International Research Journal in Global Engineering and Sciences. (IRJGES)
ISSN : 2456-172X | Vol. 2, No. 1, March-May, 2017 | Pages 19-25

Enhanced Attribute based encryption using cipher text standard policy

S.Saranya[1], Dr. V. Krishnapriya[2]

[1]Research Scholar, [2]Head of the Department , Department of Computer Science, Sri Ramakrishna college of Arts and science for women, Coimbatore

Abstract: In this paper, the main aim is to make attribute-based encryption (ABE) more suitable for access control to store data in the cloud. The main result is an extension of the decentralized CP-ABE scheme with identity-based user revocation. The revocation system is made feasible by removing the computational burden of a revocation event from the service provider, at the expense of some permanent, yet acceptable overhead of the encryption and decryption algorithms run by the users. Thus, the computation overhead is distributed over a potentially large number of users, instead of putting it on a single party (e.g., a proxy server), which would easily lead to a performance bottleneck. The encrypting this data with traditional techniques, causes that recipients must be determined formerly, moreover either they has to share the same private key or several encrypted versions (with different keys) must be stored. These undermine the possible security, efficiency and the flexibility which the cloud should provide. Attribute-based encryption (ABE) proposed is intended for one to many encryption in which cipher texts are encrypted for those who are able to fulfill certain requirements. The most suitable variant for fine-grained access control in the cloud is called cipher text policy (CP) ABE, in which cipher texts are associated with access policies, determined by the encrypted and attributes describe the user, accordingly attributes are embedded in the users' secret keys. A cipher text can be decrypted by someone if and only if, his attributes satisfy the access structure given in the cipher text, thus data sharing is possible without prior knowledge of who will be the receiver preserving the flexibility of the cloud even after encryption. All the performance and key strength has been calculated according to the user key revocation methods. Whenever the actor's attribute has been changed the key will get update itself without any conditions or delay or request from the admin. This is a cyclic process happening during every change.

Keywords – Attribute-based encryption (ABE), cipher text policy (CP-) ABE, encryption and decryption

Related Work

The application of information technology to healthcare (healthcare IT) has become increasingly important in many countries in the recent years. There are continuing efforts on national and international standardization for interoperability and data exchange[1]. Many different application scenarios are envisaged in electronic healthcare (e-health), e.g., electronic health records, accounting and billing, medical research, and trading intellectual property .In particular e-health systems like electronic health records (EHRs) are believed to decrease costs

in healthcare (e.g., avoiding expensive double diagnoses, or repetitive drug administration) and to improve personal health management in general [1]. Examples of national activities are the e-health approach in Austria , the German electronic Health Card (eHC) system . under development, or the Taiwan Electronic Medical Record Template (TMT). [2] Healthcare organizations also must comply with multiple standards and regulations regarding patient data privacy, including those issued by the Joint Commission, the Health Insurance Portability and Accountability Act (HIPAA), and individual states. [2] Accordingly, they are implementing methods to monitor and report access to critical systems and information. In addition, they recognize the need to create and enforce security policies to protect critical endpoints, such as databases containing sensitive data, like protected health information (PHI), as well as electronic medical records (EMRs), and electronic health records (EHRs). [3] An evaluation of our design shows that our ABE library performs well, has acceptable storage requirements, and is practical and usable on modern smart phones. There are multiple, parallel efforts underway to modernize medical records systems for greater efficiency, improved patient care, patient safety, patient privacy, and costs savings[3]. Data security, as it exists in many other applications, is among these challenges that would raise great concerns from users when they store sensitive information on cloud servers. These concerns originate from the fact that cloud servers are usually operated by commercial providers which are very likely to be outside of the trusted domain of the users [4].

Introduction

In several distributed systems a user should only be able to access data if a user posses a certain set of credentials or attributes. Currently, the only method for enforcing such policies is to employ a trusted server to store the data and mediate access control. However, if any server storing the data is compromised, then the confidentiality of the data will be compromised. In this paper, present a system for realizing complex access control on encrypted data that call Cipher text-Policy Attribute-Based Encryption. By using proposed techniques encrypted data can be kept confidential even if the storage server is untrusted; moreover, encrypted methods are secure against collusion attacks. Previous Attribute Based Encryption systems used attributes to describe the encrypted data and built policies into user's keys; while in our system attributes are used to describe a user's credentials, and a party encrypting data determines a policy for who can decrypt. Thus, the methods are conceptually closer to traditional access control methods such as Role-Based Access Control (RBAC). In addition, provide an implementation of our system and give performance measurements. In many situations, when a user encrypts sensitive data, it is imperative that establish a specific access control policy on who can decrypt this data. For example, suppose that the FBI public corruption of- fixes in Knoxville and San Francisco are investigating an allegation of bribery involving a San Francisco lobbyist and a Tennessee congressman. Traditionally, this type of expressive access control is enforced by employing a trusted server to store data locally. The server is entrusted as

a reference monitor that checks that a user presents proper certification before allowing him to access records or files. However, services are increasingly storing data in a distributed fashion across many servers. Replicating data across several locations has advantages in both performance and reliability. The drawback of this trend is that it is increasingly difficult to guarantee the security of data using traditional methods; when data is stored at several locations, the chances that one of them has been compromised increases dramatically. For these reasons, to require that sensitive data is stored in an encrypted form so that it will remain private even if a server is compromised.

The method is reasonably simple. Key matrix KLx2 where,

$$k_{ij} \in \{1,2,3,4,5,6,7,8\} \begin{cases} \forall i = 1,...,L \text{ ; L} \geq 16 \\ \forall j = 1,2 \end{cases}$$

This key is known only to the sender and receiver. When the first party wants to send a message M to the second party, he/she determines the key 2 L K× and every character from the message is replaced by a binary value. An eight-bit octet is generated randomly and set in a temporary vector V. The bits in the vector V from position K [1,1] to position K[1,2] are replaced by bits from the secret message. Then the resulting vector V is stored in a file. As long as the message file has not reached its end yet, move to the next row of the key matrix and another octet is generated randomly and the replacement is performed repeatedly and the resulting vector is stored in the file. The previous procedure is repeated over and over again pending the end the message. The resulting file is sent to the receiver who beforehand has the key matrix. If the key length is not enough to cover the whole message during the encryption process, the key will be reapplied over and over again until the encryption of the whole message is completed.

The Decryption process

For decrypting the received encrypted file the following steps are taken. An octet is read from the encrypted binary plain text message EBPM file, then it is set in a temporary vector V, from this vector, bits are extracted from position K(1,1) to position K(1,2) and set in a BPM file. Since the EBPM file is nonetheless not empty, the next octet is read from the EBPM file and then it is set in a temporary vector V. From this vector, bits are extracted from position K (2, 1) to position K (2, 2) and added to the binary plain text message BPM file. The above steps are repeated over and over again until the EBPM file becomes empty. Every octet form the BPM file is transformed to the corresponding character, and then is put in the plaintext file. When the EPBM is empty the plaintext file becomes the message. In case that the key length is not enough to cover the whole message during the decryption process, the key will be reapplied over and over again till the decryption of the whole message is completed.

Key Length

To show the number of possible keys, i.e., the key space when the key length is 16. The probability of replacing a string of bits whose length ranges from 1 to 8 bit in an octet is 1/64. Consequently, if the key length is 16 there are 64(16) = 7.9x10(28) possible keys. If the attacker has a cipher text and he knows that the key length is 16, there are 7.9x10(28) attempts to find the correct key, i.e. , there are 7.9x10(28)attempts to find the correct plaintext or secret message. Assuming that a super computer working in parallel is able to try 10(12) attempts per second, it will take 2.5x10(9) years to find the secret message. Note that the universe is only 10(10) years old. This eliminates brute force attack; however other types of attacks will be discussed in future work.

The aim of the algorithm is hiding a number of bits from plain text message (M) into a random vector (V) of bits. The locations of the hidden bits are determined by the key $K_{L\times 2}$]

Input:

M[plain text message], $K_{L\times 2}$ [Key array]

Algorithm Body:

First: in a plain text file, each character is sequentially replaced by its binary value.

```
i:=0
m := first digit in M file
while (m ≠ EOF) [EOF: End Of File]
i:=i mod L
Generate 8-bits randomly and set them in V
Vector
if (K[i,1]≤K[i,2]) then
                    for j=K[i,1] to K[i,2]
                      if (m ≠ EOF)
                        then do
                          V[j] = m
                          m := next m in M file
                        end do
                    next j
else
                    for j=K[i,1] downto K[i,2]
                      if (m ≠ EOF)
                        then do
                          V[j] = m
                          m := next m in M file
                        end do
                    next j
Save V in output file
i:= i+1
end while
Output: encrypted file
```

Algorithms analysis

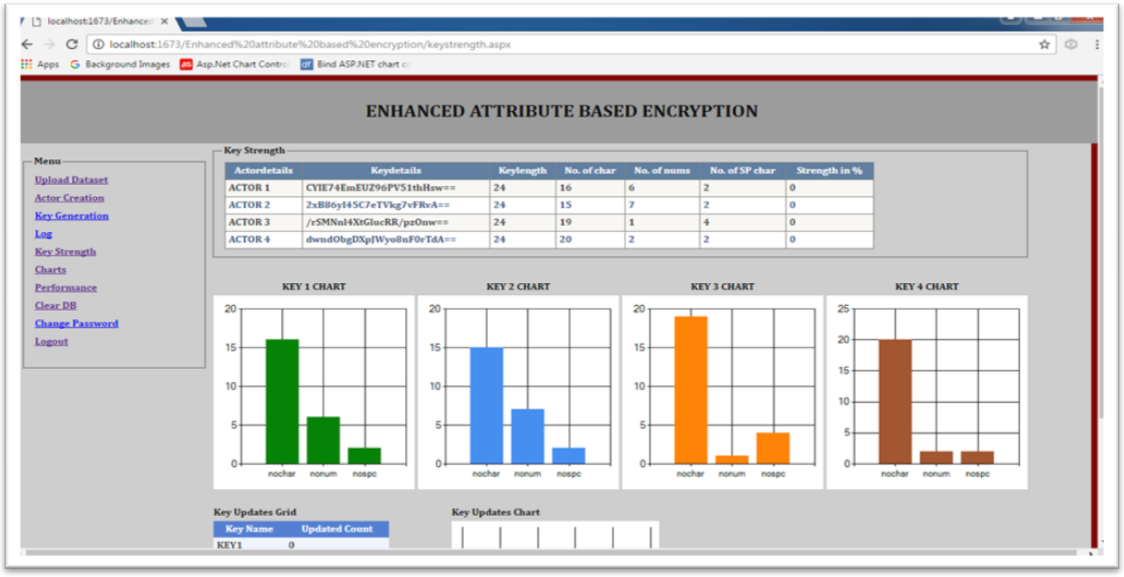
The worst case, regarding storage requirements, occurs when replacing one bit only from message to the V vector. Hence, cipher text equal eight times the size of the plain text. It have analysed worst case running times for our encryption algorithm and found that it has linear complexity of O (n).

- Key length is variable: the key length can be varied from 16 up to any larger value depending on the security level required.
- Word length is variable: the block size can be varied between 1 to 16 bits or 1 to 32 bits and so on. That is, encryption can be performed on 16, 32 or 64 bit blocks. This, in turn, can be used on different processor architectures employing 16, 32, or 64 bit registers.
- The algorithm, therefore, provides variable degrees of security. However, this improved security levels will be at the cost of increased size of the cipher text.
- The number of rounds is variable: the whole process can be repeated r times using the same key

Result and analysis

Encryption type = 64 bit key
Key Length = 23 char
Key type = Alpha numerical with special characters.
Key character = Encryption and decryption
Key Limitation = Caps alphabets = 26, Small alphabets = 26, 0 -9 numbers = 10, Special Characters = 10: result = **3.848329407410064e+135** of key combinations can be produced. It is equivalent to 30000 trillion and above combination

Fig 1.Key strength



Key strength has been decided with the number of character, numbers and special characters. As per the result most of the key is appearing in the key strength of 95 % to 99%. Which is comes under the best key character. This result represents the attribute permission for the actors available in this application. Initially all the available actors will represent in the chart. And also using grouping methods, the actor availability will be shown in actor wise. This shows the number of actors in arch group. In added with the attribute wise chart has been created for viewing the permissions. And also the chart represents the most permitted attribute to the users. This makes more data clarity to view the current status . Also the actors are un grouped to view the each user's attribute permission, with actor wise. Hence the proposed system having best performance in time, and execution process. The overall system is working good with major change in the performance between the existing system to the current system.

Conclusion

The system has been working more efficient than expectation. Before giving the formal proof, point out that from the point of view of a user, whose attributes have never satisfied the access structure defined in the cipher text, construction is at least as secure as the one by because the computation is equivalent to the decryption computation given there in this thesis. The system is similar to a decision support system that provides useful transformation to the decision makers of admin. This information helps in making decisions regarding assignment of frequently data access in the server the system required by the client based on their input in a faster manner. Since the Input given by the client is analyzed using the data mining techniques, an unknown or hidden information is retrieved from the data base. To create a system for Cipher text-Policy Attribute Based Encryption. The system allows for a new type of encrypted access control where user's private keys are specified by a set of attributes and a party encrypting data can specify a policy over these attributes specifying which users are able to decrypt. Our system allows policies to be expressed as any monotonic tree access structure and is resistant to collusion attacks in which an attacker might obtain multiple private keys. Finally, we provided an implementation of our system, which included several optimization techniques.

References

1. Securing the e-health cloud h. Lo¨ hr, a.-r. Sadeghi, and m. Winandy, "securing the e-health cloud," proc. First acm int'l health informatics symp. (ihi '10), pp. 220-229, 2010.
2. At risk of exposure - in the push for electronic medical records, concern is growing about how well privacy can be safeguarded "at risk of exposure - in the push for electronic medical records, concern is growing about how well privacy can be safeguarded," http://articles.latimes.com/2006/jun/26/health/ he-privacy26, 2006.
3. Patient controlled encryption: ensuring privacy of electronic medical records. J. Benaloh, m. Chase, e. Horvitz, and k. Lauter, "patient controlled encryption: ensuring privacy of electronic

medical records," proc. Acm workshop cloud computing security (ccsw '09), pp. 103-114, 2009.
4. Achieving secure, scalable, and fine-grained data access control in cloud computing. S. Yu, c. Wang, k. Ren, and w. Lou, "achieving secure, scalable, and fine-grained data access control in cloud computing," proc. Ieee infocom '10, 2010.
5. G. R. Blakley. Safeguarding cryptographic keys. In National Computer Conference, pages 313–317. American Federation of Information Processing Societies Proceedings, 1979.
6. D. Boneh, X. Boyen, and E.-J. Goh. Hierarchical identity based encryption with constant size ciphertext. In R. Cramer, editor, EUROCRYPT, volume 3494 of Lecture Notes in Computer Science, pages 440–456. Springer, 2005.
7. D. Boneh and M. Franklin. Identity Based Encryption from the Weil Pairing. In Advances in Cryptology – CRYPTO, volume 2139 of LNCS, pages 213–229. Springer, 2001.
8. R. W. Bradshaw, J. E. Holt, and K. E. Seamons. Concealing complex policies with hidden credentials. In ACM Conference on Computer and Communications Security, pages 146–157, 2004.
9. E. F. Brickell. Some ideal secret sharing schemes. Journal of Combinatorial Mathematics and Combinatorial Computing, 6:105–113, 1989.
10. R. Canetti, S. Halevi, and J. Katz. Chosen Ciphertext Security from Identity Based Encryption. In Advances in Cryptology – Eurocrypt, volume 3027 of LNCS, pages 207–222. Springer, 2004.
11. M. Chase. Multi-authority attribute-based encryption. In (To Appear) The Fourth Theory of Cryptography Conference (TCC 2007), 2007.
12. C. Cocks. An identity based encryption scheme based on quadratic residues. In IMA Int. Conf., pages 360– 363, 2001.
13. E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In CRYPTO, pages 537–554, 1999.
14. R. Gavriloaie, W. Nejdl, D. Olmedilla, K. E. Seamons, and M. Winslett. No registration needed: How to use declarative policies and negotiation to access sensitive resources on the semantic web. In ESWS, pages 342– 356, 2004.
15. V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute Based Encryption for Fine-Grained Access Conrol of Encrypted Data. In ACM conference on Computer and Communications Security (ACM CCS), 2006.