International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 6 | ISSN : 2456-3307

Contract Net Secure Establishment Protocol (CNSEP)

Amandeep Kaur*[1], Shaveta Jain[2]
*[1]Research Scholar, Department of CSE, GGS College of Modern Technology, PTU, Jalandhar , Punjab, India
[2]Assistant Professor, Department of CSE, GGS College of Modern Technology, PTU, Jalandhar , Punjab, India

ABSTRACT

In recent years, multi-agent systems have attracted increased interest of researchers. These multi-agent systems are composed of interacting entities known as agents that work together in a concurrent systems to solve a complex problems. These problems can be further subdivided and distributed among other agents to increase the problem solving speed. The vast amount of research has been done for agent communication, along with the semantics of agent communication. As communication is an important aspect for agent interaction in multi agent systems, this paper provides an introduction for secure communication between agents residing in an heterogeneous or distributed environments. In fact, an analytical study of literature indicates that researchers have made attempts to lay a smooth floor for the working components of semantic web but the same careful investigation also reveals the fact that the floor laid is not yet smooth. In this research paper, an attempt is made to present a standard protocol for KQML named as Contract Net Secure Establishment protocol (CNSEP). The protocol is used to perform a communication between two or many multi-agent systems. The major emphasis is done on the security of messages that is being transferred or shared between agent systems. This protocol has been developed using Contract Net Trust Establishment Protocol (CNTEP) as a reference. The main aim of this research work is to enhance existing CNTEP protocol by imposing security mechanism in it and for this a new component i.e. Security agent has been introduced.

Keywords : Multi-agent System, KQML, CNTEP

I. INTRODUCTION

The KQML is an agent communication language and a protocol that is proposed to provide communication among intelligent software agents. The need for interaction between software agents have been originated from highly dynamic, heterogeneous systems consisting of large number of autonomous nodes [1]. KQML is a high level protocol that deals with communication interpretation instead of bit streams transmission over the network system. In a distributed MAS, KQML is actively used for the interaction between agents. Being high level protocol, it is used to provide a structure that allow system designers decide 'what agents should say' while communicating instead of 'how they should say'. In a distributed system, for task execution, KQML is used to facilitates efficient inter node communication and also allows participation of agents for competitive negotiations in order to complete that task execution. For this, each agent is assigned with either manager role or contractor role within the system. A contractor agent is the one who participates for the actual execution of the tasks. And it's a manager agent who sits in the system to assign the tasks, monitor these tasks and process the result of its execution completed by contractor agents. The task message is provided by manager agent to all the contractor agents.

Thus, through the use of contract net which is a mean to create contracts as well as sub-contracts, a task is distributed among agents. This process of Contract Net is standardized under 'Contract Net Protocol' [2] [3]. The CNTEP [4] incorporates trust establishment mechanism in the existing CNP (Contract Net Protocol).

A. CNTEP Components

1. Initiator agent (IA)
2. Contractor agent(s) (CA)
3. Trust Establishment Protocol (TEP) which further consists of a :
   - ***a.*** Trust Verification agent
   - ***b.*** Agent Registration List
   - ***c.*** Trust Matrix

A. CNTEP Working

The Initiator agent (IA) requests CFP (Call for Proposal) to the respective Contractor agent (CA). In order to serve the request, the Contractor agent(s) must possess the 'Trust Certificate'. For this, they executes Trust Establishment Protocol. In Trust Establishment Protocol (TEP), 'Trust Verification Agent' demands certificate from contractor agent(s) to authenticate them and verifies that CA (Contractor agent) is a registered agent or not with either FIPA or any other organization.

Once Contractor agent(s) share the certificate with Trust Verification Agent (TVA), the TVA verifies the certificate and checks that this Contractor agent is a registered agent or not by looking into 'Agent Registration List' (ARL). And based on the response retrieved from ARL, the trust percentile of Contractor agent(s) is computed.

In enhancement to CNTEP, there is one more protocol is developed i.e. **'Reliable Contract Net Establishment Protocol'** [5] that actually worked on computing the reliability of Contractor agents. As per this, reliability of participating agents is very crucial for better communication in open multi-agent systems. As agents can join and leave the system dynamically, there is a possibility that these agents are unreliable, self interested. Before defining the actual working of RCNTEP, below are some basic terms related to this protocol.

- **TPC:** - TPC stands for Trust Percentile Certificate that is an official document of trust for an agent given by CNTEP. Without this document no agent is permissible to send their bids to composite/initiator agent.
- **RV:** - RV stands for Reliability Value of participating agent is calculated by initiator agent. It ranges between possible set of values that is {…..-2,-1, 0, +1,+2…..}. If RV of any agent exceeds the threshold then the corresponding agent is permanently blocked to bid anymore.
- **Condition:** - Condition is the clause of agents that they require at the time of task execution.
- **Result:** - Result is the fallout of an agent given after the task is executed.
- **Time:** - Time is the total duration of completion the task. This attribute can take three possible values that are Max for utmost time, Min for less time and Mod for modulate timing.

A. Reliable CNTEP Components

1. Initiator agent
2. Contractor agent
3. Bid Evaluation agent consists of 'Bid Evaluation Table'

B. Reliable CNTEP Working

As this is an extension of existing CNTEP, here, after receiving Trust Percentile Certificate from CNTEP, the contractor agents make a packet of 'Trust

Percentile Certificate' and its bid called 'Bid Packet' (BPs) and send this to Initiator agent from where it has received Call for Proposal (CFP). Initiator agent (IA) forward this BP's to BEA (Bid evaluation agent) demanding for most reliable agent. In order to evaluate the reliable agent, BEA (Bid Evaluation Agent) evaluates BET (Bid Evaluation Table) that consists of parameters such as TPC, RV, Condition, Result, Time.

The most reliable agent is returned using following formula:

$$Max(TPC + RV) + True\ (Condition) + Min\ (Time) + Ok\ (Result)$$

And finally the Initiator agent awards the task to returned agent from BEA (Bid Evaluation Agent). If the result are Ok from rewarded agent, then its RV is increased by +1, otherwise it is decreased by -1.

The figure 1 represents the high level view of RCNTEP.

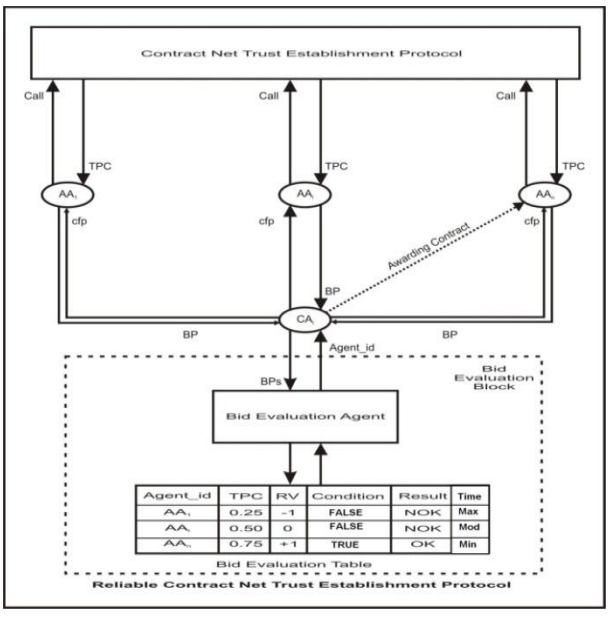

Figure 1: High Level View of RCNTEP

The organization of this document is as follows. In Section 2 (**Methodology**), the detail of proposed protocol i.e. CNSEP is presented using CNTEP and Reliable CNTEP as a reference. In Section 3 (**Results and Discussion**), performance of proposed protocol is evaluated. In section 4 (**Conclusion**), a final conclusion is provided for the work done in this protocol.

II. METHODOLOGY

Using these existing protocols i.e. CNTEP and Reliable CNTEP, this research paper works on enforcing security mechanism in these protocols by introducing CNSEP (Contract Net Secure Establishment protocol).

In order to achieve this, a new component 'Security agent' is introduced in this protocol that acts as a mediator between Initiator agent and Contractor agent(s). Along with this, to an existing BET (Bid Evaluation Table) of BEA (Bid Evaluation Agent), a new parameter is introduced i.e. Secure Algorithm that states what algorithm is used to encrypt the messages transferred between Initiator agent and Contractor agent. The stronger the algorithm, the more would be a chance for Contractor agent to assigned with task from Initiator agent in the communication process. The figure 2 below represents the high level view of CNSEP.

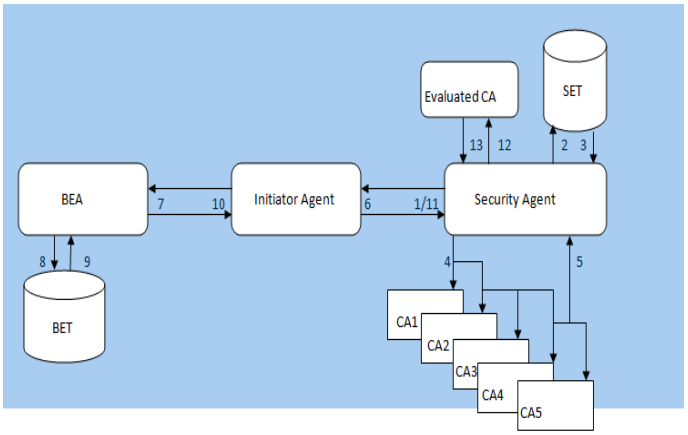

Figure 2: High Level view of CNSEP

The steps in above figure 2 are described below in the form of algorithm followed by the protocol:

STEP 1 : Initiator agent (IA) sends CFP(Call for proposal) to Security agent;
STEP 2 : Security agent searches Contractor agent in SET (Security Establishment Table);
STEP 3: Results retrieved from SET. If Contractor agent exists in SET and has not expired, move to STEP 6; else move to STEP 4;
STEP 4 : Security agent sends CFP to all Contractor Agents (CA1-CA5);
STEP 5 : All Contractor agents send their bid packets with trust percentile certificate to Security agent;
STEP 6 : Security agent sends bid packet from Contractor agent to Initiator agent;
STEP 7 : Initiator agent sends the bid packet further to BEA (Bid Evaluation agent);
STEP 8 : Bid Evaluation agent evaluates the received bid packet from all Contractor agents by looking into BET (Bid Evaluation Table);
STEP 9 : BET returns the most appropriate Contractor agent to Bid Evaluation agent based on certain parameters such as TPC, RV, Result, Condition, Time, Secure Algorithm;
i.e. Max(TPC) + Min(Time) + Best(Secure Algorithm) where Result is Ok and Condition is TRUE.
STEP 10 : Bid Evaluation agent sends the received Contractor agent from BET to Initiator agent;
STEP 11: Initiator agent sends the contractor agent details to security agent;
STEP 12 : Security sends the actual request received from Initiator agent to evaluated Contractor agent;
STEP 13 : Evaluated Contractor agent sends the response to Initiator agent via Security agent;

A. CNSEP Working

The detailed working of CNSEP is described below:

1. The Initiator agent requests CFP to Security agent. The Security agent verifies if any existing Contractor agent exists in 'Secure Establishment Table' who has been awarded contract earlier. If exists and has not been expired, the security agent directly receives the bid packet (a packet of TPC and its bid) from that Contractor agent and send it to Initiator agent. If expired or does not exist, it sends the CFP (Call for Proposal) to all active Contractor agents. Contractor agents receive the TPC from CNTEP, form a bid packet and send it back to security agent.
2. Security agent further sends the bid packet to initiator agent.
3. Initiator agent sends the received bid packet to BEA (Bid Evaluation Agent) that consults the BET (Bid Evaluation Table) to find the most suitable and secure Contractor agent using existing parameters (TPC, RV, Condition, Result, Time) and SA (Secure algorithm). Once computed , it shares the most reliable Contractor agent to Initiator agent.
4. Initiator agent shares the private token to the respective CA (Contractor agent) through Security agent.
5. Security agent maintains the details of private token and Contractor agent in Secure Establishment Table (SET) with expiration time.
6. And then after sharing the private token details to Security agent, the Initiator agent sends the random string request to Contractor agent in order to verify that it is an authenticated CA or not. If CA successfully decrypts this random string request, then IA sends the actual encrypted message to CA which CA will decrypt at its side and then send the response in encrypted form to the Security agent that communicates further the result to IA (Initiator agent).

The whole secure communication between Initiator agent, security agent and contractor agent is done by using set of KQML performatives [6] listed below [7].

1. **auth-link-request :** The Initiator agent asks the Contractor agent to send an auth-link and start

the authentication process.

2. **auth-link :** The Contractor agent wishes to authenticate itself to Initiator agent and set up a session key and message ID.
3. **auth-challenge :** The Initiator agent challenges the identity of Contractor agent in response to an auth-link. For this, Initiator agent sends an encrypted random string using private token and send to Contractor agent.
4. **reply :** The Contractor agent replies the response to Initiator agent for the encrypted random string.
5. **reply/error :** The Initiator agent if satisfied with Contractor agent response, sends the actual encrypted message otherwise reply with an error and terminate any further message communication.

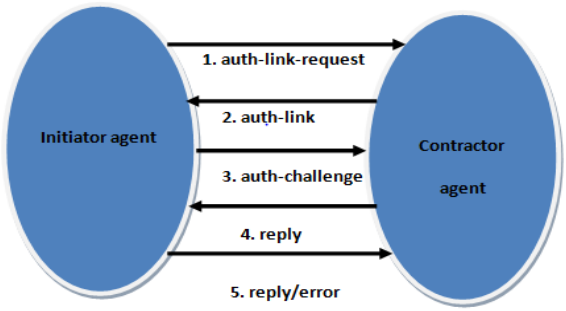

Figure 3: Authentication KQML performatives

III. RESULTS AND DISCUSSION

Multi-agent systems have become an inseparable component of KQML. CNTEP work evaluated multi-agent framework employed in cellular Network using co-ordination, performance, scalability and security as an evaluation parameter.

In this protocol i.e. CNSEP, as 'Security Agent' is employed that maintains the Security Establishment Table (SET) with Contractor Agent who can serve the request for particular period of time in session, thus the communication cost involved in serving the call for proposal by communicating with all agents again and again will greatly reduce. Also, the response will be generated at faster pace as the Contractor agent who has to serve the request is already selected in the system.

In order to compare the response time generated from existing Contract Net Trust Establishment Protocol (CNTEP) and this newly generated Contract Net Secure Establishment Protocol (CNSEP), JMETER tool has been used. Performance testing is done by considering 500 users hitting the application. From Figure 4 and Figure 5 result graphs, it is visible that CNSEP takes half the time then CNTEP during its execution.

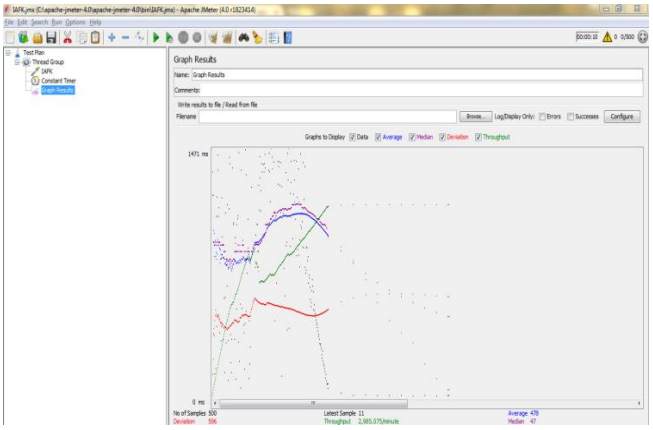

Figure 4: CNTEP (Contract Net Trust Establishment Protocol) Response Time (1471ms)

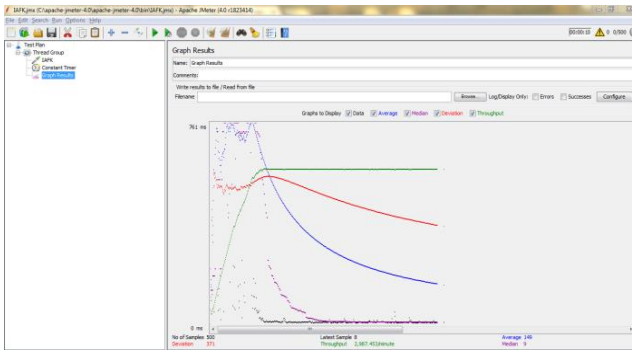

Figure 5: CNSEP (Contract Net Secure Establishment Protocol) Response Time (761 ms)

And Figure 6 represents the result graph showing complete execution time required by Contract Net

Secure Establishment Protocol i.e. selecting Contractor agent and then encrypting the message and getting response back from Contractor agent. From figure 6, it is evitable that even when 500 users hit the application, the total response time lies around 4 seconds.

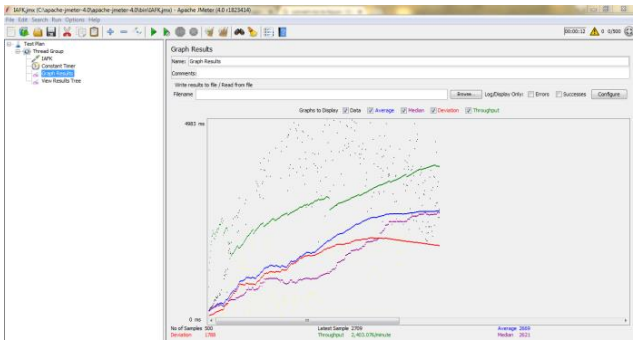

Figure 6: Total response time for CNSEP

In order to express the complexity of IAFK in general mathematical formula such as n, n^2, Log_2n etc., we must understand the tasks perform by CNSEP when any request arises. Below list represents the list of tasks performed by CNSEP:

1. Initiator agent requests CFP (Call for Proposal) to Security agent.
2. Security agent looks for Contractor agent in Security establishment table. If found, it results back Bid Packet with Trust Certificate to Initiator agent. If not found, security agent sends Call for proposal from Initiator agent to all Contractor agent which in return sends their bid packet to security agent.
3. Initiator agent receives the Bid packet from security agent and send it to Bid evaluation agent to find the most suitable Contractor agent from Bid Evaluation Table.
4. Once Contractor agent is selected from Bid Evaluation Table by considering all parameters such as Reliability Value, Security algorithm etc., bid evaluation agent sends the selected Contractor agent to Initiator agent.
5. Initiator agent sends the private token to Contractor agent through Security agent and finally communication is performed between them through encryption and decryption of messages.

In order to complete the task in complex distributed environment, the communication task needs to be sub-divided into n sub-tasks. This division leads to reduce the overall completion time for each task and also reduces the memory space requirements as task is sub-divided and is performed by descendants of Contractor agent separately.

Thus, the complexity of each task is represented as log_2n where n is number of sub-tasks. Also, this complexity is represented in one sub-system and as the number of sub-system increases, the complexity for one task will become $n\ log_2n$.

IV. CONCLUSION

This research work has presented 'Contract Net Secure Establishment Protocol' that provides the complete secure communication solution for multi-agent systems operating in distributed/ heterogeneous environments. It's an extension of existing Contract Net Trust Establishment Protocol. Its main aim is to upgrade the existing protocol with 'Security Agent' that acts as a mediator between Initiator agent and Contractor agent(s). Following the existing approach of contracting the jobs to agents based on reliability factor such as Trust Percentile Certificate, Result, Time etc., this security model includes the secure algorithm as an additional reliability parameter to determine the efficient contractor agent for performing the allocated task. Stronger the algorithm, the more it increases the secure communication over the network. Once the contractor agent is selected based on the existing and this new reliability factor, the initiator agent sends the encrypted message to

security agent and security agent further communicates the message to selected contractor agent. The security agent also stores the reliable contractor agent in Security Establishment Table for a specified time period in order to overcome the issue of selecting the contractor agent at each Initiator request. This increases the system performance and provide the timely output.

As this framework mainly focuses on the security of messages transfer over the network between agents, the security infrastructure of this framework can be further extended to handle network attacks such as 'Message replay' or 'Replay attacks' in which the attacker hacks the message from the authorized sender and replay the message again in future to retrieve some useful information from receiver which is not easily accessible.

V. REFERENCES

[1] Finin, T., Labrou, Y. and Mayfield, J. KQML as an agent communication language. In *Software Agent* , Bradshaw, J.M. (ed.), AAAI Press / The MIT Press, 1995, pp. 291-315.

[2] Reid G. Smith, December 1980, "The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver" The contract net protocol. *IEEE Transactions on Computers,* C 29(12)

[3] Kone, M.T., Shimazu, A., and Nakajima, T. The state of the art in agent communication languages. In *Knowledge and Information Systems* , 2000, pp. 259-284.

[4] Aarti Singh , Dimple Juneja and A.K.Sharma 2010, "Introducing Trust Establishment Protocol In Contract Net Protocol". In proceedings of IEEE conference on ACE'10 pages 59-63.

[5] Jagga Ankit, "A Reliable Contract Net Trust Establsihment Protocol", 2012 http://ir.inflibnet.ac.in:8080/jspui/handle/10603/10917

[6] McBurney, P. and Parsons, S. Games that agents play: A formal framework for dialogues between autonomous agents. In *Journal of Logic, Language and Information* , vol. 11(3), 2002, pp. 315-334.

[7] Chelliah Thirunavukkarasu , Tim Finin , James May Eld, " A Security Architecture for the KQML", 1995, *http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.198.7463*