuss the cutting-edge concerns of system management and information for executives in Cloud Computing. Because of the longer timescale and more intensive exploratory test, this SLR takes longer than previous surveys.

Author:Erway, A.Kupcu, C.Papamanthou, and R.Tamassia (2009)

The Seybold Report
ISSN 1533-9211
DOI 10.5281/zenodo.6686469

In flexible conveyed processing, phones can rely on appropriated calculating and data storing resources to do computationally conceived workouts like seeing, data mining, and media handling. Compact cloud increases the assignment of standard uniquely delegated framework by seeing mobile phones as organisation focal points, such as identifying organisations, in addition to providing ordinary estimate organisations. To ensure customer security in the cloud, recognised data, such as territory encourages and prosperity-related data, should be organised and stored in a secure manner. To that aim, we provide a flexible cloud data management system based on trust the board and private data separation. Finally, the game plan is demonstrated through the deployment of a pilot programme called FocusDrive, which aims to improve young people's driving success.

Author:A.Fiat and M. Naor (1994)

Another exciting innovation is cloud preparation, which is altering the way people think about ordinary Internet enrollment and, without a doubt, the entire IT sectorUsing the rapid gains in remote access innovations, dispersed figuring is expected to expand in the flexible condition.Compact dispersed registering frameworks and models are used to create these adaptive applications.Customers can store their data remotely and access high-quality on-demand cloud applications in the Mobile Cloud environment without the hassles of procuring and maintaining their own unique neighbourhood hardware and code.Data security, on the other hand, has remained a major worry and the standard block shielding circulated figuring has yet to be fully appreciated.This anxiety stems from the way sensitive data stored in open fogs is managed by corporate expert centres that are unlikely to be completely trustworthy. There are a few security and assurance challenges that need be addressed within that restriction. This section provides an overview of the appropriated processing concept, as well as a depiction of handy dispersed registering and the numerous security difficulties relevant to the flexible disseminated figuring situation.

Author: L.Cheung, J. Cooley, R. Khazan, and C. Newport (2007)

Mobile devices (e.g., tablets, wifi, computers, etc.) are rapidly evolving into an integral part of human existence.These mobile devices nonetheless require resources that are distinct from those required by a traditional information-processing device, such as PCs and workstations.Mobile Cloud Computing is the answer to addressing these problems.Another customizable framework that allows direct use of cloud resources to increase the capacity of benefit-required PDAs is proposed.The primary elements of this paradigm divide a single application's bundle into pieces.Compact apps can be run on the PDA or transferred to the cloud clone for processing. The outcomes demonstrate that when an application is run on a mobile device or in the cloud, memory is depleted and a large number of centres are closed.Most of the advantages consumed on a handy mobile phone will be reduced to about half of the memory utilised for operating applications on the adaptable.On the other hand, while Cloud Computing represents a significant advancement in the field of planning, there are some drawbacks to scattered registering, such as data security.As a result, an encryption technique with a high level of security and a small key size is in high demand.As a result, the primary focus for ensuring security is the development of new cross-breed cryptography.In

The Seybold Report
ISSN 1533-9211
DOI 10.5281/zenodo.6686469

this vein, a novel cross-breed cryptography show has been developed to achieve security in Mobile Cloud Computing.The outcomes demonstrate the potential of novel computation.Our investigation's main goal is to create a secure, versatile cloud infrastructure that would allow wireless apps to be passed on both in terms of data and figuring.

Author:Ateniese, R. Expends, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Tune (2007)

Cloud enlistment is an exciting phenomenon that is altering the way people think about the Internet and the IT business.With the advancement of remote access advancements, suitable processing is required to develop to flexible situations, where PDAs and sensors are used as data gathering centres for the cloud.

In any case, customers' concerns about data security are critical impediments to broad data collection.

These tensions stem from the fact that sensitive data is stored in open fogs, which are managed by business professional groups that are not trusted by data owners.As a result, new secure organisational structures are relying on to meet customer security concerns when using distributed registering methods.

Authors:J.Bethencourt, A. Sahai, and B. Waters (2007)

In this paper, we present a broad security framework for inspecting data accumulating in open fogs, with a special emphasis on lightweight remote gadget store and recuperate information without revealing data substance to cloud expert associations. In order to achieve this goal, our solution revolves around two research paths:To begin, we present a novel Privacy Preserving Cipher Policy Attribute-Based Encryption (PP-CP-ABE) to protect customer data. Lightweight contraptions can securely redistribute powerful encryption and unscrambling exercises to cloud authority communities using PP-CP-ABE, without revealing data or used security keys.Second, as a cryptographic access control system, we propose an Attribute Based Data Storage (ABDS) structure.The ABDS achieves data theoretical optimality by limiting estimation, accumulating, and correspondence overheads. ABDS, in particular, limits cloud organisation charges by reducing correspondence overhead for data organisations.Our implementation evaluations show the level of security and efficiency of the demonstrated course of action, such as computation and limit.

Authors: D.Boneh, X. Boyen, and E.J. Goh (2005)

In our day-to-day lives, we frequently use smartphones.Regardless, these contraptions have limitations, such as a short battery life, compelled estimation control, limited memory capacity, and an unusual framework structure.A unique framework is given in this paper to shift genuine figuring tasks from the phone to the cloud.This system employs a streamlining model to logically determine an offloading decision based on four core parameters: imperativeness, CPU usage, execution time, and memory usage.A second layer of security is added to protect the cloud-based data from any attack.The results of the tests revealed that the

ISSN 1533-9211

DOI 10.5281/zenodo.6686469

framework can select an appropriate offloading strategy for a variety of adaptive application tasks while obtaining minimal performance gains.

3. EXISTING SYSTEM

The reduction of absolute energy use while maintaining dependability and timing constraints is investigated. The analysis offered an energy-conscious wonderful task booking figuring that employed a written non-cyclic chart (showing the job need and its corresponding cost) and a basic way task approach to manage acquire the best execution solicitation of each task while limiting overall energy use. However, this model focused solely on the energy utilisation metric and ignored other important data like as memory usage, CPU usage, and remaining battery life, all of which are considered relevant measurements.

3.1 Disadvantages of existing system

Security issues

Below we address the security in mobile cloud computing at three levels:

a. Mobile terminal

It is an open operating system that offers wireless internet connection from anywhere at any time. Third-party apps and personalization are also supported. As a result, security issues in mobile terminals are quite significant, and we'll go over them in detail below, including malware, software flaws, and other considerations.

1) Malware: Malware gets access to personal information of users as they automatically downloaded and carried which remains unknown to the users. So many anti malware software have been developed but due to limited resources and capacity of mobile terminals significant computational resources are difficult to achieve. So, solutions for malware detection and prevention in mobile terminals are needed.

2) Software Vulnerabilities: In case of application software, user name and password are transferred to network by using FTP and these are stored in clear text format. This allows illegal access of mobile phones from computers on the same network and so personal information not remains secured. Whereas, in the operating system, there are code defects that, in some cases, lead to attackers destroying mobile phones.

b. Mobile network security

Mobile devices can connect to the network in a variety of ways, including using phone services, sending SMS, and using other internet services. Smart phones can also connect to the network via Wi-Fi and Bluetooth. As a result, these access modes expose users to security risks and harmful attacks.

The Seybold Report
ISSN 1533-9211
DOI 10.5281/zenodo.6686469

c. Mobile cloud

Platform stability and data and privacy protection are two areas where mobile cloud security is addressed. These two are covered in the following paragraphs: Reliability of the Platform: Because the cloud stores a large amount of sensitive information, it is always at risk of being attacked. Outside viruses, cloud users, or insiders could all be involved in these attacks. The attackers' goal is to bring down cloud services. For example, a DOS (Denial of Service) attack might shut down cloud services by damaging the platform.

d. Data and privacy protection

The ownership and management of users' data resides at separate locations and also the users do not know the exact location of the infrastructure where their data are stored. So, data protection and privacy is of great concern in mobile cloud computing environment.

- data loss or theft
- data leakage
- account or service hijacking
- insecure interfaces and APIs
- denial of service attacks
- technology vulnerabilities, especially on shared environments

4. PROPOSED SYSTEM

This paper provides a unique framework for offloading only heavy processes rather than all apps, resulting in less network traffic. The offloading choice is made dynamically at runtime using an optimization model based on four key constraints: job execution time, CPU utilisation, memory usage, and energy consumption. A new security layer is implemented that uses the AES encryption technology to encrypt the task data before transmitting it to the cloud. The experimental investigations use three distinct types of mobile applications to test this framework and demonstrate the selection of an appropriate offloading decision for improved application performance. We presented a novel technique that uses calculation offloading to unload just the serious errands of portable apps. We devised a streamlined approach for determining the offloading option.

The following are the main outcomes and pledges of this paper: This paper presents a novel structure that offloads only concentrated tasks rather than all applications, using less organisation correspondence in the process. An advancement model is designed to determine the offloading option based on the outcomes in a step-by-step manner.

The Seybold Report
ISSN 1533-9211
DOI 10.5281/zenodo.6686469

4.1 Advantages

Flexibility: Mobile cloud computing allows you to store and retrieve data from any device that is linked to the internet from anywhere in the globe. This enables for a smooth data flow whenever information is required.

Multiple Platform Support: Because cloud computing supports a variety of platforms, you can use mobile cloud computing regardless of the platform you're using to execute your applications.

Availability of data at all times: When you use mobile cloud applications, you can obtain real-time data whenever you want. This allows you to view your data whenever you want, as well as save it to the cloud if you want to surf offline.

Cost Efficiency: This service is incredibly cost-effective because there are no excessive costs associated with mobile cloud computing because it is now based on simply paying for what you use.

Backup of data: Because your phone is continually generating new data, the mobile cloud application assists you in backing up your data to the cloud when it needs to be kept secure or when it is not in use.

Data Recovery: If you lose your essential data due to a disaster, the cloud application will always allow you to restore your data from the cloud by following a certain procedure. If you are connected to the internet and have enough storage space on your device, you can recover your data from any location.

5. FRAMEWORK ARCHITECTURE

The architecture of the framework and show how its modules can communicate to achieve the design goals of the system. In addition, the linear optimization model is defined.The framework architecture consists of six modules, namely, estimator, profile, network and bandwidth monitor, decision maker,mobile manager, and cloud manager.First, the framework works at the method level, where the developers need to add an annotation (@Remote) above all intensive methods at the developing step.

Estimator: The estimator module is responsible for identifying these methods for local execution on the mobile device and remote execution on the cloud with different input sizes (stored as a sample) at the installation step.

Profile: The profile module obtains the values from estimator module for each annotated method. Then, the module creates a new file for each method and stores these values into the file. These files are updated after each running process and used by the decision maker module as a history-based file in the offloading decision.

Network and Bandwidth Monitor: This module only monitors the current status of the network and gathers cell connection state and its bandwidth, Wi-Fi connection state and its bandwidth, and signal strength of cell and Wi-Fi connection

The Seybold Report
ISSN 1533-9211
DOI 10.5281/zenodo.6686469

Decision Maker: It is, the core module of the proposed framework, contains an integer linear programming model and decision-making algorithm that predicts at runtime where the annotated methods are executed. The goal of the model is to find an application partitioning strategy that minimizes the energy consumption, transfer data, memory usage, and CPU utilization, in smartphones, subject to certain constraints.Let us assume that we have n number of annotated methods that may be offloaded to the cloud for remote execution, that is, M1, M2,….., Mn. Each method i consists of a set of parameters, namely, input size (inputi), memory usage (memoi), CPU utilization (CPUi), and battery consumption (poweri), for local execution. Other parameters, such as memory used for security(memo_seci), battery consumption used for security (power_seci), and CPU used for security (CPU_seci) are also considered if the method is

offloaded for remote execution. In this model, xi is introduced for each method i, which indicates whether the method is executed locally on the mobile device (xi=0) or offloaded for remote execution (xi=1). Theobjective function is represented as follows:

$$\text{Min } (C_{transfer} * W_{tr} + C_{memory} * W_{mem} + C_{cpu} * W_{cpu} + C_{power} * W_{power})$$

$$x \in 0, 1$$

Where:

$$C_{transfer} = \sum_{i=1}^{n} input_i * x_i$$

$$C_{memory} = \sum_{i=1}^{n} memo * (1 - x_i) + \sum_{i=1}^{n} memo\_sec_i * x_i \quad (1)$$

$$C_{cpu} = \sum_{i=1}^{n} CPU_i * (1 - x_i) + \sum_{i=1}^{n} CPU\_sec_i * x_i$$

$$C_{power} = \sum_{i=1}^{n} power_i * (1 - x_i) + \sum_{i=1}^{n} power\_sec_i * x_i$$

Ctransfe, Cmemory, Ccpuand Cpowerrepresent cost for transferring the input size, memory used, CPU used, and power consumed for method i respectively. wtr, wmemo, wCPU, and wpower are the weights for each these costs, which lead to different objectives.

The Seybold Report
ISSN 1533-9211
DOI 10.5281/zenodo.6686469

Minimize the memory used by the application methods on the mobile device. Memory cost is divided into two parts. The first part is the memory used when the method of the application is executed locally on the mobile device, whereas the second part is used to encrypt the data before transferring to the cloud in the offloading case

$$\sum_{i=1}^{n} memo * (1 - x_i) + \sum_{i=1}^{n} memo\_sec_i * x_i <= Mth \quad (1)$$

Minimize the total execution time, that is, the second constraint, for the application. The total time for executing the application methods remotely on the cloud must be less than the total time for executing the methods of the application locally on the mobile device.

$$Exe\_time\_local > Exe\_time\_cloud \quad (3)$$

$$Exe\_time\_local = C / Sm * x_i \quad (4)$$

Finally, after we solve this formulation, each method xi can be determined whether for local execution (xi=0) or offloading to the cloud (xi=1).PSO is used as best optimization algorithm with significantly better computational efficiency to solve our optimization problem. PSO provides a solution with acceptable speed in the order of tens of milliseconds.

Mobile Manager: It is responsible for sending a binary file containing the method code and its required libraries at the installation step. The mobile manager handles the execution of the method based on the model decision. If the method is executed locally on the mobile device, the files are updated with new values through the profile module. However, if the decision is to offload the method, then the mobile manager encrypts the offloaded data by using AES technique and communicate with the cloud manager module to transfer this data with the method name. Finally, the mobile manager receives and delivers the results to the application. In case of remote execution failure, the mobile manager module executes the method locally.

Cloud Manager: It is the only module deployed on the cloud side. This module is written purely in Java. Therefore, any application can benefit from the proposed framework to offload its computation to any resource that runs the Java Virtual Machine (JVM). Communication between the cloud manager and the mobile manager modules is managed by Ibi communication middleware. In the first communication, at the installation step, a binary file containing the method code and its required libraries are sent to the cloud. Then, the cloud manager receives the methods data and decrypt them in the following run. Then, the manager executes the method remotely and sends the result back to the mobile manager module with the new values to be updated by the profile module.

5.1 Framework execution flow

This section discusses the flow of execution of the proposed framework. Algorithm 1 illustrates the detailed processes of the framework and how the offloading decision for the

ISSN 1533-9211

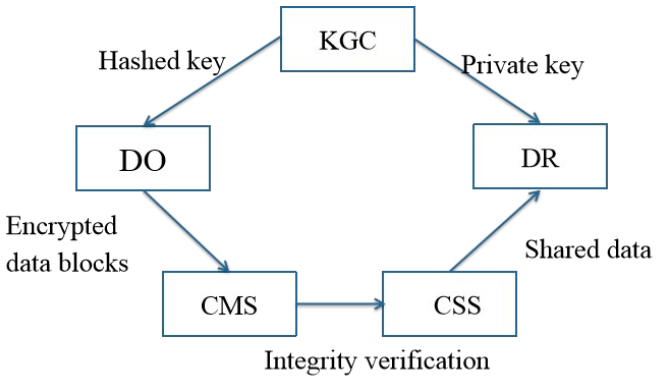
DOI 10.5281/zenodo.6686469

annotated method is made. The time complexity for this algorithm is represented by O(n) and does not consume additional resources from the mobile device. Firstly, at the developing step, the mobile application is partitioned the methods into two types. The first type is the computing-intensive methods which are annotated by developer and requires more computation resources. While the second one is the methods which depend on the device hardware and must be executed locally as discussed above. When no connection or failed connection occurs, then all of the methods are executed locally on the mobile device; otherwise, the decision maker module reads the transfer data size, memory usage, CPU utilization, and energy consumption through the profile module, where these values are stored at the installation step and updated during each run.

5.2 Intergrating the framework in mcc

Here the required steps to integrate the proposed framework in the mobile application. Our framework works on the method level and uses a Java reflection and annotation to identify the methods which can be offloaded. First, at the developing step, the developer needs to add an annotation (@Remote) above each intensive method that can be offloaded for remote execution. Thereafter, each Android application goes through three main builders to generate the APK installation file. The first builder is the Android Pre-Compiler, which generates the Java source files from your Android resources and Java source files for any service interface. Second, the Java Builder compiles the generated files from the first builder. Last, the package builder obtains all compiled files and packages them in an APK file. Our framework adds a new additional builder, called the Class and Jar Generator.

5.3 System architecture



The system model consists of four entities named Key Generator Center (KGC), Data Owner (DO), Cloud Servers (CS) and Data

Requester (DR)

The Seybold Report
ISSN 1533-9211
DOI 10.5281/zenodo.6686469

KEY GENERATION CENTER (KGC): It is responsible for generating public parameters and master key for the system and issuing private key for other entities.

DATA OWNER (DO): It is responsible to generate and encrypt the shared data, define access structures, and divide encrypted data into blocks.

CLOUD SERVERS (CS): Cloud servers come in Cloud Storage Servers (CSS) and Cloud Manage Servers (CMS) based on their roles. To save computation and communication costs of DO and DR, CMS is employed to manipulate complex computations including generating algebraic signatures of blocks, verifying data integrity of shared data and computing the intermediate data for encryption and decryption.

DATA REQUESTER (DR): It is responsible to download and decrypt the shared data for utilization. In the scheme, only the authorized DR is able to download shared data from CSS and decrypt the data.

5.4 Security requirement

In the scheme, we suppose CSS and CMS are both semi-trusted. CSS is responsible to store data and block tags for data sharing. However, once data is corrupt or lost, it might launch forge attack or replace attack for economic reasons.

DATA CONFIDENTIALITY: The shared data must keep confidential to CSS, CMS and any unauthorized DRs for privacy and security. Any disclosure of shared data is undoubtedly harmful to enterprise benefits. Consequently, it is important to ensure the confidentiality of shared data.

DATA INTEGRITY: The data should keep intact before shared by DR. It means that the data is undamaged in an unauthorized manner during storage and sharing process.

AUTHORIZED ACCESS: To achieve authorization, only DR with correct attributes can access shared data stored in CSS.

USER REVOCATION: The membership of DR must be revoked to stop his access to shared data when he leaves the organization. To achieve security of the scheme, user revocation should be required in the data sharing scheme.

DESIGN GOALS: The data sharing scheme for mobile devices is designed to achieve data privacy preservation, data security and lightweight operations.

PRIVACY PRESERVATION: The scheme should satisfy data privacy during data sharing process. As sensitive data is encrypted by data owner before outsourcing to cloud and only authorized data requesters can access the encrypted data, the shared data is private to CSS, CMS and any unauthorized Drs.

ISSN 1533-9211

DOI 10.5281/zenodo.6686469

DATA SECURITY: The scheme should achieve sensitive data security during the whole sharing process. The security requirement can be guaranteed by data confidentiality, data integrity, authorized access and user revocation in the scheme.

5.5 System implimentation

The implementation process begins with preparing a plan for the implementation of the system. After the system is implemented successfully, training of the user is one of the most important subtasks of the developer. For this purpose user manuals are prepared and handled over to the user to operate the developed system. Thus the users are trained to the operate the developed system. The implementation stage involves following Tasks.

- Careful planning
- Investigation of system and constraints.
- Design of methods to achieve the changeover.
- Training of the staff in the changeover phase.
- Evaluation of the change over method.

The maintenance phase of the software cycle is the time in which a software product useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development lifecycle.

6. LIST OF MODULES

- USER REGISTRATION MODULE
- ADMIN MODULE

6.1. User registration module

Before viewing the home page, user has to register their details like user id, password. This registration will used to avoid anonymous users. User has to select the constant factor, for every user a unique random salt will be generated. The user has to select the random function available for him during registration phase

Login: In this module user select a particular digit from his password and this is the value of the variable a, select a particular digit from his random salt password and this is the value of the variable b, enter the constant value selected during the registration phase. Finally apply the values of a, b, and constant value against the user selected function. The system also calculate the result against the user selected values, if user gives the same value as like the system generated password, the system authenticate the user to login on.

File Download: Certain files of Content are available for download from the Cloud. These files are uploaded by data owner. If user have a all rights user download the files. If doesn't download rights user unable to download cloud files user can only view the files. Here we

ISSN 1533-9211

DOI 10.5281/zenodo.6686469

are maintaining accountability of user's account. In an organization some peoples only have all rights. Else have specific rights.

6.2. Admin module

User Account: Here administrator has to register the Cloud Users. We are gathering specific details from user for registration. Admin can view the user details even edit and delete also. We use two rights the first one is downloading and another one is view for priority users. And we generate encryption key for every user. User cans logging their account using by this encryption key.

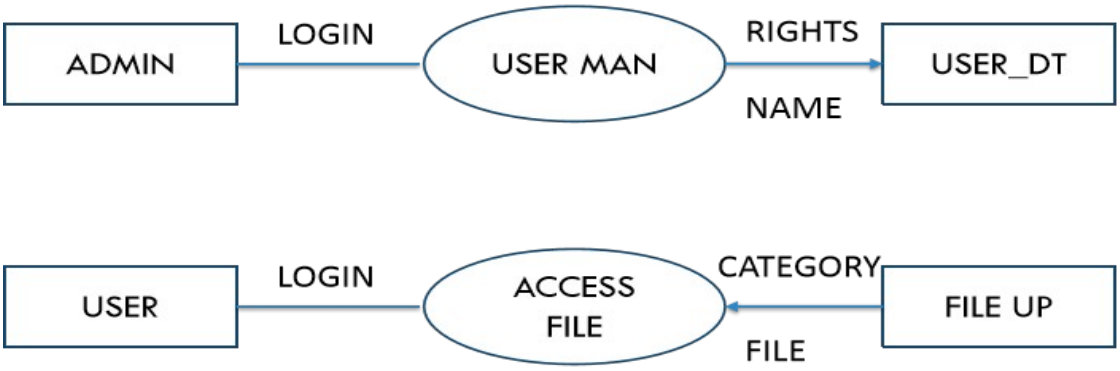
File Upload: If you use a personal computer to log on to a network and you want to send files across the network, you must upload the files from your PC. Admin upload the all files to store the cloud. Cloud computing is Internet-based computing, whereby shared resources, software, and information are provided to computers and other devices on demand, like the electricity grid.

File Download: The act or process of copying data in such a way. So admin can download what he uploaded. Upload files are stored in cloud server. So authorized user could download the files from cloud. Unauthorized user only view the files.

Cloud Visitors: In cloud where this approach is used, Users respond quickly, firmly, and respectfully when User misbehaves in cloud. Minor problems are addressed before behaviour gets out of control. Admin can track the misbehaving user using by Log file. User after the login file create the automatically. But user doesn't know about this log files. So admin can track the user. These rules help to make the cloud safe.

6.3 Data flow diagram

Level 0:

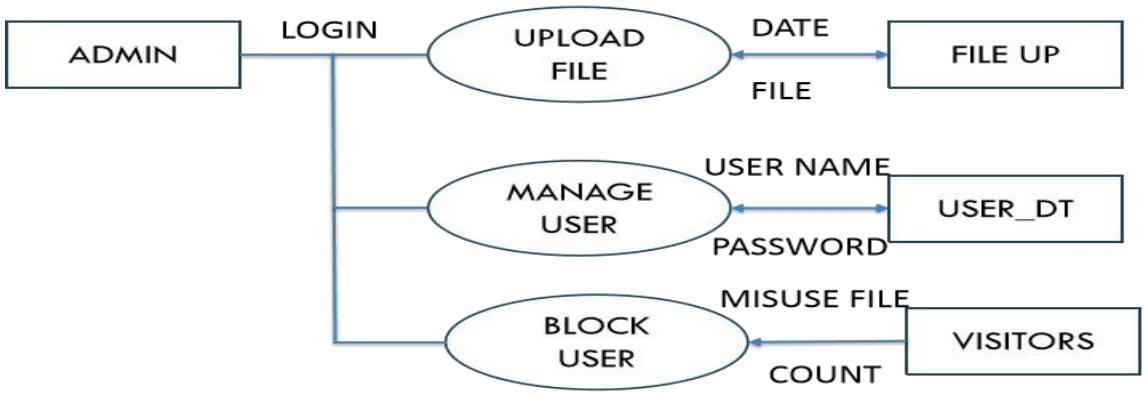


Data Flow Level 0

ISSN 1533-9211

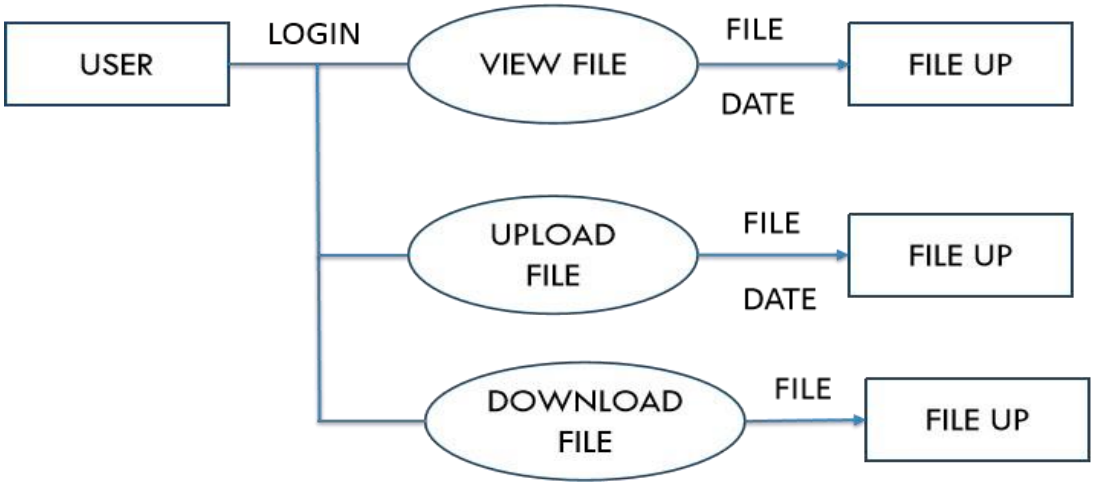
DOI 10.5281/zenodo.6686469

Level 1:



Data Flow Level 1

Level 2:



Data flow Level 2

7. SOFTWARE DESCRIPTION

The various technologies used in the software system are Client-Server Architecture

Over the years there have been 3 different approaches to application development

- Traditional Approach
- Client/Server Approach
- Component-based Approach

The Seybold Report
ISSN 1533-9211
DOI 10.5281/zenodo.6686469

In a traditional approach there was a single application that handles the presentation logic, business logic, and database interactivity. These applications were also called Monolithic applications. The drawback of this approach was that if even a minor change, extension, or enhancement was required in the application, the entire application had to be recompiled and integrated again.

Thus a 2-tiered architecture divides an application into 2 pieces:

- The GUI (client)
- Database (server)

Features of Java Programming Language: Java technology is both a programming language and a platform. The Java Programming Language the Java Programming language is a high-level language that can be characterized by all of the following buzzwords: Simple, Object Neutral, Portable, Distributed, High Performance, and Robust.

The Java Platform: A platform is the hardware or software environment in which a program runs. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware=based platforms. The Java platform has two components:

- The Java Virtual Machine
- The Java Application Programming Interface(API)

Features of the Java Technology: Write less code, Write better code, Develop programs more quickly, Avoid platform dependencies, Write one, run anywhere

J2EE in Client–Server: Database (server):The client sends a request to the server. The server being a more powerful machine does all the fetching and processing and returns only the desired result set back to the client for the finishing touches.

Client Script: HTML, DHTML, JavaScript

Middleware: JSP

Features of Servlet-Side Programming (JSP)

Java Server Pages (JSP) technology enables you to mix regular, static HTML with dynamically generated content from servlets. Many Web pages that are built by CGI programs are primarily static, with the parts that change limited to a few small locations.

MySQL: MySQL is an open-source database system with which we can do the following things: design the structure of the tables (called schema) and how they relate to one another

- add, edit and delete data
- sort and manipulate data
- produce listings based on queries

To interact with MySQL, we enter commands on a command line. These

The Seybold Report
ISSN 1533-9211
DOI 10.5281/zenodo.6686469

Commands, such as CREATE, INSERT, UPDATE, etc. are based on a

More general language called SQL (Structured Query Language).

Server Specification: The following requirements apply to the server system environment:

- Microsoft Windows XP operating system supported by MS_access
- A minimum of 512 MB RAM.
- A backup system with larger capacity (recommended).

Client Specification: The following requirements apply to the client system environment:

- Microsoft Windows XP
- 256 MB RAM

Apache Tomcat: Apache Tomcat is a server container developed by apache software foundation. Apache tomcat 6.0 implements servlet 2.5 and JSP 2.1 specification for unified expression language. Apache tomcat includes tools for configuration and management.

TESTING:

- Validation Testing
- Black Box Testing
- Unit testing
- User acceptance Testing

7.1 Hardware specification

Processor: Intel Pentium IV

Clock speed: 1.8 GHz

RAM: 256 MB

HDD: 80 GB

FDD: 1.44 MB

CD Drive: 52x Reader

Pointing device: Scroll Mouse

Keyboard: 101 Standard Key-board

7.2 Software specification

System Architecture: Java

Core Language: Jsp, Servlet

Operating System: Windows Xp

ISSN 1533-9211

DOI 10.5281/zenodo.6686469

Database: My Sql

Server: Apache Tomcat

8. CONCLUSION

The project presented a new, safe and optimized architecture for enhancing the efficiency of the download of mobile device computation into the cloud. Dynamically at work, this decision is taken based on 4 constraints: memory use, CPU use, energy consumption and runtime. The framework also adds a new safety layer which uses the AES technology to protect data from methods before the download case is transferred to the cloud. The analysis results demonstrated that the proposed architecture would boost mobile applications by reducing mobile resource consumption, such as processing time, battery use, CPU use and memory use. Finally, we conclude that the use of the proposed Framework remotely conserves mobile resources, particularly if the application requires high calculation and few transmission data.

8.1 Future work

They presented a prototype of the secure data processing model for mobile cloud computing. In the future, will focus on the follow research:

- To investigate more application scenarios that require data sharing between cloud private domain.
- To investigate the robustness of the Tri-rooted ESSI solution and security monitoring, auditing, and misuse detection in the mobile cloud system

9. REFERENCES

1. M. Alizadeh, S. Abolfazli, M. Zamani, S. Baharun, K. Sakurai, Authentication in Mobile Cloud Computing: A Survey, Journal of Network and Computer Applications, Vol. 61, pp. 59-80, February,2016.
2. N. Aminzadeh, Z. Sanaei, S. H. A. Hamid, Mobile Storage Augmentation in Mobile Cloud Computing: Taxonomy, Approaches, and Open Issues, Simulation Modelling Practice and Theory, Vol. 50, pp. 96-108, January, 2015.
3. G. Ateniese, K. Fu, M. Green, and S. Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. ACM Trans. Inf. Syst. Secur., 9(1):1–30, 2006.
4. G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Tune. Provable data possession at untrusted stores. In Proceedings of the 14th ACM conference on Computer and communications security, pages 598–609. ACM, 2007.
5. J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attributebased encryption. In SP ’07: Proceedings of the 2007 IEEE Symposium on Security and Privacy, pages 321–334, Washington, DC, USA, 2007. IEEE Computer Society.
6. D. Boneh, X. Boyen, and E.J. Goh. Hierarchical identity based encryption with constant size ciphertext. Advances in Cryptology– EUROCRYPT 2005, pages 440–456, 2005.

The Seybold Report
ISSN 1533-9211
DOI 10.5281/zenodo.6686469

7. D. Boneh, C. Gentry, and B. Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In Advances in Cryptology–CRYPTO 2005, pages 258–275. Springer, 2005.
8. D. Boneh and B. Waters. Conjunctive, subset, and range queries on encrypted data. pages 535–554. Springer, 2007.
9. I. Chang, R. Engel, D. Kandlur, D. Pendarakis, D. Saha, I.B.M.T.J.W.R. Center, and Y. Heights. Key management for secure lnternet multicast using Boolean function minimization techniques. INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, 2, 1999.
10. L. Cheung, J. Cooley, R. Khazan, and C. Newport. Collusion- Resistant Group Key Management Using Attribute-Based Encryption. Technical report, Cryptology ePrint Archive Report 2007/161, 2007. http://eprint. iacr. org.
11. L. Cheung, J. Cooley, R. Khazan, and C. Newport. Collusion- Resistant Group Key Management Using Attribute-Based Encryption. Technical report, Cryptology ePrint Archive Report 2007/161, 2007. http://eprint. iacr. org.
12. C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia. Dynamic provable data possession. In Proceedings of the 16th ACM conference on Computer and communications security, pages 213–222. ACM, 2009.
13. Fiat and M. Naor. Broadcast Encryption, Advances in CryptologyCrypto93. Lecture Notes in Computer Science, 773:480–491, 1994.
14. A. Haque, S. Islam, M. J. Islam, J.-C. Grégoire, An Architecture for Client Virtualization: A Case Study, Computer Networks, Vol. 100, pp.75-89, May, 2016.
15. D. Huang, P. Wang, D. Niyato, A Dynamic Offloading Algorithm for Mobile Computing, IEEE Transactions on Wireless Communications, Vol. 11, No. 6, pp. 1991-1995, June, 2012.
16. Kosta, A.Aucinas, P. Hui, and R. Mortier, "Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading",IEEE INFOCOM, pp. 945–953, 2012.
17. Motta, N. Sfondrini, and D. Sacco, "Cloud computing: An architectural and technological overview", International Joint Conference on Service Sciences, vol. 3, pp. 23–27, 2012.
18. M. Shiraz, A. Gani, A. Shamim, S. Khan, R. W. Ahmad, Energy Efficient Computational Offloading Framework for Mobile Cloud Computing, Journal of Grid Computing, Vol. 13, No. 1, pp. 1-18, March, 2015.
19. Shiraz, A.Gani, R. H. Khokhar, and R.Buyya, "A review on distributed application processing frameworks in smart mobile devices for mobile cloud computing," IEEE Communications Surveys & Tutorials, vol. 15, no. 3, pp. 1294–1313, 2013
20. A. ur R. Khan, M. Othman, A. N. Khan, J. Shuja, S. Mustafa, Computation Offloading Cost Estimation in Mobile Cloud Application Models, Wireless Personal Communications, Vol. 97,54 No. 3, pp. 4897-4920, December, 2017.
21. Vallina-Rodriguez and J. Crowcroft, "Energy management techniques in modern mobile handsets," IEEE Communications Surveys & Tutorials, vol. 15, no. 1, pp. 179–198, 2013.
22. 22.J. Wei, X. Hu, W. Liu, An Improved Authentication Scheme for Telecare Medicine Information Systems, Journal of Medical Systems, Vol. 36, No. 6, pp. 3597-3604, December, 2012.
23. J. Zhang, Z. Zhang, H. Guo, Towards Secure Data Distribution Systems in Mobile Cloud Computing, IEEE Transactions on Mobile Computing, Vol. 16, No. 11, pp. 3222-3235, November,2017.