IJARCCE

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

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 3, March 2017

Security in Searching Shared and Encrypted Data

Swati Virkar[1], Pritee Chinchwade[2], Shivani Ajmire[3], Prof. R.A. Badgujar[4]

Department of Computer Engineering, Jaywant Shikshan Prasarak Mandal, Bhivarabai Sawant Institute of Technology &Research, Pune, India[1,2,3,4]

Abstract: When private data content is stored in databases that are under the control of others, a formal way to protect the data, is to encrypt the data before storing in the database. To retrieve the data efficiently, a search mechanism is needed that works over the encrypted data an encryption scheme where each authorized user in the system has user own private keys to encrypt and decrypt data. The method supports keyword search which enables the server to return only the encrypted data that satisfies an encrypted query with decrypting itat client side. For strong authentication we are implementing the AES 128 with 16 bit algorithm is implemented. We can deploy this system on the cloud server.

Keywords: Encrypted Data, private data, retrieve, database.

I. INTRODUCTION

Cloud storage services, information outsourcing and sharing have become ever-present in our life. For example, a user can store data at Dropbox and share them with user friends, in the mean time she may also have access to user friends' data. Due to the private of personal data, there is an inherent need for a user to selectively share user data with different receivers. In this a user can set the access control policies and then trust on the cloud server to enforce them. Unfortunately, this approach is not realistic due to two causes. One is that the users have no means to prevent the server from accessing their data. The other is that, even if the server is busy, it may also be forced to share users' data with other parties. We formulate MPSE and its security properties and then to provide a scalable and secure construction. As our Projects main purpose is to provide high security to our clients we have implement the AES128 and also used SHA256 algorithm.Revocation is the concept that we have implemented in our project which hacks the access control from particular untrusted user and SMS API is the concept used for the OTP(one time password)which gives the opt through the message on mobile and also on Gmail. It is implements on the private cloud and public cloud.

II. ARCHITECTURE

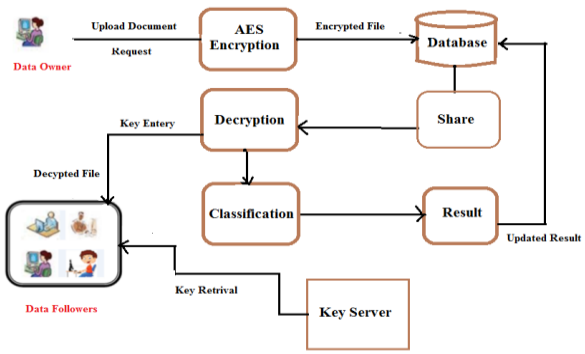


Fig.1 Architecture

Encryption:
The uploaded document can encrypt and stored at the storage server. Otherwise it is visible by all other users. This encryption can be done using the symmetric key algorithm AES.The encryption key can be generated from hash algorithm.

Searching:
Searchingcanbedonebyusingthresholdvalueofkeywordorkeywords. This checksum conversion based on CSCO algorithm. Every word in a document converts to checksum after uploading it to the server. Thenthis checksum-word list shared with users. The users who want to search send checksum of corresponding keyword. Then the document with highest rank i.e. with highest threshold value cansend back to the user. One of the im

Decryption: is also done using the inverse form of AES algorithm. Use cipher text and the public key as the inputs to AES but use the keys are in reverse order. That is, use K16 on the first iteration, K15 on the second until K1 which is used on the16th and last iteration. Decryption is done only when the private key is match with the public key.These keys distributed by the key server only if the corresponding user has the permission to access it.

III. ALGORITHM

1. AES 128

Step1 :Generate user m1 ID; Set of users m1s attribute > Domain B;
element checks in Domain manager;
Step2: Now, Generate Random Value[unique] attribute = ran;
Here we generate random value having unique attribute
Step3: Generate Random value [user] = r;
Here random value generation for particular user
Step4: Secret key= (ran +r).user m1s ID;
Secret key is the collection of two random values which are having unique attribute and unique user.

IJARCCE

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

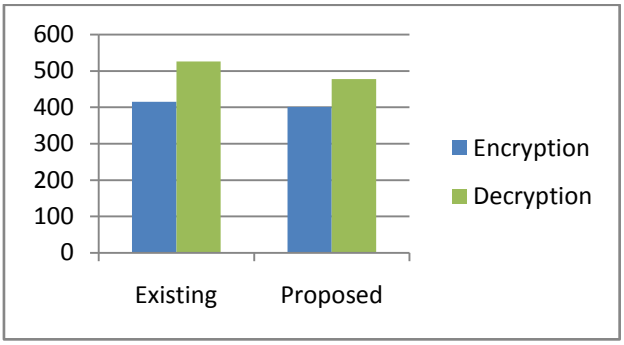
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 3, March 2017

Step5: Encrypting user data along with ID
Each user have its own ID for his encrypted data.
Step6: Encrypt (user ID. (rab+r)+data)
Encrypt that particular ID
Step7: Decrypting user data n long with
ID Decrypt the data with its ID
Step8: Decrypt (user ID. (ran+r)+data)
Decrypt the user ID with its data and random values.

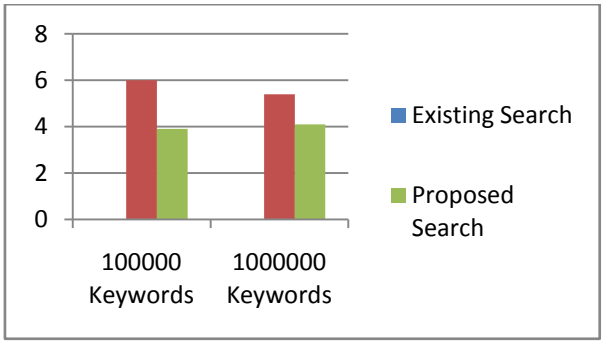
2. SHA256
Information: string required to ascertain the SHA score.
Yield: SHA score of string
Step 1: Padded with the length in such way that the outcome is numerous inleast 512 piece long.Pad the message in the usual way: suppose the length of the message M,in bits is lapped the 1 at the end of the message.
Step 2: Parse into 512 piece message squares M(1),M(2.. M(n) the messagesquare can prepare at one time.Parse the message intoN 512bit blocks M(1),M(2.. M(n).We use the big Endeansolution, so within each32bit word, the left most bit is stored in most significant bit position.utilizing the beginning hash values H(0).
Step 3: Then process the succession
H(i) = H(i1)+ CM(i) (H(i1));
Step 4: give back the H(i) SHA score of given string.

IV. RESULT ANALYSIS

Below graph shows the how much time required milliseconds for proposed as well as existing cryptography technique.



System Performance with Existing Approach



System searching Performance with Existing Approach

As Above graph shows measure the search times with various sized databases. Here used databases containing 100,000 keywords and 1,000,000 keywords. The results are shown in below figure. The new scheme requires less time for keyword searching than the old MPSE scheme time required in seconds.

V. CONCLUSION

The generated new primitive, namely AES encryption , for enabling users to selectively authorize each other to share in their encrypted data. In the formulationof system, we assume that authorization is granted on high level, namely for each ofuser which authorize to access ,search and revoke for the user data.

REFERENCES

[1] C. Bösch, Q. Tang, P. Hartel, and W. Jonker, "Selective document retrieval from encrypted database," in Proc. 15th Inf. Security Conf.(ISC), vol. 7483. 2012, pp. 224–241.
[2] Y.-C. Chang and M. Mitzenmacher, "Privacy preserving keyword searches on remote encrypted data," in Proc. 3rd Int. Conf.Appl.CryptographyNetw.Security, vol. 3531. 2005, pp. 442–455.
[3] M. Abdalla, E. Bresson, O. Chevassut, and D. Pointcheval, "Passwordbased group key exchange in a constant number of rounds," in Public Key Cryptography—PKC (Lecture Notes in Computer Science), vol. 3958, M. Yung, Y. Dodis, A. Kiayias, and T. Malkin, Eds. Berlin, Germany: Springer-Verlag, 2006, pp. 427–442.
[4] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in Advances in Cryptology—EUROCRYPT (Lecture Notes in Computer Science), vol. 3027,C. Cachin and J. Camenisch, Eds. Berlin, Germany: Springer-Verlag,2004, pp. 506–522.
[5] C. Dong, G. Russello, and N. Dulay, "Shared and searchable encrypted data for untrusted servers," in Proc. 22nd Annu.IFIP WG 11.3 Work.Conf.Data Appl. Security XXII, vol. 5094. 2008, pp. 127–143 4] F. Bao, R. H. Deng, X. Ding, and Y. Yang, "Private query on encrypted data in multi-user settings," in Proc. 4th Int. Conf. Inf. Security Pract.Experience, vol. 4991. 2008, pp. 71–85.
[6] Eu-Jin Goh eujin@cs.stanford.edu
[7] R. A. Popa and N. Zeldovich. (2013). Multi-Key Searchable Encryption.[Online]. Available: http://eprint.iacr.org/2013/508
[8] Q. Tang, "Search in encrypted data: Theoretical models and practical applications," in Theory and Practice of Cryptography Solutions for Secure Information Systems. Hershey, PA, USA: IGI, 2013, pp. 84–108.
[9] E. Bresson, O. Chevassut, and D. Pointcheval, "Dynamic group Diffie-Hellman key exchange under standard assumptions," in Advances in Cryptology—EUROCRYPT (Lecture Notes in Computer Science),vol. 2332, L. R. Knudsen, Ed. Berlin, Germany:Springer-Verlag,2002,pp. 321–336.