iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 5 **Issue:** XII **Month of publication:** December 2017

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor :6.887
Volume 5 Issue XII December 2017- Available at www.ijraset.com

Data Stored in Different Locations by using Manual Stream of Cloud Computing for Security Method

Nagasundaram S[1]
[1]*Sakthi Mariamman Engineering College, Thandalam, Chennai - 602 105.*

I. INTRODUCTION

Cloud Computing is vast and tremendous development and the security levels also enhanced. The security methodologies are different levels and types. We can use some mathematical algorithms, cryptography method and encryption methods. I have implemented one of the algorithms for storing the data in different location by the users. Data storage are different levels, that can manually and automatically. The manual storage are one of the safest storage method. The third person or hackers are not know that the location whenever we can frequently changing the location.

Keywords: Cloud Computing, Security, Storage, Location and types. In the earlier sections, we discussed the common techniques of how application developers check for a rooted device and then how an attacker can bypass some of the techniques used by the developers. In this section, we will discuss different methods being used by Android developers to store data locally, and then we will see how secure these methods are.

II. BACKGROUND

Device loss is a very common problem with mobile devices. An attacker who has physical access to the device can have access to personal as well as corporate data stored on the device. The situation could be worse if the device is rooted. Keeping this in mind, if the data storage mechanisms being used by an application are not implemented properly, it may lead to serious attacks.

A. Internal Storage

Internal storage is another way in which we can save files directly on the device. By default, files saved to internal storage are private to user application, and other applications cannot access them. When the user uninstalls a user application, these files are remove.

B. External Storage

External storage is a location that a user can use to save files. This can be a removable storage media (such as an external SD card) or an internal (non-removable) storage. External SD cards are world readable.

In the next section, let us see how developers can use Shared Preferences to store data on a device, and then we will see how an attacker can access this data from a device as well.

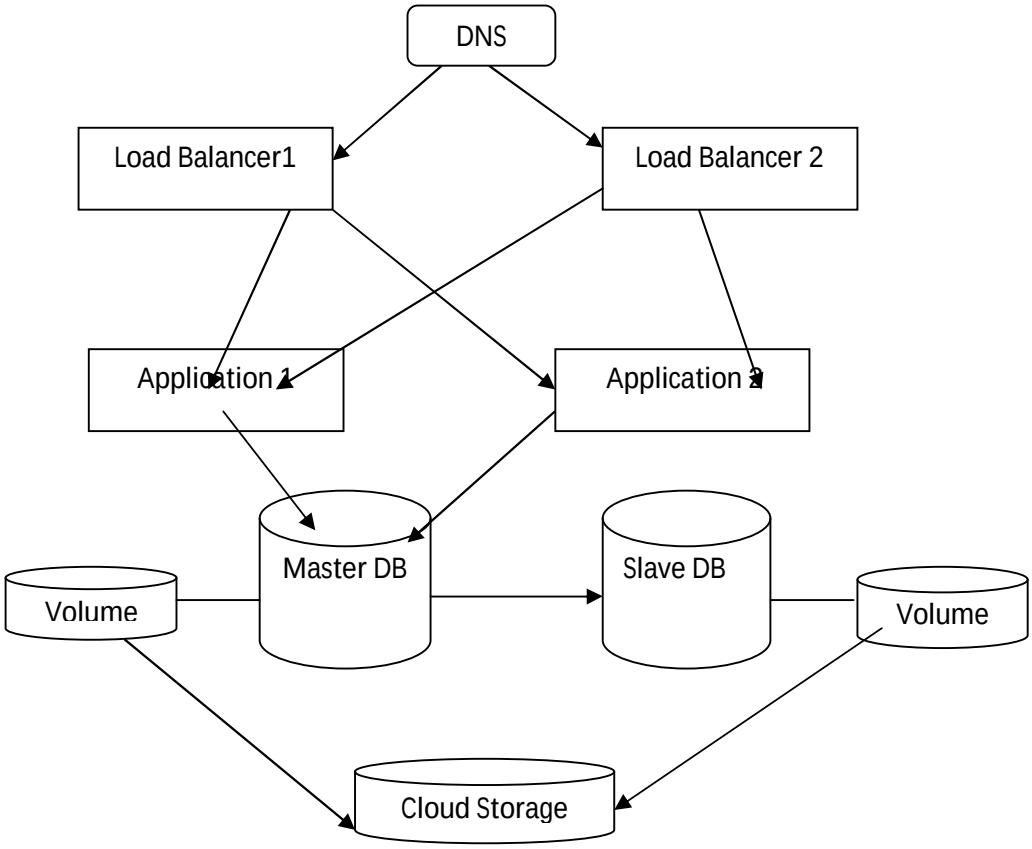


Figure.3.1.Cloud computing storage method architecture

Usually data are stored in cloud computing randomly and we have to find out where it's stored to display the location of the file systems. That is the usual procedure of storing the data value. We implement one of the concepts for the user to store some data in different locations and different paths at frequent intervals, because the third person or hacker has not identified exact location of data. This is the one of the reasons to protect our data from all unauthorized person or theft.
Data is stored as indicated. The cloud has a huge memory to store the data in different places and locations for any interval of time.

Algorithm_ DATA_STORE ( )
{
C. Initialization
CS - Cloud Server
CAM - Cloud Authentication Manager
DS - Data Storage
U_i - i^{th} *User where i= 1 to n*
for i = 1 to n
LC– location (cloud);
// User do store data
LS CS -> (Lction Str, Cld servr)
end i

The above query partitions the memory space into n^2 rectangles by drawing a horizontal and a vertical line through each of the n points. The set of dominated points is identical for each point in any rectangle, so the dominance count of the lower left-hand corner of each rectangle can be pre-computed, stored and reported for any query point within it. Range queries reduce to binary search and thus take O(lg n) time. Unfortunately, this data structure takes quadratic space. However, the same idea can be adapted to kd-trees to create more space-efficient search structure.

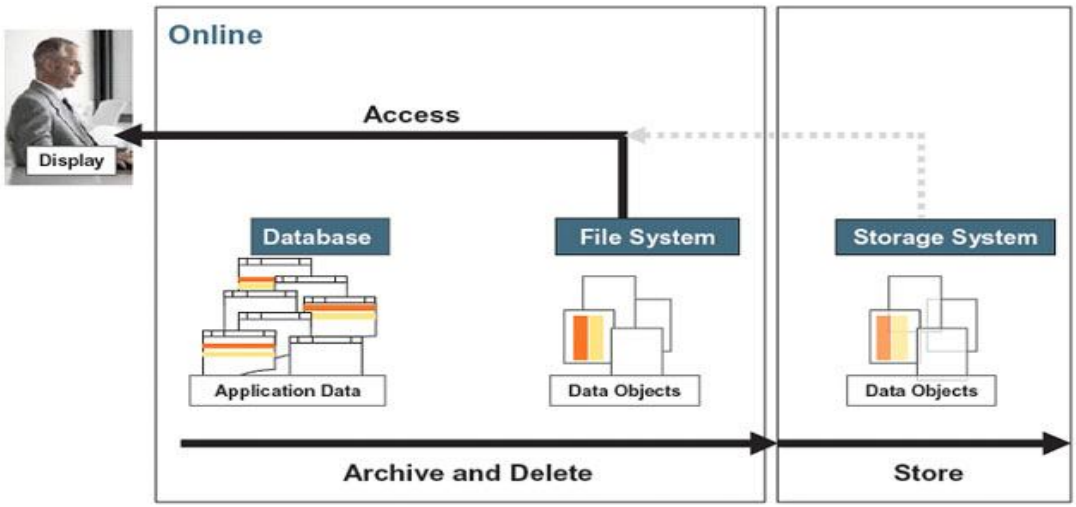


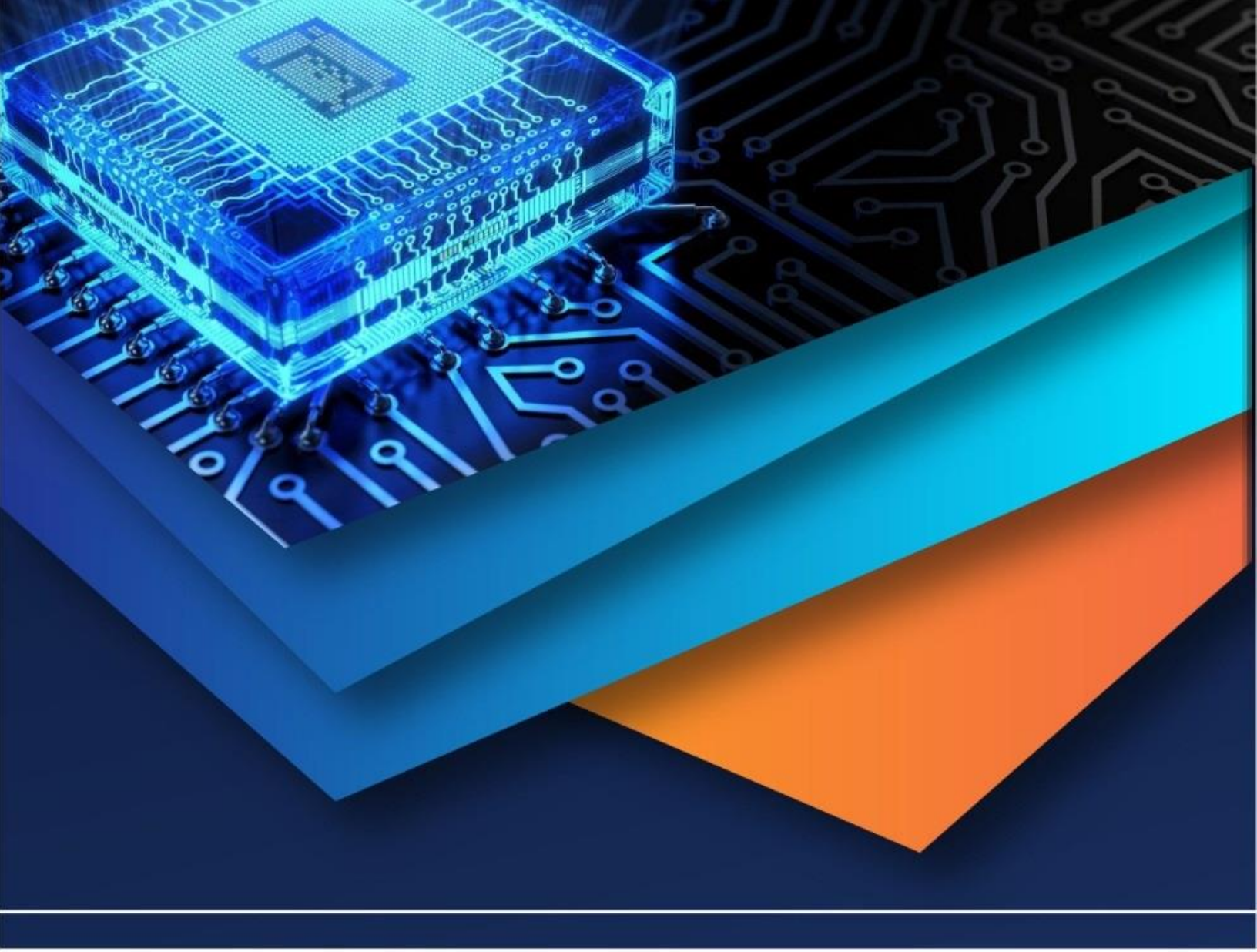
Figure. Data Archiving system

D. Implementations

Both CGAL and LEDA use a dynamic Delaunay triangulation data structure to support circular, triangular and orthogonal range queries. Both libraries also provide implementation of range tree data structures, which supports orthogonal range queries in O(k+lg 2n) time where n is the number of the subdivision and k is the number of points in the rectangular region.

New data is generated and collected faster than ever. The move towards everything digital, and the deployment of Internet of things platforms for consumers and businesses, means that data that has traditionally not been collected is now finding its way into huge data lakes and big data initiatives. This is forcing IT organizations to adapt to, and adopt, new data management strategies that are more economically practical. This includes moving from antiquated (very old) legacy systems to next-gen solutions.

In parallel, cyber security regulations are getting tighter, putting the spotlight on data privacy and forcing organizations to gain visibility into the sensitive data they store and who really has access to it. This, combined with the general perception that perimeter defenses are no longer sufficient, pushes security teams towards controls that can closely monitor the crown jewels of and business – its data.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY