Secure Multi Agent Information Filtering

Farzad Peyravi and Alimohammad Latif

***Abstract*—As the amount of accessible information is overwhelming, the intelligent information filtering systems provide helpful methods to access. The main problems are that the relevant information is spread over a big number of sources and useful information is hidden under the huge amount of useless data. We proposed a Secure Multi Agent Information Filtering (SMAIF) system based on agents and use of vector space model to find user interest in information groups supporting security issues on agents' communications. SMAIF reaches high performance in precession and recall in filtering the right information to users. Security issues in communications of secret data and users' privacy are considered in our implementation.**

***Index Terms*—Information filtering, JADE, Multi agent system, security.**

I. Introduction

With the rapid progress of computer technology in recent years, electronic information has been explosively increased. The number of books, movies, news, advertisements, and particularly on-line information, is staggering. The volume of things is considerably more than any person could observe or select in order to find the one that he or she would like [1], [2].

As the availability of the information increases, the need for finding more relevant information is growing. Human users accessing information utilize information retrieval (IR) technologies, such as search engines [3], [4]. Additionally, personalized information is provided by information filtering (IF) engines: Utilizing IF-based Recommender Systems, users are offered probably relevant items, (e.g. documents), based on similar items selected previously or further information stored in user profiles.

Information filtering agents [1], [5] are essentially new helpful tools in the ever-growing field of applications for user interfaces, which provide a comprehensive and optimal approach to operating by available information and sequential extraction of the required data for a user.

An agent usually [5]-[7] builds the profile using examples provided by the user. Thus the query is not a list of words to search for but rather combinations of words extracted from various examples.

There exists a lot of research in the field of multi agent systems in recent years [8]-[10]. It's because these systems define a new pattern in conceptualization, design and implementation of software systems. These patterns are very useful for the systems operate in open and distributed environments. The abilities of an intelligent agent are restricted only to its knowledge, computational resources, and perception [11], [12]. Evolving a multi agent solution is particularly suited to this domain since it provides the best possible match to the user's interests with the ability to quickly adapt in new situations. Moreover, security in multi agent systems has come up with many challenges due to the increased complexity of coordination between agents and interactions to user [12].

This paper describes a system named Secure Multi Agent Information Filtering (SMAIF) based on multi agent systems by use of vector space model to find user interest in information groups supporting security issues on agents' communications. By applying agents and their communications, system reaches high performance in precession and recall in filtering the right information to users. In our implementation security issues in communications of secret data and users' privacy are considered.

The outline of the paper is as follows. In Section II a brief overview about Information filtering system is summarized and then we demonstrate our designed model in Section III. Experimental results are shown in Section IV. Section V introduces security of the system and finally we conclude in Section V.

II. Information Filtering System

Information filtering system is one of the most generalized applications areas of intelligent user interfaces, which has only recently started to attract large attention. They deem that this system merely consists of a number of large information sources that should been filtered by some way to leave only information relevant to the user [2], [13]. Based on application subject, filtering systems can also be classified as cognitive, social or economic .Herewith; *Cognitive* systems choose documents based on the characteristics of their contents. *Social* systems select documents based on the recommendations and annotations of other users. *Economic* systems based on some computation of cost-benefit to the user and through some pricing mechanism [6].

The purpose of personal information [1], [6], [14] agents or assistants is to alleviate a user's information overload. This can be done by filtering information items (e.g. documents) according to a user's multiple and changing interests. Typically, an information agent builds a user profile, i.e. a representation of the user's interests, which is then used to assess the relevance of information items. Each item is assigned a relevance score, which is used, either to make the binary decision to present or not the item to the user, or to order the filtered items according to decreasing relevance. The profile should be able to adapt to changes in the user's

Manuscript received November 5, 2013; revised January 10, 2014.

Farzad Peyravi is with Electrical and Computer Engineering, Yazd University, Yazd, Iran (e-mail: fpeyravi@stu.yazd.ac.ir).

Alimohammad Latif is with Electrical and Computer Engineering, Yazd University, Yazd, Iran (e-mail: alatif@yazduni.ac.ir).

DOI: 10.7763/IJCTE.2014.V6.869

interests, which are caused by changes in the user's environment and knowledge. They are dynamic and can range from modest, but potentially fast, short-term changes, to occasionally radical, but more progressive, long-term changes. Adaptation is based on feedback from the user, which may be explicitly provided by the user, or implicitly gathered, e.g., by monitoring the user's activities.

III. System Design

A. Applying of Methodology

MAS-CommonKADS extends CommonKADS [15]-[17], for multi-agent systems (MAS) modeling, adding techniques from object oriented (OO) methodologies such as Object Modeling Technique (OMT), Object Oriented Software Engineering (OOSE) and Responsibility Driving Design (RDD) and from protocol engineering for describing the agent protocols, such as Specification and Description Language (SDL) and Message Sequence Charts (MSC96). The methodology defines the following models [18]:

Agent model (AM): specifies the agent characteristics: reasoning capabilities, skills (sensors/effectors), services, agent groups and hierarchies (both modeled in the organization model).

Task model (TM): describes the tasks that the agents can carry out: goals, decompositions, ingredients and problem-solving methods, etc.

Expertise model (EM): describes the knowledge needed by the agents to achieve their goals.

Organization model (OM): describes the organization into which the MAS are going to be introduced and the social organization of the agent society.

Coordination model (CoM): describes the conversations between agents: their interactions, protocols and required capabilities.

Communication model (CM): details the human-software agent interactions, and the human factors for developing these user interfaces.

Design model (DM): collects the previous models and consists of three sub models: network design for designing the relevant aspects of the agent network infrastructure (required network, knowledge and telemetric facilities); agent design for dividing or composing the agents of the analysis, according to pragmatic criteria and selecting the most suitable agent architecture for each agent; and platform design for selecting the agent development platform for each agent architecture.

B. Conceptualization

In this section we will describe the modeled system. The first step in this methodology is the conceptualization phase after which an elicitation task will be carried out to obtain a general description of the problem by following a user-centered approach based on use cases [18], [19]. In this approach, an actor represents a role played by a person, a piece of hardware or another system that interacts with our system. A use case corresponds to a description of the sequence of actions needed to produce an observable result useful for an actor. Table I defines the actors and their use cases modeled in our experiment.

TABLE I: Actors and Use Cases

Actors	Description	Use Cases
User	Person who works with system	Request Information / Check Retrieved Answers/ Feedback
Personal Agent	Manage user profile and user actions. Find best information matches users' profiles	Manage Profile / Find Group / Search / Answer to Query / Find Information /
List Agent	List all groups and collect information to update groups	List Groups / Collect Information / Update List /
Group Header Agent	Manage group operations, perform policies of groups. Send updated Information to List Agent	Manage Group / Send Updated Information / Update Group Information / Answer Received Queries
Database Agent	Retrieve documents and information form data bases	Retrieve Documents / Search Data Bases / Index Documents /

The outcome of this phase is a description of the different actors and their use cases. Table II describes a sample use case diagram.

TABLE II: Sample Use Case

Search Use Case	
Summery	Personal Agent searches in Agent List and finds the interested information based on User profiles
Actors	User, Group Header agent, List Agent
Precondition	Best selected information and sufficient information about the user interest
Exception	No interested information

Fig. 1 shows use case diagram for conceptualization phase.

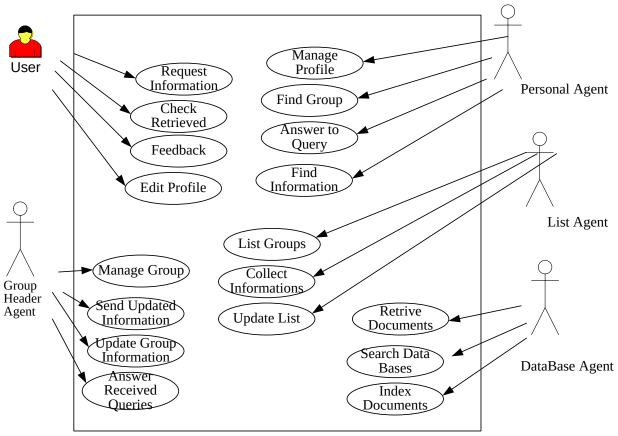

Fig. 1. Use case diagram.

C. Analysis

Central to the methodology is the agent model [12], [18], which specifies the characteristics of an agent, and plays the role of a reference point for the other models. An agent is defined as any entity – human or software – capable of carrying out an activity. The identification of agents was based on the use cases diagrams generated in the conceptualization. Such identification could be augmented in the task model. For instance, Table III presents the template for the list agent.

TABLE III: LIST AGENT TEXTUAL TEMPLATE

List Agent	
Name	List Agent
Role	Keep list of all groups, update information
Location	Inside agent society
Description	This agent manages the information about all groups and keeps list of all groups
Objective	Facilitate manages the information and create List and update it for new group
Exceptions	Group Names conflicts or absence of available information for create
Input Parameter	Group Name, Domain of groups
Output Parameter	New group, Selected group
Services	Search for specific group, create new group and update groups list
Expertise	Ability to search in group list and find by name, domain of practice and create or destroy groups
Communication	---
Coordination	Personal Agent, Group Header Agent

Let us now turn to the task model that describes the tasks that the agents can carry out. Since MAS-CommonKADS does not include a graphic structure for modeling tasks, we use UML activity diagrams to represent the activity flows and the textual template to describe the task. Fig. 2 shows the activity diagram for the list agent and Table IV demonstrates the textual template of a sample task.

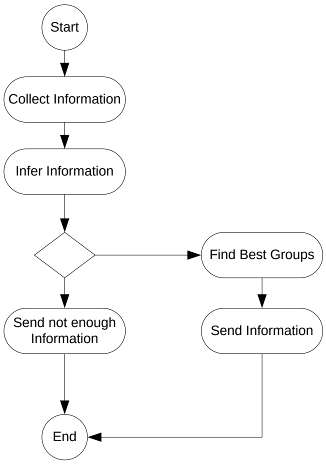

Fig. 2. List agent activity diagram.

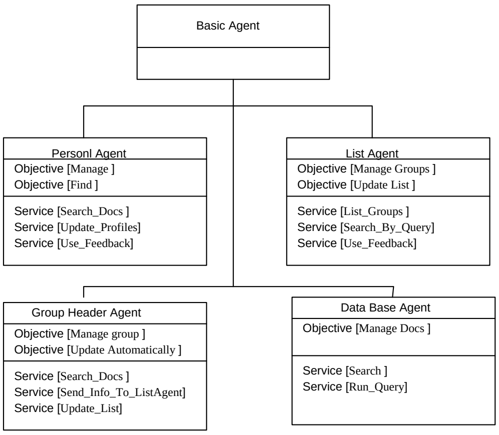

Fig. 3. Organization model.

This model aims to specify the structural relationships between human and/or software agents, and the relationships with the environment. In the organization model, we show the static or structural relationships between the agents. Following, we use a graphical notation based on OMT [12], [18] to express these relationships. Fig. 3 illustrates the class agent diagram for our agent system.

TABLE IV: DEFINITION OF INFER INFORMATION TASK

Infer Information Task	
Objective	Compare the information given by the Personal Agent to the information provided by Group Header Agent
Description	Getting the information from Personal Agent process the information and use them to determine best groups
Ingredient	Information about the profiles and list of related groups
Constraint	---
Exception	Absence of Personal Agent information or Group Header Agent information

Unlike the organization model, the coordination model shows the dynamic relationships between the agents. In this model we begin with the identification of the conversations between agents, where use cases play again an important role. At this level, every conversation consists of just one single interaction and the possible answer, which are described by means of templates as illustrated in Table V.

TABLE V: CONVERSATION TEXTUAL TEMPLATE (COORDINATION MODEL)

Search List Conversation	
Type	Requesting information
Agents	Personal Agent, List Agent
Beginner	Personal Agent
Description	List Agent receives the question and search in its list to match the best groups
Precondition	Selected groups
Post condition	Update list
Ending Condition	With interaction and receiving answer

D. *Search in System*

The challenge of finding the best information such as documents, news, email, advertising and etc. related to user's interests reduces to finding them by help of agents in our databases. The right information is grouped by their characteristics so that agents can search the right information easier. Finding them depends on cooperation of agents in their networks, So that in SMAIF, each agent helps its user maintain his personal desires and interests about his works, can be update by user or agent automatically. The nominal procedure is simple. A query from user or may automatically be sent his personal agent. The agent sends a query to the list agent. This agent has information about each group, their specifications and summery of the information they have. By use of list agent, user agent finds the right group and communicates with ach group through group header agent. Group header agent is responsible of managing group and be in touch with list agent to update its profile in list agent. Group header agent sends personal agent's request to its database agent in its group. When the right information retrieved, it will be sent to personal agent, so that user can see the desired information. Each group has a group header agent to update all information of group automatically or by use of human.

Each personal agent maintains a model or profile of user interests and each group has a profile in list agent who updates periodically or whenever group contents changes.

A query specifies what information is being sought. A response, if given, includes information or a referral to web sites or a portal. In Fig. 4 we showed the communication between each agent in proposed model.

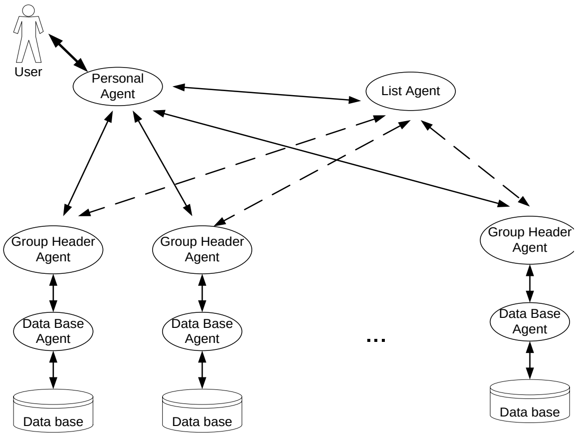

Fig. 4. Main model.

In paper we just showed some of our designed.

E. Modeling Expertise and Cooperation

Personal agent maintains a profile for its user and List agent keeps profile for each group. We capture this models via the vector space model (VSM) [7], [13] a classical information retrieval technique.

The vectors in VSM are term vectors indicating a weight for each term. We adapt VSM to locate user interests and desires [7], [20]. In our formulation, the terms correspond to different areas of interests of users and group areas activity. The interest of each user is modeled as a term vector so that group profiles. Similarly, the query is modeled as a term vector. In VSM, the similarity between two term vectors is defined as the cosine of the angle between them. We define the similarity between a query and group profile vector as the cosine of the angle between them, but scaled by the length of the group profile vector. Intuitively, for two groups with profiles in the same direction, the one with the greater relation is more desirable, whereas the traditional definition would treat them alike.

Definition 1: Given a query vector $Q = \langle q_1, q_2, ..., q_n \rangle$ and a group profile vector $E = \langle e_1, e_2, ..., e_n \rangle$, the similarity between E and Q is defined as:

$$Sim\left(Q, E\right) = \frac{\sum_{i=1}^{n} q_i e_i}{\sqrt{n \sum_{i=1}^{n} \left(q_i\right)^2}} \tag{1}$$

Definition 2: The relevance of a query vector Q to group profile P_j is computed as

$$Rel\left(Q, P_j\right) = \left(1-\eta\right)\left(Sim(Q, E_j)\right) + \eta S_j \tag{2}$$

where E_j is the interest of P_j, S_j is the cooperation of P_j, and η is the weight given to cooperation. This factor used when we have more than one list agent. In a huge systems or heterogonous system we will use many list agents so that they can communicate. This factor shows the relationship or social network which each list agent has.

In our experiments we showed the effects of η on the quality of relevancy of query vector and user.

Definition 3: Given a threshold ω_j, there is a match between user P_j and query vector Q if $Rel(Q, P_j) > \omega_j$.

IV. Experimental Results

Our test data used for SMAIF is come from standard Cranfiled. It has 1400 xml documents which contain document number, document title, authors and contents. A part of our program processed the file and inserted data (documents) in our database in a right format and terms are collected in ontology. We have 225 queries in standard Cranfiled data and the related documents (answers). Another part of our program transformed a query into term vectors.

We designed SMAIF with MAS-CommonKADS methodology and implemented by JADE toolkits. JADE is a toolkit for implementation of multi agent systems [21], [22].

A. Comparison in Simple and Agent Based Model

Two model of system is being tested, both simple (without agent) and agent based system. Precision is compared in 20 iterations. Fig. 5 shows the results. In agent based model we had 14.97% promotion in precision.

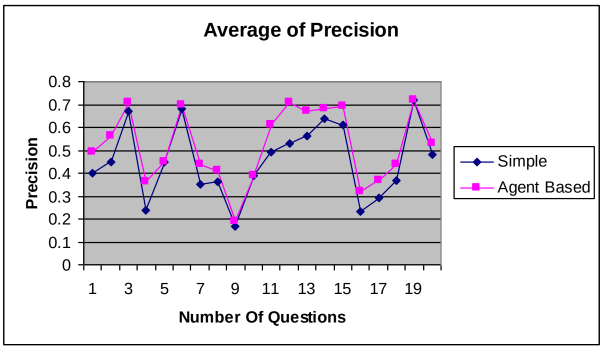

Fig. 5. Simple and agent based model.

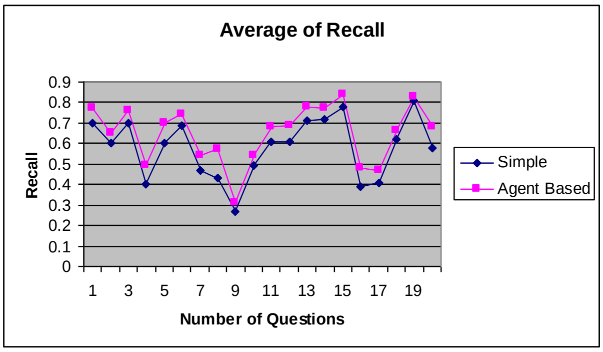

Fig. 6. Comparison on recall

In another experiment we showed average of recall between simple and agent based system. Fig. 6 shows that the agent based is 11.74% better than simple system.

The comparison on fall out of both systems has shown in Fig. 7 System in simple mode was 21.08% worst than agent based.

We had an experiment on number of groups in both systems. The result is shown in Fig. 8. When we have 5 groups, agent based system has 97.74% performance than simple system.

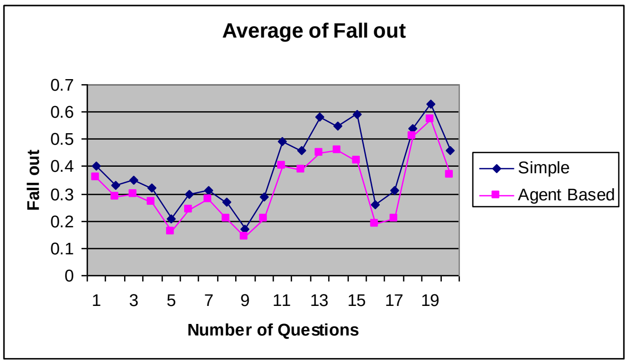

Fig. 7. Comparison on fall out.

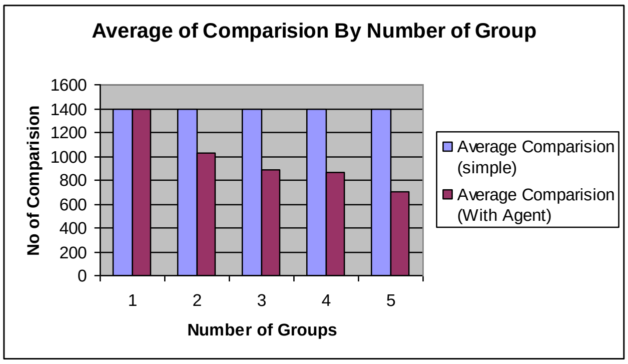

Fig. 8. Comparison on number of groups.

The last factor we had chosen to test is the importance of list agent. Fig. 9 shows that with list agent, 59.42% performance will be gained.

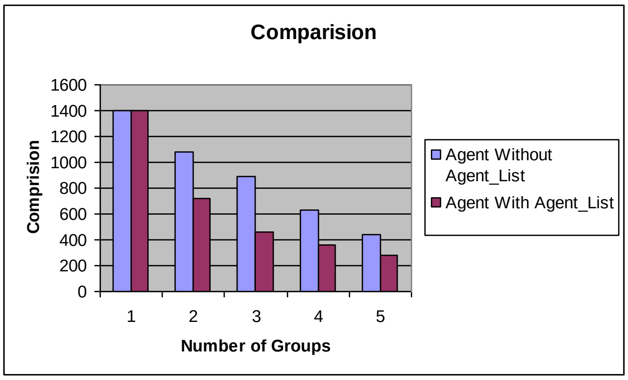

Fig. 9. Comparison on list agent.

V. Security in System

This section illustrated useful cryptography techniques due to support security issues in agent communications for example when Personal agent sends query to List agent or Group Header agent. Personal agent has to know aspects of the interests of the user that he or she might not be willing to share with other people. This situation brings forward several privacy issues in design of agents that assist the users. Thus to ensure the security subjects for agents communications we use one of the cryptography techniques, namely RSA. This algorithm provides these following security properties [23] , [25]:

- Confidentiality: assurance that communicated information is not accessible to unauthorized parties;
- Data integrity: assurance that communicated information cannot be manipulated by unauthorized parties without being detected;
- Authentication of origin: assurance that communication originates from its claimant;
- Availability: assurance that communication reaches its intended recipient in a timely fashion;
- NON-repudiation: assurance that the originating entity can be held responsible for its communications.

We use open PGP tool for encrypt and decrypt messages in planning agent of each department. This process is done as Fig. 10 [23], [24].

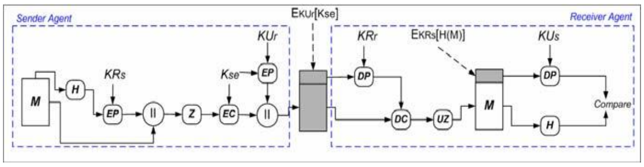

Fig. 10. Encryption/Decryption of messages using PGP.

The encryption/decryption process of messages between data communicating agents has the following steps:

1) Sender agent creates a message M.
2) SHA-1 is used to generate 160 bit hash code of message.
3) Hash code of message is encrypting using sender agent's private key.
4) This encrypted message concatenates with original message M and then zips them.
5) The zipped message is encrypted using session key that produced by sender agent.
6) Session key is encrypted with receiver's public key and concatenate with the message produced in step 5. the message produced in this step contain two part, the first part is session key which encrypted with receiver pubic key and the second part contain encrypted message using sender agent private key for determining authentication of sender agent and the original message that zipped and encrypted with session key.
7) Receiver agent decrypts the first part using its private key to obtain session key
8) Decrypt the second part using session key.
9) Unzip the message.
10) This message contains the original message M and a header for authentication check.
11) The header is decrypt with sender public key and compare to H(M) which denotes the hashed initial message for authentication check.

We used this process for important message such as queries including users' interest for list agent or group header agent. In SMAIF the encryption and decryption process is done between List Agent and Personal Agent, List Agent and

Group Header Agent and finally communication between Group Header Agent and Personal Agent. Each agent uses PGP for decryption of messages that must be sent in secure mode. For implementation we simulate them according to Fig. 4 topology. In implementations, PGP can be obtained by using "BouncyCastleProvider" [26] class in java.

SMAIF is configurable for testing data transmission in both security and non security manner and also different data requests conditions. It marks each packet send and receive time.

In experiment, two factors were considered; the first is average time spent for sending data from one agent to another one called "migration time" and the second is "response time" which is the sum of waiting for processing request, waiting for communication channel and data migration time. Effects of different data size on these factors were investigated in our experiments. First of all sending data between agents in secure manner were considered. Second, agents collaborated in non secure mode. The last method was interchanging data in secure and non secure manner. Selection policy for number of messages encrypted was 10% of total of messages from 85 interchanged between agents. Fig. 11 illustrates the effect of data size versus time factor.

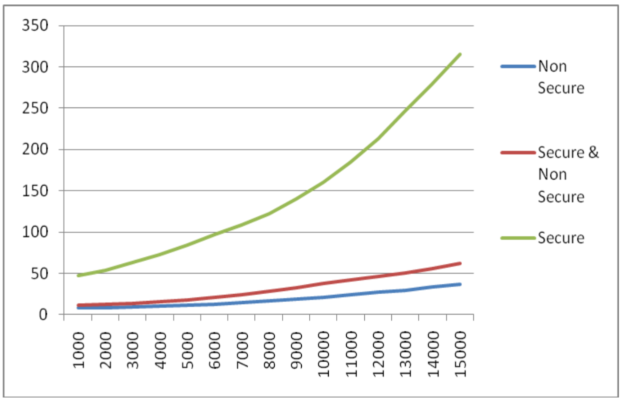

Fig. 11. Comparison of migration time based on data size.

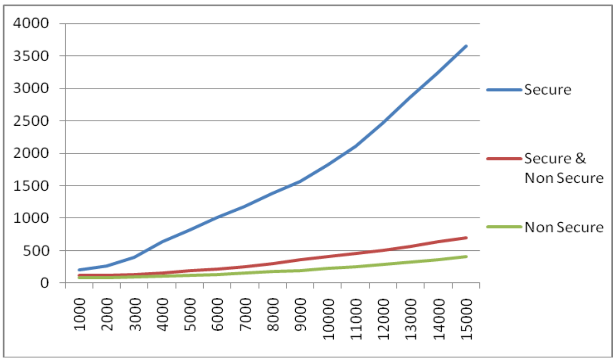

Fig. 12. Comparison of response time based on data size.

According to Fig. 11 for all data size the average time spent for data migration in secure manner is highly greater than non secure manner. Obviously, the reason is time of encryption and decryption of secure data. In the case of combination of secure and non secure data according to our assumption the average time spent for data migration is close to non secure condition. Moreover Fig. 8 depicts that for data size between 1000 to 5000 bytes the migration time in both secure and non secure is highly close to non secure manner. For data size larger than 5000 bytes migration time increases a bit for hybrid condition versus non secure one. Fig. 12 illustrates response time for three conditions versus data size according to our experiments results.

Fig. 9 depicts that for data size between 500 to 2500 bytes the response time in secure manner is relatively close to non secure manner. Data size packets larger than 2500 bytes response time for secure manner increases extremely. Furthermore like the previous experiment hybrid condition is comparatively close to non secure one.

The results imply that there is a little difference between non secure manner and the combination of secure and non secure in both data migration time and response time, so the policy of transfer agents' important messages can be applied in system and ensures the security and performance of the system simultaneously.

VI. Conclusion

In this paper we applied multi agent system to overcome information filtering challenges. We proposed a model based employing multi-agent systems and use of vector space model. Each group has some information which is in its data bases. By use of list agent (may be more than one list agent) and group header agent and the other agents we introduced, system can find the relevant and desired information in information groups.

Searching the best group is one of the most critical parts of SMAIF. The above work has opened up some interesting avenues for further research. On the theoretical side, we plan to incorporate incentives and other mechanisms to encourage the participation of users and to discourage exploitation of helpful users.

References

[1] N. J. Belkin and W. B. Croft, "Information filtering and information retrieval: Two sides of the same coin," Communication of the ACM, vol. 35, no. 12, pp. 29-38, 1999.

[2] G. Marchionini, *Information Seeking in Electronic Environments*, Cambridge Cambridge University press, 1995.

[3] P. J. Denning, *Electronic Junk*, Communication of the ACM, vol. 25, no. 3, pp. 163-165, 1999.

[4] J. Stubinz and S. Whighli, "Information retrieval system design for very high efectiveness," *IEEE Trans. on Information Retrieval*, 2008.

[5] D. Goldberg *et al.*, "Using collaborative filtering to weave an information tapestry," *Communication of the ACM*, vol. 35, no. 12, pp. 61-70, 1998.

[6] P. Resick *et al.*, "Grouplens: an open architecture for collaborative filtering of netnews," in *Proc. Conference on Computer Supported Cooperative Work*, October 1996.

[7] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*, New York: McGraw-Hill, 1995.

[8] J. Thangarajah, L. Padgham, and S. Sardina, "Modelling situations in intelligent agents," *AAMAS'06*, Hokkaido, Japan, 2006.

[9] K. Sycara, "Multiagent systems," *AI magazine*, vol. 19, no. 2, pp. 79-92, 1998.

[10] A. Rao and M. Georgeff, "BDI Agents: From theory to practice," in *Proc. First Iternational Coference on Multi-Agent Systems*, San Francisco, 1995.

[11] S. Mellouli, G. W. Mineau, and D. Pascot, "Multi-Agent system design," in *Proc. 16th Conference of the Canadian Society for Computational Studies of Intelligence*, Halifax, Canada, 2003.

[12] A. Moukas and G. Zacharia, "Evolving a Multi-agent Information Filtering Solution in Amalthaea" in *Proc. 1st International Conference on Autonomous Agents*, CA, USA, 1997.

[13] C. D. Manning, P. Raghavan, and H. Schütze, *An Introduction to Information Retrieval*, Cambridge University Press, 2009.

[14] N. R. Jennings, K. Sycara, and M. Wooldridge, "A Roadmap of Agent Research and Development," *Autonomous Agents and Multi-Agent Systems*, vol. 1, pp. 7–38, 1998.

[15] F. Peyravi and F. Taghyareh, "An Agent mediated community of practice using MAS-CommonKADS methodology," in *Proc. 1st International Conference on Digital Communications and Computer Applications (DCCA2007)*, Irbid, Jordan, 2007.

[16] F. Peyravi and F. Taghyareh, "Applying MAS-CommonKADS methodology in knowledge management problem in call centers," in *Proc. IASTED International Conference on Software Engineering (SE2007)*, Innsbruck, Austria, 2007.

[17] C. A. Iglesias *et al.*, "Analysis and design of multiagent systems using MAS-Common KADS," in *Proc. 4th International Workshop on Intelligent Agents IV, Agent Theories, Architectures, and Languages*, 1998.

[18] B. Henderson-Sellers and P. Giorgini, *Agent-Oriented Methodologies*, US: Idea Group Publishing, 2005.

[19] M. Wooldridge, N. R. Jennings, and D. Kinny, *The Gaia Methodology for Agent-Oriented Analysis and Design,* Autonomous Agents and Multi-Agent Systems, vol. 3, pp. 285-312, 2000.

[20] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613-620, 1975.

[21] F. Bellifemine, A. Poggi, and G. Rimassa. "Developing Multi-agent Systems with JADE," in *Proc. 7th International Workshop on Intelligent Agents VII. Agent Theories Architectures and Languages*, Boston, USA, 2001.

[22] Jade. Tilab. [Online]. http://www.jade.tilab.com/

[23] S. Huin, L. Luong, and K. Abhary, "Managing Enterprise Resources Planning Systems Deployment in SMEs," in *Proc. the 3rd International Conference on Project Management Society of Project Management, and College of Engineering*, Japan, 2002.

[24] K. Pashaei, F. Peyravi, and F. Taghyareh, "A secure multi-agent intelligent conceptual framework for modeling enterprise resource planning," *Current Trends in Intelligent Systems and Computer Engineering*, Springer, 2007.

[25] W. Stalling, *Cryptography and Network Security Principles and Practice*, Upper Saddle River New Jersey - USA: Prentice-Hall, 2003.

[26] Bouncycastle. [Online]. http://www.bouncycastle.org

Farzad Peyravi is currently a Ph.D. student in software engineering in Electrical and Computer Engineering Department of Yazd University. He received his M.s degree in software engineering from university of Tehran. Now, he is a member of digital image processing research group at Yazd university. His research interests include multi agent system, information retrieval, cloud computing and image processing. Now, he is working on security issues in image processing.

Alimohammad Latif obtained his M.Sc. degree in electronic engineering from Amirkabir University of Technology in 1996. He worked as a PhD. candidate at the University of Isfahan where he obtained his Ph.D. in 2011. He is currently an assistant professor at Electrical and Computer Engineering Department of Yazd University. His research interests include digital image processing, data hiding and cryptography. He is working on translation of cryptography book in Persian.