Using the Multi-Agent System to Develop a Key Agreement Protocol

Tayseer S. Atia
Lecturer, College of Software Engineering and Networking, Al-Iraqia University, Iraq
Tayseer_salman@yahoo.com

ABSTRACT
The incremental growth of computer networks caused by new users joining the network results in a confidentiality problem with the transmission of data. To solve this problem, symmetric/asymmetric key cryptography is mostly used. However, this cipher suffers from the problem of secret key maintenance and distribution across the internet which is unsecure channel. To get rid of this problem, different key exchange schemes have been developed and classified as a key redistribution scheme, key distribution scheme and key agreement scheme. In this paper, a key agreement scheme based on a multi-agent system is developed. This scheme aims to exchange the secret key between two parties. The mobile agent is the heart of this scheme where it carries the program that is responsible for generating the secret key.

Keywords: Multi-Agent, Session key, Key Agreement Protocol

1. INTRODUCTION
The increase popularity of the internet led to the privacy problems which in turn become the reason for developing cryptographic techniques.

Users or pairs of users may have Long-Lived Keys (LLK) which are pre-computed and then stored securely or secret short-lived session keys in a particular session, and then throw them away when the session has ended [1]. Techniques and procedures that supports the establishment and maintenance of keying relationships between authorized parties is another challenge faced the user of the internet [2]. The purpose of developing many key exchange protocols are to solve this challenge. Key agreement protocol allows two or more parties to exchange information among them over insecure channel [3].

An agent is an intelligent computer system that is situated, please note that the intelligence is a basic criteria for an autonomous agent in this environment in order to meet its design objectives [4]. Agents can often be characterized by a number of properties like autonomous, active, proactive, mobile [5]. Multi-agent systems (MAS) aim to create systems that interconnect separately developed agents, thus enabling the ensemble to function beyond the capabilities of any singular agent in the set-up [6].

In this paper MAS are proposed to implement a key agreement protocol. This protocol : 1) Intended to eliminate the need for the trusted third party used by previous protocols, 2) reduces the number of messages exchanged between two parties to establish public parameters, 3) employs the cryptographic techniques like the digital signature, one way hash function and encryption to provide authenticity, integrity and confidentiality and to secure the transmitted key against adversary, man in the middle attacks, and 4) uses a mobile agent to exchange a secret key between two parties in a private domain.

In this paper, the following contributions have been made:

- Design a completely java-based framework for gathering, storing and processing key generating-related data on the protocol.
- Creating a complete management system to aid the registration, updating and exchanging of key.
- Implementing a fully automated process for sending and returning the mobile agent.
- Correlate the visual basic programming language with the JADE framework for developing the system interface which offer the simplicity of the design.

Section 2 describes the prior work done on multi agent system. Section 3,4 explain the proposed protocol. Section 4 discusses the results of the simulation. section 5 explains the research conclusion.

2. RELATED WORK
Park et al. (2001) announced a new key generation scheme called one time key based system. This system uses computations to generate a key at each node using information received from previous hosts. Only, the originator is able to decode the information since he has the initial value. The main strength and weakness of this system is that there is an inter-relationship between consecutive agent keys. Therefore, in case that some intermediate agent data or key get tempered or deleted then the whole system will fail [7].

Yi et al. (2009) introduced a distributed key management method in ad hoc network based on mobile agent. Few mobile agents navigate in the network according to a visits-balance policy. Carrying secret key and network topological information, mobile agents communicate with nodes and other agents. When a new node wants to join the ad hoc network, t nodes will cooperate with each other to authenticate the new one. The problem with heavy network overload, long service time and low success ratio in traditional distributed authenticate algorithm can be solved in this way. This approach reduces the system overload, and it is very efficient and robust to transmit packets quickly. At mean
time, they completely avoid a centralized certification authority or trusted third party to distribute the public keys and the certificates, thus enhancing the tolerance of the network to compromised nodes and also efficiently saving network bandwidth. As any t nodes can reconstruct the matter private key jointly, it is infeasible at most t-1 nodes to do so. Even by collusion, the security is enhanced [8].

3. THE ARCHITECTURE OF THE PROPOSED SYSTEM

The proposed multi-agent system is distributed between the sender and the receiver (the protocol parties). In the proposed architecture, the receiver is the party who requests the generation of the session key. The sender is the party who owns the mobile agent which is responsible for generating the session key. The multi-agent system at sender side consists of three agents and they are:

- **Interface Agent (INA):** manages the communication between the sender and the receiver sides. It also manages the communication inside the sender party. This agent is a local agent.

- **Registration Agent (RA):** This agent maintains the registration database which contains the user name, password, port address, platform address. Maintenance operation over registration database includes add a new entry, delete an old entry, modify an existing entry and answering question about a specific entry. This agent is a local agent.

- **Session Key Generation Agent (SKGA):** This agent is a Mobile agent which carries the proposed pseudo random number generator algorithm. The mobile agent structure consists of the following:
  
  **Code section:** Contains the mobile agent code and the key generation algorithm.
  
  **Static data section:** contain static data such as creator identity, receiver name, platform address. Some static data are used by the generation algorithm and its digital signature.

The receiver side consists of the following two agents:

- **Initialization Agent (IA):** manages the communication inside the receiver side. It generates the key generation request message parameter, provides the initial value for the key generation agent (mobile agent) in order to generate the session key, and finally informs the request agent with the completion of the session key generation to do the required.

- **Request Agent (RTA):** The receiver side communicates with the sender through this agent. It is responsible for sending the requested messages. These messages are the key generation, registration, and the inform message to the sender side.

The proposed architecture is implemented as a hierarchal architecture using tree structure. In this tree, the father at level 1 represents the sender for a group of children at level i+1 which represent receivers. Again, fathers at level i+1 represent senders for another group of children (receivers) at level i+2 and so on. This means that each node in the hierarchy takes the role of the sender and receiver at the same time. Figure (1) shows the proposed system architecture.

![Fig 1: Proposed System Architecture](image1)

Figure (2) shows the detailed architecture of the sender and receiver nodes in the proposed system.

![Fig 2: Multi-agent System Architecture](image2)

4. THE PROPOSED PROTOCOL

The proposed key exchange protocol is classified as a key agreement protocol. The multi-agent system in this protocol have mobile architecture and the mobility is weak. It also execute in separate process and considered sender initiated. Table(1) and Table(2) summarize the messages used for remote communication between sender and receiver and local communication inside sender respectively.

Table(1) and Table(2)...

...
Table 1: Remote communication between Sender and Receiver

<table>
<thead>
<tr>
<th>Message</th>
<th>Purpose</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>KEY GENERATION</td>
<td>Request the sender for key generation</td>
<td>Rid, password encrypted with sender public key, signed with receiver private key</td>
</tr>
<tr>
<td>FINISH</td>
<td>acknowledge the sender with the completion of key generation at receiver side</td>
<td>Challenge, date, time, encrypted with sender public key, signed with receiver private key</td>
</tr>
<tr>
<td>FAIL</td>
<td>Inform the sender with generation failure</td>
<td>Failure type: 1: unauthentic</td>
</tr>
<tr>
<td>WAIT</td>
<td>from sender to receiver to request waiting for mobile agent arrival</td>
<td>Prompt</td>
</tr>
<tr>
<td>RECONFIRM</td>
<td>reconfirm key generation</td>
<td>key generation parameter</td>
</tr>
</tbody>
</table>

Table 2: Local communication inside the sender

<table>
<thead>
<tr>
<th>Message</th>
<th>Source, Destination Agents</th>
<th>Purpose</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>From interface to registration</td>
<td>Get user information matching Rid</td>
<td>Rid</td>
</tr>
<tr>
<td>POST</td>
<td>From registration to interface</td>
<td>Send user information</td>
<td>User name, host name, port address, platform name</td>
</tr>
<tr>
<td>MOVE</td>
<td>From interface to key generation</td>
<td>Request to move</td>
<td>User name, host name, port address, platform name</td>
</tr>
<tr>
<td>READY</td>
<td>From key generation to interface</td>
<td>Inform of reach</td>
<td>-------</td>
</tr>
</tbody>
</table>

The proposed key exchange protocol is summarized below:

R: receiver, S: sender. RID: registration id, E: public key encryption, SG: signature, m= message.

1. Request agent at R side sends a key generation request message, \( m = [E(RID, password)] \ || SG \).
2. Interface agent at S side performs the following:
   1. Recomputes \( H_1 = H(E(RID, password)) \)
   2. Decrypts the SG with the receiver public key

At this point, the mobile agent moves to the specified platform and sends movable message to the interface agent.

Interface agent in response to the movable message changes the state of mobile agent as movable (state = 1).

At the R when the mobile agent is reached:

1. Initialization agent performs security check for the MA, if it passes then
2. Prepares the initial value for the mobile agent to generate session key
3. Executes the mobile agent inside the specified platform
4. Stores the generated session key
5. Informs the request agent with the completion of key generation and sends initial parameter (username, time, challenge)
6. The request agent sends "finish" message to the interface agent with initial parameter
7. Mobile agent goes back to the sender side
8. Kills the MA and sends "fail" message to the sender with failure type.

In step 1, the initialization agent verifies that the agent is suitable for execution by enforcing authentication.

As soon as the MA arrives at the agent server, and before it can perform any task, it is authenticated. The initialization agent asks for the authentication details, password and an identification number.

Figure (3 a) shows the details of the interface agent local communication at the sender side, and Figure (3 b) shows the details of the interface agent remote communication.
communication with the receiver side. Figure (4) shows the behavior of registration agent, and Figure (5) shows the mobile agent life cycle.

5. SIMULATION AND RESULT

In order to simulate the proposed protocol, three different scenario are chosen. These scenarios are

1. Normal scenario
2. Malicious agent scenario
3. Malicious agency scenario

For all scenarios, it consider that the receiver Ahmed wants to share a session key with the sender Tayseer.

For the normal scenario, figure (7) shows the GUI of the proposed multi agent at the receiver side and figure (8) shows the GUI of the proposed multi agent system at the sender side. This GUI explains the current agent active in the receiver side (IA and RTA). This interface motivates the basic tasks the RTA can perform on behalf of its owner, namely key generation, registration and update. Figures 9 through 15 show the entire generation process.
Fig 7: the GUI of Sender of MobileAgent

Fig 8: the "KEY GENERATION" Message by RTA at receiver of MA

Fig 9: the "GET" Message by INA at Sender of MA

Fig 10: the "POST" Message by RA at sender of MA

Fig 11: The "Move" message by INA at sender of MA

Fig 12: "Movable" message by SKGA at sender of MA

Fig 13: GUI of the SKGA after reaching the receiver of MA

Fig 14: the session key generated by SKGA at receiver of MA
6. SECURITY ANALYSIS

The proposed system is designed in such a way to provide the maximum degree of protection for the transmitted message. The multi-agent system is resistant to the man in the middle attack because the remote communication messages are encrypted with the receiver public key and signed with the sender private key. Also, the encryption and the digital signature attached to the message contribute in preventing the modification and fabrication attack. Replay attack is unsuccessful attack on the proposed system because its costs and can be detected easily. Its cost because the attacker must provide a compatible environment like the attacked party to setup the communicational channel (MTP) between them. And its agents must be registered previously with the sender DF services. The authentication parameter provided by the MA at the receiver side implements a mutual authentication and reconfirms the validity of the current session because the RID is a parameter sent by the receiver itself to the sender in the key generation request and user name is a retrieved parameter from the sender internal database according to the received RID. Reconfirm message used by the receiver helps the sender to detect the interruption attack which may result in a malicious agency attack or malicious agent attack. Ready message sent by MA to the sender implements a dual purpose. It first informs him of MA return to change state. Second it makes the sender be aware of the MA safety in case of the absence of this notification.

7. CONCLUSION

In this paper a key agreement protocol based on MAS is proposed. The proposed protocol does not need the trusted third party. It also requires no reestablished public parameters. the number of transmitted messages are reduced between the communication parties; thus avoiding possible attack. Moreover the protocol resistant to man in the middle attack, modification, fabrication.

REFERENCES


