Jurnal Teknologi

Full Paper

INTER-CONFIDENTIALITY PROTECTION OF AGENT COMMUNICATION IN MULTI-AGENT SYSTEM BASED APPLICATIONS

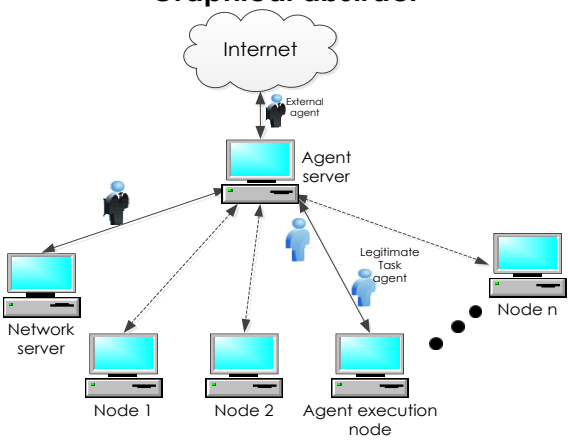
Olumide Simeon Ogunnusi, Shukor Abd Razak*, Abdul Hanan Abdullah

Department of Computer Science, Faculty of Computing, Universiti Teknologi Malaysia, 81310, UTM Johor Bahru, Johor, Malaysia

Article history
Received
15 April 2015
Received in revised form
29 September 2015
Accepted
12 November 2015

*Corresponding author
shukorar@utm.my

Graphical abstract



Abstract

Mobile agent interaction is usually vulnerable to attacks from within and outside the agent's execution environment. Also, the mobility property of mobile agents earns them the opportunity to migrate from one security domain to another. Intranet/LAN with connection to internet do, from time to time, experience agent visitation either for malicious purpose or for legitimate mission. To protect legitimate agent communication against attack by visiting agent, we propose a technique that restricts migration of the visiting agent and isolate it to a neutral host where its mission could be achieved. We refer to this technique as restriction-based access control mechanism (ResBAC). The proposed mechanism employs **certificate authentication**, **re-defining visiting agent itinerary path** and **visiting agent isolation** to accomplish the aforementioned objective. The performance of the proposed mechanism is evaluated using scenarios to determine the strength of the mechanism in term of its ability to protect agent communication against the three major threats: man-in-the-middle attack, replay attack, and passive eavesdropping.

Keywords: Agent communication, agent isolation, certificate authentication, type-space, access control model, security domain

1.0 INTRODUCTION

The mobility property of mobile agents earns them the opportunity to migrate from one security domain to another. Intranet/LAN with connection to internet do, from time to time, experience agent visitation either for malicious purpose or for legitimate mission. An occasion like this demands that the visited network domain be secure to forestall and check the possible malicious behaviour of the visiting agent. This could be achieved using an authorization policy. Some authorization policies known in literature are Discretionary Access Control (DAC), Role Based Access Control (RBAC), Mandatory Access control (MAC) [1], and Privacy-aware Role Based Access control (P-RBAC) [2]. Not until now, there is none of these policies that isolates and confines the activities of visiting agent to a neutral host such that it is deprived of direct romance with legitimate agent communication.

In order to prevent or shield visiting agent from interaction with the legitimate task agent communication, a Restriction-Based Access Control (ResBAC) mechanism is proposed to isolate visiting agent to network server and deny it the opportunity to establish communication thread with other network hosts. It is motivated by the need to ascertain that the execution environment of the task agents is devoid of

any form of interaction other than the collaboration required by the task agents to accomplish their designed objective. In the proposed model, an agent from a foreign security domain enters through the home security domain via the agent server as illustrated in Figure 1. The agent server is responsible for the scanning of the visiting agent against any malicious tendency. This scanning is necessary due to the fact that there is a possibility for a non-malicious visiting agent to have been inflicted with malicious code along its itinerary path unknown to the agent owner or its home network domain.

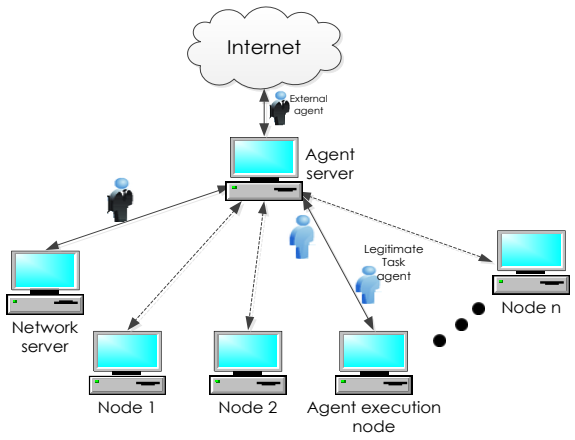


Figure 1 Restriction Based Access Control Model

The agent server could apply integrated trust-based agent admission control with standard RBAC in Gray, O'Connell [3] in the admission of visiting agent into its home security domain. However, protection of agent platform against malicious agent attack is outside the scope of this study. The study covers only attacks that are capable of violating the confidentiality of agent communication such as man-in-the-middle attack, replay attack, and passive eavesdropping.

The rest of the paper is organized as follows: Section 2 presents the related works while the proposed security mechanism is discussed in Section 3. In Section 4, the performance evaluation of ResBAC is detailed and the conclusion is drawn in Section 5.

2.0 RELATED WORKS

Access control models usually achieve secure access control of resources from the viewpoint of the system [4]. They are categorized into: Attribute-based access control; Relationship-based access control; Role-based access control; Task-based access control.

2.1 Attribute-Based Access Control Model

Attribute-based access control model defines access control policies based on various attributes of the client, data object or the environment [5, 6]. The concept of attribute-based encryption (ABE) is a promising access control technique that fulfills the requirement of access control methods that are cryptographically enforced [7]. It comes in two flavours: key-policy ABE (KP-ABE); ciphertext-policy ABE (CP-ABE). The KP-ABE attributes are used to describe the encrypted data and policies are built into user's key. Cipher text policy ABE presents a scalable means of encrypting data such that the encryptor defines the attribute set that must be possessed by the decryptor to be able to decrypt the cipher text. Hence, different users can decrypt different pieces of data per the security policy. This approach to access control effectively eliminates reliance on storage server to provide unauthorized access to data. However, due to the possibility of a group of users sharing the same attribute to data access, the approach introduces some challenges with regard to attribute and user revocation. This simply means that revocation of any attribute or any single user in an attribute group would consequently affect all other users in that group. This will definitely create a bottleneck or degrade the security mechanism.

2.2 Relationship-Based Access Control Model

This access control model is characterized by interpersonal relationships between users and expression of access control policies using these relationships in social computing domain [8, 9]. With this model, a data owner could control the release of his personal information the same manner he would control it in the conventional world [8]. The release of such information would be based on his relationship with the recipient of the data rather than the recipient's role. One consequence of this is that people can hold multiple relationships with someone. Fong [9] widen the applicability of the model to application domains rather than the social computing domain such that authorization decisions are anchored on the relationship between resource owner and the accessor of the resource in a social network maintained by the protection mechanism. The work of Carrie [8] has made relationship-based access control mechanism a general-purpose access control mechanism.

2.3 Role-Based Access Control Model

Role-based access control system [10-13] allocates a given role to each user while permission to access data is assigned to the given role. With this access control system, if different people are requiring access to a record (example is medical record), they are provided different access depending on their roles (specialties) or a specific function the person is serving at a specific time. However, variations of this model are also known in literature such as Risk-aware role-based access control [14]; claims-aware role based access control [15]. The essence of risk-aware role based access control is to provide a tool that can manage the trade-off between the risks involved in allowing unauthorized access with the cost of denying access when the consequence of inability to access

resources is severe. Claims-aware role based access control involves formulating a security token which specifies role information that corresponds to one or more roles of the entity requesting an action to be performed on a resource. The formulation entails accessing at least one or more claims with each having an expression concerning the requesting entity. The major success of role-based access control is its simplified management [4].

2.4 Task-Based Access Control Model

Access Control Model (ACM) usually does not consider the context of a given operation during authorization. Task-Based Access Control model [16, 17] was proposed to address this by changing the attention for security access control from protecting static object and subject in independent system to protecting dynamic authorization performed with the tasks executing. The attendant advantages of TBAC are: dynamic allocation of permission; multi-point access control and distributed processing. TBAC model uses 5 tuples ***(S, O, P, L, AS)*** to describe authorization. ***S*** represents the subject, ***O*** describes object, ***AS*** gives the authorization step, ***P*** specifies the permission activated by ***AS***, and ***L*** is the survival period of ***AS***. However, the notable features of TBAC are ***L*** and ***AS*** which distinguished TBAC from other ACMs.

When ***AS*** is activated, its agents begin to have central authority that it commissioned, as it also countdown its life cycle. During the lifetime of ***AS***, the permission granted to the users dependents on not only the object and subject but also on the task currently running, task status, and user's permission when task is active. Permission is frozen when task is suspended but restored when task resumes. Moreover, permission is revoked when task is terminated.

3.0 PROPOSED MECHANISM

Mobile agent is often faced with access control on entry into a security domain other than its home network domain. This is to avert the attendant risk associated with the failure to put adequate access control mechanism in place especially in a multi-agent system environment. Three security processes are usually utilized in access control namely: identification, authentication, and authorization [18]. The three processes are incorporated in the proposed mechanism. The identification and authentication (of visiting network domain) processes were performed at the agent server, while the authority to execute on the receiving network domain is granted or denied at the receiving network server. However, authentication of the visiting agent (using digital signature of agent server) is also performed at the network server to ascertain that it is sent from the agent server of receiving network domain.

On arrival of the visiting agent at the network server of the receiving network domain, an agent controller in the network server has to authenticate the signature on the visiting agent certificate by hashing the signature using 160-bit SHA-1 algorithm and compares the derived message digest with the message digest sent by agent server. If there is match, such visiting agent is allowed to execute on the platform, otherwise it is killed as shown in Figure 2. However, before permission for execution is granted, the visiting agent is isolated and its execution is confined to the network server in such a manner that prevents it from communicating with the legitimate task agents running in the execution host or their communication. The isolation of visiting agent to the network server was made possible by running it in a different type-space so that its movement and activities are limited to the space (running environment). Any attempt made by the visiting agent to reference the legitimate task agents running on execution host provokes a type error.

Authentication is a primary security issue for the establishment of secure communications. Hence in this study, the digital signature authentication process is the only technique used to authenticate the visiting agent since its security is not of any importance to its receiving security domain.

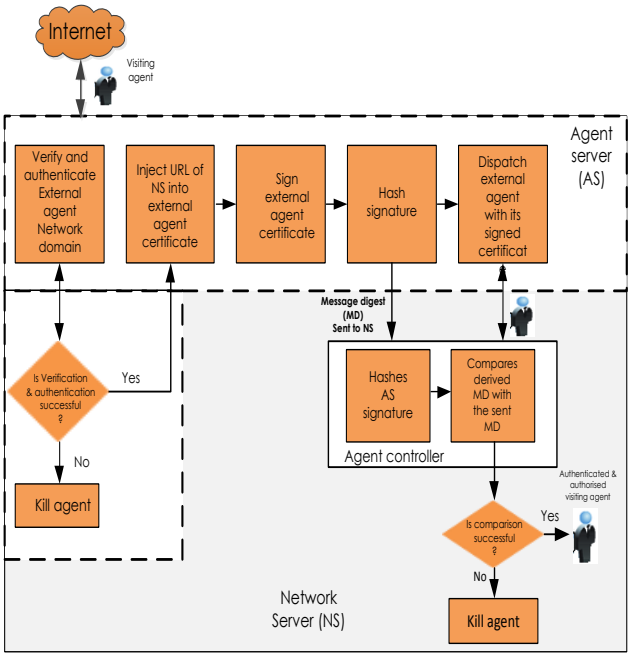


Figure 2 Workflow design model of the proposed mechanism for inter-confidentiality protection of agent communication

This technique denied the visiting agent the privilege to communicate with any other agents outside its runtime environment.

The agent server first verifies the network domain of the visiting agent to establish the identity of the domain. This is done by sending verification request to the certificate authority of the receiving security domain, who contacts the certificate authority (CA) of the sending network domain. If the verification is successful and the identity of the visiting agent is established, the agent server then signs the visiting

agent certificate, hashes the signature and sends the message digest to the agent controller of network server. It is assumed that the two communicating network domains are registered with two different certificate authorities, who are responsible for the authentication of agent migrating across network boundaries as shown in Figure 3.

Having verified the source of the visiting agent and established an identity for the agent, the second level of security is imposed. At this level, the agent server of the receiving security domain injects a new destination address into the visiting agent's certificate. The new address depicts the address of the network server of the receiving network where the visiting agent will execute. In this study, controlled migration of visiting agent within the receiving network domain, agent identification, authentication and confinement of its execution to network server are referred to as Restriction-Based Access Control (ResBAC) mechanism. The essence of this technique is to restrict the hopping of the visiting agent to the network server of the receiving security domain. It is worth noted that agent identification itself is not a primary security issue. Since security related decisions cannot be made only by presenting agent identity, the second security process, that is, authentication was employed by appending the electronic signature of agent server on the certificate of any visiting agent visiting the receiving security domain.

The most widely used access control mechanism is RBAC [FERREIRAabd, Ricardo [19]; Santos-Pereira, Augusto [18]] due to its simplicity and ease of administration. However, this authorization technique is not suitable for the proposed mechanism because the study is only interested in determining an authorized runtime environment for the visiting agent rather than the network resource access permission. The network resource access right of the visiting agent is outside the scope of this study.

3.1 Experimental Setup

In the experiment, we used a computer having Intel Core i5 CPU with 2.40GHz processor speed, 4GB RAM and 64 bits Windows operating system while Oracle virtualbox running Ubuntu Linux operating system (1GB RAM) was used for the agent execution host. The experiment was implemented using JADE framework and Java. It comprises of three JADE platforms running on the Windows and the Linus virtualboxes connected by a virtual local area network. The multiplatform was integrated using a computer running windows operating system on top of Ubutu Linux operating system version 6, which was installed using Oracle VirtualBox software. The local area network (LAN) facilitates communication between the three virtual machines. The Linux box is the Execution Host environment where the task agents run while the Agent Controller, certificate authority agent and the network server run on the Windows machine.

When an external agent visits the receiving network domain, its origin and itself are authenticated at the agent server following the procedures discussed in Sections 3.3 and 3.4. If the authentication of the visiting agent is successful, the agent server then inserts the URL of the network server into its migration path such that it views the network server as the next host to visit and execute. At the network server, the visiting agent is isolated and confined to its runtime environment so that none of its activities goes beyond the environment.

3.2 Certificate Authentication

The focus of this paper is to protect task agent communication with the execution host against possible attack by a visiting agent. This was accomplished using certificate authority model. First and foremost, the agent servers of the two communicating network domains must engage in mutual authentication which shall be explained in the next section. Thereafter, the certificate of the visiting agent itself is authenticated before it could be allowed into the receiving network domain.

3.3 Procedure for Authentication of Two Communicating Network Domains

The procedure taken for the authentication of the two communicating network domains is illustrated in Figure 3 and Figure 4, which is also summarized below:

- The security administrator instructs the agent server of the sending network domain to send a ***SYNC*** request with its signed certificate to the receiving network domain.
- The agent server of the sending network domain sends ***SYNC*** request with its signed certificate to the receiving network domain.
- The agent server of the sending network domain hashes its signature (using SHA-1 hash function) and sends the message digest to the receiving network domain.
- ***VERIFY*** operation at the receiving agent server works as follows:
  - If the certificate of the sending agent server is not with the receiving agent server, the receiving agent server request for its ***CA*** ***(CA_2)***. If ***CA_2*** does not have the certificate, it requests for it from the ***CA*** of sending network domain ***(CA_1)***.
- The receiving agent server hashes the signature of sending agent server and compares the derived message digest with the message digest sent by the sending agent server.
- The receiving agent server sends an ***ACK*** reply with its signed certificate to the sending agent server. The ***ACK*** reply acknowledges the successful authentication of the visiting agent certificate.
- The agent server of the receiving network domain hashes its signature (using SHA-1 hash function) and sends the message digest to the sending network domain.
- ***VERIFY*** operation at the sending agent server works as follows:

- If the sending agent server does not have the certificate of the receiving agent server, it verifies from its ***CA (CA$_1$)***. If ***CA$_1$*** does not have the information, it sends request to ***CA$_2$***.

The sending agent server hashes the signature of receiving agent server and compares the derived message digest with the message digest sent by the receiving agent server.

- After the mutual authentication of the two network domains, the sending agent server sends the visiting agent with its certificate to the receiving network domain.

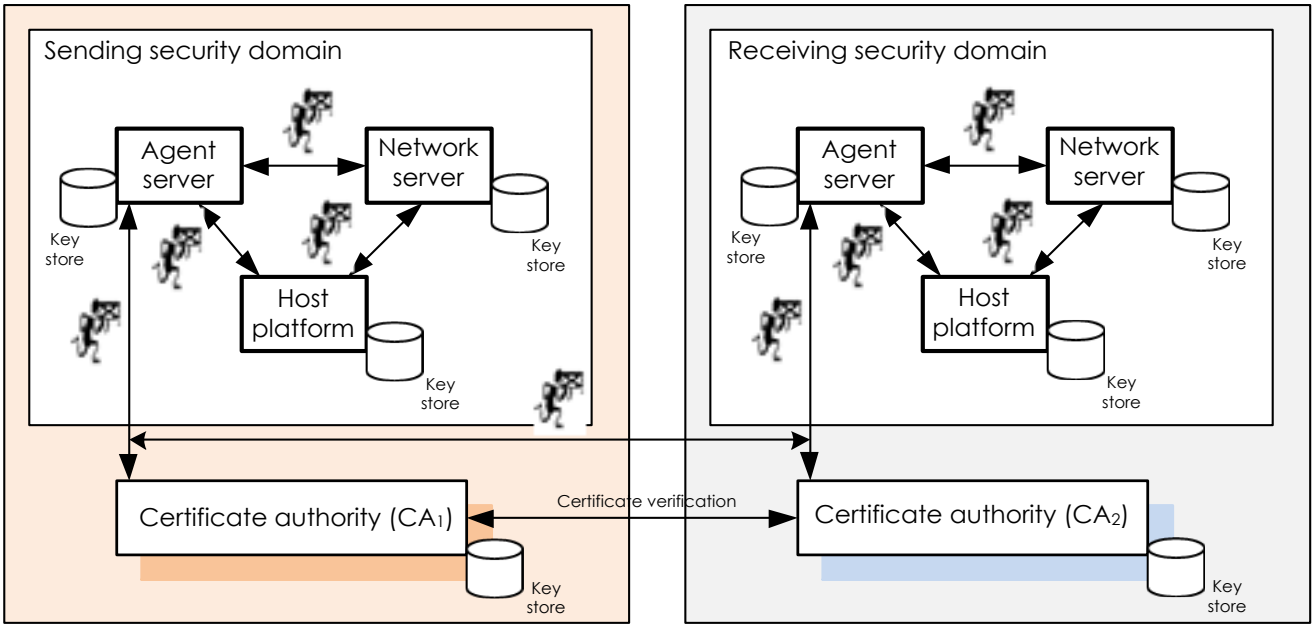


Figure 3 Network domains and visiting agent authentication process

3.4 Procedure for Authentication of Visiting Agent

The visiting agent certificate is modified to accommodate the new destination address that specifies the address of the network server as the next host to visit in its itinerary. Any agent sent to the network server for execution automatically implies that such agent is an visiting agent and all the access control policies defining the privileges of visiting agent will be invoked. Although the access control to network resources is out of scope of this study. The steps taken to authenticate visiting agent certificate are depicted in Figure 4 and summarized as followed:

At the receiving network domain,

- the security administrator (***SA***) instructs the agent server (***AS***) to sign the visiting agent certificate (***ExA$_{cert}$***).
- the ***AS*** signs the certificate of the visiting agent with its private key (***k$_i$***),
- ***SA*** requests ***AS*** to hash its signature and send the derived message digest (***MD$_1$***) to network server (***NS***).
- ***AS*** hashes its signature using 160-bit SHA-1 [20] algorithm and send *digest response* (***MD$_1$***) to network server (***NS)***, and ***AS*** dispatches the visiting agent (***ExA***) with its signed certificate to the ***NS***.

The network server carries out the following operation in order to authenticate the visiting agent:

- NS hashes the signature of ***AS*** of receiving network domain using the same 160-bit SHA-1 algorithm to obtain *digest response* (***MD$_2$***).
- It then compares ***MD$_2$*** with the one earlier sent to it from the agent server (i.e. ***MD$_1$***).

If there is match in the two *digest responses* (i.e. **MD$_1$** and **MD$_2$**), the ***NS*** then isolate ***ExA*** and hands it over to the access control mechanism (for resource access permission). The access control mechanism is expected to determine which network resource(s) to be made available for the visiting agents during execution.

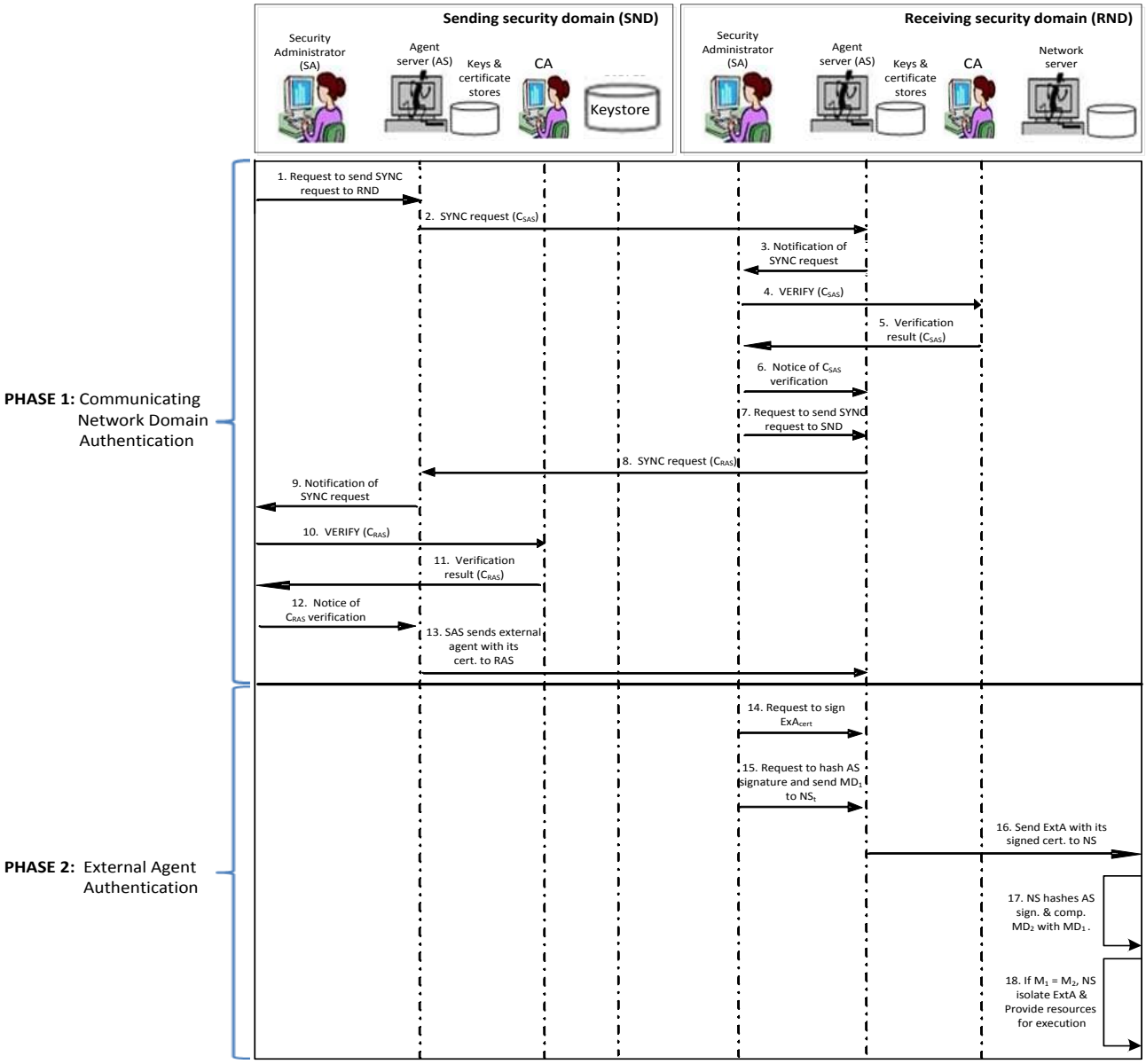


Figure 4 Authentication of network domain and visiting agent

4.0 PERFORMANCE EVALUATION AND VALIDATION OF ResBAC

There is none of the existing access control mechanisms that delve with isolating visiting agent to a neutral platform. In view of this, the proposed ResBAC mechanism was evaluated and validated on the basis of its effectiveness to combat the three basic threats to agent communication between the task agent and agent controller in the execution host, using six scenarios:

a. Man-in-the-middle-attack without the application of the proposed mechanism;
b. Man-in-the-middle-attack with the application of the proposed mechanism;
c. Replay attack without the application of the proposed mechanism;
d. Replay attack with the application of the proposed mechanism;
e. Passive eavesdropping without the application of the proposed mechanism;
f. Passive eavesdropping with the application of the proposed mechanism.

4.1 Man-in-the-Middle Attack with and without the Proposed Mechanism

Man-in-the-middle attack is one of the most important attacks upon cryptosystem. It involves an adversary making independent connections with the communication channel through which the task agent communicates with the execution host and

relays messages between them so as to believe they are communicating directly to each other over a private connection.

For example, before task agent is transmitted to ***EH***, ***AS*** asks ***EH*** for its public key [$EH(k_j)$]. If ***EH*** sends [$EH(k_j)$] to ***AS*** but ***MITM*** attacker is able to intercept it, a man-in-the-middle attack can begin. ***MITM*** attacker sends a forged message (M_f) to ***AS*** that claims to be from ***EH***, but instead includes ***MITM*** attacker's public key (Ak_j). ***AS***, believing the (Ak_j) to be [$EH(k_j)$], encrypts the secret key with ***MITH*** attacker's public key [i.e. [$E(k_s)$; Ak_j]] and sends the $E_A(k_s)$ to ***EH***. ***MITM*** attacker again intercepts $E_A(k_s)$, decrypt it (i.e. $D_A[E_A(ks)]$) using its private key (Ak_i) and re-encrypt it using the $EH(k_j)$ originally sent to ***AS*** [i.e. [$E_A[D[E(ks)]]$; $EH(k_j)$]]. When ***EH*** receives [$E_A[D[E(ks)]]$; $EH(k_j)$], the newly encrypted secret key, it believes it came from ***AS***. Now, it becomes obvious that MITM has the secret key to decrypt any message transmitted between the task agent and the execution host which can also be sniffed by the MITM as shown in Figure 5. The secret key known to ***MITM*** can be used to lunch a severe attack on future communication between the task agent and the execution host.

1. *AS: AS ← request ($EH(k_j)$)*
2. *EH: MITM ← $EH(k_j)$/* MITM intercepts */*
   /$EH(k_j)$ meant for AS */*
3. *MITM: AS ← MITM (M_f, Ak_j)*
4. *AS: [$E(S_k)$; Ak_j]*
5. *AS: MITM ← [$E(S_k)$; Ak_j] /*MITM again*/*
   */*intercepts or sniffs*/*
   /$E(S_k)$ meant for EH */*
6. *MITM: [$D_A[E(S_k)]$; Ak_j]*
7. *MITM: [$E_A[D_A[E(S_k)]]$; $EH(k_j)$]*
8. *MITM: EH ← [$E_A[D_A[E(S_k)]]$; $EH(k_j)$]*
9. *EH: TA ← request (M_f)*
10. *TA: MITM ← M_f /* MITM intercepts*/*
    */*the message for meant for EH */*
11. *MITM: [$D[M_f]$]; S_k]*

Figure 5 MITM attack during communication between agent server and execution host

Similarly, ***MITM*** can occur during the conversation between the ***TA*** and ***EH***. When the ***EH*** requests for the certificate of a ***TA*** as a proof of identity, an ***MITM*** making independent connection with the communication channel can capture and keep the request made by the ***EH*** and sends its certificate (EH_{cert}) request to the ***TA***. The ***TA*** sends its certificate (TA_{cert}) to the ***MITM***, which it keeps and forwards its certificate (A_{cert}) to the ***EH***. ***EH*** believes that the response received is from ***TA*** and hence communicate directly to the ***MITM*** ignorantly. This scenario is illustrated in Figure 6.

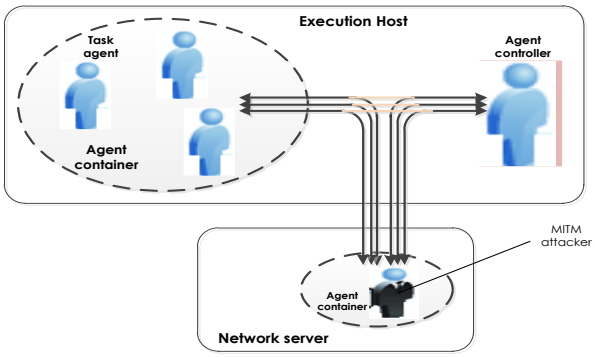


Figure 6 Man-in-the-middle attack without the proposed mechanism

The proposed mechanism isolates the ***MITM*** such that it is cut off completely from interacting or communicating with any other entity outside its container as shown in Figure 7. If the ***MITM*** attempts to launch an attack on the ***TA's*** transmission to the ***AC*** in ***EH*** or reference the legitimate ***TAs*** running on ***EH***, it provokes a type error. This makes ***MITM*** attack practically impossible and hence preserves the confidentiality of ***TA*** communication with the execution host.

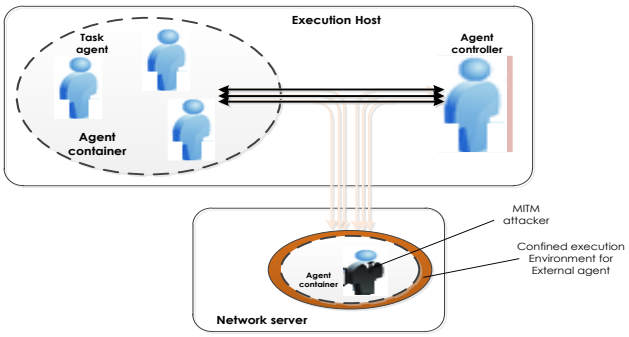


Figure 7 Man-in-the-middle attack with the proposed mechanism

4.2 Replay Attack with and without the Proposed Mechanism

A ***replay attack*** occurs when the certificate request made by the execution host on the task agent is copied by an adversary and retransmitted to the task agent. It also occurs when the certificate transmitted by the task agent intentionally to the execution host is copied and retransmitted to the execution host.

When ***TA*** arrives at ***EH***, it is mandatory the ***TA*** proves its identity to ***EH***. For this to happen, ***EH*** requests ***TA*** certificate (TA_{cert}) as proof of identity, which the agent dutifully provides after some transactions like hashing. At this time, suppose a replay attacker (***RA***) eavesdrops on the conversation and keeps TA_{cert} (or the hash value). After the conversation is over, then ***RA*** posing as legitimate ***TA*** connects to ***EH***, and sends the legitimate TA_{cert} (or hash value) read from the last session for a proof of identity as shown in Figure 8. The

EH having accepts TA_{cert} thus granting access to ***RA***. At this time, the confidentiality of TA_{cert} has been breached and such stolen certificate can be used by ***RA*** to lunch impersonation attack on the legitimate ***TA***. This scenario is illustrated in Figure 8 with the notational procedure in Figure 9.

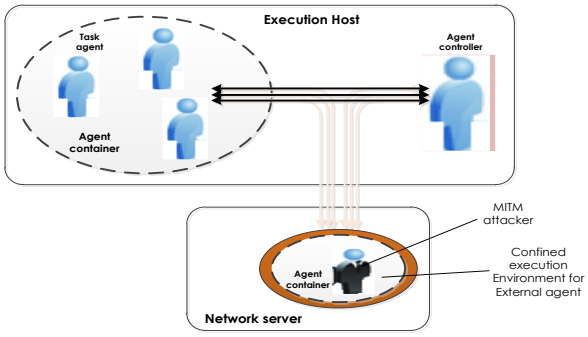


Figure 8 Replay attack without the proposed mechanism

1. *EH: TA ← request* TA_{cert}
2. *EH: RA ←* TA_{cert} */*RA connects and keeps* TA_{cert} **/*
3. *RA: EH ←* TA_{cert} */*RA transmit* TA_{cert} *to EH*/*

Figure 9 Replay attack during communication between task agent and execution host

When ***TA*** launches a replay attack on the communication between a ***TA*** and the ***EH***, the proposed mechanism prevents illegal connection with the channel through which attack can takes place. The faded connection shown in Figure 10 shows unsuccessful connection attempts made by ***RA***.

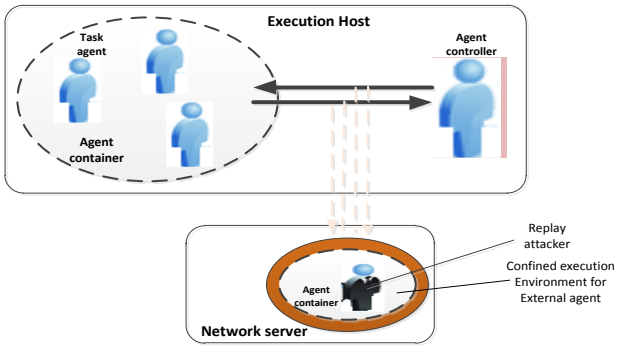


Figure 10 Replay attack with the proposed mechanism

4.3 Eavesdropping Attack with and without the Proposed Mechanism

Eavesdropping is an unauthorized real-time interception of a private communication between two or more entities. This type of communication interception does not require physical connection to the communication channel but rather the eavesdropper spies or listen to the conversations between the communicating entities.

It is worth noted that the two categories of attacks described above are also eavesdropping attacks. They are often referred to as active eavesdropping attacks. However, an eavesdropping attack can also be passive such that passive eavesdropper neither interacts with ***TA*** nor with the communication channel, but spies or listens to agent communication thereby compromises its confidentiality as shown in Figure 11 and illustrated in Figure 12. From Figure 11, it can be observed that the eavesdropper, having set to eavesdrop, records all the communication between the task agent and the execution host thereby compromising the confidentiality of task agent certificate.

1. *EA: Set spy alert /* EA is set to spy communication between EH & TA */*
2. *EH: TA ←request* TA_{cert}
3. *TA: EH ←* TA_{cert}
4. *TA: EA ←* TA_{cert} */* TA certificate is leaked to EA */*

Figure 11 Spy activity of eavesdropper during communication between task agent and execution host

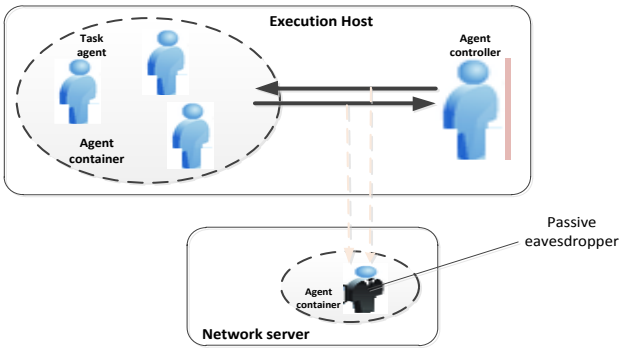


Figure 12 Eavesdropping attack without the proposed mechanism

Similar to overcoming active eavesdropping, the proposed mechanism also makes the eavesdropper deaf to the activities outside the activity-space established for the external agent as shown in Figure 13. The agent is masked from listening to the conversations outside its execution environment thereby making passive eavesdropping practically impossible for the external agent.

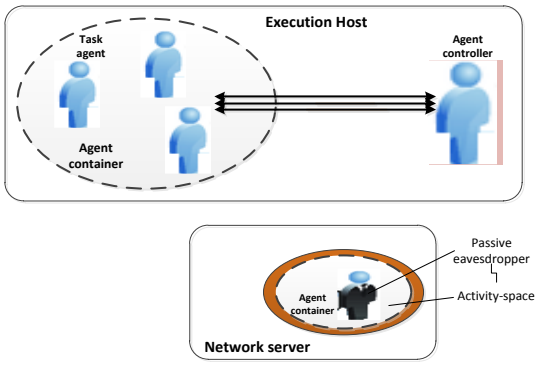


Figure 13 Eavesdropping attack with the proposed mechanism

5.0 CONCLUSION

This paper presents a restriction-based access control mechanism (ResBAC) for inter-confidentiality protection of agent communication. It was a technique adopted in this study to direct visiting agent to a desired neutral host for execution and at the same time isolates it to disable its ability to communicate with the legitimate task agents. The isolation of the visiting agent was achieved by restriction-based access control, which establishes an activity-space for the visiting agent such that none of its activities can extend beyond the boundary of the activity-space. This was used to limit what the visiting agent can do [21] especially its communication coverage.

For a visiting agent to be admitted into the receiving network domain, the sending and the receiving network domains must undergo mutual authentication. The visiting agent must also be authenticated to establish its identity at the receiving network domain. In this research, the performance of ResBAC is evaluated using six different scenarios to measure the strength and justify the efficiency of the mechanism in handling the major security threats to agent communication. The security threat comprises man-in-the-middle (MITM), replay, and passive eavesdropping attacks. The MITM and replay attacks are also called ***active eavesdropping attacks***. The main achievement of this research is the design of a novel restriction-based access control mechanism to isolate visiting agent to a neutral host to prevent possible attack on the communication between the legitimate task agents and the execution host. The mechanism is based on JADE framework and implements agent communication based on FIPA-ACL, where a malicious agent (MA) is created and equipped with the capability of probing into the communication between the task agents and the agent controller in the execution host.

Acknowledgement

We wish to acknowledge the research supports provided by Universiti Teknologi Malaysia and The Federal Polytechnic, Ado-Ekiti, Nigeria. We also wish to appreciate the efforts of the anonymous reviewers whose corrections and suggestions have added value to this article.

References

[1] Li, N. 2011. Discretionary Access Control. *Encyclopedia of Cryptography and Security*. 353-356.
[2] Ni, Q., *et al.* 2010. Privacy-Aware Role-Based Access Control. *ACM Transactions on Information and System Security (TISSEC)*. 13(3): 24.
[3] Gray, E., *et al.* 2002. Towards a Framework for Assessing Trust-based Admission Control in Collaborative Ad Hoc Applications. Dept. of Computer Science, Trinity College Dublin, Technical Report. 66.
[4] Zhao, Y. L. and C. F. Jiang. 2014. Research of Access Control Models in Personal Networks. In *Advanced Materials Research*. Trans Tech Publ.
[5] Hur, J. and D. K. Noh. 2011. Attribute-Based Access Control with Efficient Revocation in Data Outsourcing Systems. *Parallel and Distributed Systems, IEEE Transactions on*. 22(7): 1214-1221.
[6] Hu, V. C., D. R. Kuhn, and D. F. Ferraiolo. 2015. Attribute-Based Access Control. *Computer*. 2015(2): 85-88.
[7] Sahai, A. and B. Waters. 2005. Fuzzy Identity-Based Encryption. In *Advances in Cryptology–EUROCRYPT 2005*. Springer. 457-473.
[8] Carrie, E. G. 2007. Access Control Requirements for Web 2.0 Security and Privacy. In *Proc. of Workshop on Web 2.0 Security & Privacy (W2SP 2007*. Citeseer.
[9] Fong, P. W. 2011. Relationship-Based Access Control: Protection Model and Policy Language. In *Proceedings of the first ACM conference on Data and application security and privacy*. ACM.
[10] Hammoutene, M., M. Petkovic, and C. V. Conrado. 2013. Role-based Access Control. Google Patents.
[11] Alturi, V. and D. Ferraiolo. 2011. Role-Based Access Control. In *Encyclopedia of Cryptography and Security*. Springer. 1053-1055.
[12] Joshi, S. 2010. Role-Based Access Control. Google Patents.
[13] Tsai, W.-T. and Q. Shao. 2011. Role-Based Access-Control Using Reference Ontology in Clouds. In *Autonomous Decentralized Systems (ISADS), 2011 10th International Symposium on*. IEEE.
[14] Chen, L. and J. Crampton. 2012. Risk-Aware Role-Based Access Control. In *Security and Trust Management*. Springer. 140-156.
[15] Bilaney, R. P. and S. R. Devasahayam. 2014. Claims-Aware Role-Based Access Control. Google Patents.
[16] Yu, D. 2012. Role and Task-Based Access Control Model for Web Service Integration. *Journal of Computational Information Systems*. 8: 2012(7).
[17] Deng, J.-B. and F. Hong. 2003. Task-Based Access Control Model. *Journal of Software*. 14(1): 76-82.
[18] Santos-Pereira, C., *et al.* 2013. A Secure RBAC Mobile Agent Access Control Model for Healthcare Institutions. In *Computer-Based Medical Systems (CBMS), 2013 IEEE 26th International Symposium on*. IEEE.
[19] FERREIRAabd, A., *et al.* 2007. Access Control: How Can It Improve Patients' Healthcare? *Medical and Care Compunetics*. 4(4): 65.
[20] Eastlake, D. and P. Jones. 2001. *US Secure Hash Algorithm 1 (SHA1)*, RFC 3174, September.
[21] Claessens, J., B. Preneel, and J. Vandewalle. 2003. (How) Can Mobile Agents Do Secure Electronic Transactions on

Untrusted Hosts? A Survey of the Security Issues and the Current Solutions. *ACM Transactions on Internet Technology (TOIT)*. 3(1): 28-48.