Analysis of Group Key Protocols in Cloud

Suresh Yejjuparapu, Archana Raghuvamshi

***Abstract* - Cloud Computing is an emerging technology in IT industry. It is a model for enabling convenient, on-demand network access to a shared pool of computing resources on pay-per-use basis. It provides infrastructure, platform and software as services. Cloud customers can access the required services from the cloud and cloud provider provides the services and manages the cloud. Due to security issues, we need secure protocols to implement group oriented applications. Providing security to the messages in the group communication is essential. The key transfer protocols depend on the trusted Key Generations Center (KGC). This paper contains analysis of group key protocols in cloud.**

***Index Terms* - Cloud computing, Group, Group key, Key transfer protocols, KGC, pay-per-use.**

I. INTRODUCTION

A Group Key protocol [1] is a protocol; where two or more members are agree on a key in such a way that both can influence the result. If it is properly done, this avoids undesired actions on the third parties. These protocols that are useful in real-time, also do not reveal the key on which the third parties are agreed.

In most of the key exchanging systems, one party will generate the key and sends to other parties and these parties will not influence the key. By using this group key protocol avoids the key distribution problems in the key exchanging systems. If protocols are implemented with forward confidentiality then both parties influence the key.

Group applications have widely spread in the last few years. They allow group of members to work on common resources or platforms. Group communication may be a text, audio or video conference etc. Multiple users may access the application at a time.

Secure message transfer is very important in group communications. Group of authorized members can communicate by using secret key sharing [2] method. To provide data privacy, an effective method is required to all group members to generate a common secret key. Data confidentiality is very important in group communication. To provide a secure group communication, it is necessary to manage keys in the secure way for creating, updating and distribution of those keys.

Suresh Yejjuparpu, M.Tech Student, Department of CSE, University College of Engineering, Adikavi Nannaya University, Rajahmundry.

Archana Raghuvamsi, Assistant Professor, Department of CSE, University College of Engineering, Adikavi Nannaya University, Rajahmundry.

Before exchanging the confidential data, the protocol has to distribute the group key to all the members securely and efficiently. The most common method used for achieving group communication in a secure way is by encryption techniques [3].

The most used key agreement protocol in the existing work is Diffie Hellman key agreement protocol. However the Diffie Hellman algorithm is limited to provide secret key only for two entities, it is not preferable if the group contains more members. If there are more members in the group, more time is required to distribute the key. Hence it is necessary to implement to avoid this type of problem in group communication.

II. RELATED WORK

Many Group Key Agreement protocols have been proposed in literature, most being derived from the two-party Dife-Hellman (DH) key agreement protocol. While some are secure against passive attacks, others do not have a strong security proof. A Security proof shows how an attack on a protocol can be solve the problem under some strong assumptions. This type of well defined model of security protocols were first designed by Bresson et al [4].

Yung et al, proposed the first provably-secure constant round Group Key Agreement protocol inspired from the works of Burmester et al. In the same work, they also proposed a scalable compiler to transform a Group Key Agreement protocol, secure against passive attacks, into one which is secure against active attacks. Boyd et al. proposed an efficient constant round protocol where the bulk of the computation done by one member.

There are many security issues associated with the cloud computing. These can be categorized as issues faced by the cloud providers and issues face by the cloud customers. Both are providers and customers are responsible for these issues. The provider must ensure that their infrastructure is securely managed and customer's data is protected. The customers make sure their passwords regarding authentication.

When third parties want to store their personal data in public cloud, they lost their physical access to cloud servers hosting their data. This leads, a sensitive and confidential data is at the risk from the insider attacks. According to cloud security surveys, these types of attacks are greatest threat to cloud computing. So the organization must background check their employees who has physical access to data centers. And also, data centers must frequently monitor for suspicious attacks.

III. ANALYSIS of Wu et al. Group Key Protocol in Cloud

Wu et al [5], introduces necessary interpolation method background used to build parts of the proposed protocols. The aim of the polynomial interpolation method is to reconstruct the unknown function f by seeking the polynomial p_n whose graphs is in (x,y) plane through the points $(x_i,f(x_i))$, i=0,…..n.

Lagrange's Interpolation Formula [6]

For n + 1 support points (x_i, f_i), i = 0...n there is a polynomial $P_n(x)$, degrees do not exceed n, with $P_n(x_i) = f_i$, i = 0, ..., n. We will construct the interpolating polynomial $P_n(x)$ explicitly as following equation:

$$P_n(x) \equiv \sum_{i=0}^{n} f_i \prod_{k\neq i,k=0}^{n} \frac{x - x_k}{x_i - x_k}. \qquad (1)$$

Newton's Interpolation Formula

Newton's Interpolation Formula adopts divided differences to construct $P_n(x)$.

$$f_{i_0 i_1 \ldots i_k} = \frac{f_{i_1 \ldots i_k} - f_{i_0 \ldots i_{k-1}}}{x_{i_k} - x_{i_0}}. \qquad (2)$$

x_i	f_i	$k=1$	$k=2$	$\cdots$	$k=n$
x_0	$\underline{f_0}$				
x_1	f_1	$\underline{f_{01}}$			
x_2	f_2	f_{12}	$\underline{f_{012}}$		
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	
x_n	f_n	$f_{n-1,n}$	$f_{n-2,n-1,n}$	$\cdots$	$\underline{f_{012\ldots n}}$

Table 1: Divided Difference Scheme

we can calculate divided differences. And with the descending diagonal of the divided difference scheme, the coefficients $f_{i0i1\ldots ik}$ can be calculated, the interpolation problem with the Newton's interpolation formula is solved by

$$P_n(x) \equiv f_0 + f_{01}(x - x_0) + \cdots + f_{01\ldots n}(x - x_0)(x - x_1)\ldots(x - x_{n-1}). \qquad (3)$$

3.1 AUTHENTICATED GROUP KEY TRANSFER PROTOCOL BASED ON SECRET SHARING

The scheme gets key confidentiality from the security feature of Shamir's secret key sharing [7] and hash functions. It provides key authentication by providing a single authentication message. Next, the scheme handles both insider and outsider attacks. This protocol has two parts, pre distributing phase and distributing phase.

Pre-distributing phase:

KGC publishes N = pq where p and q are cryptographic primes, and publishes secure hash functions h1(x) and h2(x). Then, each member Ui registers at KGC, and shares his long-term secret (xi, yi) with KGC in a secure manner.

Distributing phase:

1. The initiator sends key sharing request to KGC with a list as {U0, U1, U2, . . . Un−1}, KGC broadcasts it.

2. Each member Ui($0 \leq i \leq n - 1$) broadcasts a random challenge $Ri \in Z_N$ to KGC as a response.

3. KGC randomly selects a group key k, and generates an interpolated polynomial f(x) with degree n to pass through (n+1) points, (0, k) and (x_i, y_i $h1(x_i, y_i,R_i)$) ($0 \leq i \leq n - 1$), where x y denotes (x + y)(mod N). Then, KGC computes additional n points Pi = (i, f(i)) for i = 0, . . . , n−1, and authentication message Auth = h_2(k,U0, . . . , Un−1,R0, . . . , Rn−1, P0, . . . , Pn−1). Finally, KGC broadcasts Auth and {Pi}n−1 i=0 .

4. Each member Ui reconstruct f(x) with his shared secret (xi, yi ⊕ h1(xi, yi,Ri)) and the broadcasted messages {Pi}n−1 i=0, recovers k = f(0). Next, Ui authenticates k with Auth.

3.2 GROUP KEY TRANSFER SCHEME IN CLOUD COMPUTING

In cloud computing, the above protocol is infeasible, because KGC and group members are weaker than computationally powerful players, hence they cannot fulfill their individual calculation. In some environment, network bandwidth is not as important as group members' computational power. In order to distribute a common key from KGC to each group member in cloud computing, asking cloud servers for computing interpolated polynomial provides an avenue for them to come up with a common secret key, i.e., KGC and group members outsource interpolation computation to cloud servers. However, this protocol cannot directly be executed in cloud computing, because both KGC and group members do not hope that the sensitive information is leaked to the public cloud servers

System initialization:

Key Generation Center randomly selects two safe primes p and q, such a way that p and q are primes such that p=(p-1)/2 and q=(q-1)/2 are also primes and computes N=pq. N is publicly known. KGC chooses two safe hash functions h1(x) and h2(x) and made publicly known.

User registration:

All Group members must register at KGC to get keys while key distribution. During registration, KGC shares a secret with members.

Key distribution:

This phase constitutes the core of the protocol and is performed whenever a group of users {U0,. . . , Un−1} decide to establish a common session key. We will first introduce the

completed key distribution procedure, and then individually introduce some core algorithms of the key distribution.

IV. ANALYSIS of aruna et al. Group Key Protocol in Cloud

The goals of Aruna et al [8] group key protocol are key authentication and key freshness. For the proper group communication, we should ensure the key freshness and a key cannot be reused. If KGC distribute the same key again, it damages the group communication. Key confidentiality ensured by an authorized group member. The authorized key distribution is done by the KGC not by any intruder.
This protocol protects the information broadcasted from KGC to group members. In this protocol the service request and challenge messages are not authenticated. An intruder can try get group key service like a group member. Intruder can also modify the information transmitted from users to KGC.
Aruna et al, have prove that none of the inside and outside attacks can successfully attack the authorized group members because the attackers cannot get the group key.

4.1 PROTOCOL DESIGN

In Aruna et al protocol has three phases same as wu et al group key protocol.

i. **Initialization of KGC**
ii. **User Registration**
iii. **Group Key Generation and Distribution**

Aruna et al, have proved the security goals mention in their protocol against the insider and outsider attacks. The two types of attacks are insider and outsider. The outsider attacker can try get some information by sending a request to KGC acting as a group member. In security analysis, aruna et al show that the outside attacker gains nothing, because the attacker cannot recover the group key, because they could not gain the individual factors of the composite number used by the KGC and the prime number difference are alone known from the Vandermonde's determinant evaluation which are the public information available to the outsiders. The individual keys are generated under cyclic permutation and cyclic code representation, getting information may not help decoding permutation and cyclic code radix.

V. PERFORMANCE ANALYSIS

In Wu et al group key transfer protocol, KGC side time consumption is, key distribution encryption1 takes O(n) and key distribution encryption2 takes $O(n^2)$. The group member side time consumption is O(n). Server side time consumption is $O(n^2)$.
More specifically, the overall time cost is $O(n^2)$ for the KGC, $O(n^2)$ for the cloud server CS1, $O(n^2)$ for the cloud server CS2 and O(n) for each group member. However, according to the original scheme, KGC takes time $O(n^3)$ to adopt the Lagrange's interpolation formula and takes time $O(n^2)$ to adopt the Newton's Interpolation Formula; for each group member, it will takes time $O(n^2)$ no matter using the Lagrange's interpolation formula or the Newton's interpolation formula.

In Aruna et al protocol, each user will get a secret while registering at KGC. Adding/removing any user does not need to update any existing shared secret. But distributing group key involving t group members, KGC has to broadcast t+1 elements to all the group members. To decrypt the secret key each member needs to compute a t degree polynomial f(x). This proposed protocol suitable for distributing group key only to the group with less number of group members.

VI. CONCLUSION

In this paper, we analyzed Wu et al protocol and Aruna et al protocol. In both protocols, KGC is the trusted third party used to generate and distribute keys. For rekeying and distributing keys both protocols used interpolation formulas and secret key sharing schemes. Both protocols guarantee the confidentiality of authentication, correctness and efficiency.
Wu et al protocol gains more performance gain in key generation and key distribution. Aruna et al protocol assumes that KGC is a mutually trusted entity, so the protocol only concentrates on the group information. This protocol works better with the small size group. If the group is large, then centralized group distribution protocols have to use.

REFERENCES

[1] Lein Harn and Changlu Lin, "Authenticated Group Key Transfer Protocol Based on Secret Sharing" IEEE Trans. Computers, Vol. 59, no.6,pp.842-846, June210.
[2] A. Shamir, "How to share a secret", Comm. ACM, vol.22, no.11, pp.612-613,1979.
[3] Willliam Stallings, Cryptography and Network Security, 4th ed. Pearson Education,2009
[4] E.Bresson, O.Chevassut, and D. Pointcheval, "Provably-Secure Authenticated Group Diffie Hellman Key Exchange." ACM Trans. Information and systems security, vol.10, no.3, pp.255-264, Aug.2007.
[5] J.Wu, Q. Liu and X.Liao, "A Secure and Efficient Outsourceable Group Key Transfer Protocol in Cloud Computing.
[6] Josef Stoer and Roland Bulirsch. Introduction to Numerical Analysis, Vol. 2, Springer, New York, 1993.
[7] A. Shamir, "How to share a secret", Comm. ACM, vol.22, no.11, pp.612-613, 1979.
[8] Aruna A, Y.V.V.N Varaprasad, "Authenticated Group Key Transfer Implementation Protocol Based On Secret Sharing".
[9] NIST – Guidelines on Security and Privacy in public cloud computing. Special Publication 800-144.
[10] J.Katz and M. Yung, " Scalable protocols for Authenticated Group Key Exchange" Vol.20, pp.85-113, 2007.
[11] M. Brumster and Y. Desmedt, "A Secure and Efficient Conference Key Distribution System" Proc. Eurocrypt'94 Workshop Advances in Cryptology, pp. 275-286, 1994.
[12] Deffie and Hellman, "New Directions in Cryptography", IEEE Trans. Information Theory, Vol.IT-22, no.6 pp.644-654, Nov,1976.
[13] Hassan T, "Privacy Aware Access control for Data Sharing in Cloud Computing Environment"

[14] Ren K, Wang C and Wang Q " Security Guidance for Critical Areas of Focus in Cloud Computing V2. 1. Cloud Security Alliance, pages 1-76, 2009
[15] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro and M. Yung, "Perfectly Secure Key Distribution for Dynamic Conferences." Information and Coputation, vol.146, no.1, pp.1-23, Oct,1998.
[16] C. Gentry, Computing arbitrary functions of encrypted data. ACM, 53(3); 97-105, 2010.
[17] C. Boyd, "On Key Agreement and Conference Key Agreement" Proc. Second Australasian Conf. Information Security and Privacy,pp.294-302,1997.
[18] Tang Chunming, D.S. Wong, Xing Hu and Dingyi Pei. "An efficient key distribution scheme in cloud computing." Cloud Computing Technology and Science, 2012 IEEE 4th International Conference on, pages 557-561, 2012.
[19] Liu Y, Cheng C, Cao J and Jiang T, "An Improved Authenticated Group Key Transfer Protocol Based on Secret Sharing." IEEE Transactions on computers, 62(11): 2335-2336, 2013.
[20] A.C.Yao, "Protocols for secure computations" in Proc. of FOCS'82, pages 160-164, 1982
[21] Endre Suli and David F. Mayers, "An Introduction to Numerical Analysis, Cambridge Press, 2003
[22] Seny Kamara and Kristin Lauter, "Cryptographic cloud storage." Financial Cryptography and Data security, pages 136-149, 2010.
[23] Ren K, Wang C and Wang Q, "Security challenges for the public cloud." Internet Computing, 16(1): 69-73, 2012.