Design Security for Internet-Based Workflow Management Systems Adopting Security Agents

Myeonggil Choi
National Security Research Institute
161, Kajong-dong, Yuseong-gu, Daejeon, 305-350
Korea

Abstract: - With the expansion of WFMS(workflow management systems) across the Internet, collaboration among enterprises increase. The security problems of Intranet-based WFMS, which is operated on Intranet, are critical, and the effectiveness of Internet-Based WFMSs is deeply influenced by security. For the security of Internet-Based WFMSs, this study introduces the Login Agent, the Security Agent, and Task Agents and presents the flexible security policy and security mechanisms. This paper analyzes security requirements for the Internet-Based WFMSs. Based on the security requirements, this study suggests the security architecture and security mechanisms for the Internet-Based WFMSs.

Key-Words: - Internet-based workflow management system, Security, Agent, Security requirements, Architecture, Security Mechanisms

1 Introduction

As the popularity of workflow management systems (WFMSs) continues to grow and invade more and more application domains such as healthcare, military, electronic commerce, security becomes a quite complex problem [8,9.10,13,14,19].

Due to the different IT environments and technology preferences, enterprises may introduce heterogeneous WFMSs that have different security systems [1,15]. The Internet makes it possible for an enterprise to collaborate with other enterprises but collaboration among enterprises increases the vulnerability of WFMSs. In addition to this problem, the components of WFMSs are deployed over large geographical areas. Therefore, the importance of security among subsystems of WFMSs should be stressed. We define Internet-Based WFMS as WFMSs collaborating with heterogeneous WFMSs through the Internet. As Internet-Based WFMSs are provided by different vendors, it is difficult to modify the security architecture and mechanisms. To overcome this problem, we adopt an agent-based approach. Agents are suitable for adoption in Internet-Based WFMS as it is easy to implement agents in heterogeneous system environments due to their independent nature. In this paper, we consider collaboration among Internet-Based WFMSs as group communication and an Internet-Based WFMS as a member of group.

This paper suggests security architecture and mechanisms, which are suitable for Internet- Based WFMS. Therefore, we introduce security features of group communication into the security mechanisms of Internet-Based WFMSs.

To design security architecture and security mechanisms for Internet-Based WFMSs, we analyze security requirements for Internet-Based WFMSs. Based on the security requirements we suggest the suitable security architecture and security mechanisms for Internet-Based WFMSs.

2 Related Works

WfMC(workflow management coalition) presents security services and security model of WFMS [19]. Elisa Bertino presents specification and enforcements of authorization constraints in WFMS [3]. This paper analyzes authorization constraints and methods to validate consistency of authorization. Vijay Atluri establishes security requirements for WFMS and shows how to adapt them to WFMS [2]. John A. Miller presents security for web-based WFMS [14]. This paper analyzes security issues, which are related to Internet-Based WFMS, focusing on secure communications and access controls.

These related works identify security problems of WFMS and present security measures that can be used in to WFMS. The above studies mainly focus on

authorization and access control in WFMS. The contributions of the above research are identification of security problems in WFMS and presentation of security solutions in WFMS.
The limitations of the above study are described below. First, the above studies mainly present authorization solutions to define roles of WFMS's participants. Second, the related works do not reflect the flexible security mechanisms of Internet-Based WFMSs.

3. An Overview of WFMS

3.1 Definitions of WFMS

It is better to briefly survey the definition of WFMS. To define WFMS, we have to review the definition of workflow. Workflow is an activity that achieves a common business objective. A workflow separates the various activities of a given organizational process into a set of well defined tasks.

According to the definition of WfMC, a WFMS is a system that supports process specification, enactment, monitoring, coordination, and administration of workflow processes through the execution of software [4,17,18]. V. Atluri defines WFMS as follows: WFMS can be considered as a computer automated infrastructure where a group of people participates together to achieve a common goal following some predefined rules and task assignments [6,7].

WFMSs are very complex pieces of software. WFMSs support advanced capabilities such as security, reliability, high performance, and transactional capabilities in heterogeneous and distributed environments in a manner that facilities interoperability with other WFMSs as well as other software systems [1,13,14].

3.2 An Example of WFMS

A simple example of Internet-Based WFMSs is illustrated in Fig.1 The collaboration between the Internet-Based WFMSs of a heavy-industry company and the Internet-Based WFMS of a steel company is presented. Although the MIS WFMS, the Production WFMS and the Global WFMS of the heavy industry company are located in the same location, the WFMS of the procurement department is separated from other WFMSs. To order material, the WFMS of procurement department sends its bills to the Global WFMS. The Global WFMS of the heavy industry company orders material bills to the WFMS of the steel company. The WFMS of the steel company forwards its order to the Order WFMS and the Delivery WFMS. On receiving the order, the Order WFMS and the Delivery WFMS process the bills of the heavy industry company.

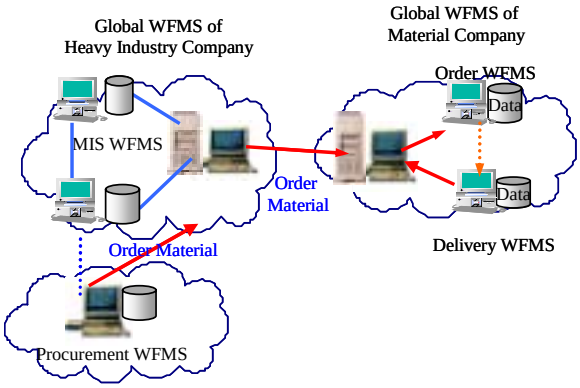

Fig.1 An example of Internet-Based WFMS

4. Security Services

Before analyzing security requirements for Internet-Based WFMSs, we review the general security services that are introduced into WFMS. The general security services are described below [29, 60].

4.1 Authentication

In most domains, it is critically important to limit access data and tasks for the authorized individuals. Before any subjects begin participating in a WFMS, they must be identified and authenticated. It is critical to identify who the user is. Security mechanisms for authentication include password, biometrics and digital signatures.

During an Internet-Based WFMS interoperates other heterogeneous Internet-Based WFMSs, a series of identification and authentication commands can be assured of their mutual identities. In particular where such systems are operating in an asynchronous manner, this may require special provisions within the interoperability protocol. Communication path is exposed to potentially insecure nodes and public data communication infrastructure that is exposed to hacking.

4.2 Access Control

Only authorized users should be allowed to execute given tasks or access given data objects. Access control must be effective in the sense that no

unauthorized users should be granted access, while at the same time no authorized user should be denied access. Security mechanisms of access control include Access Control Lists (ACLs), Role-Based Access Control (RBAC) and Multilevel Security (MLS).

4.3. Data Confidentiality

For most network-based information systems, it is important that message not be stolen or snooped. Messages are likely to contain confidential information. To effectively guarantee this, sufficiently strong encryption mechanisms should be used. With strong encryption mechanisms, no one will be able to decipher data flowing over the net except intended recipients. In practice there is a trade-off between the risk of an encrypted message being cracked and the performance of systems.

4.4. Integrity

Even if stolen message which is encrypted is useless, a malicious user may still cause problems. If the malicious user intercepts message, modifies it and then sends it along, the message will no longer be valid. To ensure the message is not tampered with, a hash function may be applied to the message forming a digest. This digest is sent along with message. If the same hash function applied to the received message disagrees with the digest, the message is assumed to have been tampered with.

4.5. Non-repudiation

In Internet-Based WFMSs, it is important to keep track of all accesses. If a user gains access illegally or an user makes a significant mistake, administrator must be able to quickly find out what has been done and by whom. Through sufficiently effective and reliable authentication and tracking, it should be possible to undeniably establish the identity of individual responsible for executing a given task. WFMSs must keep track of who executed each task and when it was executed.

5. Security Requirement for Internet-Based WFMSs

Before establishing security requirements of Internet-Based WFMSs, we should consider the scalability of security measures, comfortable management of security measures, and interoperability of security measures [6]. The security measures of Internet-Based WFMSs are able to support more users than the security measures of WFMS within an organization. The Internet-Based WFMS interact with globally distributed WFMSs. As the number of Internet-Based WFMS may increase, the security measures of the Internet-Based WFMSs could be expanded easily.

The second consideration is the comfortable management of security measures. Internet- Based WFMSs, which support all the processes of workflow, are composed of many intrinsically complex functions. The manager of Internet-Based WFMS may be accustomed to many complex functions but may not be accustomed to security measures. The comfortable management of security measures should be considered.

The third consideration is interoperability among Internet-Based WFMSs. Due to their different environments and needs, enterprises establish different security policies and implement different security measures. To effectively protect their information resources, the enterprises choose different authentication mechanisms, different crypto algorithms, and different secure communication mechanisms. To securely communicate among enterprises that have different security measures, the interoperability among enterprises should be supported.

Based on these considerations, we establish four security requirements of Internet-Based WFMSs. First, the security architecture should be less affected by changes in Internet-Based WFMSs' functions. As Internet-Based WFMSs that reflect the workflow of an enterprise frequently change, the security architecture should be independent of Internet-Based WFMSs. The security measures should be designed independently of the workflow of Internet-Based WFMSs. The independence of Internet-Based WFMSs also supports the scalability of security measures. Because the security measures are independent of Internet-Based WFMSs, it can be scaled reflecting the number of users and the performance of Internet-Based WFMSs.

Second, the security architecture should be designed to be flexible. The components of an Internet-Based WFMS can be geographically distributed. With the change in the workflow of an enterprise, the components of an Internet-Based WFMS can be changed. As the flexibility of security architecture would be less reflected by the change of Internet-Based WFMSs, the management efforts of Internet-Based WFMSs manager can be minimized.

Third, the performance of Internet-Based WFMSs should be guaranteed. Internet-Based WFMSs process important data, some of which must be processed quickly. If Internet-Based WFMSs do not comply with demands of others quickly, the enterprise may lose business opportunities. After security measures are implemented in Internet-Based WFMSs, the performance tends to be lower than ever before. The business opportunities of Internet-Based WFMSs can be influenced by performance. Therefore, the security architecture of Internet-Based WFMSs should be designed reflecting the performance of transactions.

Forth, the security architecture of Internet-Based WFMSs should be designed independently of other external security infrastructure. Each enterprise adopts different security infrastructure which is composed of certification infrastructure and public key infrastructure. Different security infrastructure makes it difficult to interoperate among Internet-Based WFMSs.

To reflect security requirements for Internet-Based WFMSs, we adopt an agent approach. It is easy to implement agents in heterogeneous system environments due to their independent nature. To implement security measures of Internet-Based WFMSs, we add agents that perform security functions to Internet-Based WFMSs.

6. Design of Security Architecture for Internet-Based WFMSs

In this section, we suggest security architecture for Internet-Based WFMSs. To access an Internet-Based WFMS, users must be authenticated. As Fig. 2 shows users log in with the Login Agent. The Login Agent consults the Security Agent to authenticate users. After being authenticated, users are able to collaborate with Task Agents according to their permitted roles. To protect collaboration among Task Agents, the communication channel should be encrypted.

The Login Agent consults the Security Agent in order to authenticate users. To increase the performance of the Security Agent, multiple Security Agents may be utilized and the Login Agents may cache some security information. The Task Agents, which interoperate with other Internet-Based WFMSs, can be physically distributed at remote areas. The detailed authentication process is suggested in later. The Security Agent stores authentication information, logs, and roles of users. The Security Agent transfers role assignment to Task Agents.

Another function of the Security Agent is to manage users' key. Without appropriate key management to generate, disseminate and destroy keys, security will become either unwieldy or ineffective. The manager of an Internet-Based WFMS is able to use a utility provided by the Security Agent to produce unique keys for users. In this suggested architecture, the processes of key generation and key distribution are not included.

7. Design of Security Mechanisms for for Internet-Based WFMSs

In this section, we suggest security mechanisms for Internet-Based WFMSs. We suggest security mechanisms for providing the following functions; authentication, member join, session key and group membership distribution, secure communication, failure detection, and member leave. Before

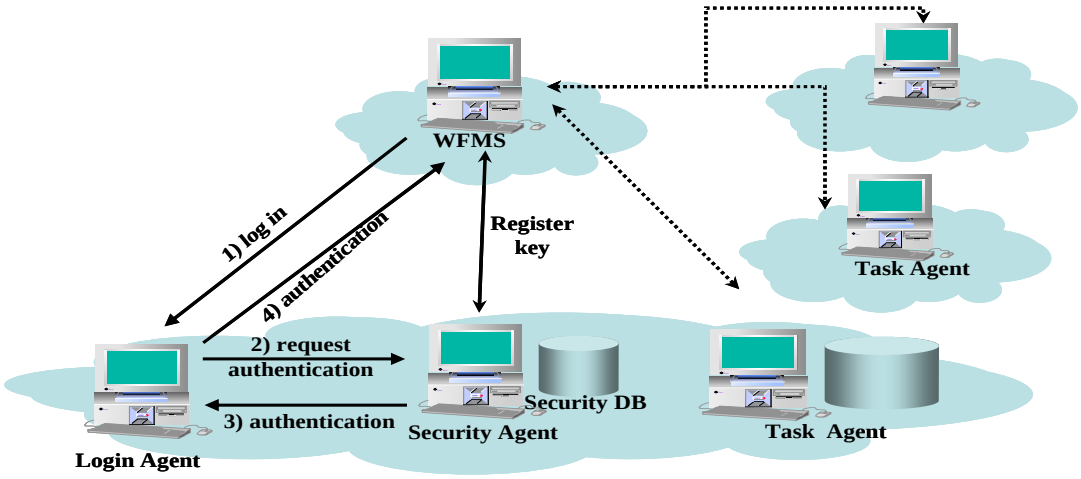

Fig. 2 The Security Architecture of Internet-Based WFMSs

describing the micro-protocols, we explain the principals in the protocols and the notation used. The security mechanism used in secure group communication is modified to be suitable for Internet-Based WFMSs [14].

The Security Agent performs as Trusted Third Party (TTP) which provides the security mechanisms for the Login Agent to authenticate a joining Internet-Based WFMS. Each potential Task Agent (A) has a shared secret K_A registered with the Security Agent. This secret key is generated and registered with the Security Agent before each potential Internet-Based WFMS who attempts to join any session.

The identity of Login Agent is known in advance to all potential Task Agents that participate in collaboration. We assume an out of band method for registering these keys and for informing everyone about the identity of the Login Agent.

In this protocol descriptions, we use the term LA to refer to the identity of the Login Agent, A to refer to a potential Task Agent. $\{X\}_k$ refers to message X encrypted under the key K. The view identifier, g, is used to uniquely tag the changing group views of collaboration. As previous section states, a group means a collection of Internet-Based WFMSs to participate in collaboration. In this architecture, the Task Agent participates in collaboration. The term SK_g refers to a session key in view g, and SK_{g+1} for the next view $g+1$. The term I, possibly with a subscript, refers to a nonce value. Key distribution protocols based on Leighton-Micali define a term π_{AB}, called a pair key, used to support secret communication between collaborating Task Agents A and B. Derived from the pair key, the Login Agent and a Task Agent (A) maintain a shared secret key $\sigma_{LA,A}$. A [policy block] is distributed by the Login Agent to each potential Task Agent during the authentication process.

The Login Agent creates an asymmetric key pair (P_{UG}, P_{rG}) during initialization. The public key (P_{UG}) is delivered to a Task Agent during authentication process. The public key is used to reduce the cost of sending secure heartbeats from the Login Agent. Note that no certificate distribution service is required in our protocol to generate or distribution the (P_{UG}, P_{rG}) asymmetric key pair.

We assume that the Task Agent willingly does not disclose its long term or session keys. All Task Agents trust Security Agent not to close their long term key.

7.1. Authentication Mechanisms

The authentication mechanisms provide facilities for the authentication of each potential Task Agents. The purposes of these mechanisms are twofold. First, the Login Agent authenticates the potential Task Agents. Second, a shared secret between the two parties is negotiated. The shared secret, called the shared secret key, later used to implement a secure channel between the two Task Agents.

This paper uses Leighton-Micali key distribution algorithm to authenticate the joining process and negotiate the shared secret [11]. The main advantage of Leighton-Micali is low cost; it uses symmetric key encryption throughout, with none of the modular exponentiation operations associated with public key cryptosystems. Public key cryptography requires significantly more computation than asymmetric algorithms.

A prospective Task Agent initiates the authentication process by sending a message to the Login Agent containing its identity and a nonce value message (1). The Login Agent then obtains the pair key π_{AB} from the Security Agent message (1), message (2). Derived from two identities and their associated long term keys, the pair key is used to establish an ephemeral secure channel between the processes. To prevent replay attacks, the Login Agent

$$(1)\ A \rightarrow LA: A, I_0 \qquad (authentication\ request)$$

$$(2)\ LA \rightarrow SA: LA, A, I_1 \qquad (pair\ key\ request)$$

$$(3)\ SA \rightarrow LA: \{[\pi_{LA,A} = \{A\}_{k_{LA}} \oplus \{LA\}_{k_A}], I_1\}_{K_{LA}} \qquad (pair\ key\ response)$$

$$(4)\ LA \rightarrow A: LA, A, \{g, A, I_{0,} I_{2,} [policy\ block], P_{UG}\}_{\sigma_{LA,A}} \qquad (authentication\ response)$$

Fig. 3 Authentication Mechanisms

verifies the encrypted nonce value I1 included in the Security Agent's response.

The Security Agent computes the shared secret key for communicating with the Task Agent (A) as follows. The Login Agent generates the value $\{A\}_{k_{LA}}$. This value is XOR-ed with the pair key $\pi_{LA,A}$ received from the Security Agent. The resulting value is a shared secret key ($\{LA\}_{k_A} = \sigma_{LA,A}$) that is used to create to a secure channel between the Login Agent and the Task Agent.

Note the each Task Agent needs not communicate with the Security Agent to obtain the shared secret key $\{LA\}_{k_A} = \sigma_{LA,A}$; it can compute it directly. The Task Agent (A) decrypts the session key with this value and validates its nonce.

After obtaining the shared secret key, the Login Agent responds with an authentication response message (4). The response contains the identities of the Login Agent and Task Agent, and a block encrypted with the shared secret key $\sigma_{LA,A}$. The encrypted block contains the view (g) and Task Agent (A) and nonce value of Task Agent (I_0), policy block [policy block], and public key (P_{UG}). Upon receiving this message, the receiver decrypts the contents and verifies the nonce I_0

7.2 Join Mechanism

The join mechanism provides a Task Agent with facilities for gaining access to collaboration. The mechanism also provides measures to ensure the reliability of the join.

The potential Task Agent (A) joins group by transmitting message (5) to the Login Agent. Upon reception of the message, the Login Agent validates the nonce value (I_2) passed to the Task Agent during the authentication process. If the nonce is not valid, the join request ignored. If the nonce is valid, the new Task Agent is accepted into group.

7.3 Join Mechanism

Note that the join message is fresh. The Login Agent knows that a correct join message is fresh and was generated by the Task Agent because of the presence of the (I_2) nonce value encrypted under the shared secret key.

Mutual authentication is achieved through the verification of the secrets the Task Agent (A) and the Login Agent (LA) shared with the Security Agent (SA). The potential Task Agent must be possession of the secret shared with the Security Agent to determine the secret key shared with Task Agent (A). The Task Agent is convinced that authentication response message is fresh by validating the nonce value I_0 sent in the original request.

$$(5) A \rightarrow LA: A, \{A, I_2\}_{\sigma_{LA,A}} \quad (join\ request)$$

Fig. 4 Join Mechanism

7.4 Rekey/Group Membership Mechanisms

The Rekey/Group Membership mechanisms provide for the distribution of membership and session keys. We note the distinction between session rekeying and session key distribution. In session rekeying, all existing group members must receive a newly created session key. In session key distribution, the Login Agent transmits an existing session key.

The session key, distribution, and rekey messages message (6-1), (6-2), (6-3) all contain a group identifier (g), the latest sequence number of Login

$$(6-1)\ LA \rightarrow A: g, S_{LA,}(A, \{g, SK_g\}_{\sigma_{LA,A}})\{H(g, S_{LA}, \{A, SK_g\}_{\sigma_{LA,A}})\}_{SK_g} \quad (key\ distribution)$$

$$(6-2)\ LA \rightarrow A: g, S_{LA,}(A, \{g, SK_g\}_{\sigma_{LA,A}}), B, C, D, \cdots$$
$$, \{H(g, S_{LA}, \{A, SK_g\}_{\sigma_{LA,A}}), B, C, D, \cdots)\}_{SK_g} \quad (key / group\ membership\ distribution)$$

$$(6-3)\ LA \rightarrow group: g, S_{LA,}(A, \{g+1, SK_{g+1}\}_{\sigma_{LA,A}}), (B, \{g+1, SK_{g+1}\}_{\sigma_{LA,B}}),$$
$$\cdots, \{H(g, S_{LA}, \{A, g+1, SK_{g+1}\}_{\sigma_{LA,A}}), \cdots)\}_{SK_{g+1}} \quad (session\ rekey)$$

Fig. 5 Rekey/Group Membership Mechanisms

Agent (S_{LA}), and a MAC calculated over the entire message $H(...)$. The group identifier and sequence number identify the current group context. The MAC ensures integrity of the message.

Session keys are distributed via session key block $(A,\{g, SK_g\}_{\sigma_{LA,A}})$. The each block of the intended Task Agent is identified by the Task Agent identifier (A). The remainder of the block is encrypted using the shared secret key $\sigma_{LA,A}$. If group identifier is decrypted by the receiver correctly, the Task Agent is assured that the block was created by the Login Agent.

Message (6-1) contains a session key block for one Task Agent. Message 6b contains a session key block for one Task Agent and enumerates the current Task Agent (B, C, D, …). In message (6-3), a session key block is generated for each Task Agent. Group membership in message (6-3) is extracted from the session key blocks.

Rekeying is performed by the transmission of message 6c. The Login Agent caches the shared secret keys, so creating this message is fast. Using the cached, shared secret key, the receivers of this message extract the session key out of the session key block and begin using it immediately. The size of this message grows linearly with group size, and is potentially large. Therefore, the size of the message is large by its nature, not as a side effect of our design.

7.5 Secure Communication Mechanisms

The secure communication mechanisms provide for the transmission of the application level traffic. Each application level message is secured using cryptographic keys distributed by the rekey/membership mechanisms, or through the use of external public key certificates.

The format of application message is dependent on the type of messaging policy. We achieve message integrity through Message Authentication Code (MAC) and confidentiality by encrypting under the session key. Message (7-1) shows the format of a message with integrity only, message (7-2) shows confidentiality only, and (7-3) shows a message with both integrity and confidentiality.

A MAC is generated by encrypting a hash of message data under the session key. A receiver confirms the MAC by decrypting and verifying the hash value. If the hash is correct, the receiver is assured that message has not been modified by some entity external to the group.

Sender authenticity message (7-4) is achieved by digital signature. The signature is generated using the private key exponent associated with the sender's certificate and verify the signature using the associated public key. Note that a byproduct of the use of digital signature is message integrity.

7.6 Failure Detection Mechanisms

Internet-Based WFMSs requires the system to tolerate attacks in which an adversary prevents delivery of rekeying messages from the Login Agent to the entire Task Agents. In such a case, some Task Agents will continue to have an old session key. A security risk exists if the old key is compromised. Also, for accurate group membership, it may be necessary for the Login Agent to be able to detect fail-stop failures of Task Agents.

We provide secure heartbeat message as the mechanisms to detect failed processes. The Login Agent detects failed processes through member heartbeats message (8). When some number of the Task Agent's heartbeats message are not received by the Login Agent, the Task Agent is assumed failed and expelled from collaboration.

The Task Agent that participating in group confirms that the Login Agent is still operating by receiving Login Agent's secure heartbeat messages (9). When heartbeat messages of the Login Agent are not received, the Task Agent can assume that the Login

$$(7-1)\ A \rightarrow group : g, A, [message], \{H(g, A, message)\}_{SK_g} \quad (with\ integrity)$$

$$(7-2)\ A \rightarrow group : g, \{A, [message]\}_{SK_g} \quad (with\ confidentiality)$$

$$(7-3)\ A \rightarrow group : g, \{A, [message], H(g, A, message)\}_{SK_g} \quad (with\ intergrity\ and\ confidentiality)$$

$$(7-4)\ A \rightarrow group : g, A, [message], H(g, A, message)\}_{C_A} \quad (with\ sender\ authenticity)$$

Fig. 6. Secure Communication Mechanisms

$$(8)\ A \rightarrow LA: g, S_A, H(g, S_A)_{\sigma_{LA,A}} \quad (member\ heartbeat)$$

$$(9)\ LA \rightarrow group: g, S_{LA}, H(g, S_{LA})_{P_{rG}} \quad (session\ heartbeat)$$

$$(10)\ A \rightarrow LA: g, A \quad (key\ retransmit\ message)$$

Fig. 7. Failure Detection Mechanisms

Agent failed.

Heartbeat message serves a dual purpose. In addition to failure detection, Task Agents use the heartbeats to ensure that they have the most current group state. The sequence number of the Login Agent ensures the heartbeat is fresh. The presence of the group identifier allows a group member to be certain that they are using the most recent session key. The heartbeats are encrypted to ensure that an adversary cannot fake heartbeats. Without these protections, an adversary may by able to prevent delivery of new session keys and trick members from continuing to transmit under an old session key indefinitely.

A Task Agent that fails to receive current session key or group membership information can attempt to recover by sending a key retransmit message (10). The key retransmit message indicates to the Login Agent that the Task Agent wishes to get the most key/group membership distribution message (6-1), (6-2), or (6-3) in response to the key retransmit message. In this case, the process will be able to recover by installing the most recent session key and group membership.

The goal of failure detection mechanisms at this level only provides mechanisms for reliable detection of Login Agent's failure and not recovery from its failure. Mechanisms for detection of failure can be used to implement recovery algorithms using primary backup or replicated approaches at higher levels.

7.7 Leave Mechanisms

This mechanism provides an interface for a Task Agent to gracefully exit the group. A Task Agent sends message 11 to indicate that it is exiting the group.

The leave mechanism has a secondary purpose. Using the micro-protocol, a Task Agent may request the ejection of other Task Agents from the collaboration. To request an ejection, the requester places the identity of the Task Agent in the $\{g, B\}_{SK_g}$ block (as B) The Login Agent receiving a message with this format will eject the Task Agent in accordance with some local policy.

$$11.\ A \rightarrow LA: A, \{g, A, \{g, B\}_{SK_g}\}_{\sigma_{LA,A}} \quad (leave\ request)$$

Fig. 8 Leave Mechanism

8. Conclusion

For workflow to be useful in domains in which high security is effective, Internet-Based WFMSs must combine established approaches to security with ways that achieve the desired level of security while maximizing usability and minimizing cost.

This paper analyzes security requirements for Internet-Based WFMSs. Based on the security requirements for Internet-Based WFMSs, we suggest security architecture and security mechanisms for Internet-Based WFMSs. To implement the security architecture of Internet-Based WFMSs, we adopt the Login Agent, the Security Agent, and the Task Agent. We use agents because it is easy to implement them in heterogeneous system environments due to their independent nature.

We present a flexible security mechanisms based on micro-protocols. In particular, the security mechanisms provide authentication mechanisms, secure communication mechanisms, membership mechanisms, and rekey mechanisms for Internet-Based WFMSs. The feature of mechanisms is that it is very flexible and can be used in Internet-Based WFMS environments.

References:

[1] Mike Anderson, "Workflow Interoperability – Enabling E-Commerce", *WfMC White Paper*, 1999.

[2] Vijay Atluri, "Security for Workflow Systems", *Information Security Technical Report*, Vol.6, No.2, 2001, pp.59-68.

[3] R. Baskerville, "Information System Security Design Methods: Implementation for Information

Systems Development", *ACM Computing Survey*, Vol.5, No.4, 1993, pp.375-414.

[4] Elisa Bertin, Elena Ferrari, "The Specification and Enforcement of Authorization Constraints in Workflow Management Systems", *ACM Transactions on Information Systems and System Security*, Vol.2, No.1, February, 1999, pp.65-104.

[5] N.G.Duffield, P.Goyal, A.Greenberg, P.Mishra, K.K.Ramakrishnan, and J.E. van der Merwe, "A Flexible Model for Resource Management in Virtual Private Networks", *ACM SiGCOMM*, 1999.

[6] Mary Ann Davidson, "Security for E-Business", *Information Security Technical Report*, Vol.6, No.2, 2001, pp.80-94.

[7] N.G. Duffield, P.Goyal, A. Greenberg, P.Mishra, Ramakrishnan, and J.E. V. der Merwe, "A Flexible Model for Resource Management in Virtual Private Network", in *ACM Sigcomm*, San Diego, Cal., USA, Aug., 1999.

[8]L.Gong, "Enclaves: Enabling Secure Collaboration over the Internet", *Proceedings of 6th USENIX UNIX Security Symposium*, 1996, pp.149-159.

[9] Ehud Gudes, Martin S.Oliver and Reind P.van de iet, "Modeling, Specifying and Implementing Workflow Security in Cyberspace", *Journal of Computer Security*, Vol.7, No.4, 1999.

[10] W-K. Huang and V.Atluri, "SecureFlow: a Secure Web-Enabled Workflow Management System", *4th ACM Workshop on Role-based Access Control*, OCT. 1996.

[11] T.Leighton, S.Micali, "Secret-Key Agreement without Public-Key Cryptography", *In Proceedings of Crypto 93*, 1994, pp.456-479.

[12] Patrick McDaniel, Peter Honeyman, "Antigone: Flexible Framework for Secure Group Communication", *Proceedings of the 8th USENIX Security Symposium*, 1999, pp.99-114.

[13] J.A.Miller, D.Palaniswami, A.P.Sheth, K.J.Kochut, and H.Singh, "WebWork:METEEOR2's Web-Based Workflow Management System", *Journal of Intelligent Information Systems, Special Issue on Workflow*, Vol.10, 1997, pp. 185-215.

[14] John A. Miller, Mei Fan, Amit P.Sheth and Krys J.Kochut, "Security in Web-Based Workflow Management Systems", *Theis, University of Georgia*, 1999.

[15] P. Muth, J. Weissenfels, G. Weikum, "What Workflow Technology Can Do for Electronic Commerce", *Proc. EUROMED NET Conference*, 1998.

[16] Rolf Oppliger, *Internet & Intranet Security*, Artech House, 1998.

[17] Workflow Management Coalition, "Workflow Reference Model", *Technical Report*, 1994.

[18] Workflow Management Coalition, "Workflow and Internet: Catalysts for Radical Change", *White Paper*, June, 1998.

[19] Workflow Management Coalition, "Workflow Security Considerations", *White Paper*, 1998.