IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 4, April 2016

Storing & Retrieval of Encrypted Binary Data to &From Cloud Storage with Android Application

Gunjan Pathekar[1], Dr. Kalyani.A.Akant[2]

PG Student, Department of Computer Science, G.H. Raisoni College of Engineering, Nagpur, India [1]

Professor, Department of Information Technology, G.H. Raisoni College of Engineering, Nagpur, India [2]

Abstract: This paper describes the implementation of a secure cloud storage application. Cloud service providers often have access to the files of their customers. To prevent this and other security risks, a novel variant of the RSA encryption algorithm is employed. The application is web-based and therefore platform-independent. It is possible to create, open and edit text files directly in the browser. The files are encrypted and decrypted only locally on the computer of the user. A special role in the implementation plays the user's Smartphone which is used as a security token. The project comprises an Android application that basically generates and holds the decryption key. To open and decrypt a file, the user has to scan a QR code which is shown on the computer screen.

Keywords: Cloud Platform, Android Application, Security, AES, RSA, Token Generation.

I. INTRODUCTION

Cloud services are used to save and stores the data in cloud. Server stores the data and it can be accessed through the internet. The data cannot be connected to a single device. When we use any Cloud storage, a file can be created on a PC and later edited on a laptop without transferring the file from the user's PC to the laptop manually. Therefore cloud service provider is being provided with the data. In some cases, the service provider can also potentially access the user's file. The cloud storage service Drop box 1, for example, does encrypt the data. But Drop box holds the decryption key and can access that data. By using such services, the provider has to be trusted. This is not necessary when using the approach of this paper. The user's files get encrypted before they are transferred to the server. For the encryption, AES algorithm and a variant of the RSA 2 algorithm is being used. For decrypting the data, the client has to scan a QR code with Smartphone. A feature of this Approach is that the private key of the user is actually located on the Smartphone. So even if an attacker places malware on the user's PC, he wouldn't be able to gain full access to all files of the user. Other cloud storage services that encrypt the files locally are accessible to such an attack if the decryption key is detectable on the user's machine.

In this Internet generation, everyone wants there data to be intact and secured. Many Free Cloud storage service providers are such as Microsoft Sky Drive, Google Drive, Rapid Share, Drop box, Amazon Cloud Drive etc. But the data in these services is mostly in the unencrypted format. Suppose even if the data is in encrypted format, the keys for encryption can be produced by the software of the organization and the organization can access user data if required. So the end user has to think that his data should be safeguarded. The user should firstly encrypt his data and secondly upload same copy of those files on the cloud like Google Drive.

This project provides an encryption of 256 bit AES for the .png image files.

II. RELATED WORK

In 2014, Hongwei Li, Dongxiao Liu, Yuanshun Dai, Tom H. Luan, And Xuemin developed the searchable encryption for multi-keyword ranked search over the storage data. Specially, by considering the large number of outsourced documents (data) in the cloud. They utilized the relevance score and k-nearest neighbor techniques to develop an efficient multi-keyword search scheme which can return the ranked search results based on the accuracy. Within this framework, they leverage an efficient index that further improves the search efficiency, and adopt the blind storage system to conceal access pattern of the user which searches it. Security analysis shows that our scheme can achieve con dentiality of documents and index, trapdoor confidentiality, trapdoor unlink ability, and concealing access pattern of the search user. Finally, using extensive simulations, they show that the proposal can achieve much improved efficiency in terms of search functionality and search time compared with the existing proposals. In 2015, C. Anusha, M. Srilakshmi, Dr. S. Prem Kumar proposed a new cloud storage encryption schema which allows cloud storage providers to protect user privacy. Since authorities cannot tell the obtained secrets are whether true or false, the cloud storage providers ensure that the user privacy is still securely provided. The given scheme believes cloud storage service providers or trusted third parties handling key management are trusted and cannot be hacked. Sometimes may intercept the communication between users and cloud storage providers and then constrain storage providers to discharge user secrets by using government power or other means. In this case the encrypted data which are affected to be known and storage providers are requested to release user secrets.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 4, April 2016

The proposed Deniable CP-ABE design is to build an Audit free cloud storage service. The deniability feature makes coercion invalid, and the ABE property assures secure cloud data sharing with a fine grained access controlled mechanism. In 2014, Baojiang Cui, Zheli Liu and Lingyu Wang authors proposed a scheme for much easy encryption, by introducing the novel approach of key aggregate searchable encryption (KASE) and establishes the concept through a concrete KASE design, in which a data user needs to allot a single key to a user for sharing a large number of documents, and the user only needs to acknowledge a single trapdoor to the cloud for querying the shared documents. The security analysis and performance evaluation both of them confirms that our proposed schemes are provably secure and practically efficient. In 2014,Manel Gherari, Abdelkrim Amirat, Mourad Oussalahand Ridda Laouar proposed Smart Mobile Cloud Architecture (SMCA). They consider this new architecture as referential that allows MCC users to have a full alertness of both contexts (Cloud and Mobile) at the same time. They introduced a new concept called Smart Cloud Gate (SCG), which directs to profile both mobile applications and the Cloud to extract knowledge that can be used as a criteria to select the convenient services, which will be advised to mobile applications and give each different application the appropriate view of the Cloud. In 2013 , Kan Yang **,** Xiaohua Jia, Kui Ren, Bo Zhang proposed DAC-MACS (Data Access Control for Multi-Authority Cloud Storage) which is an efficient and secure data access control scheme with efficient decryption. Specifically, they constructed a new multi-authority CP-ABE scheme with efficient decryption and also built an efficient attribute revocation method that can achieve both forward security and backward security. The reasoning and the simulation conclusion show that our DAC-MACS is highly effective and provably secure under the security model. In 2013Zhibin Zhou and Dijiang Huang, presented a holistic security framework to secure the data storage in public clouds with the appropriate spotlights on lightweight wireless gears store and retrieve data without disclosing the data matter to the cloud service providers.

To achieve this goal, the solution focuses on the following two research directions: First, they presented a novel Privacy Preserving Cipher Policy Attribute-Based Encryption (PP-CP-ABE) to protect users' data. Using PP-CP-ABE, light-weight devices can securely redistribute heavy encryption and decryption activities to cloud service providers, without acknowledging the data content and used security keys. Second, they proposed an Attribute Based Data Storage (ABDS) system as a cryptographic access control mechanism. ABDS accomplishes information theoretical optimality in terms of decrease computation, storage and communication overheads.

Especially, ABDS minimizes cloud services charges by lessen communication overhead for data managements. The performance assessments show the security strength and efficiency of the presented solution in terms of computation, communication, and storage.

III. PROPOSED METHODOLOGY

This demo comprises of an Android Application that is connected to the Google drive. To develop an Android Application which will enable to upload the Data Files on Cloud File Repository like Google Drive? The Android Application will use Unique Serial No. hardware of the device to For Encryption. The database will store the required information for Individual Encryption Salt and user details. Middle tier (application server) handles the data processing operations between Device and database server to provide the Encrypted File Upload. This proposed Application will Upload Files on Google Drive in encrypted format. Three-tier client-server architecture will be used. It enables an application from client machine to forward commands to database from the use of middleware service. The database processes command request and then sends a feedback to the middleware service which then sends the reply to client. It is efficient because middle tier (application server) of the architecture handles the data processing operations between client and database server.
To provide data security while uploading data in cloud i.e Google drive. The client uploads the data in the Google drive cloud with the help of key i.e IMEI NUMBER of his android mobile phone. The uploaded data will be in the encrypted form that will only be decrypted by the key that is only known to the client. Even the Google managers cannot see the original data, they can see in encrypted format only. If a hacker hacks a Gmail account he will only see the encrypted data and our personal data will be totally safe. In a normal case a hacker hacks a Gmail can also steal the important credential data from our Gmail's Google drive. So basically data is encrypted & safe.

There are basically 2 things A) We will have to create an Activity named ListFilesActivity.java to show our files that we had uploaded. It gives us an upload button for uploading files. The Main Android Methods that are needed to be called are:

1) OnCreate() Method:- Here 4 work is done such as:-
• It will connect the Layout with the Activity.
• It initializes the List.
• It initializes the Refresh Button.
• It initializes the Upload Button

AES ALGORITHM Encryption using 128 bit key size
1. Start
2. 128 bit plain text block is sent.
3. Add a round key, here 4bit word length (w0, w1, w2, w3) is added.
4. Round 1 starts, here 4bit word length (w4, w5, w6, w7) is added.
5. Round 2 starts, here 4bit word length (w8, w9, w10, w11) is added.
6. Similarly the process continues up to 10 rounds,
7. Round 10 starts, here 4bit word length (w40, w41, w42, w43) is added.
8. We get a 128bit cipher text block.
9. End

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 4, April 2016

AES ALGORITHM Decryption using 128 bit key size

1. Start
2. 128 bit Cipher text block is sent.
3. Add a round key, here 4bit word length (w40, w41, w42, w43) is added.
4. Round 1 starts, here 4bit word length (w39, w38, w37, w36) is added.
5. Round 2 starts, here 4bit word length (w35, w34, w33, w32) is added.
6. Similarly the process continues up to 10 rounds,
7. Round 10 starts, here 4bit word length (w0, w1, w2, w3) is added.
8. We get a 128bit plain text block.
9. End

IV. RESULTS

Figure 1 shows the list of the image files to be uploaded on the Google drive.

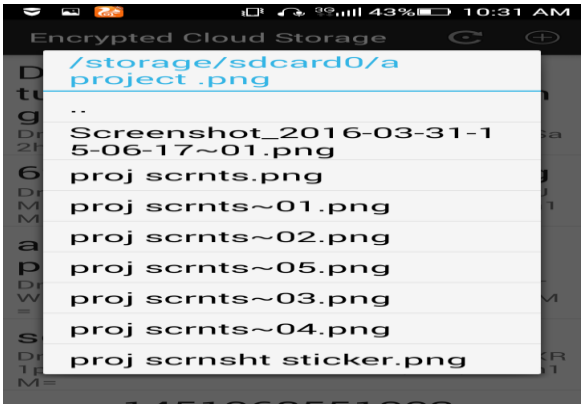

Fig 1: List of .png files from which a single file is to be encrypted by selecting it.

Figure 1 shows the view of Android App in which after clicking the upload button the user gets alist of .png format files, from which he selects a single file for encrypting.

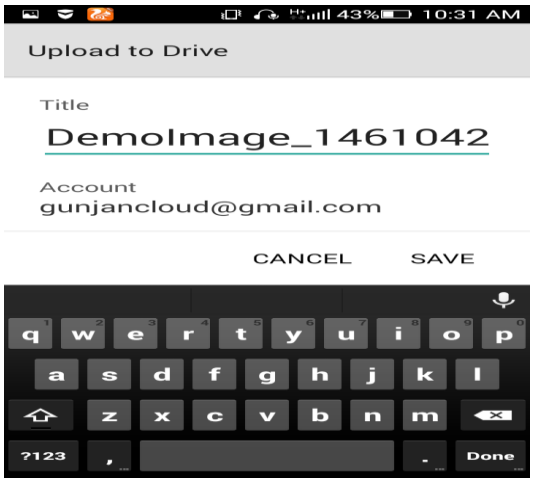

Fig 2: Selected .png File being uploaded to the drive for encryption.

The copy of the selected file gets uploaded on the Google Drive. The user has to rename the new image file and click on save.

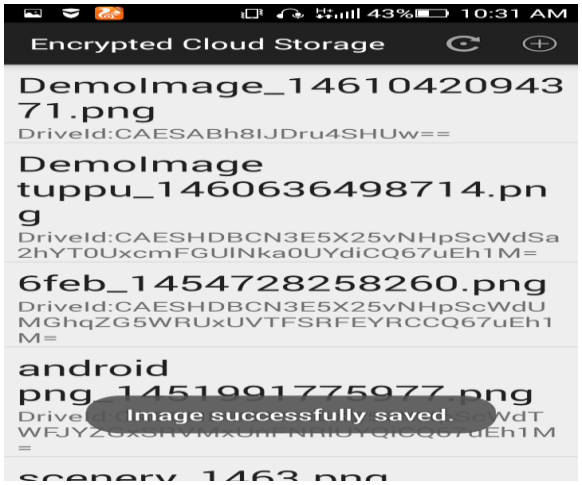

Fig 3: .png File Encrypted to the cloud & saved successfully.

The copy of our image file is saved in the cloud Google Drive in encrypted format.

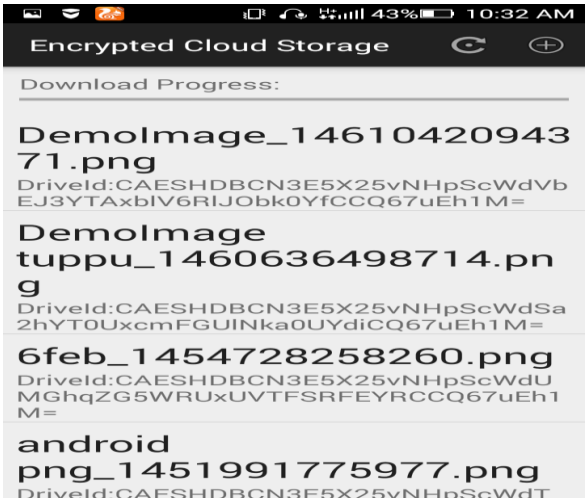

Fig 4: Download Progress of the encrypted image for decryption.

Figure 4 shows the download progress of our encrypted image file. Here the file is getting decrypted. Then we get the downloaded file in our android phone.

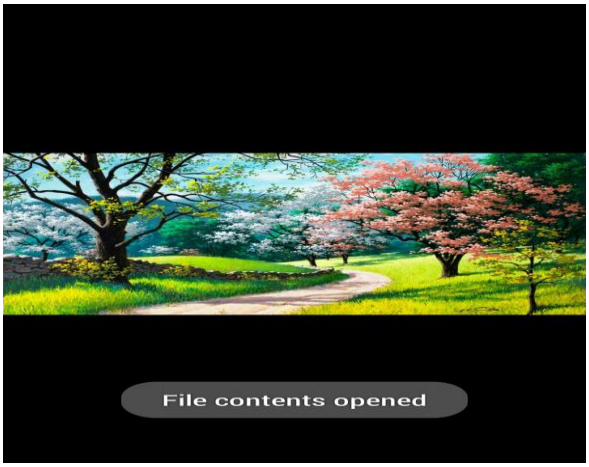

Fig 5: .png file decrypted & opened in our android phone

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 4, April 2016

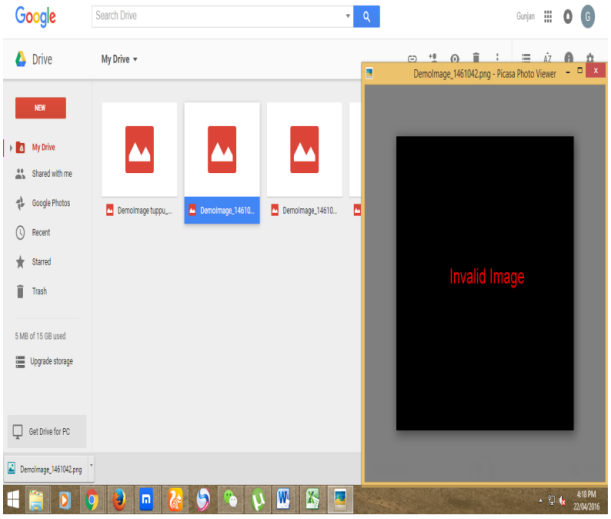

Fig 6: Unable to open the encrypted image in client's PC in his Google Drive

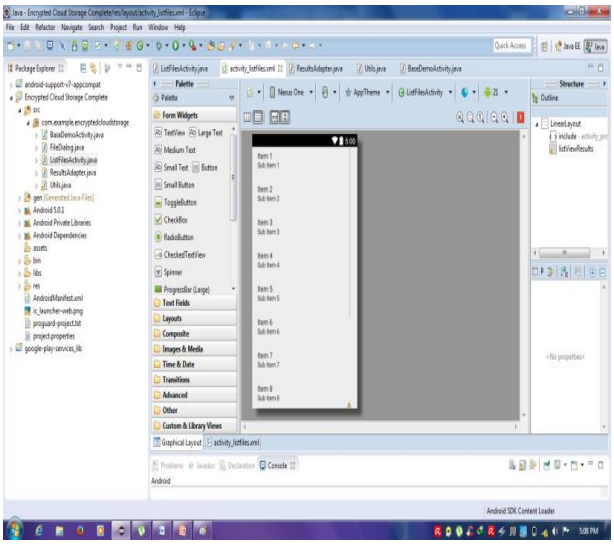

Fig7: Graphical layout of the android App.

The view and the layout are bind together with the activity.

V. CONCLUSION

A cloud storage web application which is based on AES algorithm and a new variant of the RSA encryption algorithm has successfully been implemented. The concept provides a high level of security; encryption and decryption are both performed on the client's side. The decryption key is neither stored on the user's machine, nor has the key to be typed in manually. Though, there are some limitations of this implementation, especially that only text files can be saved on the server. For a wide-spread use of the application, a user should be able to save all types of files. This could be realized by creating a standalone program instead of using the browser, since a JavaScript file executed in a browser is currently not able to save to the file system. This is necessary, because the data received by the server has to be processed on the client for decryption.

A constraint of the concept itself is that the Smartphone basically can't be used to retrieve files from the server. Another Smartphone would be needed to scan the QR code, which is of course not viable for practical use. Skipping the step of scanning the QR code and reducing the concept to only use the Smartphone and the server is no solution either. The security benefits of the scheme would be removed and a simpler encryption method could be used as well. AES, for example, has a much better performance in computation time than RSA [13]. To every file, a encrypted AES key could be stored on the server. Since only small text files are encrypted in the implementation of this thesis, this method wasn't necessary here. But for larger files, the RSA algorithm would probably be too slow. Despite of some constraints, the approach of this thesis provides security and also preserves the usability. Scanning a QR code is quickly done and user doesn't have to remember another password for the decryption process. Also, the fact that many users choose weak password, can't be exploited here.

REFERENCES

[1] Zhibin Zhou and Dijiang Huang, "Efficient and Secure Data Storage Operations for Mobile Cloud Computing", 8th International Conference on Network and Service Management (CNSM 2012).
[2] Kan Yang ,"DAC-MACS: Effective Data Access Control for Multiauthority Cloud Storage Systems", ieee transactions on information forensics and security, vol. 8, no. 11, november 2013.
[3] Taiwo Ayodele, Dennis Adeegbe, "Cloud Based Emails Boundaries and Vulnerabilities", Science and Information Conference 2013.
[4] Kan Yang,"DAC-MACS: Effective Data Access Control for Multiauthority Cloud Storage Systems", ieee transactions on information forensics and security, vol. 8, no. 11, November 2013.
[5] Taiwo Ayodele, Dennis Adeegbe, "Cloud Based Emails Boundaries and Vulnerabilities", Science and Information Conference 2013
[6] Barker, Barker, W., Burr, W., Polk, W., Smid, M. U.S. Department of Commerce, National Institute for Standards and Technology (NIST)(2007, March 8). Recommendation for Key Management – Part 1: General (NIST SP800-57.
[7] U.S. Department of Commerce, National Institute for Standards and Technology (NIST). (2001, May 25)Security Requirements for Cryptographic Modules (FIPS Publication Number 140-2).
[8] Blum, D. (2010, March). Using Encryption to Protect Sensitive Data in Cloud Computing Environments Retrieved from Gartner database.
[9] Barker, E.Burr, W., Dang, Q., Jones, A., Polk, T., Rose, S., Smid, M. U.S. Department of Commerce, National Institute for Standards and Technology (NIST)(2009, December). Recommendation for Key Management –Part 5: Application-Specific Key Management Guidance (NISTSP 800-57).
[10] Barker, Barker, W., Burr, W., Polk, W, Smid, M. U.S. Department of Commerce, National Institute for Standards and Technology (NIST)(2007, March 8). Recommendation for Key Management – Part 1: General (NIST SP800-57).

BIOGRAPHY

Gunjan Pathekar received the bachelor's degree in Information Technology engineering from the Nagpur University, Maharashtra, India in 2013. Presently she is pursuing her master's in mobile technology in the department of Computer Science from G. H. Raisoni College of Engineering, Nagpur Maharashtra, India. Her research interests include android platform, cryptography, cloud security etc.