A Trustable Software Agent Framework for the Examination Paper Preparation and Moderation Process (EPMP)

Abdulrazaq Nathim Abdulrazaq
Universiti Tenaga Nasional
Jalan IKRAM-UNITEN, 43009
Kajang, Selangor, Malaysia

Mohd Sharifuddin Ahmad
Mohd Zaliman M. Yusoff
Universiti Tenaga Nasional
Jalan IKRAM-UNITEN, 43009
Kajang, Selangor, Malaysia

Moamin A Mahmoud
Universiti Tenaga Nasional
Jalan IKRAM-UNITEN, 43009
Kajang, Selangor, Malaysia

ABSTRACT
This paper develops a trustable software agent framework to simulate the possible situations of trust between agents and propose novel solutions that could be provided by software agent technology. The technology is applied to the security and trust process to overcome the serious issues that could be faced by using software agent technology. This study demonstrates the use of autonomous software agents to secure the Examination Paper Preparation and Moderation Process (EPMP) domain and apply security and trust mechanisms on the agents that access and perform various domain tasks. Three levels of trust are designed, which correspond to the EPMP domain's groups of people who are the Examination Committee, Moderator, and Lecturer. Each level has its own trust requirements that should be fulfilled by the agent in order to access the level, otherwise the agent is blocked and access to the level is denied. This work implements the proposed framework for the domain and observes the results of the security and trust mechanisms. It then compares the performance before and after the implementation to see the effectiveness of the framework.

Key words
Intelligent Software Agents, Multi-agent Systems, Trust, Trustable Agents, Trustable Framework.

1. INTRODUCTION
In the world of the software intelligent agents, building trust is to make an intelligent agent trust another agent and delegate part of their tasks in a heterogeneous distributed multi-agent environment. In general, trust is "a generalized expectancy that the word, promise oral or written statement of another individual or group can be relied upon" [1].

In the environment of software agents, the agents could be depended upon to do as instructed. Trust is "the condition in which one exhibits behavior that makes one vulnerable to someone else, not under one's control" [2] [3].

Generally, trust toward a specific agent is generated through recognition and experience under repeated transactions with that agent. In addition, reputation plays a significant role in the software agent environment; it is the socialized trust which can be propagated through a social network of agents. It helps the agents to trust the target agent without any direct interaction with the target agent. The benefits of introducing trust and reputation into multi-agent system include [4]:

- Trust can eliminate and disregard much of unnecessary communications which are necessary in many interaction protocols thus greatly improve the performance of the multi-agent systems.
- An agent can decide in easily based upon the evaluation of the trustworthiness of another agent.
- Trust is a type of soft security which complements the traditional hard security like encryption, authorization, and authentication. An agent exists in complex heterogonous environment must possess both two securities in order to be safe and effective.

While trust could be a very useful tool, it cannot be applied under all conditions. In some circumstances, trust has no significance and meaning [5]. Reviewing the preconditions for the applicability of trust might give a better understanding of its function and lead to a better definition. The conditions are [6]:

- Uncontrollability: The more can control the entity that rely on, the more can predict its behavior; the more is capable of determining the end-result (and increase the knowledge about of the expected behavior). So, the less the entity is controllable, the more that needs trust to cope with uncertainty.
- Partially Monitored: This condition is related to the previous one. Monitoring everything the entity does, uncertainty about the quality of the end result is significantly reduced, thereby reducing the need for trust.
- Private Information: This is information that is private to an agent. Examples of private information are the agent's capabilities, its trusting-behavior (or decision making rules), and its disposition towards other agents. In the absence of private information, an agent would be highly predictable (the agent becomes a white box), making trust unimportant.
- Dependency: Fully independent agents do not need to rely on other agents to attain their goals. Dependency is therefore a requirement of trust.

Goal-Oriented: The utility (or welfare) of an agent is always relative to a goal. Without goals, nothing is at stake, and an agent could not care less if a task is performed well or not. Consequently, trust is irrelevant in the absence of goals [7].

This research aims to solve trust issues that arise in a specific domain of a university. In such domain, not all information should be widely accessible to everybody. The parties involved need full control on how their data are used, who has access and who are blocked. In Mahmoud [8] work, a collaborative framework in multi-agent systems is developed, the goal of which is to complete the examination paper preparation and moderation (EPMP) within a stipulated time. The EPMP domain uses intelligent software agents to do various tasks, which raises a trust issue. The issue here is trust with the other agents who are performing the tasks for the

requesting agent. If the other agents are malicious agents, then they could pose a threat and create problems in the domain.

Nowadays security issues such as counterfeiting, identity theft, piracy and hacking have spread widely. For example 'hacking' is a serious problem faced by organizations, companies, and even the governments of the world. In this era, stealing of confidential information from the organizations or companies can cause serious security problems and financial losses.

For a domain such as the EPMP, it is easy to hack and steal the examination paper, make changes, or even do damages on the document. The stolen examination paper would be a serious problem not just for the faculty but for the university. The university's reputation would be jeopardized and it would affect its reputation among competing universities. Consequently, a serious issue of trust and security in the domain arises and needs to be solved by applying some security or trust mechanism to avoid trust-related problems.

The objectives of this research are, (a) To explore the use of intelligent software agents in trust and security process; (b) To develop a framework for trustable software agent system based on the trust and security theories; and (c) To study, analyze, implement and validate the framework for trustable software agent system.

2. RELATED WORK

In order to sustain the interactions between software agents, trust and trustworthiness is a crucial requirement. Several research in psychology, sociology, economics and computer science viewed it as relevance. This section provides an extensive review on related trust models developed over the years which implement trust in an intelligent agent system.

Several well-known online trading Web sites such as eBay, SPORAS, TRAVOS, and Amazon are among various reputable systems developed so far. Reputation element in these systems is a cumulative rating evaluation for both parties in trading. Nevertheless, these elements have been known as very simple and are weak under some primitive attacks. The researcher who provides a trust model in dispersed systems with main focus on software agents is Marsh [9]. He modeled trust as a real number ranging from -1 to +1. However, as Marsh highlights, the model has some troubles in computing trust values at extreme values as well as zero ones. The model also has some limitations in evaluating trust as a negative value. As a matter of fact, the algebraic notations and operations that have been delivered are not competent enough to handle the negative values.

Castelfranchi [10] led a research group on trust which advocates a socio-cognitive approach to trust. They highlighted that trust is a behavior, decision and a mental attitude towards another agent [10].

In daily life, trust is a mental perception where it is based on measurement of past actions as well as on the expectation of future actions.

As a decision (the act of entrusting a task), trust puts a part of the trusting agent's welfare on the line and thus involves risk. Nevertheless, the favorable transaction history with another agent does not guaranteed that this will continue in the future.

Finally, trust can also be regarded as behavior that emphasizes on the actions (delegating and monitoring) of trusting agents and the relation between them. This relation usually strengthens as time progresses.

Yu and Singh [11] highlighted the possibility of applying Dempster Shafer evidence theory. Reputation is measured by the rating propagation and considering the agent's neighbors. The propagated standards are prejudiced by the neighbors. They model trust by using recommender systems. On the other hand, Schillo [12] proposes to build a TrustNet based on adopting game theory and using probability theory together for updating beliefs about other agents.

Furthermore, Mui et al. [13] highlighted an interesting probabilistic model for reputation based on Bayesian network. They defined a framework based on game theory for understanding the relative strength of dissimilar notions of reputation. Zacharia and Maes [14] published a framework where agents involve in some communities and developed a temporal kind of reputation assessment based on the performances and recommendations.

Namin et al. [15] proposed an efficient trust model based on collecting encrypted hash values via a defined view named as "Trust Path". It is a combinatorial method in measuring objects number as well as adopting basic cryptographic techniques to handle a secure counting that resists against some attacks. In fact, attacking and breaking the model in terms of cryptography is computationally hard. This characteristic arises from the one-way property of hash functions.

3. THE EXAMINATION PAPER PREPARATION AND MODERATION PROCESS (EPMP)

The Examination Paper Preparation and Moderation Process (EPMP) [16] is a work process developed to prepare and moderate examination question papers in the faculty of Information Technology. The EPMP consists of an Examination Committee, Lecturers and Moderators who are collectively responsible for the implementation and completion of the work process. The EPMP is a very important process because it deals with students' examination papers. The procedures associated with it must be implemented in an efficient and timely manner. The EPMP is normally activated twice in every academic year stipulating the deadlines for starting, document submissions and completion. The committee, moderators and lecturers must cooperate between themselves manually to implement the tasks completely. The peculiarity of the process is such that members of the faculty can be the committee, moderators and lecturers at the same time.

The process starts when the Examination Committee sends out an instruction to lecturers to start prepare examination papers. A Lecturer then prepares the examination paper, together with the solutions and the marking scheme (Set A). He then submits the set to be checked by an appointed Moderator.

The Moderator checks the set and returns them back with a moderation report (Set B) to the Lecturer. If there are no corrections, the Lecturer submits the set to the Examination Committee for further actions. Otherwise, the Lecturer needs to correct the paper and resubmit the corrected paper to the Moderator for inspection. If corrections have been made, the Moderator returns the paper to the Lecturer. Finally, the Lecturer submits the paper to the Committee for further processing. The lecturer and moderator are given deadlines to complete the process. Figure 1 shows the process flow for the EPMP.

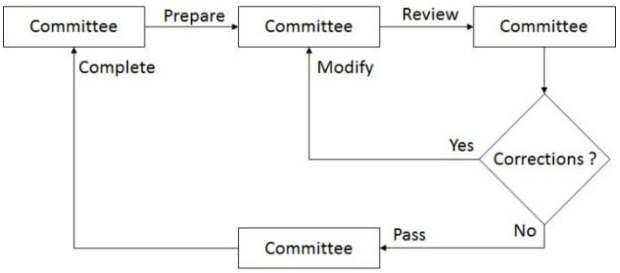

Fig 1: The EPMP Process Flow

Lack of security and threats from unauthorized or entrusted agents could make the EPMP vulnerable to attacks. The EPMP agents use the Transmission Control Protocol/Internet Protocol (TCP/IP) suite and utilize the communication ports to pass and receive messages from other agents without any security, which would raise issues for the domain.

When a specific agent is requested to provide one of the services it can offer, it can effectively provide the service and act in a benevolent way or, on the other hand, it can provide a bad service, acting fraudulently or maliciously. This agent which behaves as such is an intruder agent.

4. THE FRAMEWORK DEVELOPMENT PROCESSES

The steps to develop the framework are discussed below:

- **Initial Trust:** In the proposed system, an Initial Trust is conceived as an organization which needs software agents to perform various tasks on its behalf and obtainable from a trusted supplier. The supplier is initially assumed to be trustworthy and have the following qualities: Security of supply, consistency of products, cooperation and innovation. To confirm the assumption, the organization consults another organization that trusted the supplier verify that the supplier is trusted and could be relied upon. With these characteristics and qualities, the supplier is believed to be trustworthy.

  After creating the agents, the supplier sends the numbered agents to the organization's administrator to undergo "Agent Registration".

- **Agent Registration:** In this step, the administrator receives the agent from the supplier. The agent is registered and given a Name, ID, and a Secret Answer (SA). The SA is a phrase which an agent uses to respond to a question when it attempts to access a particular domain. The agent is given a very short time to answer the question to access a specific domain.

- **The Trust Model:** This section proposes a model of trust in software agent system implemented on the EPMP domain. The levels of the domain's security are classified into:

  - **Regular level:** It is the lowest security level in which the agent can access resources with minimum information. In this level, when the agent needs to access the environment of (EPMP domain) the agent must provide its Name and ID. When the agent gives its credentials (Name and ID) the domain checks whether the credentials are valid. If the credentials are valid, the agent is allowed entry into the domain.

  - **Secret level:** The Secret level is the intermediate security level in which the agent is required to give additional information because of tighter security at this level. In this level, there are two domains: Lecturer and Moderator domains. To access these domains, the agent provides its Name, ID and the Secret Answer. The domain checks if the agent's information are valid. In case the agent's information are false, it is blocked and denied entry into the domain. Otherwise, the agent is allowed entry into the domain. The agent should provide the secret answer within a short period, otherwise it is blocked.

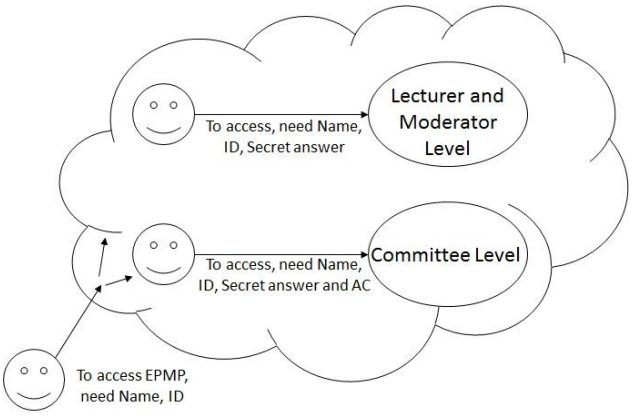

Fig 2: Access levels

  - **Top Secret level:** The Top Secret level is the highest security level in which the agent access by giving accurate information because of high security content. This level is the Examination Committee (EC) level, which has the most responsibility within the domain and the entire environment. When the agent needs to access the domain, firstly it enters its Name, ID, and the secret answer. The EC agent checks if the information are valid. The difference between the Top Secret level and the other levels is that the Top Secret level has an additional procedure of granting the agent the authorization to access the domain. The EC agent sends a request to the administrator to solicit information on the status of this agent whether it is a super-agent or a normal agent. If the agent is super-agent, the EC agent generates an Access Code (AC), which is a unique digit random number (maximum 9 digits), and sends the AC directly to the agent, which it uses to access the domain. Otherwise if the agent is a normal agent, it is denied entry into the domain.

5. IMPLEMENTING A TRUSTABLE AGENT FRAMEWORK

The design is based on the following stages.

- **Supplying Agents (Initial Idea):** The idea of agents as entities that are created and supplied to a receiver organization is conceived, in which an organization that needs software agents to perform various tasks on its behalf can get them from a trusted supplier (S). The supplier, in this case, must be trustworthy and has qualities such as security of supply, consistency of products, and cooperation. The supplier creates and sends the agents to the organization. Each agent is given a creation number:

S supply (C) ^ (a)
(C ∪ a) → A

where (**S**) is the supplier, (**C**) is creation number given to an agent, (**a**) is a normal agent, and (**A**) is the agent with the creation number.

- **Supplier:** A supplier, (**S)**, is an organization or person that creates and provides agents to others.
- **Agent (a):** An agent, (**a)**, is defined as an entity which performs a set of actions autonomously and continuously in a particular environment to achieve its goals.
- **Agent (A):** It is the agent that has been tagged with a registration number and is ready to register.
- **Creation Number (C):** It is a unique number that is given by the supplier to the agent in order to distinguish it from other agents.

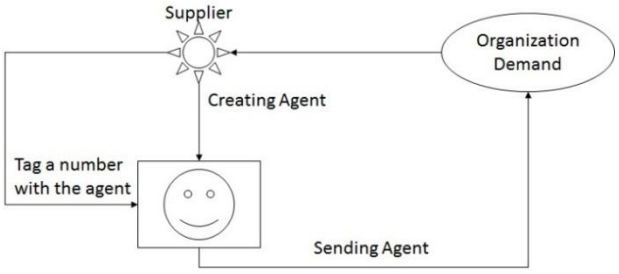

Fig 3: Supplier and Agent

The organization's administrator receives the agents and the registration process follows.

- **Registration:** The administrator receives the agents from the supplier with a list of agents' creation numbers and checks the agents and their numbers. If they match, the agents are allowed to register, otherwise, they are blocked.

  If: $A_1, A_2, \ldots\ldots, a_n \in L$, *then proceed to register*
  Or: $a_1, a_2, \ldots\ldots a_n \notin L$, *then block the registration*

  Where (**A**) is the agent with creation number from the supplier, (**L)** is the list**;** (**a**) is a normal agent without creation number.

The list (L) contains the agents and their creation number, this list is sent by the supplier and received by the administrator.

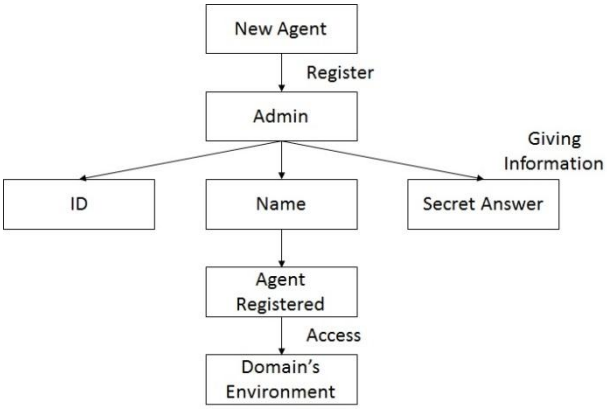

Fig 4: Registration Procedure

The agents' registration process requires the agents to register their Name, ID, and Secret Answer. Upon registration, the agents are ready to do their tasks as stipulated by the EPMP process, which entails the agents to enter the domain and then to a particular level in the domain that needs them. There are three levels that the agents can enter. The following illustrates each level and how the agent can access it.

- **Domain level (EPMP Domain):** The domain level is considered as a regular level (low level) because it is the lowest security level in which the agent can access resources with minimum information.
  - Low level: As a part of the trust model, this level is vulnerable to an attack with low-level risk. Attacks may come from malicious agents (**ma**) or other resources.
  - Attack: An attack is a sequence of cooperation and defection used by a malicious agent, **ma**, to achieve or maintain a trustworthy status as maintained by a trusted agent, (**ta**), with which it is interacting.

    When the agent needs to access this level (EPMP domain), the agent must enters its Name and ID, after which the agent can directly access the EPMP domain.

    If A = [ID ∪ N] True ⇒ ta, Then access the domain

    Else A = [ID ∪ N] False ⇒ ma, Access denied and block agent.

    The agent, which accesses the domain is considered as a trusted agent (**ta**), where: (**A**) is the agent, (**ID**) is the identification (random number) and N is the user name.
  - Trusted agent (**ta**): Is the agent who has gained the trust from the domain to do tasks inside the domain with full security and trustworthiness.
  - Malicious agent (**ma**): Is the agent which intentionally attack, disrupt and destroy trust relationships, businesses or services and put them under risks. Figure 5 shows the process of accessing the EPMP domain.

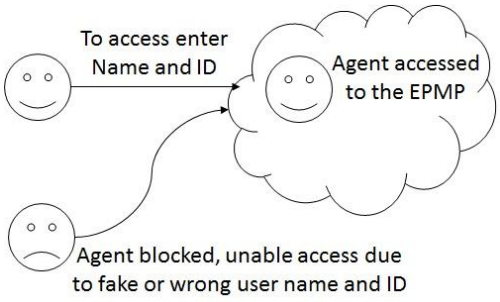

Fig 5: Agent accessing EPMP domain

- **Secret level (Lecturer and Moderator level):** The secret level is the intermediate security level, which is also vulnerable to an attack. The agent access resources by giving more information than the regular level because of tighter security at this level. In this level, there are two domains considered: Lecturer and Moderator. In this case, the agent should provide its Name, ID and the Secret Answer.
    - **Secret Answer** (**SA**): It is a phrase created from random letters. This phrase is used by the agent to gain access to the Secret Level.

      If A1 = [ta1∪ SA] True ⇒ ta2, Then access the domain

      Else A1 = [ta1∪ SA] False ⇒ ma, Access denied and blocked agent.

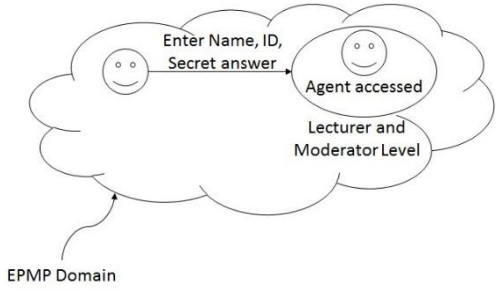

Fig 6: Agent accessing Lecturer and Moderator level

- **Top Secret level (Committee level):** The Top Secret level is the highest security level in which the agent can access by giving accurate information. This level applies to the Examination Committee (**EC**) level of the EPMP domain which has the most responsibility within the domain and the entire environment. When the agent needs to access the Committee level, there is a procedure which the agent has to do in order to gain access to the level.

  In the Committee level, there is an agent who controls the level (EC agent). When the agent requests for access, the (**EC**) agent sends a request to the administrator to confirm whether this agent is a super-agent or a normal agent. If the administrator responds with a super-agent, the (**EC**) agent generates an Access Code (AC) and sends the AC directly to the agent, which the agent use to access the domain. Otherwise, the (**EC**) agent will not generate an Access Code.
    - **Access Code (AC):** It is a unique random number (maximum 9 digits) generated by the (**EC**) agent. This AC is used by an agent to access the Committee level**.**
    - **EC Agent:** It is an agent who controls the Committee level. This agent is responsible for generating Access Code (**AC**) for other agents.

      If A2 = [ta2 ∪ SA ∪ AC] True ⇒ ta3. Then access the domain

      Else A2 = [ta2 ∪ SA ∪ AC] False ⇒ ma, Access denied

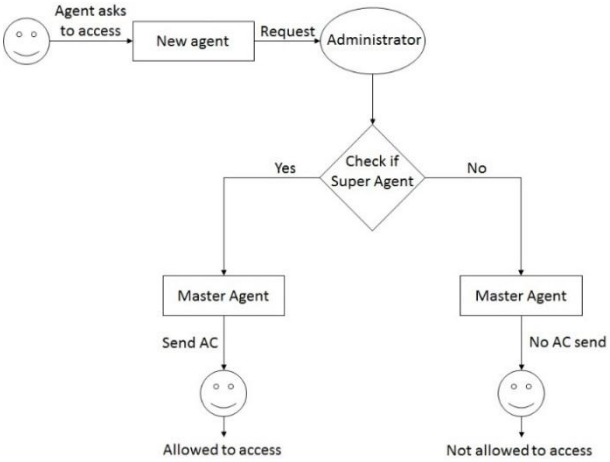

Fig 7: Agent accessing Committee level

6. TESTING

The test explains the framework's concepts by clarifying the code in separate stages.

- **Creating Agents:** Creating agents involves giving them the registration number. The code segments below shows:

  (Name, S_port, R_port)

  where (Name) is the initial name for the agent given by the supplier, (S_port) is the sending port, which is the registration number, and (R_port) is the receiving port. The list of agents and their registration numbers (S_port) are saved in a text file.

- **Registration:** When entering the information of the new agent in the registration, the agent is given a new Name, ID, and Secret Answer.

  (Name, S_port, R_port)

  The following information is based on the code implementation:

  Name: jon_bash
  Id: 37
  Secret Answer: ground_zero

  The agent is now ready to enter the EPMP domain and perform the required tasks.

  In case there are malicious agents that try to register, they are not able to do so because they do not have the registration number (port number). If any attempt to enter the domain with faked number, they will be blocked.

  They will also be recorded in the block list, meaning that those malicious agents will not be able to try any

attempts in the future to access the domain because they are in the block list.

- **Accessing EPMP Domain (Regular Secret Level):** After the agent gets it's Name, ID, and Secret Answer it is able to access the EPMP domain. To access the EPMP domain, the agent needs to enter the Name and the ID as shown in the code segment below:

  access_epmp (Name,S_port,R_port,User_name,User_id)

  Given the information for the agent that registered previously, e.g.:

  (a,30,100,jon_bash,37).

  The result of this entry is:

  welcome (`Access Allowed to EPMP Domain`)

  which means that the agent is trusted and allowed access to the domain; otherwise the agent is not trusted and is not allowed to access:

  sorry(`Access Denied to EPMP Domain`).

- **Accessing Lecturer and Moderator level (Secret level):**
  In this level, if the agent has a task to perform, the agent needs to enter the Name, ID, and the Secret Answer. The Secret answer as defined earlier is a phrase created from random letters, which is used by the agent to gain access to the secret level.

  An implementation of such entry is as shown below:

  access_l_m(Name,S_port,R_port,User_name,User_id,User_secret_answer)

  Giving the information:

  (a,30,100,jon_bash,37,ground_zero).

  If the information is correct, the result is:

  welcome (`Access Allowed to Lecturer and Moderator Domain`);

  Otherwise, the agent is not allowed access as shown below:

  sorry (`Access Denied to Lecturer and Moderator Domain`).

- **Accessing the Examination Committee (Top Secret Level):** The Top Secret level is the highest security level at which the agent can access by giving accurate information because of the high security content. In this level, the agent needs to enter the Name, ID, and the Secret Answer as in the Secret level plus an access code to access this level. If the agent enters the information:

  access_c(Name,S_port,R_port,User_name,User_id,User_secret_answer).

  (a,30,100,jon_bash,37,ground_zero).

  There is an additional condition required to enable the agent to access the level. When the agent enters the information in order to access the Top Secret Level, the Master agent is "an agent who controls the Committee level, this agent is responsible for generating Access Code (AC) to other agents." The Master agent sends the information of the agent who requests to access and check if this agent is a super-agent or a normal agent.

  If it is a super-agent then an Access Code (AC) is generated for it to use to access the level. Below is how the code is:

  get_ac(47564).

  Otherwise if the agent is a normal agent, it is not allowed access:
  sorry(`Access Denied to Committee Domain`)

7. RESULTS

The test results above demonstrate that the trust framework reduces the problem of security that surrounds the EPMP domain and the threats that could pose serious risks to the domain.

The implementation of the model uses Win-Prolog and its extended module Chimera, which has the ability to handle multi-agent systems [17]. The powerful logical inference capability of Prolog can be exploited to develop an inference engine specific to a particular domain.

The results show that any 'normal' agents are unable to access the EPMP domain in contrast with Mahmoud's work [8] where the agents can access the domain easily without any security. These agents could be malicious agents or viruses.

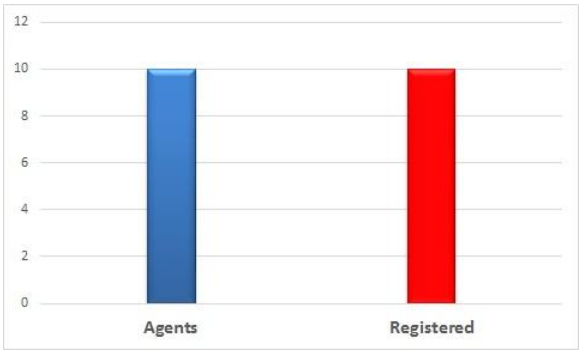

Fig 8: Agents registered in Mahmoud's work

Figure 8 shows that the agents that access and registered in Mahmoud's work are the same. It is inferred that any agents (malicious or non-malicious) can access the domain easily.

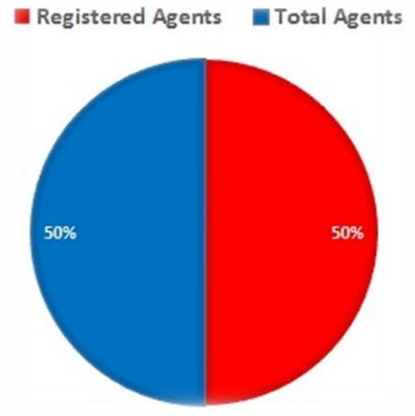

Fig 9: Percentage of Agents registered in Mahmoud's work

By implementing the trustable agent framework starting from sending the agents from the trusted supplier and during the registration process, accessing the three levels, the threats facing the EPMP domain are significantly reduced. Figure 10 shows the number of agents that registered and the agents that are not registered (blocked).

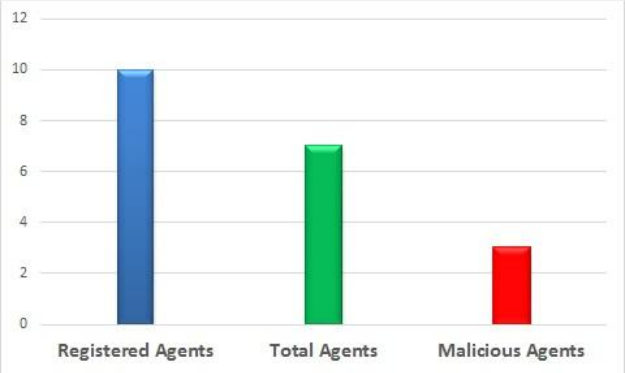

Fig 10: Agents registered and malicious agents

After the registration process, the agents that registered are ready to access the domain. In the Regular level (EPMP level) the agents can access this level by entering their Name and ID. These agents are not allowed to access the domain if the Name and/or the ID are/is wrong. This gives the domain greater security.

In the second level, the Secret level (Lecturer and Moderator level), the requirement to enter this domain is more than the first level; the agents can access this level by entering their Name, ID and Secret Answer. These agents are not allowed to access the domain if the Name, ID and/or the Secret answer are/is incorrect. This gives this level more security.

The third level is the Top Secret level (Examination Committee level) and the requirement to enter this level is more than the second level. The agents can access this level by entering their Name, ID, and Secret answer but they need the confirmation, i.e. the Access Code (AC). With the AC it can access the Examination Committee level, which is the Top Secret level.

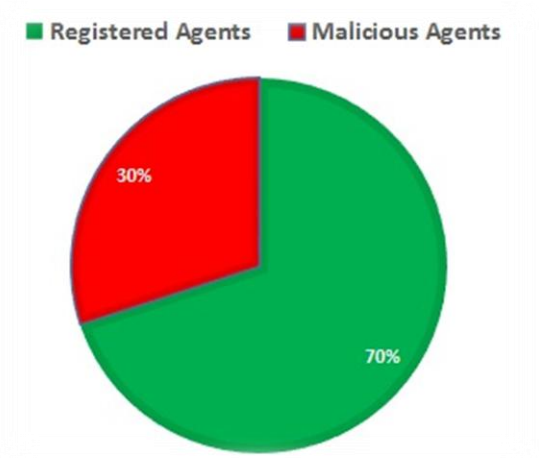

Fig 11: Percentage of Agents registered and malicious agents

Figure 12 shows that not all agents can access the Examination Committee level. Only the agents which are considered as super agents and have the AC have access to this level. This gives this level better security from malicious agents.

Fig 12: Super agents and Normal agents

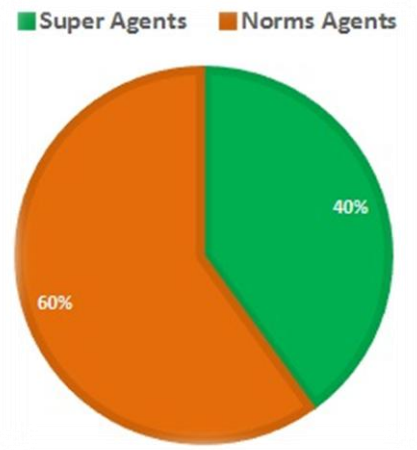

Fig 13: Percentage of Super agents and Normal agents

8. CONCLUSIONS AND FURTHER WORK

This paper develops a trustable software agent framework to overcome the issues of trust and security. The framework is developed based on the trust and security theories. It uses ID, Name, Secret Answer and the Access Code (AC) as security attributes. It then explores the use of intelligent software agents in trust and security process, studies, analyses, and implements the trust and security framework on the EPMP. Win-Prolog and its agent module, Chimera, are exploited to implement the Trustable Software Agents Framework.

In this framework, an agent attempts to access three levels of domain. The first level is called the Regular level, the second is the Secret level or Lecturer and Moderator level, and lastly the third level is the Top Secret level or Examination Committee level.

In each level there is a security and trust procedure that differs from the other two levels. In the first level, the agent needs to supply its Name and ID to access the level. In the second level, the agent needs to provide its Name, ID, and the Secret Answer, and in the third level the agent needs to provide its Name, ID the Secret Answer and the Access Code (AC) from the EC agent. In order to gain that AC, the condition is that the agent should be a super-agent, otherwise it is unable to gain the AC.

This research is inspired by the work of Mahmoud [8] who worked on the EPMP domain that used intelligent software agents to do various tasks, which raises the trust issue. The issue here is trust with the other agents which performs the

tasks for the requesting agent. Due to the autonomy of each agent, the agent could perform actions which may be construed as malicious to the other agents and pose a threat and create problems in the domain. To enhance the system, this work develops the trustable software agent framework, which overcomes the threat and security issues that are vulnerable to the EPMP domain.

The following points are based on observing the results before and after the implementation of the trustable software agent framework:

- The threats that are jeopardizing the EPMP domain have been reduced.
- Malicious agents and other threats are unable to access the domain without the trust information that is given to the trustable agents during the registration procedure. Before implementing the framework, accessing the domain level was easy for any agent.
- The domain and its levels are secured and agents and their corresponding human counterparts are safe to proceed with the work at their levels.

The use of Cryptography as a future work on the framework would increase the security of the domain. By implementing any of the Cryptography Mechanisms such as RSA, DSA or MD5 and encrypting the information that the agent uses to access the domain, would strengthen the security of the domain and reduce the attempts of malicious access by other agents to the domain.

9. REFERENCES

[1] Rotter J. B., "Interpersonal trust, trustworthiness, and gullibility," American Psychologist, vol. 35, no. 1, pp. 1, 1980.

[2] Zand D. E., "Trust and managerial problem solving," Administrative Science Quarterly, pp. 229-239, 1972.

[3] Patrick A., "Building trustworthy software agents," IEEE Internet Computing, vol. 6, 2002.

[4] Lu G., Lu J., Yao S. et al., "A review on computational trust models for multi-agent systems," The Open Information Science Journal, vol. 2, pp. 18-25, 2009.

[5] Stranders R., "Argumentation based decision making for trust in multi-agent systems," IEEE Transactions on Knowledge and Data Engineering, vol. 6, no. 1, pp. 64-71, 2006.

[6] Ramchurn S. D., Huynh D., and Jennings N. R., "Trust in multi-agent systems," The Knowledge Engineering Review, vol. 19, no. 01, pp. 1-25, 2004.

[7] Castelfranchi C., and Falcone R., "Social trust: A cognitive approach," Trust and deception in virtual societies, vol. 2005, no. 11 July, pp. 55-90, 2001.

[8] Mahmoud M. A., "Development of an application for a collaborative framework in multi-agent systems", College of Graduate Studies, UniversitiTenagaNasional, Selangor, 2010.

[9] Marsh S. P., U. o. S. D. o. C. Science, and Mathematics, Formalising trust as a computational concept: University of Stirling, 1994.

[10] Falcone R., Pezzulo G., Castelfranchi C. et al., "Why a cognitive trustier performs better: Simulating trust-based contract nets." pp. 1394-1395.

[11] Yu B., Singh M. P., and Sycara K., "Developing trust in large-scale peer-to-peer systems." pp. 1-10.

[12] Schillo M., Funk P., and Rovatsos M., "Using trust for detecting deceitful agents in artificial societies," Applied Artificial Intelligence, vol. 14, no. 8, pp. 825-848, 2000.

[13] Mui L., Mohtashemi M., and Halberstadt A., "A computational model of trust and reputation." pp. 2431-2439.

[14] Zacharia G., and Maes P., "Trust management through reputation mechanisms," Applied Artificial Intelligence, vol. 14, no. 9, pp. 881-907, 2000.

[15] Namin A., Wei R., W., "An Efficient trust model for multi-agent systems." pp. 1-6.

[16] Uniten, "Guidelines For Good Practice (Revised Version)," UniversitiTenagaNasional - College of Information Technology, August 2006.

[17] http://www.lpa.co.uk/chi.html.