International Journal of Innovative
Computing, Information and Control
Volume **8**, Number **11**, November **2012**

 ISSN 1349-4198
pp. **7955–7968**

A TRUST MECHANISM FOR WEB SERVICES REGISTRIES DATA EXCHANGING

Chi-Hua Chen[1], Ding-Yuan Cheng[1], Bon-Yeh Lin[1,2], Yin-Jung Lu[1,2] and Chi-Chun Lo[1]

[1]Institute of Information Management
National Chiao Tung University
No. 1001, University Road, Hsinchu 300, Taiwan
cclo@faculty.nctu.edu.tw; { chihua0826; kewas.cheng }@gmail.com

[2]Telecommunication Laboratories
Chunghwa Telecom Co., Ltd.
No. 12, Lane 551, Min-Tsu Road, Sec. 5, Yang-Mei, Taoyuan 326, Taiwan
{ bylin; yinjung0407 }@cht.com.tw

Received July 2011; revised December 2011

Abstract. *In recent years, Web Services (WS) have been becoming more popular, as more people realize its benefits. However, the WS-Security specification only guarantees the security on end-to-end level. We propose the architecture of Trust-based Web Services Registries Data Exchanging (TWSRDE) which is four-tier architecture composed of the users, Web Service Providers (WSP) in Simple Object Access Protocol (SOAP) server, Enhance Web Services Registry (EWSR), and extra Web Services Registries (WSReg). The EWSR provides the trust mechanism which considers physical space and concept space to infer the trust value of WS for WSReg data exchanging.*
Keywords: Service oriented architecture, Web service registry, Trust mechanism

1. **Introduction.** In recent years, *Web Services* (WS) have been becoming more popular, as more people realize its benefits. The *Service-Oriented Architecture* (SOA) is often referred to as "message-oriented" services which is composed of *Web Service Registries* (WSReg), *Web Service Providers* (WSP), and *Web Service Requesters* (WSReq) by *Universal Description, Discovery and Integration* (UDDI), *Simple Object Access Protocol* (SOAP), and *Web Services Description Language* (WSDL). WSP can publish WS on WSReg, and WSReq can request WSReg to invoke the WS. However, the current security mechanism of SOA which is based on WS-Security only considers about the protection during transmission level without trust level of WS. To provide trust mechanisms in WS, the design principles and functionalities for WSReg are as follows.

(1) WS-Security specification is adopted to protect contents during end-to-end message delivery.

(2) The feedback and recommendation mechanisms are integrated into WSReg to obtain the trust value of WS.

(3) The trustworthy WS is provided to WSReq.

In this study, we propose the architecture of *Trust-based Web Services Registries Data Exchanging* (TWSRDE) which is 4-tier architecture composed of the users, WSP, *Enhance Web Services Registry* (EWSR), and extra WSReg. The EWSR provides the trust mechanism to infer the trust value of WS for WSReg data exchanging. Moreover, the trust mechanism integrates the physical space information (i.e., *Non-Repudiation* (NR), *Certificates Authorization* (CA), *Integrity* (Int), and *Confidentiality* (Con) [1]) and concept space

information (i.e., User Feedbacks (UFs) [2-4]) to improve the security foundations as described in the WS-Security specification which only guarantees the security on end-to-end level [5]. Our proposed method can provide the reliable WS to the WSReq.

The remainder of the paper is built as follows. In Section 2, we provide background knowledge through the description of related technologies, such as the concept of requirements of web services registry and trust mechanism. In Section 3, we analyze the web services security and propose a trust mechanism for web services registries data exchanging. Section 4 illustrates the experimental results. Finally conclusions are given in Section 5.

2. Background and Related Work.

In this paper, we propose the TWSRDE which aims to consider the following factors: (i) SOA and (ii) trust value of WS. Necessary research background and relevant technology include: (1) security requirements of WSReg and (2) trust mechanism.

2.1. Security requirements of WS registry.

Dustdar and Treiber [5] proposed a view based on WSReg. They indicated that the requirements of WSReg are interoperability, reliability, security, fault tolerance, scalability, availability, expressiveness of query language, expressiveness of WS description, transient WS, and management. The different types of WSReg are built with the different security requirements. Gil and Artz discussed the trust factors which are popularity, authority, user expertise, direct experience, recommendation, recency, incentive, and deception [6]. Therefore, the trust factors of WS are as follows.

(1) Popularity: If a WS is used by many WSReq, it tends to be more trusted.

(2) Authority: A WS is more trusted if it is published by the trustworthy WSP.

(3) User expertise: A WSReq with expertise usually makes better judgments regarding a WS and concludes whether or not it is trustable.

(4) Direct experience: The direct interaction of a WSReg with a WS provides reputation information, a record of whether or not trust was well placed in the past.

(5) Recommendation: Feedbacks from other WSReg for a WS provide indirect reputation information.

(6) Recency: Content, associations, and trust change with time.

(7) Incentive: Information may be more believable if there is motivation for a WS to provide accurate information.

(8) Deception: Some WSP and their WS may have deceptive intentions.

2.2. Trust mechanism.

The data exchanging in different WSReg in SOA and the resource sharing between different nodes in social network are similar. In the social network, the nodes would not be reliable absolutely. For resource sharing between service nodes, there may be several malice nodes which would share unsecure or incorrect data. How to choose a trustworthy node to process the resource sharing is very important. RASSA algorithm [7] chooses node with high reputation. Each node has a reputation value. However, the reputation value calculation is according to the three factors which are data integrity, trust, and reliability on each node. Moreover, the nodes set the weighting of the three factors as α, β, and γ. [7] defines the $Score_{BA}$ which is the score about node B judging on node A. The scores about node B judging on node A for data integrity, trust, and reliability are expressed as $Score_{x_{BA}}$, $Score_{y_{BA}}$, and $Score_{z_{BA}}$, respectively.

$$Score_{BA} = \alpha \times Score_{x_{BA}} + \beta \times Score_{y_{BA}} + \gamma \times Score_{z_{BA}} \tag{1}$$

If node A provides its resource to n nodes, the other node would judge on node A. The evaluation formula which is mentioned in [7] is presented as

$$\text{Reputation}_A = a \times \text{Reputation}_A + b \times \frac{\sum_{i=1}^{n} (Scorei_A \times Uptimei_{AVE})}{\sum_{i=1}^{n} Uptimei_{AVE}} \tag{2}$$

Pirzada et al. proposed Trust-based Routing in [8]. They use the weighted averages method which can calculate the trust value of each node by following equation.

$$T_{xy} = W(P_A) \times P_A + W(P_P) \times P_P \tag{3}$$

Although, the previous research provided a simple and convenience rating method, they did not mention how to calculate the score judged by another node. Hence, this study proposes a trust mechanism which could provide a fair and righteous evaluation for each WSReg.

Wang et al. discussed the trust systems based on centralized management architecture (e.g., reputation evaluation on eBay) and decentralized architecture (e.g., peer-to-peer (P2P) trust evaluation) for reputation-oriented trustworthy computing [9-11]. Although the decentralized architecture does not require extra costs to set up separate servers, it is costly in terms of network communication when the requesting peer broadcasts a request to other peers repeatedly and computes trust value locally. Moreover, all peers cannot possess and store a transaction history with the target service provider in the decentralized architecture. Therefore, the centralized architecture which is suitable for WS can provide the central trust management server (i.e., WSReg) to store trust history data and compute the trust value of WS. However, Wang et al. only considered the number of positive ratings and negative ratings, and the malice WS cannot be discovered and avoided quickly by using this approach [9-11]. For this reasoning, we will consider more security requirements and design the trust mechanism in the centralized architecture.

2.3. **Summary.** The former studies only take care of the end-to-end security of the WS. They did not pay attention to evaluating the trust level of WS. Some literature has mentioned this need but did not propose concrete and efficient mechanisms to solve the problem. By notice that the trust mechanisms developed in social network fields are simple and efficient, we adopt these mechanisms and modify them to evaluate the trust value of WS. This paper aims to achieve the following goals.

(1) Search, discovery, and invoking WS are facilitated in a secure manner.

(2) A trust mechanism considers both physical and concept space factors to evaluate the trust level of WS.

3. The Design of Trust Mechanism.

In this section, we propose the trust mechanism and present it in detail.

3.1. **Security requirement of WS.** For evaluating the trust level of WS, we classify the trust factors into two spaces. One is physical space and another is concept space. In physical space, we evaluate the factors according to the security protocols used in network layer, transport layer, and application layer such as IPSec, SSL, and WS-Security. On the other hand, we describe the factors in concept space and develop a feedback mechanism from the WSReq to adjust and reflect the trust level of WS.

3.1.1. *Physical space factors.* In physical space, we consider four security demands of WS as follows: (1) *Non-Repudiation* (NR), (2) *Certification Authority* (CA), (3) *Integrity* (Int), (4) *Confidentiality* (Con). At present, applying WS-Security specification can fulfill these demands to protect the WS data exchanging between WSP, WSReq, and WSReg. We use the following notations to describe the security features provided by the WS j in WSReg i. The notation used in this paper is summarized in Table 2 in Appendix A.

$$NR(S_{i,j}) = \begin{cases} 1, & \text{if WS } j \text{ in WSReg } i \text{ provides NR} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

$$CA(S_{i,j}) = \begin{cases} 1, & \text{if WS } j \text{ in WSReg } i \text{ provides CA} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

$$Int(S_{i,j}) = \begin{cases} 1, & \text{if WS } j \text{ in WSReg } i \text{ provides Int} \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

$$Con(S_{i,j}) = \begin{cases} 1, & \text{if WS } j \text{ in WSReg } i \text{ provides Con} \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

3.1.2. *Concept space factors.* It is not enough for the relying parties to tell whether the WS is trustworthy or not by only considering physical space factors. Therefore, we propose concept space factors to evaluate the trust level of WS. These factors are popularity, authority, user expertise, direct experience, recommendation, recency, incentive, deception and storage time. These factors are quantified and the values are stored in WSReg. These values can be analyzed to infer the trust level of WS.

(1) **Popularity.** If a WS is used by many WSReq, it tends to be more trusted.

(2) **Authority.** A WSP who already has a fundamental honor usually provides more trustworthy WS.

(I) If the WSP already has authority, then the WS which is provided by the WSP may be trustworthy. For example, WS which is provided by IBM tend to gain higher trust value than WS provided by an unknown company.

(II) If the WS provided by the WSP are all trustable, the WSP will be strengthened the authority.

(3) **User expertise.** A WSReq with expertise usually makes better judgments regarding a WS and concludes whether or not it is trustable.

(I) When the WSReq is an expert, he can make better judgments regarding a WS.

(II) WSReq who uses the WS many times will be regarded as an expert.

(4) **Direct experience.** The trust value of the WS is computed according to the historical interactive experiences and feedback between WSReq and WSP in the same WSReg.

(5) **Recommendation.** Feedbacks from other WSReg for a WS provide indirect reputation information.

(6) **Recency.** The weighting of the newer feedback from WSReg is more trustworthy.

(7) **Incentive and deception.** Our trust mechanism considers the incentive to encourage the well-behaved WSP and combines the deception mechanism to punish the misbehaved WSP. This mechanism will be executed dependent on the result of user feedback.

(8) **Storage time.** The honor value of the time-honored brand is higher.

(I) If the storage time of WSP is longer, the WSP is more reputable.

(II) If the storage time of WSReq is longer, the WSReq is more reputable.

(III) If the storage time of WS is longer, the WS is more trustworthy.

3.2. **A trust mechanism for WS.** To verify the WS, WSP, and WSReg which are trustworthy or not, we propose a trust mechanism that considers physical space factors and concept space factors. The procedures are shown as Figures 1 and 2.

Step 1: Setting initial values. We set the initial trust value of a WS by considering the honor of WSP and the four physical space factors mentioned in Section 3.1.1.

Step 2: Adjusting honor values and trust values. This trust mechanism considers the concept space factors which are popularity, authority, user expertise, direct experience, recommendation, recency, incentive, deception and storage time. Therefore, we can adjust the trust value to commit the trust level of WS according to WSReq feedback.

Step 3: Adjusting reputation values and trust values for data exchanging. The WSReg will exchange trust information of WS with others. The trust value of a specific

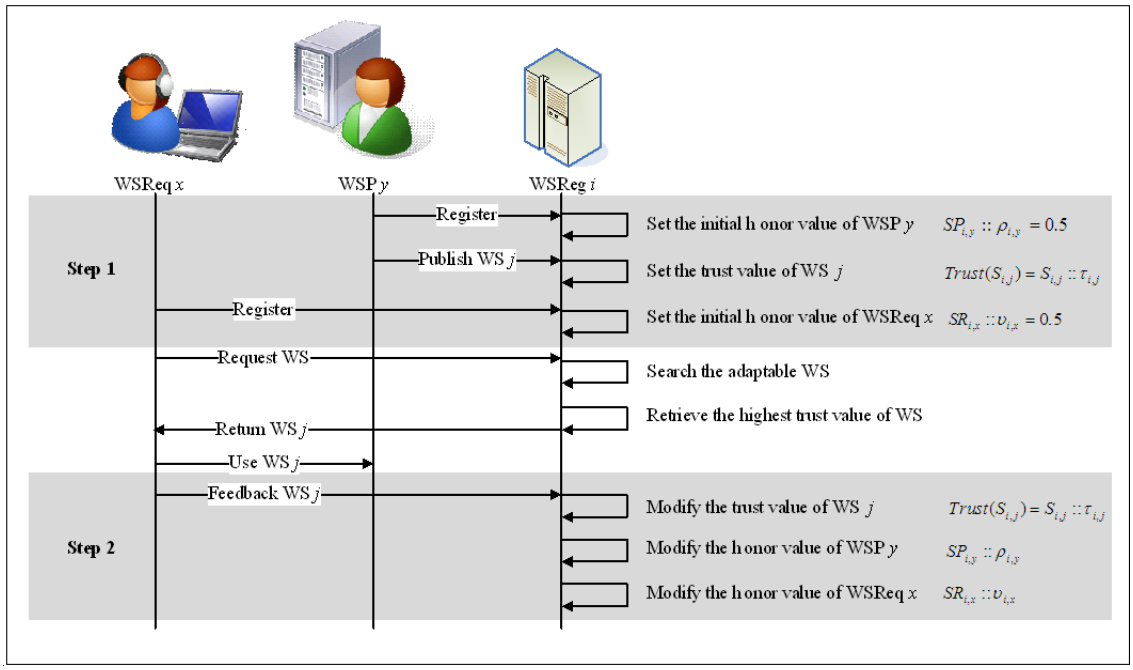

FIGURE 1. The procedure of trust mechanism for computing the honor values of WSReq x and WSP y and the trust value of WS j in WSReg i

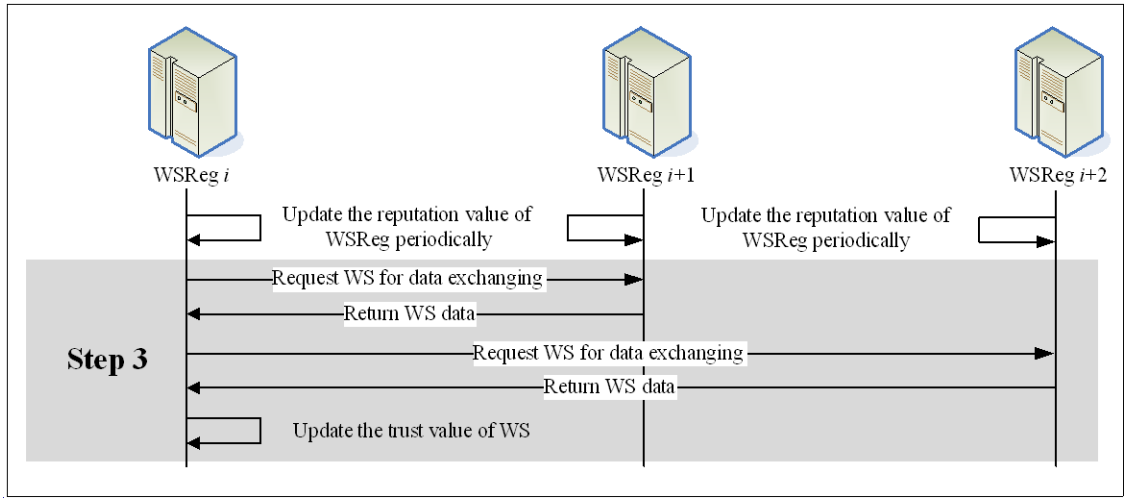

FIGURE 2. The procedure of trust mechanism for computing the reputation value of WSReg i and the trust value of WS with data exchanging

WS will be stored in many WSRegs. Therefore, the WSReq can query and update the trust value of the WS through data exchanging.

3.2.1. *Setting initial values (Step 1).* WS-Security is a flexible and feature-rich extension to SOAP to apply security to WS. By accommodating WS-Security XML Signature and Encryption, WS can meet the physical space factors.

3.2.1.1. The honor value of WSP. The initial honor value of a new WSP y in WSReg i is defined as the following.

$$Honor(SP_{i,y}) = SP_{i,y} :: \rho_{i,y} = 0.5 \tag{8}$$

$Honor(SP_{i,y}) = SP_{i,y} :: \rho_{i,y}$: The honor value of WSP y in WSReg i.

3.2.1.2. The honor value of WSReq. The initial honor value of a new WSReq x in WSReg i is defined as the following.

$$Honor(SR_{i,x}) = SR_{i,x} :: \upsilon_{i,x} = 0.5 \tag{9}$$

$Honor(SR_{i,x}) = SR_{i,x} :: \upsilon_{i,x}$: The honor value of WSReq x in WSReg i.

3.2.1.3. The trust value of WS. The trust value of a new WS j in WSReg i is defined as the following.

$$\begin{aligned} Trust(S_{i,j}) &= S_{i,j} :: \tau_{i,j} \\ &= W_{NR} \times NR(S_{i,j}) + W_{CA} \times CA(S_{i,j}) + W_{Int} \times Int(S_{i,j}) \\ &\quad + W_{Con} \times Con(S_{i,j}) + W_{WSP} \times Honor(SP_{i,j}) \end{aligned} \tag{10}$$

$$\forall\ 0 < W_{NR} < 1,\ 0 < W_{CA} < 1,\ 0 < W_{Int} < 1,\ 0 < W_{Con} < 1,\ 0 < W_{WSP} < 1,$$
$$W_{NR} + W_{CA} + W_{Int} + W_{Con} + W_{WSP} = 1$$

$Trust(S_{i,j}) = S_{i,j} :: \tau_{i,j}$: The trust value of WS j in WSReg i.

W_{NR}, W_{CA}, W_{Int}, W_{Con}, W_{WSP}: The weighting of NR, CA, Int, Con, and the honor value of WSP.

3.2.2. *Adjusting honor values and trust values (Step 2).*

3.2.2.1. The adjusted honor value of WSP. The adjusted honor value of WSP y in WSReg i is defined as the following.

$$SP_{i,y} :: \rho_{i,y} = W_{old} \times Honor_{old}(SP_{i,y}) + W_{new} \times Honor_{new}(SP_{i,y}) \tag{11}$$

$SP_{i,y} :: \rho_{i,y}$: The honor value of WSP y in WSReg i.

$Honor_{old}(SP_{i,y})$: The original honor value of WSP y in WSReg i.

$Honor_{new}(SP_{i,y})$: The new honor value of WSP y in WSReg i.

W_{old} and W_{new}: the weighting of original and new honors value of WSP y in WSReg i.

$$\begin{aligned} Honor_{new}(SP_{i,y}) &= W_{count} \times \frac{SC(SP_{i,y}) = \sum_{j=1}^{SC(R_i)} SW(S_{i,j}, SP_{i,y})}{SC(R_i)} \\ &\quad + W_{ST} \times \frac{\sum_{j=1}^{SC(R_i)} S_{i,j} :: \tau_{i,j} \times SW(S_{i,j}, SP_{i,y})}{SC(SP_{i,y})} \\ &\quad + W_{TC} \times \frac{\sum_{j=1}^{SC(R_i)} TC(S_{i,j}) \times SW(S_{i,j}, SP_{i,y})}{\sum_{j=1}^{SC(R_i)} TC(S_{i,j})} + W_{LT} \times \frac{Max(SPL_{i,y})}{Max(L_i)} \end{aligned}$$

$$\forall\ 0 < W_{count} < 1, 0 < W_{ST} < 1,\ 0 < W_{TC} < 1,\ 0 < W_{LT} < 1,$$
$$W_{count} + W_{ST} + W_{TC} + W_{LT} = 1,$$
$$SW(S_{i,j}, SP_{i,y}) = \begin{cases} 1, & \text{if service } j \text{ of registry } i \text{ is provided by} \\ & \text{provider } y \\ 0, & \text{otherwise} \end{cases}, \tag{12}$$
$$LT(S_{i,j}) = S_{i,j} :: \beta_{i,j},$$
$$SPL_{i,y} = \left\{ \begin{array}{l} S_{i,1} :: \beta_{i,1} \times SW(S_{i,1}, SP_{i,y}), \ldots, \\ S_{i,SC(R_i)} :: \beta_{i,SC(R_i)} \times SW(S_{i,SC(R_i)}, SP_{i,y}) \end{array} \right\},$$
$$L_i = \left\{ S_{i,1} :: \beta_{i,1}, S_{i,2} :: \beta_{i,2}, \ldots, S_{i,SC(R_i)} :: \beta_{i,SC(R_i)} \right\}$$

$SC(R_i)$: The number of WS in WSReg i.
$SC(SP_{i,y})$: The number of WS published by WSP y in WSReg i.
$SW(S_{i,j}, SP_{i,y})$: Is WS j belongs to WSP y in WSReg i.
$TC(S_{i,j})$: The number of transaction belongs to WS j in WSReg i.
$LT(S_{i,j})$: The storage time of WS j in WSReg i.
$Max(SPL_{i,y})$: The longest storage time of all WS belongs to WSP y in WSReg i.
$Max(L_i)$: The longest storage time of all WS in WSReg i.
W_{count}, W_{ST}, W_{TC}, and W_{LT} stood for the weighting of number of WS, satisfying value of WS security, number of transaction, and storage time of WSP.

3.2.2.2. The adjusted honor value of WSReq. The adjusted honor value of WSReq x in WSReg i is defined as following.

$$SR_{i,x} :: \upsilon_{i,x} = W_{old} \times Honor_{old}(SR_{i,x}) + W_{new} \times Honor_{new}(SR_{i,x}) \tag{13}$$

$SR_{i,x} :: \upsilon_{i,x}$: The honor value of WSReq x in WSReg i.
$Honor_{old}(SR_{i,x})$: The original honor value of WSReq x in WSReg i.
$Honor_{new}(SR_{i,x})$: The new honor value of WSReq x in WSReg i.
W_{old} and W_{new}: The weighting of original and new honors value of WSReq x in WSReg i.

$$\begin{aligned}
Honor_{new}(SR_{i,x}) \;=\; & W_{count} \times \frac{\sum_{j=1}^{SC(R_i)} \bigcup_{k=1}^{TC(S_{i,j})} TW(S_{i,j,k}, SR_{i,x})}{SC(R_i)} \\
& + W_{TC} \times \frac{\sum_{j=1}^{SC(R_i)} \sum_{k=1}^{TC(S_{i,j})} TW(S_{i,j,k}, SR_{i,x})}{\sum_{j=1}^{SC(R_i)} TC(S_{i,j})} + W_{LT} \times \frac{Max(SRL_{i,x})}{Max(L_i)} \\
& \forall\ 0 < W_{count} < 1,\ 0 < W_{TC} < 1,\ 0 < W_{LT} < 1, \\
& W_{count} + W_{TC} + W_{LT} = 1, \\
& TW(S_{i,j,k}, SR_{i,x}) = \begin{cases} 1, & \text{if transaction } k \text{ of WS } j \text{ of WSReg } i \\ & \text{is consumed by WSReq } x \\ 0, & \text{otherwise} \end{cases}, \\
& LT(S_{i,j}) = S_{i,j} :: \beta_{i,j}, \\
& SRL_{i,x} = \left\{ \begin{array}{l} S_{i,1} :: \beta_{i,1} \times TW(S_{i,1,k}, SR_{i,x}), \ldots, \\ S_{i,SC(R_i)} :: \beta_{i,SC(R_i)} \times TW(S_{i,SC(R_i),k}, SR_{i,x}) \end{array} \right\}, \\
& L_i = \left\{ S_{i,1} :: \beta_{i,1}, S_{i,2} :: \beta_{i,2}, \ldots, S_{i,SC(R_i)} :: \beta_{i,SC(R_i)} \right\}
\end{aligned} \tag{14}$$

$S_{i,j,k}$: The transaction k of the WS j in WSReg i.
$SC(R_i)$: The number of WS in WSReg i.
$TW(S_{i,j,k}, SR_{i,x})$: The transaction k of WS j belongs to WSReq x in WSReg i.
$TC(S_{i,j})$: The number of transactions of WS j in WSReg i.
$LT(S_{i,j})$: The storage time of WS j in WSReg i.
$Max(SRL_{i,x})$: The longest storage time of all transactions belongs to WSReq x in WSReg i.
$Max(L_i)$: The longest storage time of all WS in WSReg i.
W_{count}, W_{ST}, W_{TC}, and W_{LT} stood for the weighting of number of WS, satisfying value of WS, number of transactions, and storage time of WSReq.

$$content(S_{i,j,k}) = content\_feedback_{i,j,k} \times SR_{i,x} :: \upsilon_{i,x}\ \forall\ 0 \le content\_feedback_{i,j,k} \le 1 \tag{15}$$

$content_{i,j,k}$: The content satisfying value of transaction k of WS j in WSReg i.
$content\_feedback_{i,j,k}$: The content satisfying value of transaction k of WS j which is feedback from WSReq x in WSReg i.

$$security(S_{i,j,k}) = security\_feedback_{i,j,k} \times SR_{i,x} :: \upsilon_{i,x}\ \forall\ 0 \le security\_feedback_{i,j,k} \le 1 \tag{16}$$

$security_{i,j,k}$: The security satisfying value of transaction k of WS j in WSReg i.

$security\_feedback_{i,j,k}$: The security satisfying value of transaction k of WS j which is feedback from WSReq x in WSReg i.

3.2.2.3. The adjusted trust value of WS. The trust value of WS j in WSReg i is defined as the following.

$$S_{i,j} :: \tau_{i,j} = W_{old} \times Trust_{old}(S_{i,j}) + W_{new} \times Trust_{new}(S_{i,j}) \tag{17}$$

$S_{i,j} :: \tau_{i,j}$: The trust value of WS j in WSReg i.

$Trust_{old}(S_{i,j})$: The original trust value of WS j in WSReg i.

$Trust_{new}(S_{i,j})$: The new trust value of WS j in WSReg i.

$$\begin{aligned} Trust_{new}(S_{i,j}) \;=\; & W_{content} \times \frac{\sum_{k=1}^{TC(S_{i,j})} content(S_{i,j,k})}{TC(S_{i,j})} + W_{security} \times \frac{\sum_{k=1}^{TC(S_{i,j})} security(S_{i,j,k})}{TC(S_{i,j})} \\ & + W_{TC} \times \frac{TC(S_{i,j}) = S_{i,j} :: \alpha_{i,j}}{Max(T_i)} + W_{LT} \times \frac{LT(S_{i,j}) = S_{i,j} :: \beta_{i,j}}{Max(L_i)} \\ & + W_{success} \times (1 - FR(S_{i,j})) \end{aligned}$$

$$\begin{aligned} & \forall\ 0 < W_{content} < 1,\ 0 < W_{security} < 1,\ 0 < W_{TC} < 1, 0 < W_{LT} < 1,\ 0 < W_{success} < 1, \\ & W_{content} + W_{security} + W_{TC} + W_{LT} + W_{success} = 1, \\ & T_i = \left\{ S_{i,j} :: \alpha_{i,1}, S_{i,j} :: \alpha_{i,2}, \ldots, S_{i,j} :: \alpha_{i,SC(R_i)} \right\}, \\ & L_i = \left\{ S_{i,j} :: \beta_{i,1}, S_{i,j} :: \beta_{i,2}, \ldots, S_{i,j} :: \beta_{i,SC(R_i)} \right\} \end{aligned} \tag{18}$$

$TC(S_{i,j})$: The number of transactions belongs to WS j in WSReg i.

$content(S_{i,j,k})$: The content satisfying value of transaction k of WS j in WSReg i.

$security(S_{i,j,k})$: The security satisfying value of transaction k of WS j in WSReg i.

$LT(S_{i,j})$: The storage time of WS j in WSReg i.

$Max(T_i)$: The maximum frequency of transactions of all WS in WSReg i.

$Max(L_i)$: The maximum storage time of all WS in WSReg i.

$SC(R_i)$: The number of WS in WSReg i.

$FR(S_{i,j})$: The transaction failure rate of WS j in WSReg i.

$W_{content}$, $W_{security}$, W_{TC}, W_{LT}, and $W_{success}$ stood for the weighting of content satisfying, security satisfying, number of service transaction, service storage time, and access service success, respectively.

3.2.3. *Adjusting reputation values and trust values for data exchanging (Step 3).*

3.2.3.1. The reputation value of WSReg. The number of WS published in WSReg i is $SC(R_i)$. The reputation value of WSReg i is defined as the following.

$$Reputation(R_i) = R_i :: r_i = NR(R_i) \times CA(R_i) \times \frac{\sum_{j=1}^{SC(R_i)} S_{i,j} :: \tau_{i,j}}{SC(R_i)} \tag{19}$$

$Reputation(R_i) = R_i :: r_i$: The reputation of WSReg i.

$$NR(R_i) = \begin{cases} 1, & \text{if WSReg } i \text{ provides NR} \\ 0, & \text{otherwise} \end{cases} \tag{20}$$

$$CA(R_j) = \begin{cases} 1, & \text{if WSReg } i \text{ provides CA} \\ 0, & \text{otherwise} \end{cases} \tag{21}$$

The trust value that belongs to one WS in WSReg i is higher score when the WSReg i supports NR and CA.

3.2.3.2. The trust value of WS for data exchanging. The trust value of exchanging WS j in WSReg i is defined as the following.

$$S_{i,j} :: \tau_{i,j} = W_{old} \times Trust_{old}(S_{i,j}) + W_{new} \times Trust_{new}(S_{i,j}) \tag{22}$$

$S_{i,j} :: \tau_{i,j}$: The trust value of WS j in WSReg i.

$Trust_{old}(S_{i,j})$: The original trust value of WS j in WSReg i.
$Trust_{new}(S_{i,j})$: The new trust value of WS j in WSReg i.

$$Trust_{new}(S_{i,j}) = \frac{\sum_{e=1}^{n} S_{e,j} :: \tau_{e,j} \times Reputation(R_e)}{\sum_{e=1}^{n} Reputation(R_e)} \quad \forall\ R_i \neq R_e \tag{23}$$

$Reputation(R_e) = R_e :: r_e$: The reputation of WSReg e.

4. **System Architecture and Evaluation.** In this section, we design architecture of TWSRDE and evaluate the trust mechanism which considers the physical space factors and concept space factors described in Section 3.

4.1. **System architecture.** The architecture of TWSRDE adopts hybrid architecture which is combined the advantage of central architecture and distributed architecture and shown as Figure 3. The users can use all kinds of terminal devices to access the web-based applications in TWSRDE. WSP can publish their WS (e.g., booking ticket service) to WSReg. In TWSRDE, the EWSR provides trust mechanism and returns the trust value to users. The Intelligent Agents (IAs) in EWSR exchange the foreign services data with extra WSReg.

4.1.1. *Users.* The users can use all kinds of terminal devices to send their request to TWSRDE. TWSRDE will use the trust mechanism to recommend the WS with higher trust value for users.

4.1.2. *Web service providers.* WSP build SOAP environment such as AXIS2 to provide some services for user invocation. After building services, WSP can publish the information of business, services, and binding templates to WSReg through heterogeneous networks. For security, we can modify the AXIS2 API (such as upload.jsp) to build the hash code of service by MD5 algorithm [12-14].

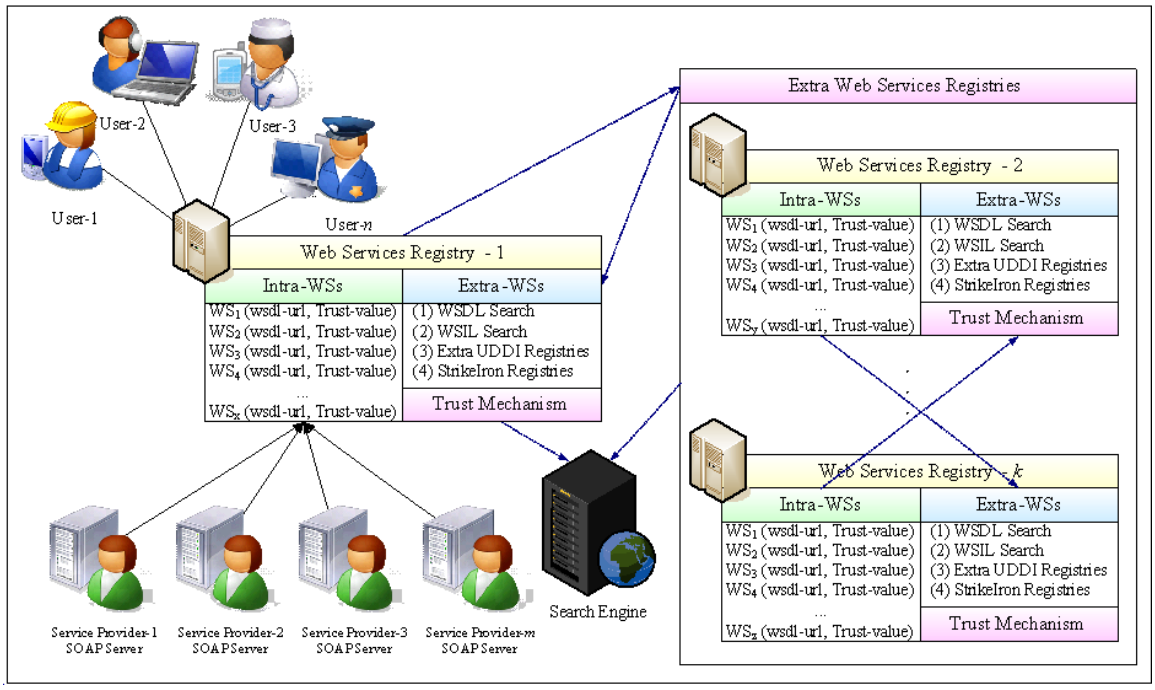

FIGURE 3. The system architecture of TWSRDE

4.1.3. *Enhance web services registry.* The aim of semantic web is to locate services automatically based on the functionalities. UDDI is helpful to discovery web services with semantic web. Therefore, we use the JUDDI to build UDDI environment which provides Business Entities, Service Entities, Binding Templates, and tModels to represent the detail of business and its services. Services in JUDDI can be searched by name, by location, by business, by bindings or by tModels. However, JUDDI does not support any inference based on the taxonomies referred to by the tModels. Integration of semantic web and JUDDI will solve this problem. And then, EWSR which is proposed in this paper can retrieve the detail and relationship of those services in JUDDI by UDDI4J APIs for the semantic inference easier [12-14]. Moreover, EWSR combines the trust mechanism to calculate and update the trust value according to user's feedback for malice WS avoidance.

4.1.4. *Extra web services registries.* EWSR can support the several WS discovery approaches (e.g., WSDL Search, WSIL Search, UDDI Query, and Use StrikeIron Registry [12-14]) to exchange the WS data with the extra WSReg.

4.2. **Evaluation.** In experiments, there are 20 WSP and 8000 WSReq in WSReg. We assume that the WS inter-request time function, inter-publish time function, feedback value function are uniform distribution functions, and the rate of malice WS is 60%. Moreover, we simulate the results of WS transactions with five cases which are (1) the different rates of malice WS, (2) the different numbers of WSP, (3) the different weighting of trust factor increments for WSP, (4) the different weighting of trust factor increments for WSReq, and (5) the different weighting of trust factor increments for WS. The simulation parameters are shown in Table 1.

Table 1. Simulation parameters

Parameter	**Value**
Number of WSP	20
Number of WSReq	8,000
Mean WS inter-request time	1 (cycle time)
Mean WS inter-publish time	1 (cycle time)
Rate of malice WS	60%
Feedback for normal WS	(50%, 100%]
Feedback for malice WS	[0%, 50%)

4.2.1. *Case 1: The different rates of malice WS.* In this section, we adopt the parameters as Table 1 to simulate the WS transactions with and without trust mechanism. The results which are shown in Figure 4 indicate the lower ratio of transaction of malice WS with trust mechanism.

We also simulate the different rates of malice WS which are from 10% to 90%. Figure 5 shows that the results of the WS transactions with the different rates of malice WS. The ratio of transaction of malice WS is larger at the start time when the number of malice WS is larger. However, this problem can be solved by the increasing feedbacks through trust mechanism.

4.2.2. *Case 2: The different numbers of WSP.* We adopt some parameters as Table 1 and set that the numbers of WSP are 10, 20, 40, 80, and 160 to simulate the WS transactions with the different numbers of WSP. The results of WS transactions with the different numbers of WSP are shown in Figure 6. The simulation results show that the ratio of transaction of malice WS is lower when the simulation time is longer. Although the

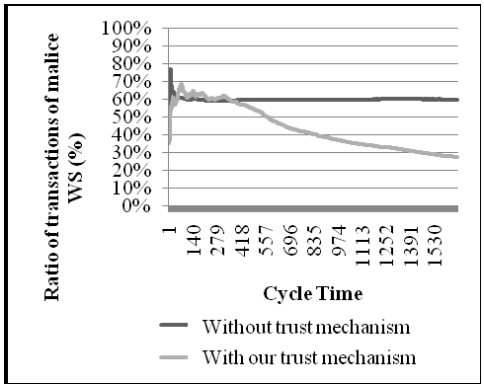

FIGURE 4. WS transactions with and without trust mechanism

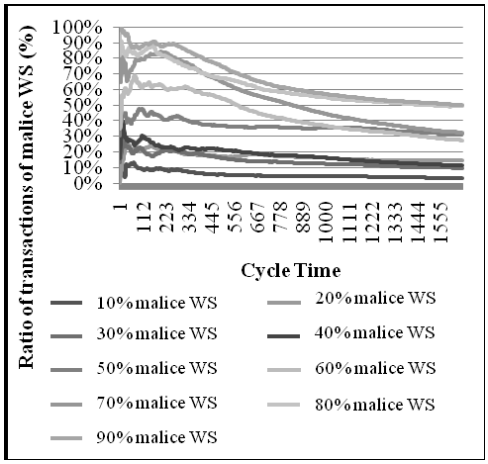

FIGURE 5. WS transactions with the different rates of malice WS

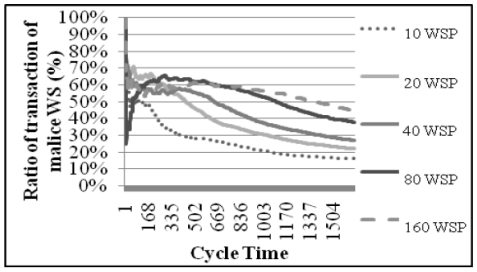

FIGURE 6. WS transactions with the different numbers of WSP

number of malice WS is larger when the number of WSP is larger. This problem can be solved by the increasing feedbacks from WSReq.

4.2.3. *Case 3: The different weighting of trust factor increments for WSP.* The results of WS transactions with the different weighting of trust factor increments for WSP are shown in Figure 7. The simulation results show that the priority of all different importance weightings is $W_{LT} > W_{count} > W_{ST} > W_{TC}$. Therefore, we can know that the storage time is longer; WSReq can avoid requesting and invoking the malice WS through our trust mechanism.

4.2.4. *Case 4: The different weighting of trust factor increments for WSReq.* The results of WS transactions with the different weighting of trust factor increments for WSReq are shown in Figure 8. The simulation results indicate that the priority of all different

importance weightings is $W_{LT} > W_{TC} > W_{count}$. The storage time is similar with the results as described above. Moreover, the simulation results show that the numbers of malice WS are more than the normal WS. The trust mechanism can adjust the trust value and recommend users the normal WS according to the transactions and feedback.

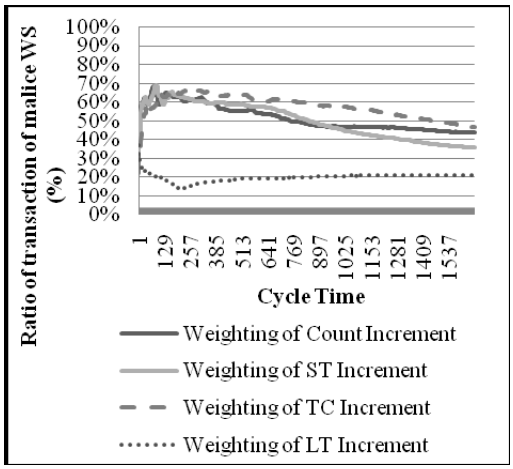

FIGURE 7. WS transactions with the different weighting of trust factor increments for WSP

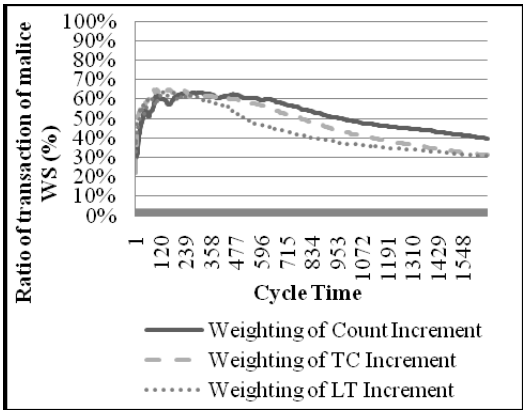

FIGURE 8. WS transactions with the different weighting of trust factor increments for WSReq

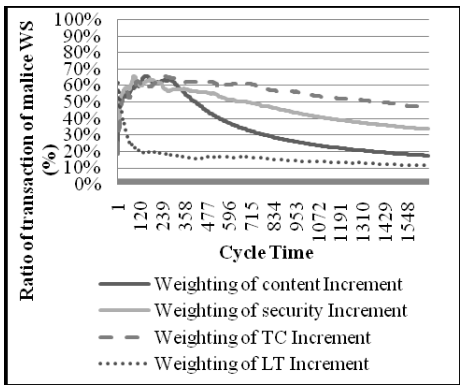

FIGURE 9. WS transactions with the different weighting of trust factor increments for WS

4.2.5. *Case 5: The different weighting of trust factor increments for WS.* The results of WS transactions with the different weighting of trust factor increments for WS are shown in Figure 9. The simulation results indicte that the feedback of WS content satisfying value and security satisfying value can reduce the transaction times of request malice WS. The priority of all different importance weightings is $W_{LT} > W_{content} > W_{security} > W_{TC}$. By our proposed trust mechanism cannot only overcome the shortcomings of the WS-Security, but also find the reliable WS.

5. **Conclusions and Future Work.** According to the shortcoming of WS-Security specification, we propose an architecture of TWSRDE which combines a novel trust mechanism to consider the physical space information (i.e., NR, CA, Int, and Con) and concept space information (i.e., UFs) to improve the security foundations as described in the WS-Security specification which only guarantees the security on end-to-end level. Our proposed method can provide the reliable WS to the WSReq. TWSRDE includes users, WSP in SOAP server, EWSR, and extra WSReg. WSReg combines the IAs which provide the heterogeneous WS resources search to fetch new WS information and process the analysis of WS reliability for WS data exchanging with different WSReg.

Although the proposed trust mechanism considers physical space information and concept space information, it only can calculate the trust value of single WS. This trust mechanism cannot support the trust value calculation for complex WS. Therefore, the security requirements for complex WS can be considered in the future.

Acknowledgment. The research is supported by the National Science Council of Taiwan under the grant Nos. NSC 100-2811-H-009-011 and NSC 100-2622-H-009-001-CC3.

REFERENCES

[1] G. Yee and L. Korba, Security personalization for Internet and web services, *International Journal of Web Services Research*, vol.5, no.1, pp.1-23, 2008.

[2] Z. Wu and A. C. Weaver, A privacy preserving enhanced trust building mechanism for web services, *Proc. of the 3rd Annual Conference on Privacy, Security and Trust*, Canada, 2005.

[3] Z. Wu and A. C. Weaver, Dynamic trust establishment with privacy protection for web services, *Proc. of the 3rd IEEE International Conference on Web Services*, Florida, 2005.

[4] Z. Wu and A. C. Weaver, Token-based dynamic trust establishment for web services, *Proc. of the 43rd ACM Southeast Conference*, USA, 2005.

[5] S. Dustdar and M. Treiber, A view based analysis on web service registries, *Distributed and Parallel Databases*, vol.18, no.2, pp.147-171, 2005.

[6] Y. Gil and D. Artz, Towards content trust of web resources, *Journal of Semantic Web*, vol.5, no.4, pp.227-239, 2007.

[7] Y. M. Liu, S. B. Yang, W. M. Chen and W. Dong, The research of the reputation-aware super node selection algorithm in P2P system, *Journal of the Graduate School of the Chinese Academy of Sciences*, vol.25, no.2, pp.197-203, 2008.

[8] A. A. Pirzada, A. Datta and C. McDonald, Propagating trust in ad-hoc networks for reliable routing, *Proc. of the International Workshop on Wireless Ad-Hoc Networks*, 2004.

[9] Y. Wang, D. S. Wong, K. J. Lin and V. Varadharajan, Evaluating transaction trust and risk levels in peer-to-peer E-commerce environments, *Information Systems and E-Business Management*, vol.6, no.1, pp.25-48, 2008.

[10] Y. Wang and K. J. Lin, Reputation-oriented trustworthy computing in E-commerce environments, *IEEE Internet Computing*, vol.12, no.4, pp.55-59, 2008.

[11] Y. Wang, K. J. Lin, D. S. Wong and V. Varadharajan, Trust management towards service-oriented applications, *Service Oriented Computing and Applications*, vol.3, no.2, pp.129-146, 2009.

[12] C. C. Lo, T. H. Kuo, H. Y. Kung, H. T. Kao, C. H. Chen, C. I. Wu and D. Y. Cheng, Mobile merchandise evaluation service using novel information retrieval and image recognition technology, *Computer Communications*, vol.34, no.2, pp.120-128, 2011.

[13] C. C. Lo, C. H. Chen, D. Y. Cheng and H. Y. Kung, Ubiquitous healthcare service system with context-awareness capability: Design and implementation, *Expert Systems with Applications*, vol.38, no.4, pp.4416-4436, 2011.

[14] B. Y. Lin, C. H. Chen, H. C. Chang and C. C. Lo, A network behavior analysis system for cloud computing service, *Information – An International Interdisciplinary Journal*, vol.14, no.3, pp.931-938, 2011.

Appendix A

The notation used in this paper is summarized in Table 2.

Table 2. Nomenclature

Notation	Meaning
$S_{i,j}$	The WS j in WSReg i
$Honor(SP_{i,y})$	The honor value of SP y in WSReg i
$Honor(SR_{i,x})$	The honor value of WSReq x in WSReg i
$Trust(S_{i,j})$	The trust value of WS j in WSReg i
$SC(R_i)$	The number of WS in WSReg i
$SC(SP_{i,y})$	The number of WS published by WSP y in WSReg i
$SW(S_{i,j}, SP_{i,y})$	Is WS j belongs to WSP y in WSReg i
$TC(S_{i,j})$	The number of transaction belongs to WS j in WSReg i
$LT(S_{i,j})$	The storage time of WS j in WSReg i
$Max(SPL_{i,y})$	The longest storage time of all WS belongs to WSP y in WSReg i
$Max(L_i)$	The longest storage time of all WS in WSReg i
$S_{i,j,k}$	The transaction k of the WS j in WSReg i
$TW(S_{i,j,k}, SR_{i,x})$	The transaction k of WS j belongs to WSReq x in WSReg i
$Max(SRL_{i,x})$	The longest storage time of all transactions belongs to WSReq x in WSReg i
$content_{i,j,k}$	The content satisfying value of transaction k of WS j in WSReg i
$content\_feedback_{i,j,k}$	The content satisfying value of transaction k of WS j which is feedback from WSReq x in WSReg i
$security_{i,j,k}$	The security satisfying value of transaction k of WS j in WSReg i
$security\_feedback_{i,j,k}$	The security satisfying value of transaction k of WS j which is feedback from WSReq x in WSReg i
$content(S_{i,j,k})$	The content satisfying value of transaction k of WS j in WSReg i
$security(S_{i,j,k})$	The security satisfying value of transaction k of WS j in WSReg i
$Max(T_i)$	The maximum frequency of transactions of all WS in WSReg i
$FR(S_{i,j})$	The transaction failure rate of WS j in WSReg i
$Reputation(R_e)$	The reputation of WSReg e