ISSN 2278-7763

An Identity Based Key Exchange Protocol in Cloud Computing

Venkateswara Rao Molli[1], Omkar Nath Tiwary[2]

[1]Information System and Network Engineering,St Joseph University in Tanzania,Dar-Es-Salam,Tanzania; [2] Information System and Network Engineering,St Joseph University in Tanzania,Dar-Es-Salam,Tanzania.
Email: venkateshmolli@gmail.com,tiwary45@gmail.com

ABSTRACT

Workflow systems often use delegation to enhance the flexibility of authorization; delegation transfers privileges among users across different administrative domains and facilitates information sharing. We present an independently verifiable delegation mechanism, where a delegation credential can be verified without the participation of domain administrators. This protocol, called role-based cascaded delegation (RBCD), supports simple and efficient cross-domain delegation of authority. RBCD enables a role member to create delegations based on the dynamic needs of collaboration; in the meantime, a delegation chain can be verified by anyone without the participation of role administrators. We also propose the Measurable Risk Adaptive decentralized Role-based Delegation framework to address this problem.

Describe an efficient realization of RBCD by using aggregate signatures, where the authentication information for an arbitrarily long role-based delegation chain is captured by one short signature of constant size. RBCD enables a role member to create delegations based on the need of collaboration; in the meantime anyone can verify a delegation chain without the participation of role administrators. The protocol is general and can be realized by any signature scheme. We have described a specific realization with a hierarchical certificate-based encryption scheme that gives delegation compact credentials**.**

Keywords : *Delegation framework, Delegation credentials, Encryption.*

1 INTRODUCTION

Cloud Computing is a term that is often bandied about the web these days and often attributed to different things that -- on the surface -- don't seem to have that much in common. So just what is Cloud Computing? We analysis it called a service, a platform, and even an operating system. Some even link it to such concepts as grid computing -- which is a way of taking many different computers and linking them together to form one very big computer.

In this cloud computing model the major role has been given to the service provider the admin person must be there because his authorization signature is required to provide a service to clients. Since pay was maintained the server side has concern about security.

The main challenge addressed in this paper is the verification of role-based authorization chains in decentralized environments, which has not been much studied in the existing literature. We have presented the RBCD model and its associated cryptographic operations for convenient verification of delegation chains. RBCD enables a role member to create delegations based on the need of collaboration; in the meantime anyone can verify a delegation chain without the participation of role administrators. Our protocol is general and can be realized by any signature scheme. We have described a specific realization with a hierarchical certificate-based encryption scheme that gives delegation compact credentials.

In our RBCD, given a privilege, two types of entities can delegate the privilege to others: 1) the resource owner of the privilege and 2) a member of a role who is delegated the privilege Decentralized role-based delegation allows users from administratively Independent domains to be dynamically joined according to the needs of the tasks. We have also explored the applications of RBCD for efficient and flexible trust establishment in decentralized and pervasive environments.

2 PROCEDURE FOR PAPER SUBMISSION

2.1 Cloud computing

A basic definition of cloud computing is the use of the Internet for the tasks you perform on your computer. The "cloud" represents the Internet. The simplest thing that a computer does is allow us to store and retrieve information. We can store our family photographs, our favorite songs, or even save movies on it. This is also the most basic service offered by cloud computing. Most cloud computing infrastructures consist of services delivered through common centers and built on servers. Clouds often appear as single points of access for all consumers' computing needs. Commercial offerings are generally expected to meet quality of service (QoS) requirements of customers.

A cloud service has three distinct characteristics that differentiate it from traditional hosting. It is sold on demand, typically by the minute or the hour; it is elastic -- a user can

ISSN 2278-7763

have as much or as little of a service as they want at any given time; and the service is fully managed by the provider (the consumer needs nothing but a personal computer and Internet access). Significant innovations in virtualization and distributed computing, as well as improved access to high-speed Internet and a weak economy, have accelerated interest in cloud computing.

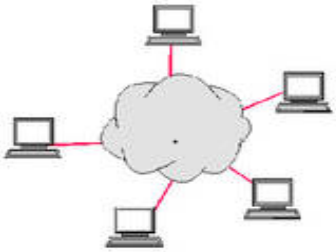

Fig 1.cloud computing

With a cloud computing model, the company doesn't own the physical infrastructure. This helps to avoid capital expenditures, and the infrastructure is essentially rented from a third party. The company uses resources and pays for only the specific resources they use. This can improve utilization rates, because servers aren't typically left idle. The rise in large bandwidth possibilities for companies is responsible, in part, for the rise in cloud computing. It's reasonable, in many applications, to get the same response times from remote servers as it is from local infrastructure.

One of the primary benefits of cloud computing is that it can reduce both capital expenditures on infrastructure, as well as operational expenditures on infrastructure maintenance and engineering. In this way, companies pay only for what they use, or they subscribe to a service over time. Cloud computing has a relatively low barrier to entry, a low management overhead, and rapid access to a wide range of application types.

Cloud computing providers generally offer service level agreements (SLAs) to insure that the company is able to access the systems they need and that uptime will be similar, or even higher, than locally-based solutions.

2.2 Delegation& Decentralization

When an organization retains its authority and responsibility at the higher levels, it is said to be centralized. When an organization delegates its authority and responsibility to the lower levels, it is said to be decentralized. In a special-operations-forces organization, the degree of centralization or decentralization is determined not so much by the specific manner in which the commander has organized his subordinate units as it is by the policies that he has established to guide his operations. In centralization, a limited amount of authority is delegated. In decentralization, a significant amount of authority is delegated to lower levels. Each form has its advantages and disadvantages and is affected by a number of factors, such as size of organization and the amount of geographic dispersion. If the organization is very large, diversified or geographically dispersed, the limitations of expertise and personal resources will generally lead to decentralization of authority to the heads of these different businesses. Innovative enterprises, where speed and adaptability to change are characteristics of the business, tend towards decentralization.

Delegation is the process that makes management possible, because management is the process of getting results accomplished through others. A manager should provide team members with the information they require to do a good job, communicating with them frequently, and giving them clear guidelines on the results that are expected. Further, managers must also take the "relationship responsibility" for those with whom they work. Delegation is the preferred approach to managing and coaching people who have high skill and high will to complete the specific task at hand.

At a certain point, there are just too many facets to running a successful business to continue doing it alone. In an increasingly complex business environment, with all the trends affecting business today, such as globalization, the information technology explosion, strategic alliances, increased mergers and acquisitions, heightened competition, and higher expectations of nearly every customer, it just isn't possible to still be that one person in control of everything. Bringing in others to manage is an absolute necessity for survival now.

Owners and managers should concentrate on the activities they do that bring the most value to their organization. You must perform only "essential activities" that give the company its competitive advantage over other companies in the industry. Learn to do less and manage more. The delegation task is in finding the right persons and giving them the right work. The sheer volume of management responsibilities necessitates delegation. Always drop unnecessary work altogether; concentrates only on the tasks that nobody else can do. Necessary tasks that can be done by others should be delegated. Often the need to delegate is sparked by rapid business expansion, particularly as a result of acquisition.

2.3 Overview of RBCD

We propose a model for the delegation of authority in role-based TM systems, called RBCD. The main goal of this model is to allow flexible transfer of privileges and sharing of resources in decentralized environments. Our model allows a role member to delegate his/her privileges to users who may belong to different organizations as opposed to restricting this delegation ability to role administrators in traditional access

ISSN 2278-7763

control models. In addition, our RBCD model allows delegatees to further extend the delegated privileges to other collaborators. The challenge that arises in realizing this goal in a decentralized environment is that the public key of an intermediate delegator may not be known by a verifier or the resource owner. Therefore, the delegation credential signed by that delegator may not be trusted by the verifier.

The distributed cascaded delegation problem essentially de-signs a delegation mechanism that efficiently verifies a hierarchical delegation chain. In the cascaded delegation model, a delegation recipient *E* may further extend the delegated privilege to another entity *E~,* and the delegation credentials of *E* are passed to entity *E~,* along with the delegation certificate signed by *E* as the issuer. The public key of the next delegate is encoded in the delegation credential, which naturally forms a chain of trust. Therefore, trusting the original delegator means HCBE_AGGREGATE can take any number of signatures. That the delegatee's public keys are authorized by the delegation. In addition, the authorization chain is stored in delegation credentials and does not have to dynamically be discovered. However, previous cascaded delegation protocols support neither multiple administrative domains nor the use of roles in the delegation. We give support to both in our RBCD model.

In our RBCD, given a privilege, two types of entities can delegate the privilege to others: 1) the resource owner of the privilege and 2) a member of a role who is delegated the privilege. A role *r* is delegated a privilege by receiving a delegation credential *C* that explicitly assigns the privilege to role *r.* Members of the role *r* are allowed to further delegate the privilege to another role *r~* as follows. A member *D* of the role *r* uses the delegation credential *C* to generate a delegation credential *C~*. *C~* comprises multiple component credentials, which include the credential of the current delegation authorization, the credential *C* from the preceding delegation, and the role membership credential of the delegator *D.* The verifier can make the authorization decision based on delegation credential *C~* and the role membership credential of the requester. Verification can be done by any party without the participation of any role administrators, which we call independent verifiability.

The length of a delegation chain in RBCD refers to the number of delegators involved. A privilege *P* is delegated by an entity *E* to a role *r*1. A member *D* of role *r*1 further delegates the same privilege *P* to role *r*2. The delegation chain of privilege *P* involves entity *E,* role *r*1, entity *D,* and role *r*2. Role *r*2 receives the privilege *P* as the result of the delegation chain. The length of the chain is two.

Decentralized role-based delegation allows users from administratively independent domains to be dynamically joined according to the needs of the tasks.

3 EXISTING SYSTEM

The previous cascaded delegation protocols support neither multiple administrative domains nor the use of roles in the delegation. The previous delegation, it is impossible for others to find out what the role signature or the extension signature is: Existing delegation models assume that the delegation is issued by administrators. However to enable the flexible resources sharing each and every authorized user have the rights to accesses the delegation but in existing system it cant do that because this will be issued by administrator only. Drawbacks in Existing system:

[1] Traditionally, the number of signatures required for the verification of a delegation chain is linear to the number of entities of the chain.

[2]The Existing system is not a role based model but in proposed we use delegation in a role based model. It can't be simplified to support individual delegation.

4 PROPOSED SYSTEM

In this proposed System We have to enable flexible resource sharing, the decision of introducing new role members into collaboration needs to dynamically be made by members of existing roles, without the involvement of administrators. In the meantime, the shared information needs to adequately be protected against unauthorized or unqualified users. These goals drive us to reexamine conventional assumptions in role-based delegation. The main challenge addressed in this paper is the verification of role-based authorization chains in decentralized environments, which has not been much studied in the existing literature.We have presented the RBCD model and its associated cryptographic operations for convenient verification of delegation chains.[1]We present a role-based delegation mechanism that supports the efficient verification of multistep delegation chains. It has two main features: (i) flexible delegation and (ii) simple verification.[2]We give a detailed protocol specification for public-key signing and management, which ensures the integrity of shared resources. The protocol name is RBCD.[3]Our implementation needs only one aggregate signature, which is a significant improvement in efficiency over the existing delegation chain protocols.[3]our delegation model is role based, it can be simplified to support individual delegations.

5 EQUATIONS

5.1 HCBE Preliminaries

Here, a brief overview of the necessary cryptographic knowledge. The Hierarchical Certificate-Based Encryption (HCBE) scheme is a public-key cryptosystem, where messages are encrypted with public keys and decrypted with the corre-

ISSN 2278-7763

sponding private keys.
HCBE_AGGREGATE (*sn, info_, sig*2, . . . , *sign*). This algorithm is run by Bob, who uses his private key *sn* and the public key certificates on his chain to compute an aggregate signature, which will be used as his decryption key. The inputs to this algorithm are Bob's private key *sn*, the string *info_* that contains the information of Bob, and a number of signatures1 that contains the public-key certificate Signatures associated with his certification chain.
HCBE_CERT_OF_CA (*si, infoi*+1). The CA at the *i*th level runs this algorithm to certify the public key of the CA at level *i* + 1 by computing a signature. The first input is the private key of CA*i*, and the second input is a string *infoi*+1 that contains the public key *si*π of the signer and the public key *si*+1π of CA*i*+1. The string *infoi*+1 may also include information such as the expiration date.

RBCD _ Initiate
RBCD _ Extend

5.1.2 RBCD_Initiate

The resource owner D0 delegates the privilegeD0.priv to members of an affiliated role A1.r1. The privatekey sD0 corresponds to the public key PD0 of entity D0. EntityD0 takes the following steps.

- Set the string info1 = PD0_D0.priv_A1.r1_PA1. The string info1 contains the publickey PD0 of the owner of the delegated privilege, the delegated privilege D0.priv, the role A1.r1 that receives the privilege, and the publickey PA1 of the administrator of the role A1.r1.
- Run HCBE_CERT_OF_CA (sD0, info1), which outputs an extension signature X1.
- Define a string tuple chain1 as [D0.priv, PD0 , A1.r1, PA1].
- Set the partial delegation credential C1 for the role A1.r1 as (X1, chain1).
- Put credential C1 on a credential server.

5.1.3 RBCD_Extend

An entity Di, whose role is Ai.ri, further delegates D0.priv to role Ai+1.ri+1. Di uses his private key sDi , his role signature RDi , and the delegation credential Ci of the role Ai.ri to compute a partial delegation credential Ci+1. Entity Di takes the following steps.

- Parse the credential Ci as (SAgg, chaini), where SAgg is the aggregate signature of credential Ci, and chain is the corresponding string tuple. Signature SAgg is a function of the preceding extension and role signatures on the delegation chain. The string tuple chaini contains the components of the delegation chain.
- Set the string infoi+1 = PD0_D0.priv_Ai+1.ri+1_ PAi+1, where PD0 is the public key of the resource owner of the delegated privilege, D0.priv is the delegated privilege, and Ai+1.ri+1 is the role that receives the privilege and the public key PAi+1 of the role administrator Ai+1.
- Run HCBE_AGGREGATE (sDi, infoi+1,RDi, SAgg), which outputs an aggregate signature S_Agg.
- Define the string tuple chaini+1 of credential Ci+1 as the string tuple chaini appended with public key PDi, the role name Ai+1.ri+1, and the public key PAi+1.
- Set credential Ci+1 as (S_Agg, chaini+1).
- Put the partial delegation credential Ci+1 for the role Ai+1.ri+1 on a credential server.

6 MODUL DESCRIPTION

6.1 Server cloud

In this module we going to design a server which itself have to act as a server. The server has to provide the service to the client so that it can get amount for the time being used by the client. It has its own login page to identify client. Server is the person that have to monitor the cloud, authentications have to given to the clients.

6.2 Client design

The client is in the need to access data from server so, for that first it have to prove its identity to the server , then it have to pay for the time of use of the software from server, It can also select its own version of software. The work is in the web page of the server. It can select the cloud to which it is going to connect.

6.3 Delegation property

If the server is busy with other works so it delegate its job to other trusted party thus the delegated person can able to do the job of the server. Then if the delegate is not there then the delegation is gone to the third person thus a delegation chain has been maintained so that the client can able to know whom is now providing service.

6.4 Algorithm implementation

The delegation properties have to be in chain. The signature of the admin is required for authenticating individual clients. Here the signature of the persons in the delegation process also involved in the delegation chain. So that aggregation property is maintained, RSA algorithm is used to safeguard delegation chain.

7 Helpful Hints

7.1 Figures and Tables

ISSN 2278-7763

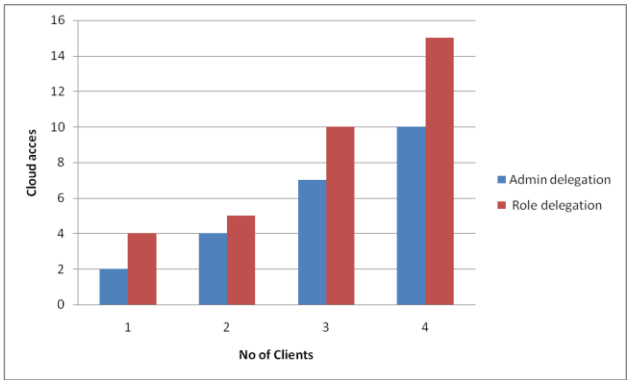

Figure 7.1 Performance analyses Graph

The graph is drawn based on comparing with admin delegation and role delegation.The role delegation performance and efficiency is more. In Admin delegation the number of clients served is less. In Client delegation the number of clients served is more.Access the cloud in based on two ways one is admin delegation and second one is role delegation. In role delegation the existing clients became a role and provided access of cloud instead of administrator.
The tabulated below from the table. Here can see that role delegation is efficient.In role delegation already existing client give the authentication to new clients to access cloud. In admin delegation authentication is given to existing clients.

Table 7.1 Performance analysis

No of Clients	Admin delegation	Role delegation
4	2	4
5	4	5
10	7	10
15	10	15

7 End Sections

7.1 ACKNOWLEDGMENTS

We have taken efforts in this journal. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend my sincere thanks to all of them. We are highly indebted to Directors, principal, and member of St.Joseph University In Tanzania. Our thanks and appreciations also go to my colleagues in developing the Journal and people who have willingly helped me out with their abilities.

7.2 CONCLUSION

We have presented the RBCD model and its associated cryptographic operations for convenient verification of delegation chains. RBCD enables a role member to create delegations based on the need of collaboration; in the meantime, anyone can verify a delegation chain without the participation of role administrators. Our protocol is general and can be realized by any signature scheme. We have described a specific realization with a hierarchical certificate-based encryption scheme that gives delegation compact credentials.
As future work, we intend to further improve and simplify our mechanisms providing administrator alert, when administrator login in to the system. The alerts such as who are the clients are login and how much data they forward to others.

7.3 FUTURE ENHANCEMENT

The project has covered almost all the requirements. Client's requirements and improvements can easily be done since the coding is mainly structured or modular in nature. Improvements can be appended by changing the various delegation properties, delegation chin extension. Project can extend to provide more security related concept rather than using hashing techniques. The disk space is available to provide all type of services in web concepts.

REFERENCES

[1] P.S. Barreto, H. Y. Kim, B. Lynn, and M. Scott, "Efficient algorithms for pairing-based cryptosystems," in *Proc. Crypto,* vol. 2442, *Lecture Notes in Computer Science,* 2002, pages 354–368.

[2] R. Bhatti, J. Joshi, E. Bertino, and A. Ghafoor, "X-GTRBAC Admin: A decentralized administration model for enterprise-wide access control," in *Proc. ACM SACMAT,* 2004, pages 78–86.

[3] M. Blaze, J. Feigenbaum, and A. D. Keromytis, "KeyNote: Trust management for public-key infrastructures," in *Proc. Security Protocols Int. Workshop,* 1998, pages 59–63.

[4] M. Blaze, J. Feigenbaum, and J. Lacy, "Decentralized trust management," in *Proc. IEEE Symp. Security Privacy,* May 1996, pages 164–173.

[5] D. Boneh and M. K. Franklin, "Identity-based encryption from the Weil pairing," in *Proc. CRYPTO,* vol. 2139, *LNCS,* 2001, pages 213–229.

[6] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps," in *Proc. Eurocrypt,* 2003, pages 416–432.

[7] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "A survey

ISSN 2278-7763

of two signature aggregation techniques," *CryptoBytes,* vol. 6, no. 2, pages 1–9, 2003.

[8] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," in *Proc. Asiacrypt,* vol. 2248, *LNCS,* 2001, pages 5 14–532.

[9] D. Clarke, J.-E. Elien, C. Ellison, M. Fredette, A. Morcos, and R. L. Rivest, "Certificate chain discovery in SPKI/SDSI," *J. Comput. Security,* vol. 9, no. 4, pages 285–322, Jan. 2001.

[10] P. Devanbu, M. Gertz, C. Martel, and S. Stubblebine, "Authentic third-party data publication," *J. Comput. Security,* vol. 11, no. 3, pages 291–314, 2003.

