ISSN No. 0976-5697

Volume 3, No. 4, July- August 2012

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Collaborating Services of Dynamic Authentication in Service-Oriented Architecture - Based Business Processes

Apparao R*
Assistant professor
Vardhaman College of Engineering
Kacharam, Shamshabad – 501 218,
Hyderabad, Andhra Pradesh, India
Apparao1000@gmail.com

J Naga Padmaja
Assistant Professor
khammam Institute of Technology & Science
Khammam– 5007001,
Andhra Pradesh, India
srija26@gmail.com

A Krishna Chitanya
Associate Professor
Vardhaman College of Engineering
Kacharam, Shamshabad – 501 218,
Hyderabad, Andhra Pradesh, India
Chaituit2004@gmail.com

C Satya Kumar
Assistant Professor(Sr. Grade)
Vardhaman College of Engineering
Kacharam, Shamshabad – 501 218,
Hyderabad, Andhra Pradesh, India.
csatyakumar@gmail.com

Abstract: Modern distributed business applications are embedding an increasing degree of automation and dynamism, from dynamic supply-chain management, enterprise federations, and virtual collaborations to dynamic service interactions across organizations. Such dynamism leads to new challenges in security and dependability. In Service-Oriented Architecture (SOA), collaborating services may belong to different security realms but often need to be engaged dynamically at runtime. If a cross-realm authentication relationship cannot be generated dynamically at runtime between heterogeneous security realms, it is technically difficult to enable dynamic business processes through secure collaborations between services. A potential solution to this problem is to generate a trust relationship across security realms so that a user can use the credential in the local security realm to obtain the credentials to access resources in a remote realm. However, the process of generating such kinds of trust relationships between two disjoint security realms is very complex and time consuming, which could involve a large number of extra operations for credential conversion and require collaborations in multiple security realms. In this paper, we propose a new cross-realm authentication protocol for dynamic service interactions. This protocol does not require credential conversion or establishment of authentication paths.

Keywords: Authentication, inter-Organizational Security, Multi-party interactions, Service-Oriented Architecture, Web Services

I. INTRODUCTION

This will explain about the inter-organizational business processes, with the emerge of service-oriented technologies; dynamism and flexibility are becoming the core characteristics of modern inter-organizational business processes, such as business application integration, distributed auction services. With in service oriented architecture (SOA), an organization may encapsulate and publish its applications as services, and select and interact at runtime with the services provided by the other organizations However, for both user and vendor organizations, this raises immediate problems of security, trust and dependability [3].Until these problems are addressed and solved satisfactorily, the potential of automatic inter organizational business processes will be severely restricted. In dynamic and distributed environment it is often difficult for a complex business process to follow a static business specification. The execution order of its activities at runtime usually unpredictable, and some occasions, the actual execution of a process can be "one-of--a kind".

The applications and services involved in a complex business process are typically heterogeneous, provided by different organizations. Since each organization has its own security mechanisms and policies to protect its local resources, the business process has to operate amongst multiple Message confidentiality: Message confidentiality ensures the sender that the message can be read only by an intended receiver.

Message authentication*:* Message authentication ensures the receiver that the message was sent by a specified sender and the message was not altered en route. To provide these two functions, one-time session keys need to be shared among communication entities to encrypt and authenticate messages. Thus, before exchanging communication messages, a key establishment protocol needs to distribute one-time secret session keys to all participating entities. The key establishment protocol also needs to provide confidentiality and authentication for session keys.

There are two types of key establishment protocols: key transfer protocols and key agreement protocols [9], [10]. Key transfer protocols rely on a mutually trusted key generation center (KGC) to select session keys and then transport session keys to all communication entities secretly. Most often, KGC encrypts session keys under another secret key shared with each entity during registration. In key agreement protocols, all communication entities are involved to determine session keys. The most commonly used key agreement protocol is Diffie-Hellman (DH) key agreement protocol. In DH protocol, the session key is determined by exchanging public keys of two communication entities. Since the public key itself does not provide any authentication, a digital signature can be

attached to the public key to provide authentication. However, DH public key distribution algorithm can only provide session key for two entities; not for a group more than two members

In this approach, the confidentiality of group key is ensured using any encryption algorithm which is computationally secure. Our protocol uses secret sharing scheme to replace the encryption algorithm. A broadcast message is sent to all group members at once. The confidentiality of group key is information theoretically secure. In addition, the authentication of broadcasting message can be provided as a group authentication. This feature provides efficiency of our proposed protocol.

II. EXISTING SYSTEM

A. A Secure and Efficient Conference Key Distribution System:

We present a practical interactive conference key distribution system based on public keys, which is `proven' secure provided the Diffie-Hellman problem is intractable. The system authenticates the [5][6] users and allows them to compute their own conference key. A certain number of interactions is required, but the number of rounds is independent of the number of conference users. All users involved perform the same amount of computation and communication. Our technique for authentication can be extended and used as the basis for an authentication scheme which is `proven' secure against any type of attack, provided the discrete logarithm problem is intractable.

B. Scalable Protocols for Authenticated Group Key Exchange:

We consider the problem of authenticated group key exchange among n parties communicating over an insecure public network [2][3]. A number of solutions to this problem have been proposed; however, all prior provably-secure solutions do not scale well and, in particular, require O(n) rounds. Our main contribution is the rst scalable protocol for this problem along with a rigorous proof of security in the standard model under the DDH assumption; our protocol uses a constant number of rounds and requires only O (1) \full" modular exponentiations per user. Toward this goal (and adapting work of Bellare,Canetti, and Krawczyk), we trust present an efficient compiler that transforms any group key-exchange protocol secure against a passive eavesdropper to an authenticated protocol which is secure against an active adversary who controls all communication in the network. This compiler adds only one round and O(1) communication (per user) to the original scheme. We then prove secure | against a passive adversary | a variant of the two-round group key-exchange protocol of Burmester and Desmedt. Applying our compiler to this protocol results in a provably-secure three-round protocol for authenticated group key exchange which also achieves forward secrecy.

DH public key distribution algorithm can only provide session key for two entities, not for a group more than two members. The main disadvantage of this approach is to require every user to store a large size of secrets.

III. OBJECTIVE OF THE PROTOCOL

In this paper, the[1] confidentiality of group key is ensured using any encryption algorithm which is computationally secure. And our proposed protocol uses secret sharing scheme to replace the encryption algorithm. A broadcast message is sent to all group members at once. The confidentiality of group key is information theoretically secure. In addition to this, the authentication of broadcasting message can be provided as a group authentication.

A. Model:

Group key transfer protocol relies on one trusted entity, KGC, to choose the key, which is then transported to each member involved. Each user is required to register at KGC for subscribing the key distribution service. The [10] KGC keeps tracking all registered users and removing any unsubscribed users. During registration, KGC shares a secret with each user. In most key transfer protocol, KGC encrypts the randomly selected group key under the secret shared with each user during registration and sends the cipher text to each group member separately. An authenticated message checksum is attached with the cipher text to provide group key authenticity. In this approach, the confidentiality of group key is ensured using any encryption algorithm which is computationally secure. Our protocol uses secret sharing scheme to replace the encryption algorithm. A broadcast message is sent to all group members at once. The confidentiality of group key is information theoretically secure. In addition, the authentication of broadcasting message can be provided as a group authentication. This feature provides efficiency of our proposed protocol.

IV. SECURITY ANALYSIS

We first consider two types of adversaries in our proposed protocol, insider and outsider.

Adversaries can be categorized into two types. The first type of adversaries is outsiders of a particular group. The outside attacker can try to recover the secret group key belonging to a group that the outsider is unauthorized to know. This attack is related to the confidentiality of group key. In our proposed protocol, anyone can send a request to KGC for requesting a group key service. The outside attacker may also impersonate a group user to request a group key service. In security analysis, we will show that the outside attacker gains nothing from this attack since the attacker cannot recover the group key.

The second type of adversaries are insiders of a group who are authorized to know the secret group key, but inside attacker attempts to recover other member's secret shared with KGC. Since any insider of a group is able to recover the same group key, we need to prevent inside attacker knowing other member's secret shared with KGC.

A. Security Goals:

The main security goals for our group key transfer protocol are: Key freshness, key confidentiality, and key authentication [4] [10].

Key freshness is to ensure that a group key has never been used before. Thus, a compromised group key cannot cause any further damage of group communication. Key confidentiality is to protect the group key such that it can only be recovered by authorized group members, but not by

any un-authorized user. Key authentication is to provide assurance to authorized group members that the group key is distributed by KGC, but not by an attacker.

In our protocol [7] [8], we only focus on protecting group key information broadcasted from KGC to all group members. The service request and challenge messages from users to KGC are not authenticated. Thus, an attacker can impersonate a user to request for a group key service. In addition, attacker can also modify information transmitted from users to KGC without being detected. We need to analyze security threats caused by these attacks. So, we will conclude that none of these attacks can successfully attack to authorized group members since attackers can either obtain the group key or share a group key with authorized group members. User/message authentication and key confirmation can be easily incorporated into our protocol since each user has shared a secret key with KGC during registration.

V. PROPOSED PROTOCOL

Group key transfer protocol relies on one trusted entity, KGC, to choose the key, which is then transported to each member involved. Each user is required to register at KGC for subscribing the key distribution service. The KGC keeps tracking all registered users and removing any unsubscribed users. During registration, KGC shares a secret with each user. In most key transfer protocol, KGC encrypts the randomly selected group key under the secret shared with each user during registration and sends the ciphertext to each group member separately. An authenticated message checksum is attached with the ciphertext to provide group key authenticity. In this approach, the confidentiality of group key is ensured using any encryption algorithm which is computationally secure. Our protocol uses secret sharing scheme to replace the encryption algorithm. A broadcast message is sent to all group members at once. The confidentiality of group key is information theoretically secure. In addition, the authentication of broadcasting message can be provided as a group authentication. This feature provides efficiency of our proposed protocol.

Each user needs to register at KGC to subscribe the group key transfer service and to establish a secret with KGC. Thus, a secure channel is needed initially to share this secret with each user. Later, KGC can transport the group key and interact with all group members in a broadcast channel. The confidentiality of group key distribution is information theoretically secure, that is, the security of this transfer of group key to each group member does not depend on any computational assumption. The authentication of the group key is achieved by broadcasting a single authentication message to all group members.

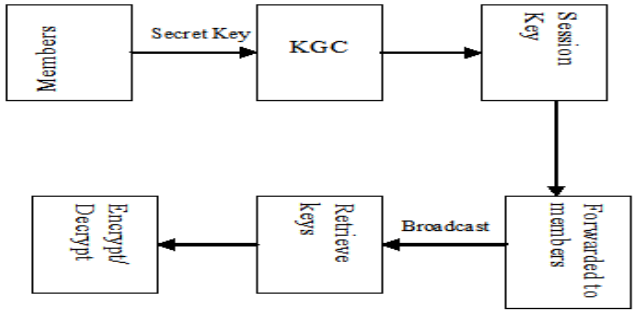

Figure 1 Architecture of the protocol

a) Initialization of KGC: The KGC randomly chooses two safe primes p and q (i.e., primes such that p1 = (p-1)/2 and q1 = (q-1)/2 are also primes) and compute n = pq. n is made publicly known.

b) User Registration: Each user is required to register at KGC for subscribing the key distribution service. The KGC keeps tracking all registered users and removing any unsubscribed users. During registration, KGC shares a secret (xi yi) with each user Ui, where xi yi E z.

c) Key Generation and Distribution: Upon receiving a group key generation request from any user, KGC needs to randomly selects a group key and access all shared secrets with group members.[10] KGC needs to distribute this group key to all group members in a secure and authenticated manner. All communication between KGC and group members are in a broadcast channel. For example,

We assume that a group consists of t members,{u1,u2…ut} and shared secrets are (xi yi) for I =1; . . . ; t.

The key generation and distribution process contains five steps.

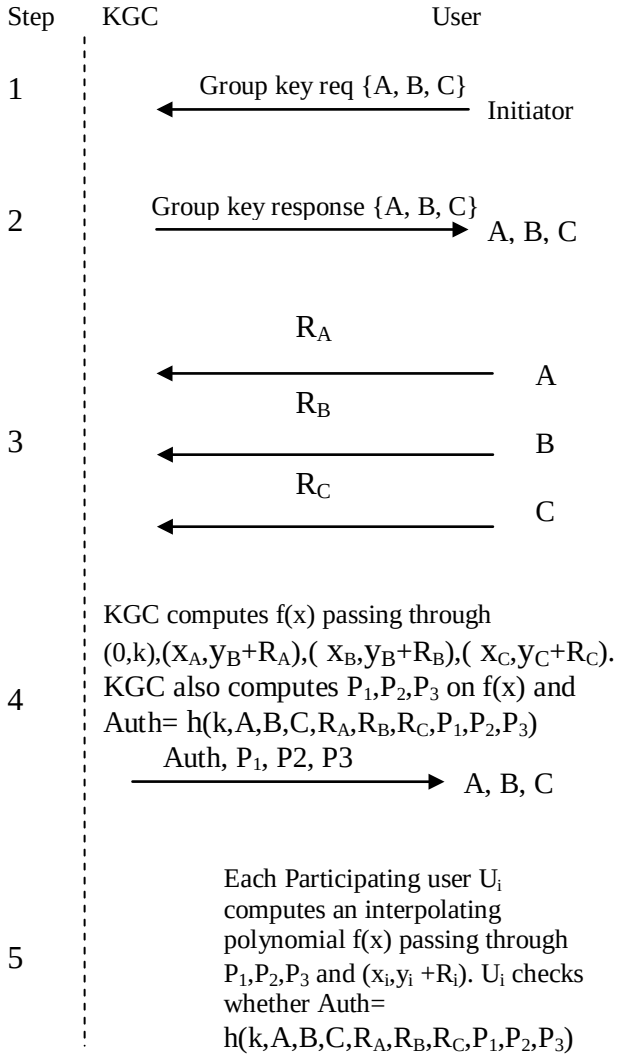

Figure 2 Key Generation and Distribution Process

Step 1. The initiator sends a key generation request to KGC with a list of group members as (U1; U2; . . . ; Ut).
Step 2. KGC broadcasts the list of all participating members, (U1; U2; . . . ; Ut), as a response.
Step 3. Each participating group member needs to send a random challenge, R1 E Z_to KGC.
Step 4. KGC randomly selects a group key, k, and generates an interpolated polynomial
Step 5. For each group member, Ui, knowing the shared secret (x_i,y_i©R_i) and t additional public points, Pi, for I = 1; . . . ; t, on f(x), is able to compute the polynomial f(x) and recover the group key k = f(0). Then, Ui computes h (k, U1, U2….Ut, R1…..RT, P1…..Pt) and checks whether this hash value is identical to Auth. If these two values are identical, Ui authenticates the group key is sent from KGC.

VI. CONCLUSION

We have proposed an efficient group key transfer protocol based on secret sharing. Every user needs to register at a trusted KGC initially and preshare a secret with KGC. KGC broadcasts group key information to all group members at once. The confidentiality of our group key distribution is information theoretically secure. We provide group key authentication. Security analysis for possible attacks is included.

VII. REFERENCES

[1]. [1] G.R. Blakley, “Safeguarding Cryptographic Keys,” Proc. Am. Federation of Information Processing Soc. (AFIPS'79). Nat'l Computer Conf., Vol.48,pp.313-317.

[2]. [2] S. Berkovits, “How to Broadcast a Secret,” Proc. Eurocrypt ’91 Workshop Advances in Cryptology Springer-Verlag, 1992, pp.101-113.

[3]. [3] R. Blom, “An Optimal Class of Symmetric Key Generation Systems,” Proc. Eurocrypt ’84 Workshop Advances in Cryptology, Vol. 209, Springer.

[4]. [4] “Perfectly Secure Key Distribution for Dynamic Conferences,” C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, pp. 449-455, 1998.

[5]. [5] C. Boyd, “On Key Agreement and Conference Key Agreement,” Proc. Second Australasian Conf. Information Security and Privacy, (ACISP '97), pp. 294-302.

[6]. [6] J.C. Cheng and C.S. Laih, “Conference Key Agreement Protocol with Non Interactive Fault-Tolerance over Broadcast Network, vol. 59 no. 6,pp. 842-846,2010.

[7]. [7] A. Fiat and M. Naor, “Broadcast Encryption,” Proc. 13th Ann. Int’l Cryptology Conf. Advances in Cryptology, pp. 480-491, 1994.

[8]. [8] H. Harney, C. Muckenhirn, and T. Rivers, “Group Key Management Protocol (GKMP) Architecture, 1997. RFC 2094.

[9]. [9] K.H. Huang, Y.F. Chung, H.H. Lee, F. Lai, and T.S. Chen, “A Conference Key Agreement Protocol with Fault-Tolerant Capability,” Computer Standards and Interfaces. Volume 31,2009.

[10]. [10] J. Katz and M. Yung, “Scalable Protocols for Authenticated Group Key Exchange,” J. Cryptology. LNCS 2729, pp.110-125, Springer-Verlag, 2003.

