National Journal of Multidisciplinary Research and Development
ISSN: 2455-9040
Impact Factor: RJIF 5.22
www.nationaljournals.com
Volume 3; Issue 1; January 2018; Page No. 1137-1140

Impact of predicate based encryption on big data cloud controls

Ashish Yadav[1], Deepak Chaudhary[2], Dr. Rohit Singhal[3]
[1] M.Tech Scholar, Department of CSE, IET Alwar, Rajasthan, India
[2] Assistant Professor, Department of CSE, IET Alwar, Rajasthan, India
[3] Professor, Department of CSE, IET Alwar, Rajasthan, India

Abstract
As demand for large amount of data storage increased with variety and need for speedy access to data their comes in Big Data. Big data come in for data mining and to understand pattern and trend of data. It becomes little bit of impossible to store gigabytes of information so, organisations apprehend to store these large data on cloud, because of processing power and scalability in terms of pay as you use. Due to increasing demand of security for data and access Structures, data owner are not able to store this amount of data on their own without any security infrastructure. In this paper we propose a way of encryption for big data access control.

Keywords: predicate, big data, encryption

1. Introduction
With the rapid development of technologies and social network sites, data generation has increased in our day today life and this large amount of data is called as the big data. The increasing data pose many challenges from storage to computation, since traditional tools cannot manage this large amount of data. Cloud computing, due to its flexible, scalable and economic resources, is a natural fit for storing, processing and sharing the data.
In ABE, a user has a set of Predicate in addition to its unique ID. There are two classes of ABEs. In Key-policy ABE or KP-ABE (Goyal*et al.*), the sender has an access policy to encrypt data. A writer whose Predicate and keys have been revoked cannot write back stale information. The receiver receives Predicate and secret keys from the authority and is able to decrypt information if it has matching attributes. In Cipher text-policy, CP-ABE, the receiver has the access policy in the form of a tree, with Predicates as leaves and monotonic access structure.
Lewko and Waters proposed a fully decentralized ABE where users could have zero or more Predicate from each authority and did not require a trusted server. In all these cases, decryption at user's end is computation intensive. So, this technique might be inefficient when users access using their mobile devices. To get over this problem, Green *et al.* proposed to outsource the decryption task to a proxy server, so that the user can compute with minimum. However, the presence of one proxy and one key distribution Center makes it less robust than decentralized approaches. Both these approaches had no way to authenticate users, anonymously. Yang *et al.* presented a modification of, authenticate users, who want to remain anonymous while accessing the cloud. To ensure anonymous user authentication Attribute Based Signatures were introduced by Maji *et al.* This was also a centralized approach. A recent scheme by the same authors takes a decentralized approach and provides authentication without disclosing the identity of the users. However, as mentioned earlier in the previous section it is prone to replay attack.
In this paper, we focus on how to securely share data contents to a certain group of people during a particular time period in cloud-based systems, and propose a cryptographic approach

2. Proposed Work
The proposed model consists of following entities: cloud server, data owner, users, attribute authority for each user, attribute authorities bulletin board.

Cloud server
The server stores data for data owners and provides data access to users. The server is also responsible for storing attribute authority bulletin board where the update keys of the attribute are stored and then it is responsible for removing the old update keys which are based on the old time stamp.

Data owner
The data owner defines the access policies on attributes from different attribute authorities and encrypts data using the session key sunder these access policies before outsourcing data to cloud servers. The session keys are in turn encrypted using TDAAC algorithm. The access policies are embedded in the cipher text and access keys. Data owners are responsible for using the global identifier for each user. These data owners do not rely on the server to enforce the access policy.

User
Each user will be assigned with a global identity. Whenever request is made to access particular data with data owner the credentials are checked and global identity is given, using this identity user will be given a secret key to access particular data from the cloud server. However, the secret key of a user is insufficient to decrypt a cipher text encrypted under the

access policy at time ti even when the corresponding attributes satisfy the access policy. The user has to obtain a set of update keys at each time slot ti from the corresponding authorities who have published in attribute authority bulletin board which are stored in the cloud server.

Attribute and user revocation

To deal with the attribute revocation problem, we follow the ideas in identity-based encryption revocation [10] and divide the time into slots. At each time slot ti∈ T, for any attribute and user that belongs to attribute set, attribute authority generates the update keys according to access tree and update list so that only the users who possess attributes at time slot ti are able to obtain valid update key that belongs to attribute. Thus, by setting the update list and publishing the rresponding update key the authority can achieve dynamic change of attribute and user.

The main contributions of this paper are the following

1. Data owner upload the data on Cloud server depends on the server IP and the port that provides the back bone to the entire project
2. Distributed access control in such a way that the only authorized user can have access to the source of data.
3. Confidentiality in regards of the Identity of the user.
4. The architecture is decentralized.
5. The access control and authentication are both collusion resistant, meaning that no two users can collude and access data or authenticate themselves, if they are individually not authorized.
6. Implementation of SHA-3 to overcome drawbacks of SHA1 & SHA-3
7. The proposed scheme is resilient to replay attacks. A writer whose attributes and keys have been revoked cannot write back stale information.
8. The protocol supports multiple read and write on the data stored in the cloud.
9. Easy control in comparison of the previous projects with validators and alphanumeric security feature.
10. Data can be downloaded on if the Unique ID match with the file uploaded.
11. Email, blog and chat features to be provided such that ideas can be shared among the users (For their convenience).

On presenting her id the trustee gives her a token. A creator on presenting the token to one or more KDCs receives keys for encryption/decryption and signing. SKs are secret keys given for decryption, Kx are keys for signing. The message MSG is encrypted under the access policy X. The access policy decides who can access the data stored in the cloud. The creator decides on a claim policy Y, to prove her authenticity and signs the message under this claim. The cipher text C with signature is c, and is sent to the cloud. The cloud verifies the signature and stores the cipher text C. When a reader wants to read, the cloud sends C. If the user has Predicate matching with access policy, it can decrypt and get back original message. Write proceeds in the same way as file creation. By designating the verification process to the cloud, it relieves the individual users from time consuming verifications. When a reader wants to read some data stored in the cloud, it tries to decrypt it using the secret keys it receives from the KDCs. If it has enough Pre dicate matching with the access policy, then it decrypts the information stored in the cloud.

3. Secure Hash Algorithm

The basis of keccak is the 'state' which is a 5 x 5 array of words. A word is a 64-bit value. So, the representation would be 5 x 5 x 64 cube with each small section of the code to be bit. And the steps involved in the implementation of SHA 3 (Keccak Algorithm:-

i) θ (Theta Function)

we have taken 'sum' of one column and put it with the 'sum' of other column and then 'adding ' that into a bit. Well any addition in this is actually xor. Now the first part of the code example takes each sheet (which is an array size 5) and xor all of them together, creates an 'xor' lane that is put that into an array of size 5(called C). Now the next part of the code is to create a 'dual xor' lane. Now lets say you want to modify A [0] [x]. You can't C [4] ^ C [1] since you need the 2nd bit of C [4] xored with the 1st bit of C [1]. So a modified version of C [1] is created that is shifted one bit to the left. The now modified 2nd bit of C [1] is the 1st bit of the original C [1]. These 'dual xor' lanes are put into D(an array size 5) Then the last step in the code that that D[x] is xored with every element such that A[x] [0.4].

ii) ρ (Rho)

Note: Rho and Pi are done together in the python code but separated here.

In this step there is a set array called 'r' that rotates each lane in the state so that if the look at the left-most, top-most lane(which rotates one bit) in the picture the 1st bit becomes the 2nd bit and the last bit has become the first bit.

iii) π (Pi)

The reason was that I was thinking that x = y = 0 was in the bottom-left corner. It's not. It's the center element. So the element below the center element is B[0][4] and the one to the left of the center is B[4][0]. The math for this function is fairly easy to understand when you look at the pseudo-code

iv) χ (Chi)

This operation also a simple operations of bit-and/not/xor. Here is a representation. If A(B[0][0]) is the word you are trying to transform into Å(post Chi) then Y(B[1][0]) is the next element to A's left and Z(B[2][0]) then the equation is (A xor ((not Y) and Z)) = Å

The last step does not have a image to go with it but is based on a fixed 'Round Constant' matrix and mixed with certain values like the current round number and the size of the lanes

5. The you can return B which is the modified A for one round. Those are the basic round steps for SHA-3. Hopefully this helps someone understand a little more

4. Experimental Result

In this we are going to support the experiment proposed with the help of result oriented snapshots of the program and analysis.

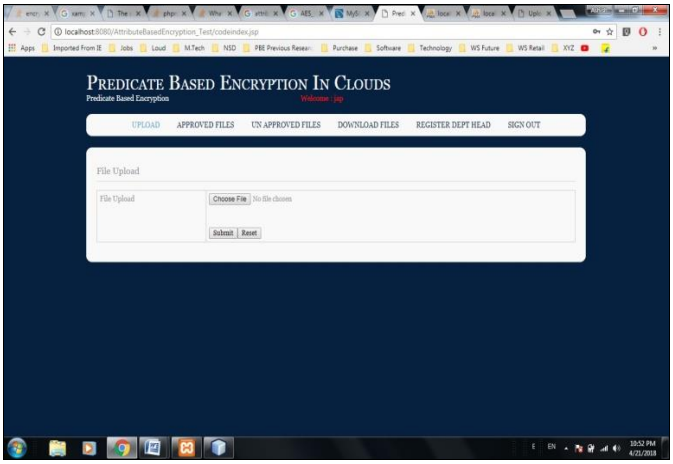

Fig 1: Upload file to be encrypted by proposed Algo.

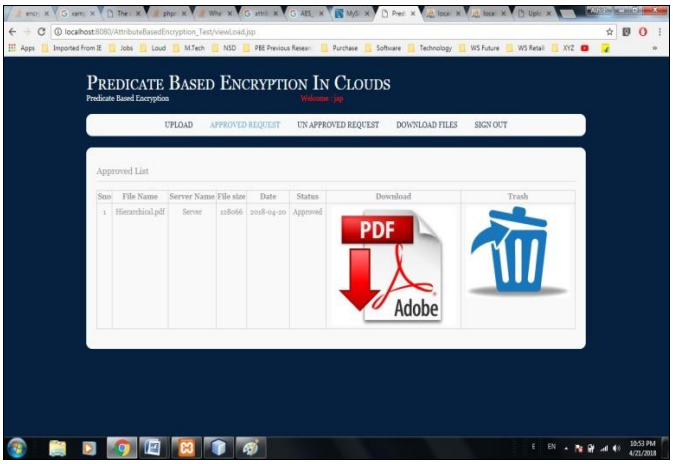

Fig 2: Approved section where files are approved by senior heads, and files are stored in encrypted form only.

Figure 1 & 2 are the part of user module that define the working of the project. In which file upload and store patterns are to defined that leads them to store in encrypted form directly to the cloud servers.

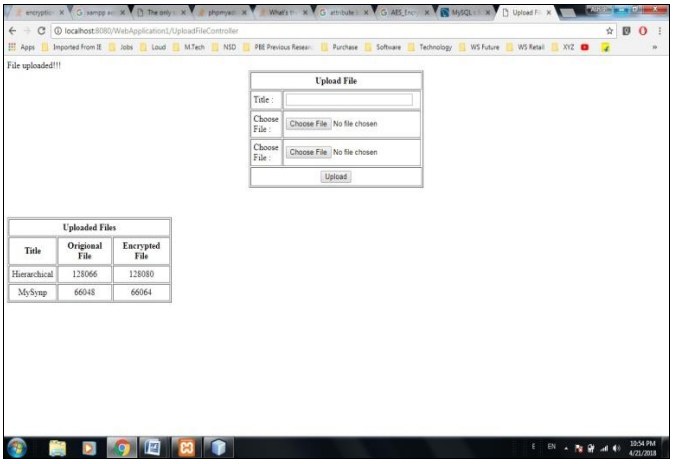

Fig 3: Below figure used to upload Original & Encrypted file for comparison.

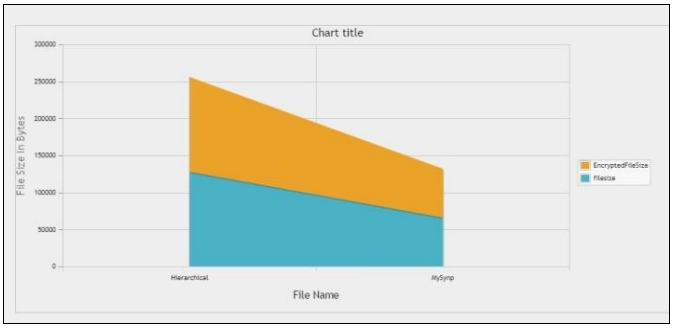

Fig 4: Graph Userid and corresponding DID,Fileid

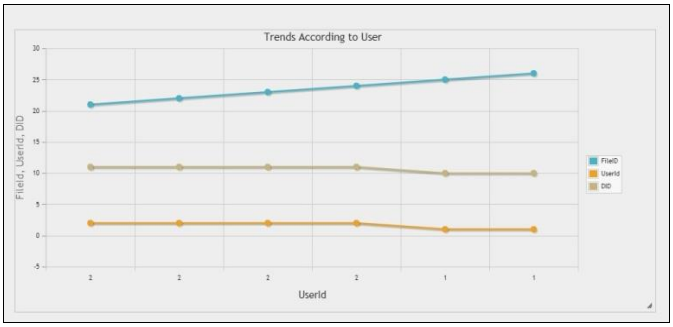

Fig 5: Graph compare Original & Encrypted file using graph

Figure 3, 4 & 5 are showing the module 2 that is the comparison phase deals with comparing two file one is plain text file and the other is cipher text file content.

5. Conclusion

In this paper, we analyze different attribute-based encryption schemes: ABE, KP-ABE, CP-ABE, and ABE with non-monotonic access structure. KP-ABE and CP-ABE have formed the base for many other access methods depending on monotonic or non-monotonic access structure, but these ethos lack in flexibility and efficiency and hence these were unable to use in enterprise. To overcome the limitations of these methods we have proposed a secure time domain ABE scheme by embedding time into both the cipher texts and the keys, such that the users who hold sufficient attributes in a specific time period can decrypt the data and we have also proposed an method to achieve dynamic modification of access policies which supports efficient on-demand user/attribute revocation for data access permissions that are stored in cloud server.

6. Future Scope

In a future work, we will implement TDAAC algorithm on real cloud based big data systems and explore the time-domain access control scheme in standard model and we prove the security of TDAAC against adversaries.

One more problem that is record keeping using Block Chain. That is a block chain is a decentralized, distributed and public digital ledger that is used to record transactions across many computers so that the record cannot be altered retroactively without the alteration of all subsequent blocks and the collusion of the network.

7. Reference

1. Yang K, Jia X. Expressive, efficient, and revocable data access control for multi-authority cloud storage. IEEE Transactions on Parallel and Distributed Systems. 2014; 25(7):1735-44
2. Minu George. International Journal of "Advanced Research in Computer and Communication Engineering" on "A Survey on Attribute Based Encryption Scheme in Cloud Computing, 2013.
3. Cheng-Chi Lee. International Journal of Network Security, on A Survey on Attribute-based Encryption Schemes of Access Control in Cloud Environment. 2013; 15(4):231-240.
4. Guojun Wang. Hierarchical attribute-based encryption nd scalable user revocation for sharing data in cloud servers in Elsevier Journal, 2011.
5. Junzuo Lai. Information Forensics and Security, IEEE ransactions on Attribute Based Encryption With erifiable Out Sourced Decryption. 2013; 8(8).
6. Boyen X. Mesh signatures, in Eurocrypt, ser. Lecture Notes in Computer Science, vol. 4515. Springer. 007, 210-227.
7. Chaum D van Heyst E. Group signatures, in Eurocrypt. 1991, 257-265.
8. Maji HK, Prabhakaran M, Rosulek M. Attribute-based signatures: Achieving attribute privacy and collusion-resistance, IACR Cryptology ePrint Archive, 2008.
9. Attribute-based signatures, in CT-RSA, ser. Lecture Notes in Computer Science, vol. 6558. Springer. 2011, 376-392 1.
10. Beimel A. Secure Schemes for Secret Sharing and Key Distribution, Ph D Thesis. Technion, Haifa, 1996.
11. Sahai A, Waters B. Fuzzy identity-based encryption, in EUROCRYPT, ser. Lecture Notes in Computer Science, vol. 3494. Springer. 2005, 457-473.