A Data Sharing Protocol to Minimize Security &Privacy Risk of Cloud Storage in Big Data Era

E.BOOMIKA

M.C.A, C.RAMYA, M.C.A, M.E., ASSOCIATED PROFESSOR,
A.R.J COLLEGE OF ENGINEERING AND TECHNOLOGY, MANNARGUDI.

Abstract: The maturity of cloud computing technology in terms of reliability and efficiency, a large number of services have migrated to the cloud platform. To convenient access to the services and protect the privacy of communication in the public network, three-factor Mutual Authentication and Key Agreement (MAKA) protocols for multi-server architectures gain wide attention. Most of the existing three-factor MAKA protocols don't provide a formal security proof resulting in various attacks on the related protocols, or they have high computation and communication costs. And most of the three-factor MAKA protocols haven't a dynamic revocation mechanism, which leads to malicious users can not be promptly revoked. To address these drawbacks propose a provable dynamic revocable three-factor MAKA protocol that achieves the user dynamic management using Schnor signatures and provides a formal security proof in the random oracle. Security analysis shows that our protocol can meet various demands in the multi-server environments. Performance analysis demonstrates that the proposed scheme is well suited for computing resource constrained smart devices. The full version of the simulation implementation proves the feasibility of the protocol.

Keywords: cloud computing, Protocol, Architecture, Network.

I.INTRODUCTION

Cloud computing technology has been completely commercialized. It can not only improve service efficiency but also reduce costs. More and more companies are putting their services on the cloud platform for development, management and maintenance. This not only reduces the local maintenance burden for these enterprises, but also provides unified security and operation management for all services on the third party cloud platform, as Although third-party cloud platforms have more powerful technologies and more standard technical specifications to ensure that the servers run in a relatively secure environment, users and servers communicate in the public network. Therefore, authentication and key agreement are critical for the communication security. The use of mutual authentication and key agreement (MAKA) protocols not only prevent attackers from abusing server resources, but also prevent malicious attackers posing as the Server to obtain the user's information. Therefore, the MAKA protocols have been extensively studied since port proposed a password-based authentication protocol Earlier MAKA protocols are designed for single-server architecture. As Internet users grow exponentially, the number of cloud servers rendering different services has also grown significantly. For the single-server architecture, it is difficult for users to maintain a variety of passwords for each server. To improve user experience, many scholars propose more flexible MAKA protocols for multi-server environments. Combined with the unified management features of the cloud platform, such protocols can be conveniently applied. On the other hand, users usually utilize simple letters or numbers as their passwords, and even a large number of users directly use the default password if the smart devices don't require the user to modify the password mandatory. This module is designed for authorized users to view the files. The users who are registered are the authorized users. If the authorized users enter into this software, the "Valid User" alert will be displayed and they can view files. If the unauthorized user enters into this module, the "Invalid User" alert will be raised

II. PROGRAM DESIGN LANGUAGE

ASP.NET is more than the next version of Active Server Pages (ASP); it is a unified Web development platform that provides the services necessary for developers to build enterprise-class Web applications. While ASP.NET is largely syntax compatible with ASP, it also provides a new programming model and infrastructure for more secure, scalable, and stable applications. You can feel free to augment your existing ASP applications by incrementally adding ASP.NET functionality to them. ASP.NET is a compiled, NET-based environment; you can author applications in any .NET compatible language, including Visual Basic .NET, C#, and JScript .NET. Additionally, the entire .NET Framework is available to any ASP.NET application. Developers can easily access the benefits of these technologies, which include the managed common language runtime environment, type safety, inheritance, and so on. Developers can choose from the following two features when creating an ASP.NET application, Web Forms and Web services, or combine these in any way they see fit. Each is supported by the same infrastructure that allows you to use authentication schemes, cache frequently used data, or customize your application's configuration, to name only a few possibilities. Each of these models can take full advantage of all ASP.NET features, as well as the power of the .NET Framework and .NET Framework common language runtime. These features and how you can use them are outlined as follows: Accessing databases from ASP.NET applications is an often-used technique for displaying data to Web site visitors. ASP.NET makes it easier than ever to access databases for this purpose. It also allows you to manage the database from your code. For more information, see Accessing Data with ASP.NET. ASP.NET, in turn, is built on the .NET Framework, so the entire framework is available to any ASP.NET application. Your applications can be authored in any language compatible with the common language runtime, including Microsoft Visual Basic, Visual C#, and Java Script .NET. The following sections provide an overview of the features offered by ASP.NET.

III.SYSTEM DESIGN

The Structured Query Language (SQL) is the language of databases. All modern relational databases, including Access, FileMaker Pro, Microsoft SQL Server and Oracle use SQL as their basic building block. In fact, it's often the only way that you can truly interact with the database itself. At this point, you might be thinking that you're not a programmer and learning a programming language is certainly not up your alley. Fortunately, at its core, SQL is a simple language. It has a limited number of commands and those commands are very readable and are almost structured like English sentences. Before we get started, it's important that you have a basic understanding of how databases work. If you're comfortable with terms like table, relation and query, feel free to plow right ahead! If not, you may wish to read the article before moving on. Let's look at an example. If you were considering removing items from your store that were priced over $5, you might want to retrieve this information from your database. There's one simple lesson you should take away from our discussion at this point: SQL is like English. Don't worry about how you construct SQL statements; we'll get to that in the rest of our series. Just realize that SQL isn't as intimidating as it may first appear. We'll be discussing a number of topics in the coming weeks. Be sure to check this page as we add new articles. I'll replace the items in the list below with links to the relevant articl

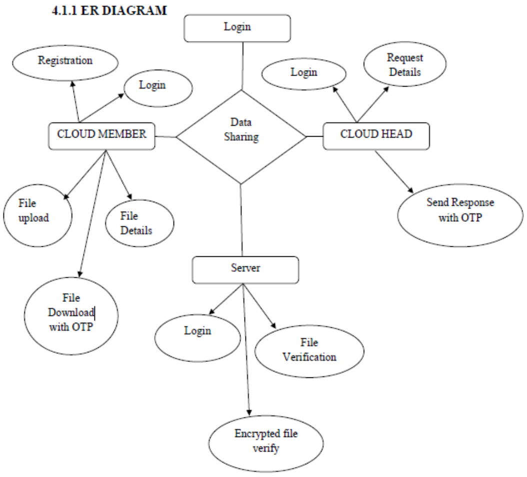


Figure 3.1 Data model

Applications Manager DB2 Server Monitoring capability helps database administrators (DBAs) monitor the availability and performance of production databases. It is an agentless database monitoring software that provides out-of-the-box performance metrics for ensuring the IBM DB2 database server runs efficiently. The database monitoring tool provides a web client that helps you to visualize the network of DB2 Servers. It provides in-depth monitoring data that helps you make educated decisions about usage patterns, plan capacity and alert you of impending problems.

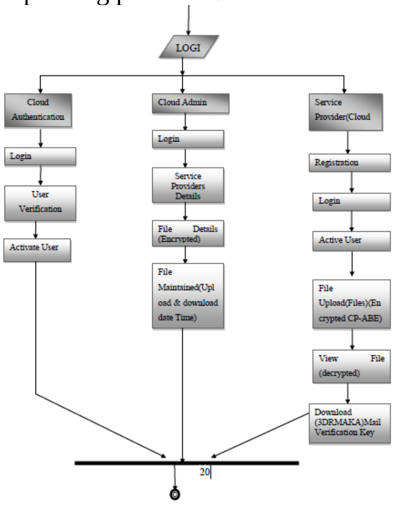
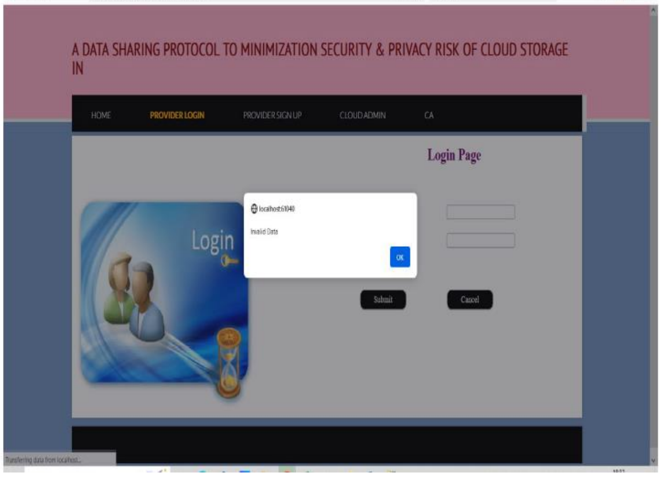


Figure 3.2 Activity Diagram

Out-of-the-box monitoring of MS SQL availability and performance. Monitors performance statistics such as memory usage, buffer manager statistics, connection statistics, cache details, sql statistics, etc., Alerts can be configured for these parameters. Based on the thresholds configured, notifications and alerts are generated if the SQL Server or any specified database within the server is not accessible. Actions are executed automatically based on configurations. Performance graphs and reports are available instantly. Reports can be grouped and displayed based on availability, health, and connection time.

IV.TESTING AND VALIDATION

System testing involves user training system testing and successful running of the developed proposed system. The user tests the developed system and changes are made according to their needs. The testing phase involves the testing of developed system using various kinds of data.An elaborate testing of data is prepared and the system is tested using the test data. While testing, errors are noted and the corrections are made. The corrections are also noted for the future use. The users are trained to operate the developed system. System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points. Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.


A DATA SHARING PROTOCOL TO MINIMIZATION SECURITY & PRIVACY RISK OF CLOUD STORAGE IN
HOME
PROVIDER LOGIN
PROVIDER SIGN UP
CLOUD ADMIN
CA
Login Page
Login
OK
Submit
Cancel

localhost:61040

Invalid Data

Transferring data from localhost...

Figure 4.1 Sample of test case without error

During the deployment design phase of the solution life cycle, you design a high-level deployment architecture and a low-level implementation specification, and prepare a series of plans and specifications necessary to implement the solution. Project approval occurs in the deployment design phase.The whole process has been designed for the user side to enable the standard level of security to their important information and data that has been stored into the cloud. It develops the administrators' performance evaluation in a better way.

Text condition	Text table	Actual results	Expected result	Final result
Username	user	Refer the user	System accepts, refer admin login	PASS
password	456123789	Refer the user	System accepts the data	PASS

Figure 4.2 Testing status

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

V.SYSTEM ARCHITECTURE

This module is designed for authorized users to view the files. The users who are registered are the authorized users. If the authorized users enter into this software, the “Valid User” alert will be displayed and they can view files. If the unauthorized user enters into this module, the “Invalid User” alert will be raised. This module is used to help the server to view details and upload files with the security. Admin uses the log key to the login time. Before the admin logout, change the log key. The admin can change the password after the login and view the user downloading details and the counting of file request details on flowchart. The admin can upload the file after the conversion of the encrypted file format.

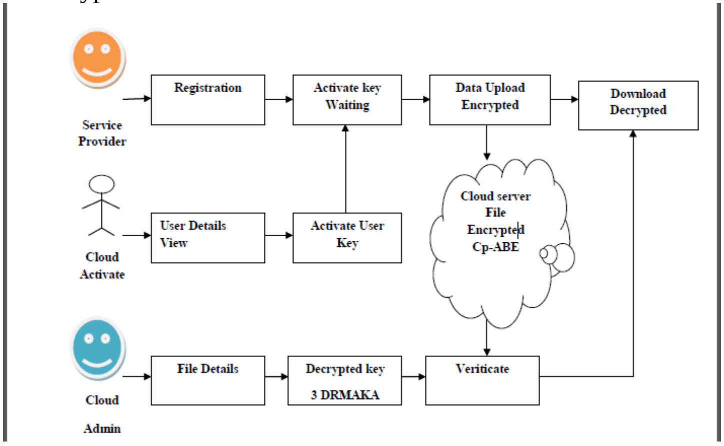


Figure 5.1 System Architecture

If Authenticated user name and authenticated key is given the file can be downloaded by the user. The upload the file and download the file in cloud system at generated the key. The verified and key can be provided the cloud system. To convenient access to the services and protect the privacy of communication in the public network, three-factor Mutual Authentication and Key Agreement (MAKA) protocols for multi-server architectures gain wide attention. This module is used to help the client to search the file using the multiple key words concept and get the accurate result list based on the user query. The user is going to select the required file and register the user details and get activation code in mail from the email before enter the activation code

VI.CONCLUSION

A novel collaborative key management protocol is presented. With help of that secure key management is guaranteed which is easier to deploy and compare with previous multi-authority schemes. Initially, the file is encrypted using key parameter. It is combination of both cloud server and key authority's public keys. Key generation is done by 3DRMAKA algorithm & provides the security while transmission of both the data and key. In this project presented an efficient 3DRMAKA based mutual authentication, unlinkablity and one-time password with a zero-knowledge scheme for a cloud environment. Proposed scheme assumes a new setting where users keep their passwords far away from the service provider in the cloud. This feature has been gained a good chance to service provider to increase time processing. Furthermore, our proposed scheme resists insider attacks, replay attacks and parallel session attacks. Also, our work has many virtues, including freely chosen password, user anonymity, mutual authentication, session key agreement and does not require the synchronized file. In performance evaluation, our scheme has been proven to obtain strong security.

VI.REFERENCES

1. Design of Mutually Authenticated Key Agreement Protocol Resistant to Impersonation Attacks for Multi-Server Environment[AlavalapatiGoutham Reddy, Eun-Jun Yoon-2016]
2. Robust Biometrics-Based Authentication Scheme for Multiserver Environment [Debiao He, Member, IEEE, and Ding Wang -2014]
3. Anonymous Authentication for Wireless Body Area Networks With Provable ecurity[Debiao He, SheraliZeadally, Neeraj Kumar, Member, IEEE, and Jong-Hyouk Lee, Senior Member, IEEE-2017]
4. Anonymous Authentication for Wireless Body Area Networks With Provable ecurity[Debiao He, SheraliZeadally, Neeraj Kumar, Member, IEEE, and Jong-Hyouk Lee, Senior Member, IEEE-2017]
5. A Generic Framework for Three-Factor Authentication: Preserving Security and Privacy in Distributed Systems [Xinyi Huang, Yang Xiang, Member, IEEE, AshleyChonka -2011]
6. SikharPatranabis, YashShrivastava, "Provably Secure Key-Aggregate Cryptosystems with Broadcast Aggregate Keys for Online Data Sharing on the Cloud", Vol. 66, No. 5, 2017.
7. Yuqi Wang, and Kun She, "A Practical Quantum Public-key Encryption Model", Third International Conference on Information Management(ICIM), 2017.