Applying Secret Sharing Schemes to Service Reputation

Akbar S. Namin[1], Ruizhong Wei[2], Weiming Shen[1,3], Hamada Ghenniwa[3]
[1]*Integrated Manufacturing Technologies Institute, National Research Council, Canada*
akbar.siami@nrc.gc.ca, weiming.shen@nrc.gc.ca
[2]*Department of Computer Science, Lakehead University, Canada*
wei@ccc.cs.lakeheadu.ca
[3]*Department of Electrical and Computer Engineering, University of Western Ontario, Canada*
hghenniwa@eng.uwo.ca

Abstract

A successful result of choosing, invoking, integrating and composing services in a collaborative environment depends directly on locating reliable services with respect to their performances, histories, feedbacks from their customers, and generally their reputations. Nevertheless, lacking a secure mechanism to create countable reputation information may lead service providers to exaggerate their performances. We propose to apply Threshold Schemes under the context of service reputation towards providing an unconditionally secure "reputation" credits and systems for services and their providers. We argue that applying such a secure mechanism for reputation of services is not forgeable. As a result, any service requester will be able to rely on valuable reputation information and chooses the most reliable services with confidence.

Keywords: Service Oriented Computing, Software Agents, Security, Cryptography, Secret Sharing Schemes, Service Reputation, Internet.

1. Introduction

Service-oriented computing, an emerging paradigm for integrating legacy applications over the Internet, is recognized as a major discipline for future distributed computing.

Despite embodying revolutionary techniques by service-oriented computing in service integration through heterogeneous environments, both service providers and consumers have faced new challenges. Among them is trustworthiness in a collaborative, dynamic, and risky environment (e.g., the Internet). Customers for running their businesses need to collaborate with some other service providers that are capable of fulfilling the desired services. Regarding the inevitable impact of delivering services in designated time and manner, customers need to locate and collaborate with more reliable business partners. This issue can be a hard problem if there is no information or references in terms of reputation of a service and its provider.

Current standard of locating a service in service-oriented paradigm and specifically under Web services context is designed as "*Universal Description, Discovery, and Integration (UDDI)*" standard. UDDI enables customers to search for a desired service in a nonflexible and limited manner. Regarding our interest in reputation, UDDI does not provide any information in terms of reliability, history, consumers' satisfaction, and reputation of services and their providers. Therefore, due to lack of reputation information and more significantly the reliability of reputation information itself, UDDI cannot satisfy and assist service consumers in finding an appropriate business partner. Moreover, another problem would have been arisen. The problem changes to "*who*" and "*how*" would be responsible to update the information. In a distributed environment, such as the Internet, in which there is a little or no trust among participants, the service providers likely inflate their services' performance. As another possibility, a service provider may issue a faked good/bad reputation for other services that have been considered as businesses competitors.

To provide an *unconditionally secure* reputation model for services in a risky environment such that entities might act as malicious participants, we adopt some cryptographic techniques known as "*Secret Sharing Schemes*" generally, and "*Metering Schemes*" specifically [11, 12]. As some examples of applying metering schemes in various application domains, we may consider counting the accurate measure of the number of access to a page, a site, or a Web service, identifying the highly requested services, and providing some mechanisms for advertising businesses. As an extension to these application domains, we apply metering schemes in service reputation context to provide a secure mechanism for providing reliable reputation information.

"*Secret Sharing Schemes*" are combinatorial approaches to provide security in distributed computing

environments among participants. "*Threshold*" scheme, a well-known type of secret sharing schemes, provides a mathematically proven secure solution for distributing shares to participants such that by pooling a specific number of shares, or threshold value, the participants reconstruct the original secret value. It has been proven that any coalition of the participants lower than the threshold value is not able to reveal the original secret value. More technically, they have absolutely "*no knowledge*" about the original secret.

However, in terms of business level of service consumers and their business levels in market, simple threshold schemes are not sufficient. As an example, a recommendation letter received from a well-known business company with high quality assurance and business activity is much more worthy than a recommendation letter received from an unknown company. As another interesting model by using threshold scheme, is giving more shares to some participants and treating them as very important participants; but, generally in a threshold scheme all participants and their business levels are considered equal. We consider both models either giving more shares to some requesters or giving shares based on classification of requesters through this paper.

General access structure is known as a generalization of threshold schemes in which a dealer generates shares based on some groups of participants. However, we do not use the general access structure in our model, since its concept is suitable for other purposes. We only adopt the idea of classification and divide the participants into some groups named as "*Reputation Structures*" with respect to their business levels.

Forming suitable groups or reputation structures is not straightforward and requests some knowledge and reasoning about members, and their business levels. Moreover, rapid changing of companies as service consumers and their business levels in different time frames asks for other supporting technologies.

Agent-Oriented computing is emerged as a promising paradigm for dealing with cooperation, coordination and decision-making in distributed applications. Software agents have been developed with sophisticated interaction patterns. They are efficient in enforcing automatic and dynamic collaborations. The "*reasoning"* and "*problem solving"* capabilities of software agents provide the feasibility of solving some difficult problems in developing a secret sharing scheme. The creating of reputation structures and reasoning about their members are among some challenges that software agents are capable of fulfilling them.

This paper proposes an unconditionally secure model for services reputation by adopting secret sharing schemes and using software agents. To describe the model, we need to extend UDDI and equipped it with some knowledge regarding reputation. The rest of the paper is organized as following. Section 2 reviews related work. Section 3 introduces secret sharing schemes briefly. Section 4 presents an extension for UDDI standard. We present a service reputation model in Section 5. Section 6 provides a simple example. Section 7 discusses some aspects of the model and Section 8 provides conclusion.

2. Related work

Service reputation has been discussed in some research papers. Some researchers consider reputation as trust relationships among participants. Therefore, more approaches are proposed in some other contexts mainly multi-agent systems. Although, the concept of trust and reputation has very close meaning, however, they are different in usages. We do not address them in here and we just review some literature related to reputation.

The most on hand cases of developing reputation systems are well-known online trading Web sites such as eBay and Amazon. Reputation mechanism in these systems is a cumulative rating measure for both parties in trading. However, these mechanisms have been known as very naive and are vulnerable under some primitive attacks. As an example, Resnick et al. [13] shows the Pollyanna effect of the e-trading reputation systems. This effect has been addressed as disproportionately positive and rare negative feedbacks. Also, Dellarocas [3] has shown some attacks on these systems as well.

Reputation is defined as "an opinion or view of one about something" by Sabater et al. [14]. The reputation has been modelled as "individual", "social", and "ontological". Individual refers to impression of one's behaviour to others. In social, the reputation of an individual will be influenced by the community of the individual which belongs to. Context-based reputation is defined as ontological one.

Mui et al. [9, 10] have proposed an interesting probabilistic model for reputation based on Bayesian network. They have described a framework based on game theory for understanding the relative strength of the different notions of reputation.

Yu et al. [17] have discussed the feasibility of applying Dempster Shafer evidence theory. Reputation is evaluated based on the rating propagation and considering the agent's neighbours. Also, the propagated values are weighted by the neighbours. In fact, they model reputation by using recommender systems. In their model, agents collect information about other agents including their recommending potential and trustworthy partners.

A conceptual model based on Web Service Agent Proxy (WSAP) and a third party named as "Reputation and Endorsement System (RES)" to save reputation information has been proposed by Maximilein et al. [7, 8]. WSAP monitors services' activities and help in

future usages. WSAP collects information on the services to and convey this information to the RES. Each service consumer keeps the reputation information locally and as a result the collected reputation information is not available for publicity.

We are interested in collecting reputation information, which is not only reliable but also globally acceptable. Therefore, secret sharing schemes and their techniques in some application domain such as metering schemes have been chosen to address the above challenges. Metering schemes have been proposed mainly to address some secure payment models for metering a Web site. Masucci et. al. has explored the concept of metering schemes in a rigorous way. We refer the interested readers to [1, 2, 4, 5, and 6]. However, we adopt these schemes to provide a reliable model for measuring service reputation. Moreover, the proposed model employs two emerging paradigms (service orientation and agent orientation), as enabling technologies, in designing a secret sharing-based model for metering services reputation.

3. Secret sharing schemes

Secret sharing schemes are cryptographic techniques to provide security among active participants in a distributed computation environment. In a secret sharing scheme, a dealer D has a secret. The dealer D divides and distributes the secret values among a set of participants such that only some specific subsets of participants named as "*qualified*" are able to reveal and reconstruct the secret value [16]. Let K be the key set, S the share set, and P the set of participants. When a dealer D wants to share a secret $k \in \mathrm{K}$, s/he will give some subset shares of $S_i = \{s \mid s \in S\}$ to each participant $P_i \in \mathrm{P}$. A subset of participants will attempt to determine the value of secret k from the shares they collectively hold [16].

As a special case of secret sharing schemes is a *threshold scheme* that has been proposed by Shamir [15].

3.1. Shamir's *(t, n)* threshold scheme

Shamir's threshold scheme [15] is based on a mathematical fact that each polynomial $y = f(x)$ of degree $t-1$ is uniquely determined by t points (x_i, y_i) with distinct x_i, where $y_i = f(x_i)$.

The main purpose of a threshold scheme is to reconstruct a secret key from a certain number of shares, as threshold value, such that any coalition of less than the threshold value is unable to reconstruct the secret key. It consists of following steps:

The shares generator, or the dealer D, chooses a random number $k \in \mathrm{K}$ as a secret key.

The dealer D chooses a "large enough" prime number $q > Max(k, n)$ such that k, as a secret, should be divided into n shares as the number of participants. The purpose of choosing a large prime number q is for interpolation in modular arithmetic number system. The set of integer numbers (module a prime number q) forms a field $\mathbb{Z}_q$ in which interpolation is possible.

The dealer D defines $a_0 = k$ and chooses $t-1$ random, independent coefficients $a_1, ..., a_{t-1}$ in $\mathbb{Z}_q$ that form a polynomial:

$$f(x) = a_0 + a_1 x + a_2 x^2 + ... + a_{t-1} x^{t-1} \pmod{q}$$

The dealer D chooses n random distinct numbers x_i in $\mathbb{Z}_q$ and computes n values y_i where $y_i = f(x_i)$ for all $1 \le i \le n$. The pairs $s_i = (x_i, y_i) \in S$ are n secret shares.

Any group of size t or more may pool their shares and compute coefficients a_j, $0 \le j \le t-1$ of $f(x)$, using Lagrange interpolation. Therefore, they can compute $f(0) = a_0 = k$ and reveal the secret.

Any group of size $t-1$ or less has no information about k (perfectly secrecy).

As we mentioned earlier using simple threshold scheme is not sufficient services reputation, since the value of shares are treated equal. We use some classification on participants and generate some reputation structures to overcome this limitation.

4. An extension to UDDI

UDDI provides some meta-data regarding services, their providers, and how to interact and use them. Any service provider likely to offer some services registers through UDDI. Consequently, any service requester looking for a specific service, searches UDDI repository, locates the desired service, and finds out how to use it.

Running business processes successfully requests involved parties contribute effectively. Hence, the overall result of collaboration depends on performance of each involved party. As a result, choosing a suitable service provider as a business partner is crucial and has main impact on the ultimate result. Current standard of UDDI does not provide any information regarding choosing and reputation of any individual service. In terms of collaboration, customers should be equipped with a measure regarding the performance and reputation of service providers. We consider an extension for UDDI standard to overcome this

challenge. We equip an UDDI with two other components. More precisely we define two other components namely as *"Reputation Knowledge*" and "*Reputation Agent*" as complementary for UDDI. As an alternative, the reputation knowledge component might be mixed up with UDDI database.

The information sited in reputation knowledge is an updateable data source regarding service both providers and requesters' reputation. The information assists consumers by providing a measurable unit for reputation of services as well as providing some information for reputation agent to reason about the business levels of service requesters. It is important to note that, we consider reputation information for both service providers and requesters and consider them customers generally.

The reputation agent is responsible for classifying the service requesters into some levels with respect to their business levels. The reputation agent reasons about the business levels of service requesters and decides about the values of their feedbacks or recommendations. Therefore, we are able to classify all customers and form the reputation structures, which are useful in the model.

By keeping the reputation information on the UDDI's side some benefits may achieve. Firstly, any service requester may search for the desired services through UDDI and achieves their reputation at the same time. Therefore, there is no need for further search through other resources. Secondly, the service requesters may compare the reputation of similar service providers instantly and decide more effectively in a minimum time. Thirdly, the reputation information is available for the publicity in UDDI known as a trusted data source.

Despite locating the reputation information on UDDI and protecting them from modification, other problems still exist:

Who has privileges to update this information?

How securely this information will be updated?

As a general approach, service consumers are allowed to announce their satisfactions of using a service. But a service provider may employ some faked service consumers, named as "*corrupted*", to inflate satisfaction for using services. Identifying corrupted customers from "*valid"* customers will be handled by the reputation agent.

As another significant problem is overloading UDDI with communicating, interacting, collecting, validating, and constructing reputation information achieved from million of thousands customers. In a massive distributed environment with enormous customers, it is not a feasible idea to overload reference source such as UDD. As a more suitable approach is allowing service providers reconstruct the requirements for achieving extra reputation by themselves. In following we describe how to achieve this approach.

5. A secret sharing model for service reputation

This section proposes an unconditionally secure model for service reputation. Figure 1 depicts a scenario, in which some clients, as participants, by searching through service discovery, achieve some knowledge about a desired service. The service discovery offers some service satisfaction points as shares to the clients. The clients by using the service and in case of satisfaction of using the service award one or more satisfaction points to the service provider.

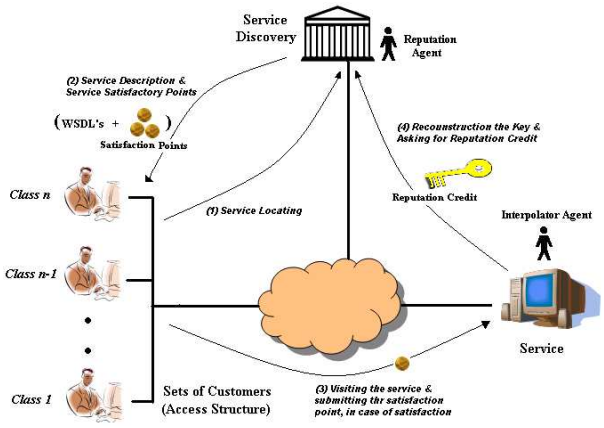

Figure 1 - A secret sharing model for services reputation

The main techniques used in this model are adopting threshold scheme as well as classifying clients with respect to their business levels. Forming the later issue needs some knowledge about the participants, their qualities, business levels. As an appropriate technology for addressing these issues, is employing software agents and their unique capabilities in reasoning. The model employs two agents namely as "*interpolator agent*" and "*reputation agent*".

The reputation agent reasons about business levels, and classifications of clients. As a result, it generates the required satisfaction points and supports UDDI. Consequently, UDDI hands out these shares to customers to submit to service providers as reward. Eventually, the service providers collect these shares in order to claim for more reputation.

The *interpolator agent*, sited on the service provider's side, by receiving a set of satisfaction points, interpolates and reveals the secret value or key and hands it over to the reputation agent on UDDI. The reputation agent by receiving the revealed secret value verifies it and increases the globally accessible reputation of the service by a standard measurable unit.

As an essential requirement of the model is that any service consumer has to hand out different satisfaction points to the service provider in different time frames. It might be feasible by requesting more shares from the reputation agent at discovery time. As a result, a service

consumer may either involve in more classes or reputation structures or possesses more shares in a specific class. As a matter of fact, each client, by possessing *m* satisfaction points, is able to submit and recommend a service provider for at most *m* times.

We represent the set of *w* service consumers as participants, as $\mathrm{P} = \{P_1,...,P_w\}$, K is the set of all possible keys, and S is the set of all possible shares or satisfaction points. To describe the model, we divide the scenario into some stages as following.

5.1. Initialization stage

In order to model a secure and globally accessible service reputation mechanism, a trusted entity to take the responsibilities of stages is needed. As an appropriate candidate is the trusted UDDI, with the proposed extension. In addition to regular information, UDDI and specifically the "Reputation Agent" provide some satisfaction points. Initially the reputation agent reasons and classifies customers with respect to their business levels.

5.1.1. Identifying clients' business levels. Suppose *w* clients $P_1,...,P_w$ have sent their requests to UDDI in order to find out the functionality and reputation of a service *j*. For simplicity we assume that all participants have sent their requests at the same time. The general case is easily extendable by the reputation agent via creating an open list of participants rather than a closed one. Therefore, the reputation agent may create shares and assigns them later to coming next requests.

We denote $\mathrm{P} = \{P_1,...,P_w\}$ for $1 \le i \le w$ as the set of participants. Also, we represent C_{P_i} for $1 \le i \le w$ as the business level, of the participant P_i , which is specified by reasoning the reputation agent. Suppose that UDDI has defined *n* business levels. We denote by ρ_l^j the set of all participants for *j*-th service with business level equal to $l, 1 \le l \le n$, classes:

$$\rho_l^j = \{P_i \mid P_i \in \mathrm{P}, C_{P_i} = l, 1 \le l \le n\}$$

We denote the set of all participants assigned in particular classes for service *j*:

$$\rho^j = \{\rho_l^j \mid 1 \le l \le n\}$$

5.1.2. Creating the reputation structures. The reputation agent, by reasoning about subsets of ρ^j creates some reputation structures such that the members of each reputation structure are chosen from all different classes of ρ_l^j 's. As an example of a simple reputation structure will be combination of one high level business level client with five low business levels of clients. It assures that for revealing a secret key, a recommendation from one customer with high business level is required. In fact, the members of the reputation structure are one member from ρ_s^j and five members from ρ_d^j such that $s \ne d$ and $d < s$. The reputation agent creates h reputation structure $\Gamma_1^j, \Gamma_2^j, ..., \Gamma_h^j$ and we denote the set of all created reputation structures for j-th service as:

$$\Gamma^j = \{\Gamma_m^j \mid 1 \le m \le h\}$$

Table 1 represents the created reputation structures and their properties. Each reputation structure is a subset of participants with equal or different business levels. In addition, each reputation structure has a specified length, as threshold value, and a specified secret key that is supposed to be revealed by the participants in involved in the reputation structure.

Reputation Structures	Members	Threshold / Secret Key
Γ_1^j	$\sigma_1 = \{P_i \mid P_i \in \rho_l^j, 1 \le l \le n\}\,;\ \mid \sigma_1 \mid = r_1$	$(t_1, r_1) / k_1$
Γ_2^j	$\sigma_2 = \{P_i \mid P_i \in \rho_l^j, 1 \le l \le n\}\,;\ \mid \sigma_2 \mid = r_2$	$(t_2, r_2) / k_2$
Γ_3^j	$\sigma_3 = \{P_i \mid P_i \in \rho_l^j, 1 \le l \le n\}\,;\ \mid \sigma_3 \mid = r_3$	$(t_3, r_3) / k_3$
…	…	…
Γ_h^j	$\sigma_h = \{P_i \mid P_i \in \rho_l^j, 1 \le l \le n\}\,;\ \mid \sigma_h \mid = r_h$	$(t_h, r_h) / k_h$

Table 1 - The created reputation structures

As an example, the access structure Γ_1^j consists of r_1 number of participants such that any coalition of t_1 number of members is able to reveal the secret value k_1 . Hence, each reputation structure forms a *(t, r)* – Shamir's threshold scheme. For the sake of generality, we denote each *(t, r)* model as (t_m, r_m) - threshold scheme for $1 \le m \le h$ such that $t_m \le r_m$. The later inequality assures the *robustness* of the model.

5.1.3. Reasoning about threshold and secret values for generated reputation structures. The "Reputation Agent", based on business level of each customer, reasons about the length of each structure, or threshold, and its members. As an example of reasoning will be choosing members of a reputation structure from participants with different business levels. As another example will be forming a structure with a few customers and with high business levels. In contrary the collection of the satisfaction points of a large number of

middle class customers might be needed to increase the reputation of a service.

Therefore, the reputation agent picks up some "*random*" values k_m as secret keys for each reputation structure Γ_m^j for $1 \le m \le h$ with threshold t_m for service *j*. In addition, the reputation agent chooses some large prime number q_m such that $q_m > Max(k_m, r_m)$. The values of k_m are the secret keys to be revealed by the service. Finally, the reputation agent chooses some random polynomials in $\mathbb{Z}_{q_m}$. The reputation agent sends out the values of q_m to the interpolator agent sited on the service provider's side for interpolation purposes.

5.2. Generating satisfaction points

The reputation agent generates satisfaction points as much as threshold value for each reputation structure. The generated shares for each structure Γ_m^j for $1 \le m \le h$ are a function of the secret key $k_m \in K$. In fact, the reputation agent divides the value of each k_m into r_m pieces as $s_1, s_2, ..., s_{r_m} \in S$. As generating shares in Shamir's (t_m, r_m) -Threshold Scheme, and the reputation agent picks up h random polynomials with degree $t_m - 1$, for $1 \le m \le h$, as:

$$q_m(x_m) = a_0^m + a_1^m x_m + ... + a_{t_m-1}^m x_m^{t_m-1} (mod\ q_m)$$

The values of $a_0^m = k_m$ are the secret keys. Furthermore, the reputation agent evaluates the following r_m values for each reputation structure:

$$y_\theta = q_m(x_\theta) \quad (mod\, q_m) \quad \text{for} \quad 1 \le \theta \le r_m$$

The reputation agent computes r_m values of $y_\theta = q_m(x_\theta)$. The pairs $s_\theta = (x_\theta, y_\theta)$ are r_m secret satisfaction points.

5.3. Applying hash function to satisfaction points

In order to verify the validity of each satisfaction point, the reputation agent uses a hash function on generated satisfaction points and sends the hash values to the interpolator agent. The reason for using a hash function rather than a digital signature is efficiency of a hash function. Also, digital signatures are useful when a receiver needs to verify the content of a message; however, in this mode, the service provider needs to verify the received shares rather than contents.

The interpolator agent, by receiving hash values, stores them in a table. The interpolator agent, by receiving any satisfaction point, applies the hash function on the received share and compares the equality of the computed and saved hash values. The equality guarantees the validity of the share. By receiving each satisfaction point, the interpolate agent removes the corresponding hash value from the table. In fact, removing the hash value emphasizes that a satisfaction point can be used only once. The hashing algorithm can be any one of well known hash functions such as SHA-1.

5.4. Sending reputation structures to the interpolation agent

In order to interpolate the secret values, the interpolation agent needs to have enough knowledge about the created reputation structures. The required knowledge might consist of the participants of each structure and its threshold value. In other words, the reputation agent, besides sending the hash values of the generated shares and the prime numbers q_m, sends the reputation structures Γ_m^j, its threshold value t_m and value of r_m, for $1 \le m \le h$. In case of using threshold scheme without creating reputation structure, the threshold value will be enough.

Sending this information is a continuous process. In fact, reasoning about the reputation structures for any service needs some knowledge that the reputation agent acquires through communication with customers. The reputation agent may modify the structures for a service while communicating with its customers. We discuss about this issue through next sections.

5.5. Offering encrypted satisfaction points to clients

The reputation agent uses an encryption algorithm to protect the satisfaction points while sending to customers through the Internet. The encryption takes place by using the clients' public keys over a secure channel such as SSL (*Secure Socket Layer*). Then only the service consumer that has the desired secret key can decrypt the encrypted satisfaction point. The encryption prevents the model from any malicious eavesdropper's attack.

5.6. Satisfaction of using a service

In case of satisfaction of using a service, the client submits a satisfaction point to the interpolator agent. To prevent any faked claim regarding increasing reputation by service provider, the client must submit different satisfaction points for different time frames. The

interpolator agent uses a hash function to verify the satisfaction points. Consequently, the interpolator agent collects them and removes the corresponding entries from its storage.

5.7. Interpolating the secret values of reputation structures

Since the reputation structures, their members, and threshold values are known to the interpolator agent, therefore it is able to interpolate the secret values. More preciesly, the interpolator agent, by collecting t_m shares for each Γ_m^j and for $1 \le m \le h$, is able to construct the system of equations and reveals the secret key $k_m \in K$ of a (t_m, r_m) - threshold scheme. In fact, the interpolator agent is dealing with *h* threshold (t_m, r_m) - Shamir's schemes and *h* different values of keys $k_m \in K$.

In other words, the interpolator agent creates a system of polynomials of degree t_m -1 such that the share values $s_\theta = (x_\theta, y_\theta)$ for $1 \le \theta \le r_m$ are the second parts of each equations. Since the degree of the each polynomial is t_m – 1, t_m numbers of s_θ are enough to solve the system and find out the values of $a_0^m, a_1^m, ..., a_{t_m-1}^m$ and eventually the value of the secret key $k_m = a_0^m$.

5.8. Verifying the revealed secret keys

By revealing each secret key, the interpolator agent submits it to the reputation agent in order to claim for more reputation. The reputation agent verifies the values of the secret keys and decides for increasing the reputation of the service provider.

The reputation mechanism can be any measurable method. As an example, it can be the set of integer values such that zero represents no reputation and values more than zero represent the specified reputation units.

6. Discussion

We adopt secret sharing schemes mechanisms to generate some subsets of clients capable of revealing the secret values. Creating reputation structures, indeed, is a continued process that needs an experimental approach to figure out the members. Deciding about membership of clients in business levels and forming the reputation structures, needs some knowledge about capabilities of each member. As we discussed earlier the most effective technology for fulfilling this reasoning is using software agents. Software agents are not only useful to form reputation structures, but also they have ability to handle dynamic revision of the created structures. In following we clarify this issue.

In case of needs to change the number of shares in a threshold scheme, the only requirement is to define a new polynomial with the same free terms [15]. Changing the number of shares is based on the reasoning capabilities of the reputation agent in identifying the needs for changing the reputation mechanism with respect to the customers and services' capabilities. The reputation agent, by reasoning about the clients either may assign them to an existing structure, as new members, or creates a new structure. As a result, the reputation agent may create some new random polynomials for the revised / created reputation structures and follow the same scenario.

As another interesting issue will be changing the number of existing participants of a structure. The reputation agent, regarding the activities of clients or services may decide to increase the number of clients of a reputation structure (i.e. in fact the reputation agent decreases the generated shares sensitivities). Therefore, the reputation agent creates a new random polynomial such that not only values for the existing satisfaction points and the secret value are not changed but also the new generated satisfaction points are added to the modified reputation structures.

The limitation of the model is that the service provider cannot claim for reputation unless collects a specific numbers of satisfaction points as threshold value. It may be considered as a shortcoming of this model; however, the security of model is guaranteed.

7. Conclusion

The emerging service-oriented computing, as a new paradigm for service integration, has opened up new view of distributed computing such that the entities are able to offer their services over the Internet regardless of platforms and developed languages. By looking at capabilities as services, new challenges have been emerged. One of the most important issues is trust and reliability of services.

This paper proposes an unconditional secure model to address a secure reputation model for service-oriented computing. We adopt secret sharing schemes and specially threshold scheme. To address the recommendations achieved from different customers with different business levels. We classify the customers into some classes as reputation structures. Even though, it is possible to handle the values of recommendations by providing more shares to some customers with higher business levels; however, the classification is still inevitable. By classifying the participants into some

structures and adopting threshold scheme on these structures, the participants of each structure are able to reconstruct the secret values.

The model needs an intelligent component to handle the dynamic generation and modification of the reputation structures, reasoning and assigning customers to each structure, generating initial shares, and interpolating the secret values. Intelligent software agents are software components capable of reasoning and problem solving. The capability of an agent in solving a mathematical problem such as the ability to reason about the number of generated shares and setting the reputation structures are among concerns of this paper.

References

[1] C. Blundo, D. Bonis, and B. Masucci, "Dynamic Multi-Threshold Metering Schemes", *Selected Areas in Cryptography*, 2000, pp. 130-143.

[2] C. Blundo, D. Bonis, and B. Masucci, "Bounds and Constructions for Metering Schemes", *Communications of Information and Systems,* 2002, 2(1), pp. 1-28.

[3] C. Dellarocas, "Immunizing Online Reputation Reporting Systems against Unfair Ratings and Discriminatory Behavior", *In Proceeding of the 2nd ACM Conference on Electronic Commerce, 2000.*

[4] B. Masucci, and D. Stinson, "Metering Schemes for General Access Structures", *ESORICS*, 2000, pp. 72-87.

[5] B. Masucci, and D. Stinson, "Efficient Metering Schemes with Pricing", *DISC*, 2000, pp. 194-208.

[6] B. Masucci, "Secure Metering on the Web", http://udsab.dia.unisa.it/dottorato/TESI/tesi-masucci.pdf, PhD Thesis, 2000.

[7] E. Maximilien, and M.P. Singh, "Reputation and Endorsement for Web services", *ACM SIGecom Exchange*, 2001, 3(1), pp. 24-31.

[8] E. Maximilien, and M.P. Singh, "Conceptual Model for Web Service Reputation", *ACM SIGMOD Record 31*, 2002, no 4, pp. 36-41.

[9] L. Mui, M. Mohtashami, C. Ang, P. Szolovits, and A. Halberstadt, "Ratings in Distributed Systems: A Bayesian Approach", *11th workshop on Information Technologies and Systems (WITS)*, New Orleans, 2002.

[10] L. Mui, M. Mohtashami, and A. Halberstadt "A Computational Model for Trust and Reputation," *35th Hawaii International Conference on System Sciences*, 2002.

[11] M. Naor, B. Pinkas, "Secure and Efficient Metering Scheme", *ASIACRYPT*, 1998, pp. 388-398.

[12] W. Ogata, K. Kurosawa, "Provably Secure Metering Scheme", *ASIACRYPT*, 2000, pp. 388-398.

[13] P. Resnick, R. Zekhauser, "Trust Among Strangers in Internet Transactions: Empirical Analysis of eBay's Reputation System", *NBER Workshop on Empirical Studies of Electronic Commerce Paper*, 2000.

[14] J. Sabater, C. Sierra, "A Reputation Model for Cregarious Societies", *4th Workshop on Deception, Fraud, and Trust in Agent Societies*, 2001.

[15] A. Shamir, "How to Share a Secret", *Communication of ACM*, 1979, 22(11), pp. 612-613.

[16] D. Stinson, "An Explication of Secret Sharing Schemes", *Designs, Codes, and Cryptography*, 1992, no 2, pp. 357-390.

[17] B. Yu, M.P. Singh, "Towards a Probabilistic Model of Distributed Reputation Management", *4th Workshop on Deception, Fraud and Trust in Agent Societies*, Montreal, Canada, 2001.