SKQML: A Secure Multi-agent Communication Language

Zhu LieHuang   Cao YuanDa   Liao LeJian
School of Computer Science and Technology
Beijing Institute of Technology, Beijing, 100081
P.R. China

Wang DaZhen
Department of Computer Science and Technology
Hubei University of Technology, Wuhan, 430068
P.R. China

Abstract: KQML is one of the most universal agent communication languages, but KQML have not defined security specifications to provide secure communication among agents. Firstly, this paper proposes a multi-agent secure communication protocol and three multi-agent group re-keying protocols. Secondly, it extends the base KQML to secure KQML which supports the secure communication protocol and multi-agent group re-keying protocols.

Key-Words: Multi-agent  KQML  Secure Communication  Re-keying Protocol  Group Communication

1 Introduction

Multi-agent technology has had extensive applications in distributed systems. To solve distributed problem using multi-agent, the agents in the same system must communicate and collaborate each other across open environment such as Internet, in which security is a critical issue. It is very important to ensure the security of the communication data among each agent. However, the popular agent communication language, such as KQML[1] and FIPA ACL[2], has not defined the standards for secure agent communication, which make multi-agent technology in secure to be applied to many key fields. In order to prevent the communication among agent from possible attacks, a multi-agent secure communication protocol should satisfy the following security requirements:

1) Confidentiality: ensure only authorized agents can access to the transmitted secure message.
2) Integrity: ensure non-authorized agent cannot modify the transmitted secure message.
3) Authenticity: ensure the source of the transmitted secure message be identified accurately and the identifier cannot be forged.
4) Non-repudiation: ensure the sender and the receiver both can not repudiate the communication process.

KQML-Based PKI[3] which takes into account of security based on KQML can ensure the integrality, authenticity and non-repudiation of the transmitted message among agents through digital signature. But KQML-Based PKI cannot ensure the confidentiality which is the foremost security factor in most application fields.

Petr Novák and Milan Rollo[4] embeds a X-Security layer into FIPA ACL to dispose secure...
communication among agents. X-Security layer encrypt the communication data using the receiver public key to offer peer-to-peer secure communication mechanism. But X-Security needs encrypt and send $n-1$ times for each message for a multi-agent system with $n$ agents. So the computation quantity increases rapidly with the number of agents. In addition, public key cryptography is not secure against chosen message attack and cannot encrypt message long term.

In this paper, we use group re-key protocol, such as GKMP[5], HBT[6], HKT[7], OFT[8] to ensure all agents in a multi-agent system share the same group key. Then the sender agent can encrypt, and the receiver agent can decrypt, the communication data using the same group key, which ensure the confidentiality. Message authentication code based on the group key can ensure the integrality and authenticity.

2 Multi-agent secure communication protocol

Figure 1 shows the rationale of multi-agent secure group communication. In figure 1, KMA denotes the key management agent, $A_1,A_2,..A_n$ denotes each communication agent which share the same group key $k_0$. $A_i$ sends encrypt message to other agents as follows.

$$A_i \rightarrow \{A_1,A_2,..A_n\} : E_{k_0}(m)$$

Where $x \rightarrow y : z$ denotes agent $x$ sending $z$ to agent(s) $y$ (i.e., $y$ can be an agent set). $E_{k_0}(m)$ denotes the encrypted message of $m$ with $k_0$.

![Fig.1 Multi-agent secure communication principle](image)

Agent $A_1 \in \{A_2,A_3,..A_n\}$ decrypts $E_{k_0}(m)$ with $k_0$ and gets $m = D_{k_0}(E_{k_0}(m))$ when receiving $E_{k_0}(m)$. Where $D_{k_0}(E_{k_0}(m))$ denotes decrypting $E_{k_0}(m)$ with $k_0$. The non-authorization agent cannot get $m$ because of having no $k_0$.

Computing MAC (message authentication code) with $k_0$ and appending the MAC to the message can provide integrality and authenticity as follows.

$$A_i \rightarrow \{A_2,A_3,..A_n\} : m | MAC(k_0,m)$$

Where $MAC(k_0,m)$ denotes the MAC of $m$ with $k_0$, $m | MAC(k_0,m)$ denotes appends $MAC(k_0,m)$ to the end of $m$.

$A \in \{A_2,A_3,..A_n\}$ computes the $MAC(k_0,m)$ after receiving $m | MAC(k_0,m)$. Then $A$ compares $MAC(k_0,m)$ with the received $MAC(k_0,m)$. $A$ can make certain that the received message has not been modified and forged if the result is equal. $MAC(k_0,m)$ can be computed with the following formula.

$$MAC(k_0,m) = E_{k_0}(MD(m))$$

Where $MD(m)$ denotes the MDC (Message Digest Code) which is the fixed length code of $m$ using message digest algorithm, such as MD5 and SHA.

3 Multi-agent re-keying protocols

GKMP, HBT, OFT, OKCT and HOFT use a group key manager server for all agents in a same multi-agent system share the same group key, and make following re-keying policy.

1) Time sensitive re-keying policy: re-keying the group key timely to prevent group key being broken.

2) Leave sensitive re-keying policy: re-keying the group key when an agent leaves off the multi-agent system so as to prevent the left agent decrypting the latter message.

3) Join sensitive re-keying policy: re-keying the group key when a new agent joins the multi-agent system so as to prevent the new joining agent decrypting the former message.

3.1 Time sensitive re-keying protocol

In figure 2, KMA sends time sensitive re-keying
message package to each agent \( A \in \{ A_1, A_2, \ldots, A_n \} \). \( A \in \{ A_1, A_2, \ldots, A_n \} \) renews \( k_0 \) to new group key \( k_1 \) using group re-key algorithm, such as GKMP, HBT, OFT, OKCT and HOFT. \( A_1 \) encrypts message \( m \) to \( E_{k_1}(m) \) using the new group key \( k_1 \) and send \( E_{k_1}(m) \) to each agent \( A \in \{ A_2, A_3, \ldots, A_n \} \).

\[
\text{KMA} \rightarrow \{ A_1, A_2, \ldots, A_n \}:
\]

\[
\text{time sensitive re-keying package}
\]

\[
A_1 \quad A_2 \quad \cdots \quad A_n
\]

\[
E_{k_1}(m) \quad E_{k_1}(m) \quad E_{k_1}(m)
\]

\[
\text{Fig.2 Time sensitive re-keying protocol}
\]

**3.2 Leave sensitive re-keying protocol**

In figure 3, \( A_1 \) sends a leaving request message to KMA. KMA send leave sensitive re-keying package to other agent \( A \in \{ A_1, A_2, A_3, \ldots, A_n \} \) after receiving the leaving request package. \( A \in \{ A_1, A_2, A_3, \ldots, A_n \} \) renews \( k_1 \) to the new group key \( k_2 \). \( A_1 \) encrypts message \( m \) to \( E_{k_2}(m) \) using the new group key \( k_2 \) and send \( E_{k_2}(m) \) to each agent \( A \in \{ A_2, A_3, \ldots, A_n \} \).

\[
\text{KMA} \rightarrow \{ A_1, A_2, A_3, \ldots, A_n \}:
\]

\[
\text{leave sensitive re-keying package}
\]

\[
A_1 \quad A_2 \quad A_3 \quad \cdots \quad A_n
\]

\[
E_{k_2}(m) \quad E_{k_2}(m) \quad E_{k_2}(m)
\]

\[
\text{Fig.3 Leave sensitive re-keying protocol}
\]

**3.3 Join sensitive re-keying protocol**

In figure 4, \( A_{n+1} \) send joining multi-agent system request message to KMA. KMA send join sensitive re-keying package to the other agent \( A \in \{ A_1, A_2, \ldots, A_{n+1} \} \) after receiving the joining request package. \( A \in \{ A_1, A_2, \ldots, A_{n+1} \} \) renews \( k_2 \) to the new group key \( k_3 \). \( A_1 \) encrypts message \( m \) to \( E_{k_3}(m) \) using the new group key \( k_3 \) and send \( E_{k_3}(m) \) to each agent \( A \in \{ A_2, A_3, \ldots, A_{n+1} \} \).

\[
\text{KMA} \rightarrow \{ A_1, A_2, \ldots, A_{n+1} \}:
\]

\[
\text{join sensitive re-keying package}
\]

\[
A_1 \quad A_2 \quad A_3 \quad \cdots \quad A_{n+1}
\]

\[
E_{k_3}(m) \quad E_{k_3}(m) \quad E_{k_3}(m)
\]

\[
\text{Fig.4 Join sensitive re-keying protocol}
\]

**4 Secure KQML**

**4.1 KQML**

KQML (Knowledge Query and Manipulation Language) is the most popular agent communication language. KQML provides a common framework for the communication and collaboration in multi-agent system. KQML includes three layers which are communication, message and content. A common expression of KQML is as follows:

\[
\text{<performat} \\
\quad \text{sender <word> //message sender} \\
\quad \text{receiver <word> //message receiver} \\
\quad \text{language <word> //language of content layer} \\
\quad \text{reply-with <word> //the anticipant response} \\
\quad \text{in-reply-to <word> //the message triggering} \\
\quad \text{ontology <word> //ontology used by current} \\
\quad \text{content <word> //the content of current message} \\
\]

The most outstanding characteristic of KQML is the extensibility which makes KQML possible to extends its function easily just through defining different ontologies, adding performatives, and adding parameters.
4.2 Extends KQML to satisfy multi-agent re-keying protocol

In order to satisfy multi-agent re-key protocol, we add several performatives which includes join, leave, reject, rekeying_expired, rekeying_join, and rekeying_leave.

When Agent $A_x$ request to join the multi-agent system, $A_x$ send the following message to KMA.

```
(join
  :sender $A_x$
  :receiver KMA
  :reply-with $A_x$ _Join_ Group
  :content "the re-key algorithm list $A_x$ can perform")
```

KMA judges whether $A_x$ is entitled to join the multi-agent system when receiving the joining request message. If $A_x$ is forbidden to join, KMA send $A_x$ the following reject message.

```
(reject
  :sender KMA
  :receiver $A_x$
  :in-reply-to $A_x$ _Join_ Group
  :content "reject $A_x$ joining multi-agent system")
```

If $A_x$ is allowed to join, KMA judge whether the re-key algorithm is in re-key algorithm list $A_x$ proposed. If not, KMA sends reject message to $A_x$. otherwise KMA sends rekeying_join performative to all agents as follows

```
(rekeying_join
  :sender KMA
  :receiver $\{A_1,A_2,...A_n\} \bigcup \{A_x\}$
  :in-reply-to $A_x$ _Join_ Group
  :rekeying-algorithm "re-key algorithm"
  :key-seq "group key serial"
  :content "multi-agent re-key package")
```

Each agent $A \in \{A_1,A_2,...A_n\} \bigcup \{A_x\}$ computes the new group key based the multi-agent re-key in the received rekeying_join message and renews the group key and group key serial.

When Agent $A_x$ requests to leave the multi-agent system, $A_x$ sends leave message to KMA as follows.

```
(leave
  :sender $A_x$
  :receiver KMA
  :reply-with $A_x$ _leave_ Group
  :content "the reason of $A_x$ leaving")
```

KMA constructs new multi-agent re-key package after receiving the leave message and sends the following rekeying_join message to each agent except $A_x$ as follows.

```
(rekeying_leave
  :sender KMA
  :receiver $\{A_1,A_2,...A_n\} \bigcap \{A_x\}$
  :key-seq "group key serial"
  :rekeying-algorithm "re-key algorithm"
  :content "multi-agent re-key package")
```

4.3 Extends KQML for multi-agent secure communication

By setting language parameter to GroupSec, the receiver agent knows whether the content in the message should be decrypted or/and authenticated.

The following message indicates that the content of the performative should be decrypted.

```
(performative // ask-all, ask-one, and so on.
  :sender $A_x$
  :receiver $\{A_1,A_2,...A_n\} \bigcap \{A_x\}$
  :language GroupSec
  :type Encryption //only be encrypted
)
The following message indicates that the content of
the performative should be authenticated.

(performative
  :sender A
  :receiver {A, A, ...A}_n − {A_i}
  :language GroupSec
  :type Authentication  //only authentication
  :key-seq “group key serial”
  :algorithm “message digest algorithm”
    //MD5, SHA
  :content “(m|MAC(k, m))”)

The following message indicates that the content of
the performative should be both decrypted and
authenticated.

(performative
  :sender A
  :receiver {A, A, ...A}_n − {A_i}
  :language GroupSec
  :type Encryption&Authentication
  :key-seq “group key serial”
  :algorithm “the combination of encryption
    algorithm and message
    digest algorithm”
    //3DES_MD5, AES_SHA, and so on
  :content “E_k(m|MAC(k, m))”)

5 Conclusion
This paper extends the base KQML to secure KQML
with a multi-agent secure communication protocol and
three multi-agent group re-keying protocols. Secure
KQML ensure the confidentiality, integrity, and
authenticity of the transmitted message. By adding
certificate authority agent and corresponding
performatives, secure KQML can also ensure the
non-repudiation.

References:
Agent Communication Language. In Proceedings
of the 3rd International Conference on Information
and Knowledge Management, New York: ACM
communication language. Specification. FIPA,
October 1997.
agent: KQML-Based PKI. In Proceedings of the
2nd International Conference on Autonomous
Communication Security in Multi-Agent Systems.
In: Multi-Agent Systems and Applications III,
Management Protocol (GKMP) Architecture,
Internet Engineering Task Force, July 1997. RFC
2094.
for Multicast: Issues and Architectures, Internet
Group Communications Using Key Graphs,
IEEE/ACM Transactions on Networking, VOL.8,
NO.1, 2000, pp.16-30.
Management for Large Dynamic Groups: One-Way
Function Trees and Amortized Initialization. IETF
Internet draft, August 2000.