Formal Security Validation of Cloud Security Protocols using Avispa Tool

Sai Srujan Rao V
Department of Information Technology,
SRM University, Kattankulathur, Chennai, India

Dharani J
Department of Information Technology,
SRM University, Kattankulathur, Chennai, India

***Abstract* — Cloud computing has attracted a lot of attention in the recent years by providing different kind of services to the users through the internet. Basically in cloud, users can store cloud objects and access those objects from the cloud storage. To provide the access privilege to the user as well as storing cloud objects in encrypted form is a challenging task Fine grained access control in cloud is also required as cloud servers are usually provided by commercial providers who are not in the same trusted domain with the users, as well as sometimes cloud object owners are also from different trusted domain, as per security requirement cloud object owners privacy also has to be maintained. To solve all these type of security problems various solutions are proposed in literature, but little attention has been paid to diagnosis all those proposed schemes In this current project we have studied all the security related protocols specifically fine grained access control protocols used for cloud infrastructure. We then compare all those proposed schemes as per security wise. We have used AVISPA (Automated validation of Internet security Protocols and Applications) tool for building and analyzing formal security models of the proposed schemes for formal security validation.**

General Terms--*Cloud computing, Access control, Avispa, Validation.*

I. INTRODUCTION

Cloud computing is a promising computing paradigm which recently has drawn extensive attention from both academic and industry. Cloud computing is the delivery of computing services over the Internet [1]. Cloud services allow individuals and businesses to use software and hardware that are managed by third parties at remote locations. Cloud computing is also facing many challenges that, if not well resolved, may impede its fast growth. While there are benefits, there are privacy and security concerns too. Data is travelling over the Internet and is stored in remote locations. In addition, cloud providers often serve multiple customers simultaneously. All of this may raise the scale of exposure to possible breaches, both accidental and deliberate. Loss of control over data and dependence on the cloud computing provider these two issues can lead to a number of legal and security concerns related to infrastructure , identity management , access control[4], risk management, regulatory and legislative compliance auditing logging, integrity control as well as Cloud Computing provider dependent risks. Few works from the proposed cloud security protocols "Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing" proposed by SHUCHENG YU [7], DAC-MACS: Effective Data Access Control for Multi authority cloud storage systems proposed by KAN YANG [5] are validated by using avispa tool[8]. The rest of this paper is organized as follows. In section II, we discuss and review the related works. In section III, we discuss the validation in existing schemes. In section IV, we will describe the analysis of existing schemes. In Section V, possible future works will be discussed, before conclusion in section VI.

II RELATED WORKS

The cloud security protocols "Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing" proposed by SHUCHENG YU, uses key policy attribute based encryption in which key is encrypted by symmetric key and the whole data and key is encrypted by using the policies. In Key-policy ABE or KP-ABE, the sender has an access policy to encrypt data. A writer whose attributes and keys have been revoked cannot write back stale information. The receiver receives attributes and secret keys from the attribute authority and is able to decrypt information if it has matching attributes. In this protocol model, there are 3 entities Data Owner, Cloud Server, User.

The communication between these entities are performed by

Data owner selects a unique ID for the data file and randomly selects a symmetric data encryption key and encrypt the data file using DEK.Define a set of attribute I for the data file and encrypt DEK with I using KP-ABE. Finally, each data file is stored in the cloud. Data owner assigns the new user a unique identity w and an access structure when the user wants access the file from a cloud, cloud server sends cipher text C .On receiving C, the user first decrypts it with his private key. Then he verifies the signature $\delta O,(P,SK,PK)$. If correct, he accepts (P,SK,PK) as his access structure, secret key, and the system public key. When the received access structure and access policies are same, the user gets file encrypted by symmetric key. User decrypts the file with symmetric key and retrieves the file.

In Ciphertext-policy, CP-ABE [2], the receiver has the access policy in the form of a tree, with attributes as leaves and monotonic access structure with AND, OR and other threshold gates. In this CP-ABE we considered the setting where ciphertexts are associated with sets of attributes, whereas user secret keys are associated with policies. In this paper we

implement an effective and secure data access control scheme for Multi authority cloud storage system. Attribute revocation method for multi-authority CP-ABE scheme that achieves both forward security and backward security. It is efficient in the sense that it incurs less communication cost and computation cost of the revocation.

Forward Security: The revoked user (whose attribute is revoked) cannot decrypt the new cipher text that is encrypted with new public key .Backward Security: The newly joined user can also decrypt the previous published cipher texts that are encrypted with previous public key if it has sufficient attributes. Entities involved in this paper they are Certification authority, Attribute authority, Owner, Cloud server, User. The CA is a global trusted certificate authority in the system. It sets up the system and accepts the registration of all the users and AAs in the system. The CA is responsible for the distribution of global secret key and global public key for each legal user in the system. Every AA is an independent attribute authority that is responsible for issuing, revoking and updating user's attributes according to their role or identity in its domain. Every AA has full control over the structure and semantics of its attributes. Each AA is responsible for generating a public attribute key for each attribute it manages and a secret key for each user associates with their attributes. The cloud server stores the owner's data and provides data access service to users. It generates the decryption token of a cipher text for the user by using the secret keys of the user issued by the AAs. The server also does the cipher text update when an attribute revocation happens. The data owners define the access policies and encrypt the data under the policies before hosting them in the cloud. They do not rely on the server to do the data access control. Instead, the cipher text can be accessed by all the legal users in the system. But, the access control happens inside the cryptography. That is only when the user's attributes satisfy the access policy defined in the cipher text, the user can decrypt the ciphetext. Each user is assigned with a global user identity from the CA. Each user can freely get the ciphertexts from the server. To decrypt a ciphertext, each user may submit their secret keys issued by some AAs together with its global public key to the server and ask it to generate a decryption token for some ciphertext. Upon receiving the decryption token, the user can decrypt the ciphertext by using its global secret key. Only when the user's attributes satisfy the access policy defined in the ciphertext, the server can generate the correct decryption token. Then, the user can use the content key k to further decrypt theencrypted data component.

III.VALIDATION IN EXISTING SCHEMES

In computer security and cryptanalysis, data can be transmitted or stored by a computer system can be vulnerable to attacks. A common approach (brute-force attack) is to try repeated guessing for the key. In Dictionary attack, the search space can be built in a variety of ways. Most attackers will grab list of words for a variety of topics and languages. The validation of the protocols is important now days because whether the protocol is vulnerable to an attacks. For validation of the existing protocols we we analyse the formal security of all the existing access control schemes using the AVISPA tool, called the Automated Validation of Internet Security Protocols and Applications. The AVISPA tool provides a suite of applications for building and analysing formal models of security protocols. Protocol models are written in the High Level protocol Specification Language, or HLPSL.The AVISPA tool comprises four back-ends: OFMC, CLAtSe, SATMC, and TA4SP[9].OFMC performs several symbolic techniques to explore the state space in a demand-driven way. CL-AtSe provides a translation from any security protocol specification written as transition relation in intermediate format into a set of constraints which are effectively used to find whether there are attacks on protocols. SATMC builds a propositional formula and then the formula is fed to a state-of-the-art SAT solver to verify whether there is an attack or not. Finally,TA4SP is a back-end which approximates the intruder knowledge by using regular tree languages.

IV. ANALYSIS OF PROPOSED SCHEMES

In this section, we describe analysis of YU paper we have chosen the back-ends OFMC and Cl-AtSe for an execution test and a bounded number of sessions model checking. For the replay attack checking, the back-ends check whether the legitimate agents can execute the specified protocol by performing a search of a passive intruder. After that the back-ends give the intruder the knowledge of some normal sessions between the legitimate agents .For the Dolev-Yao model check, the back-ends check whether there is any man-in-the-middle attack possible by the intruder. We have simulated all the discussed existing schemes under both the back-ends OFMC and Cl-AtSe. The formal verification analysis of the yu scheme shown in ensures that it is secure against replay and man-in-the-middle attacks.In another paper also we validate the results using AVISPA tool We have chosen the back-ends OFMC and Cl-AtSe for an execution test and a bounded number of sessions model checking. For the replay attack checking, the back-ends check whether the legitimate agents can execute the specified protocol by performing a search of a passive intruder. After thatthe back-ends give the intruder the knowledge of some normal sessions between the legitimate agents .For the Dolev-Yao model check, the back-ends check whether there is any man-in-the-middle attack possible by the intruder.We have simulated all the discussed existing schemes underboth the back-ends OFMC and Cl-AtSe. The formal verification anlysis of the Kan Yang scheme shown inensures that it is secure against replay and man-in-the-middle attacks.

We have compared the results of the formal security analysis of all schemes for in Table. From this table it is clear that the Shucheng Yu scheme and the Kan Yang scheme are safe.

Scheme	**Results Using Ofmc and Atse**
Shucheng Yu	Safe
Kan Yang	Safe

V.POSSIBLE FUTURE WORK

In future, there would be implementations of several protocols in cloud computing for accessing of the data. These protocols are to be validated for better usage in accessing the data.

VI.CONCLUSION

In this paper, we have validated two existing cloud security protocols using the Avispa tool, which one of the protocols is better to use for accessing the data. Finally, we have analyzed the results of the tool by using two backend ofmc and atse, both results state that the protocols are safe.

ACKNOWLEDGMENT

I am grateful to the principal and management of SRM University for extending all the facilities and constant encouragement for carrying out this work. Also heartily thank Dharani.J for giving me an opportunity to complete this research. You have been a tremendous mentor for me. I would like to thank you for encouraging my research. Your advice on both research as well as on my career have been priceless.

REFERENCES

[1] P. Mell and T. Grance, "The NIST definition of cloud computing," National Institute of Standards and Technology, Tech. Rep., 2009.

[2] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attributebased encryption," in Proceedings of the 2007 IEEE Symposium on Security and Privacy (S&P'07). IEEE Computer Society, 2007, pp. 321–334.

[3] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive,efficient, and provably secure realization," in Proceedings of the 4th International Conference on Practice and Theory in Public Key Cryptography (PKC'11). Springer, 2011, pp. 53–70.

[4] V. Goyal, A. Jain, O. Pandey, and A. Sahai, "Bounded ciphertext policy attribute based encryption," in Proceedings of the 35th International Colloquium on Automata, Languages and Programming (ICALP'08). Springer, 2008, pp. 579–591.

[5] K. Yang, X. Jia, and K. Ren, "DAC-MACS: Effective Data AccessControl for Multi-Authority Cloud Storage Systems," IACR Cryptology ePrint Archive, p. 419, 2012

[6] S. Ruj, A. Nayak, and I. Stojmenovic, "DACC: Distributed Access Control in Clouds," in Proceeding of the 10th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom'11). IEEE, 2011, pp. 91–98.selection," Proc. of the 2009 Int. Conf. on Security and Management, July 2009, pp. 188- 194.

[7] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute Based Data Sharing with Attribute Revocation," Proc. ACM Symp. Information,Computer and Comm. Security (ASIACCS), pp. 261-270, 2010 G.

[8] AVISPA: Automated validation of internet security protocols and applications (2003) FET Open Project IST-2001-39252.www.avispa-project.org

[9] Basin D, M¨odersheim S, Vigan`o L (2003) An on-the-fly model-checker for security protocol analysis. In: Proceedings of ESORICS'03. Lecture notes in computer science, vol 2808.Springer, Berlin Heidelberg New York, pp 253–270