1-1-2005

Analysis and modelling of trust in distributed information systems

Weiliang Zhao
University of Wollongong, wzhao@uow.edu.au

Vijay Varadharajan
Macquarie University

George Bryan
University Of Western Sydney

Analysis and Modelling of Trust in Distributed Information Systems

Weiliang Zhao[1], Vijay Varadharajan[1,2], and George Bryan[1]

[1] School of Computing and Information Technology,
University of Western Sydney
{wzhao, g.bryan}@cit.uws.edu.au
[2] Department of Computing,
Macquarie University
vijay@ics.mq.edu.au

Abstract. In this paper, we consider the analysis and modelling of trust in distributed information systems. We review the relations of trust relationships in our previous work. We discuss trust layers and hierarchy based on formal definition of trust relationship. We provide a set of definitions to describe the properties of trust direction and trust symmetry under our taxonomy framework. In order to analyze and model the scope and diversity of trust relationship, we define trust scope label under our taxonomy framework. We provide some example scenarios to illustrate the proposed definitions about properties of trust relationship. The proposed definitions are new elements of the taxonomy framework for enabling the analysis and modelling of trust. We provide some discussions about the life cycle of trust relationships. The proposed trust structure and properties are currently being used in the development of the overall methodology of life cycle of trust relationships in distributed information systems.

1 Introduction

Trust has been studied in multiple dimensions in the computing world. As a concept of security, trust was firstly introduced in trusted systems [1] and trusted computing [2]. Marsh has tried to formalize trust as a computational concept [3]. Several community-based reputation systems [4,5], trust negotiation systems [6,7,8] and trust management systems [9,10,11]have been proposed.

Trust plays an important role in distributed information systems. The properties of trust and how to define/model trust relationships are important concerns in the analysis and design of distributed information systems. Our main objective of this research is to develop a sound understanding of trust and to create a powerful set of tools to analyze and model trust relationships in distributed information systems. In our earlier work [12], we have outlined a formal definition of trust relationship. The target of the formal definition of trust relationships is not only to reflect many of the commonly used notions of trust but also to provide a taxonomy framework where a range of useful trust relationships can be

S. Jajodia and C. Mazumdar (Eds.): ICISS 2005, LNCS 3803, pp. 106–119, 2005.

expressed and compared. The research in [12] only provides a starting point for the analysis and design of trust relationships. We provide a set of definitions for the properties of trust direction and trust symmetry between involved entities in distributed environments. We provide trust scope label and rules to compare trust scope labels. The operations, definitions and rules are enabling tools in the analysis and design of trust relationships in distributed information systems. We provide examples of scenarios to show users how to understand and use the proposed definitions about trust relationships in the real world. The research provided in this paper is an important part of the overall methodology of life cycle of trust relationships in distributed information systems.

The remainder of the paper is organized as follows. In section 2, we provide the definition of trust relationship. In section 3, we describe a set of operations and definitions for relations of trust relationships in distributed environments. In section 4, we discuss the trust layers and hierarchy. In section 5, we provide a set of definitions for trust direction and trust symmetry. We employee the Microsoft's domain trust as a regressive scenario example to illustrate the definitions in this section. In section 6, we discuss the scope and diversity of trust relationships. In section 7, we provide some discussions about our overall methodology of the life cycle of trust relationships in distributed environments. Finally section 8 provides some concluding remarks.

2 Definition of Trust Relationship

We have given a formal definition of trust relationship with a strict mathematical structure in our previous work [12]. This definition of trust relationship has a broad expressive power and it is the cornerstone of our trust taxonomy framework. All trust notions proposed in this paper is based on this definition. The definition of trust relationship is expressed as:

Definition 1. *A trust relationship is a four-tuple* $T =< R,\ E,\ C,\ P >$ *where:*

- R *is the set of trusters. It contains all the involved trusters. It is a non empty set.*
- E *is the set of trustees. It contains all the involved trustees. It is a non-empty set.*
- C *is the set of conditions. It contains all conditions (requirements) for the current trust relationship. Normally, a trust relationship has some specified conditions. If there is no condition, the condition set is empty.*
- P *is the set of properties. The property set describes the actions or attributes of the trustees. It is a non-empty set. The property set can be divided into two sub sets:*
  - *Action set: the set of actions that the trusters trust that trustees will and can perform.*
  - *Attribute set: the set of attributes that trusters trust that trustees have.*

The formal definition of trust relationship can reflect the commonly used notions of trust and provides a taxonomy framework. When trust relationships are used,

the full syntax (four-tuple $< R,\ E,\ C,\ P >$ must be followed. Trust relationship T means that under the condition set C, truster set R trust that trustee set E have the properties in set P. The definition of trust relationship provides a starting point for capturing different forms of commonly understood notions of trust. The above strict definition of the trust relationship is the basis for the discussions of all properties of trust in this paper.

3 Relations of Trust Relationships

In the this section, we provide some operations and definitions about the relations of trust relationships. The relations of trust relationships play an important role in the analysis and design of trust relationships in distributed information systems. From the nature of trust relationship and its mathematical structure, some new trust relationships can be derived based on the existing trust relationships. The operations of using two existing trust relationships to generate a new trust relationship under specific constraints and operations of decomposing one existing trust relationship into two new trust relationships under specific constraints are defined as follows:

OPERATION 1. *Let* $T_1 = (R_1,\ E_1,\ C_1,\ P_1)$ *and* $T_2 = (R_2,\ E_2,\ C_2,\ P_2)$. *There is a set* $T = (R_1 \cap R_2,\ E_1 \cap E_2,\ C_1 \cup C_2,\ P_1 \cup P_2)$. *If* $R_1 \cap R_2 = \emptyset$ *or* $E_1 \cap E_2 = \emptyset,\ T = \emptyset$.

If $R_1 = R_2$ and $E_1 = E_2$, the operation becomes:

OPERATION 1A. *Let* $T_1 = (R,\ E,\ C_1,\ P_1)$ *and* $T_2 = (R,\ E,\ C_2,\ P_2)$. *There is a set* $T = (R,\ E,\ C_1 \cup C_2,\ P_1 \cup P_2)$.

If $R_1 = R_2,\ E_1 = E_2$ and $C_1 = C_2$, the operation becomes:

OPERATION 1B. *Let* $T_1 = (R,\ E,\ C,\ P_1)$ *and* $T_2 = (R,\ E,\ C,\ P_2)$. *Then there is a set* $T = (R,\ E,\ C,\ P_1 \cup P_2)$.

OPERATION 2. *Let* $T_1 = (R_1,\ E_1,\ C,\ P)$ *and* $T_2 = (R_2,\ E_2,\ C,\ P)$. *There is a set* $T = (R_1 \cup R_2,\ E_1 \cap E_2,\ C,\ P)$.

If $E_1 = E_2$, the operation becomes:

OPERATION 2A. *Let* $T_1 = (R_1,\ E,\ C,\ P)$ *and* $T_2 = (R_2,\ E,\ C,\ P)$. *There is a set* $T = (R_1 \cup R_2,\ E,\ C,\ P)$.

OPERATION 3. *Let* $T_1 = (R_1,\ E_1,\ C,\ P)$ *and* $T_2 = (R_2,\ E_2,\ C,\ P)$. *There is a set* $T = (R_1 \cap R_2,\ E_1 \cup E_2,\ C,\ P)$.

If $R_1 = R_2$, the operation becomes:

OPERATION 3A. *Let* $T_1 = (R,\ E_1,\ C,\ P)$ *and* $T_2 = (R,\ E_2,\ C,\ P)$. *There is a set* $T = (R,\ E_1 \cup E_2,\ C,\ P)$.

OPERATION 4. *Let* $T =< R,\ E,\ C,\ P >$. *If there are* $R_1,\ R_2$ *and* $R = R_1 \cup R_2$, *then there are trust relationships* $T_1 =< R_1,\ E,\ C,\ P >$ *and* $T_2 = < R_2,\ E,\ C,\ P >$.

OPERATION 5. *Let* $T =< R,\ E,\ C,\ P >$. *If there are* $E_1,\ E_2$ *and* $E = E_1 \cup E_2$, *then there are trust relationships* $T_1 =< R,\ E_1,\ C,\ P >$ *and* $T_2 = < R,\ E_2,\ C,\ P >$.

OPERATION 6. *Let* $T =< R,\ E,\ C,\ P >$. *If there are* $P_1,\ P_2$ *and* $P = P_1 \cup P_2$, *then there are trust relationships* $T_1 =< R,\ E,\ C,\ P_1 >$ *and* $T_2 = < R,\ E,\ C,\ P_2 >$.

This operation has the following special case:

OPERATION 6A. *Let* $T =< R,\ E,\ C,\ P >$ *and there are* $P_1,\ P_2,\ C_1, C_2$ *and* $P = P_1 \cup P_2,\ C = C_1 \cup C_2$. *If* C_1 *is the condition set for* P_1 *and* C_2 *is the condition set for* P_2, *then there are trust relationships* $T_1 =< R,\ E,\ C_1,\ P_1 >$ *and* $T_2 =< R,\ E,\ C_2,\ P_2 >$.

All operations can be used to generate new trust relationships from the existing trust relationships under some specific constraints. The **Operation 1** deals with any two trust relationships and a new trust relationship is generated, if the result is not $\emptyset$. The **Operation 1A, 1B, 2A, 3A** deal with how to use two trust relationships to generate new trust relationship under some specific constraints. The **Operation 4, 5, 6 and 6A** deal with how to decompose one trust relationship into two trust relationships under some specific constraints. **Operation 1A** and **Operation 6A** are inverse operations. **Operation 1B** and **Operation 6** are inverse operations. **Operation 2A** and **Operation 4** are inverse operations. **Operation 3A** and **Operation 5** are inverse operations.

In the discussion of trust relationships, we have defined the equivalent, primitive, derived, direct redundant and alternate trust relationships and have classified the direct redundant trust relationships into different types. They are as follows:

Definition 2. *Let* $T_1 =< R_1,\ E_1,\ C_1,\ P_1 >$ *and* $T_2 =< R_2,\ E_2,\ C_2,\ P_2 >$. *If and only if* $R_1 = R_2$ *and* $E_1 = E_2$ *and* $C_1 = C_2$ *and* $P_1 = P_2$, *then* T_1 *and* T_2 *are equivalent, in symbols:*

$$T_1 = T_2 \iff R_1 = R_2 \text{ and } E_1 = E_2 \text{ and } C_1 = C_2 \text{ and } P_1 = P_2$$

Definition 3. *If a trust relationship cannot be derived from other existing trust relationships, the trust relationship is a primitive trust relationship.*

Definition 4. *If a trust relationship can be derived from other existing trust relationships, the trust relationship is a derived trust relationship.*

Note: Trust relationships are predefined in information systems. A derived trust relationship is always related to one or more other trust relationships. For an independent trust relationship, it is meaningless to judge it as a derived trust relationship or not.

Definition 5. *Let* $T =< R, E, C, P >$. *If there is trust relationship* $T' = < R', E', C', P' >$ *and* $T \neq T'$, $R \subseteq R'$, $E \subseteq E'$, $C \supseteq C'$, $P \subseteq P'$. T *is a direct redundant trust relationship.*

We now discuss several special cases of direct redundant trust relationships based on the single tuple of trust relationship. We believe that these special cases play important roles in the analysis and design of trust relationships.

TYPE 1*: DRLR (Direct Redundant of Less Trusters)*
Let $T =< R, E, C, P >$. *If and only if there is a trust relationship* $T' = < R', E, C, P >$ *and* $R' \supset R$, T *is a DRLR trust relationship.*

T is DRLR trust relationship means that there is another trust relationship with super set of trusters and all other tuples are same as peers in T.

TYPE 2*: DRLE (Direct Redundant of Less Trustees)*
Let $T =< R, E, C, P >$. *If and only if there is a trust relationship* $T' = < R, E', C, P >$ *and* $E' \supset E$, T *is a DRLE trust relationship.*

T is DRLE trust relationship means that there is another trust relationship with super set of trustees and all other tuples are same as peers in T.

TYPE 3*: DRMC (Direct Redundant of More Conditions)*
Let $T =< R, E, C, P >$. *If and only if there is an alternate trust relationship* $T' =< R, E, C', P >$ *and* $C' \subset C$, T *is a DRMC trust relationship.*

T is DRMC trust relationship means that there is another trust relationship with a subset of conditions and all other tuples are same as peers in T.

TYPE 4*: DRLP (Direct Redundant of Less Properties)*
Let $T =< R, E, C, P >$. *If and only if there is a trust relationship* $T' = < R, E, C, P' >$ *and* $P' \supset P$, T *is a DRLP trust relationship.*

T is DRLP trust relationship means that there is another trust relationship with super set of properties and all other tuples are same as peers in T.

Definition 6. *Let* $T =< R, E, C, P >$, $T' =< R, E, C', P >$ *and* $C \neq C'$. T *and* T' *are alternate trust relationships of each other.*

An alternate trust relationship means that there is an alternate condition set for the same truster set, trustee set and property set. Perhaps, there are multiple alternate trust relationships. In distributed computing, multiple mechanisms and multiple choices are necessary in many situations and it is the main reason why we define and discuss alternate trust relationships here.

Scenario Example: Consider an online e-commerce service called FlightServ, which can provide flight booking and travel deals. FlightServ is designed using web services. FlightServ connects with customers, airlines, hotels and credit card services (some of these may also be web services). The whole system could be very complicated, but in this example, we only consider some basic trust relationships in the system. In the system, customers are classified into normal flyers and frequent flyers. Originally, some trust relationships are modelled as follows:

TS2- 1. *Airlines trust normal flyers can make their airline bookings, if they have address details & confirmed credit card information.*

TS2- 2. *Airlines trust frequent flyers with no condition that frequent flyers can make their airline bookings.*

TS2- 3. *Hotels trust normal flyers can make their hotels booking, if they have address details & confirmed credit card information.*

TS2- 4. *Hotels trust frequent flyers can make their hotels booking, if they have address details & confirmed credit card information.*

TS2- 5. *Credit card services are trusted by all possible entities without any condition that the credit card services will give the correct evaluation of credit card information.*

TS2- 6. *Credit card services are trusted by all possible entities without any condition that the credit card services will keep the privacy of credit card information.*

For the above trust relationships in the system, based on definitions and operations in section 3, we have the following analysis:

- All above trust relationships are primitive.
- Using the **Operation 3A**, trust relationships **TS2-3** and **TS2-4** can be merged to a new trust relationship **TS2-(3)(4)**: "Hotels trust customers if they have address details & confirmed credit card information that customers can make their hotels booking". If **TS2-(3)(4)** has been defined in the system, **TS2-3** and **TS2-4** becomes DRLE trust relationships and will be removed out of the system.
- Using the **Operation 1B**, trust relationships **TS2-5** and **TS2-6** can be merged to a new trust relationship **TS2-(5)(6)**: "Credit card services are trusted by all possible entities without any condition that the credit card services will give the correct evaluation of credit card information & the credit card services will keep the privacy of credit card information". If **TS2-(5)(6)** has been defined in the system, **TS2-5** and **TS2-6** becomes DRLP trust relationships and will be removed out of the system.

We hope that the above scenario example can provide a general picture of modelling trust relationships in distributed environments. We believe that these operations and definitions are useful but they are not sufficient for the overall methodology of modelling trust relationships in distributed environments. In following sections, we will expand the taxonomy framework and discuss the classification of trust, trust layers, direction and symmetry of trust and the life cycle of trust relationships in distributed environments.

4 Trust Layers and Hierarchy

Some researchers have tried to identify different forms of trust relationships [13]. Grandison et al [13] have given a bottom-up classification and used the terms as

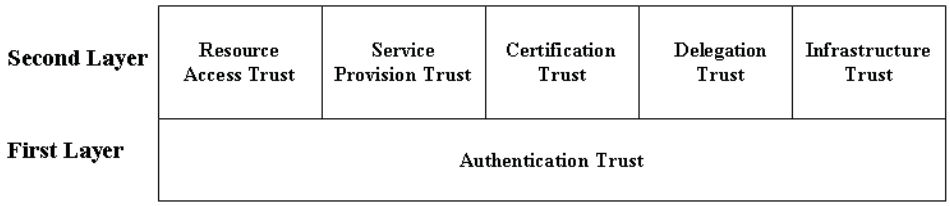

Fig. 1. Trust Layers

resources access trust, service provision trust, certification trust, delegation trust and infrastructure trust. From the view point of establishment or evaluation of trust relationships, all the above trust types must build on a more basic trust relationship which is the authentication trust or identity trust.We will categorize the trust relationships into two layers. Authentication is on layer one and other types are on layer two. We will give a hierarchy of trust based on the nature of the four tuples of trust relationship.

Authentication has continuously been an important topic in information security community. There are many popular authentication schemes such as X.509 and PGP. Authentication trust belongs to a separate layer and all other trust types belong to another layer above the authentication trust. This is illustrated in Figure 1. Note that trust types of layer two may not be necessarily specified in terms of an identity. Anonymous authorization belongs to access trust and it is an example that there is no specified identity. Anonymous authorization can be implemented using certificates with capabilities. The real identity of the involved trustee will not be revealed. For example, a customer has a certificate for accessing some resources on the Internet. The customer's behaviors of accessing the resources can be recorded. If it is desirable that the customer cannot be identified, the related access trust is a kind of anonymous access trust. Particularly for the resource access trust and service provision trust, the anonymous authentication is desirable in some cases. In such a situation, the layer of authentication still needs to provide a mechanism to deal with the same entity as the trustee in the whole scope of the trust process. Normally, there is a temporary and dynamic identification which will be uniquely connected with the involved trustee in the scope of the trust process.

Authentication trust is the only type of trust at layer one. At layer two, trust relationships can be classified in different ways. In the following, we will give another kind of classification which is different from the bottom-up classification of Grandison et al. Based on strict definition of trust relationship, trust relationships at layer two can be classified according to the nature of the trustees in trust relationship $< R,\ E,\ C,\ P >$. If E is an infrastructure, the trust relationship belongs to infrastructure trust. If E is not an infrastructure, the trust relationship belongs to non-infrastructure trust. Non-infrastructure trust relationships can be classified based on the ownership of the property set. If the trusters have the ownership of the property set, the trust relationship belongs to access trust. If the trustees have the ownership of the property set, the trust relationship belongs to provision trust. If some properties are owned by trustees and some other

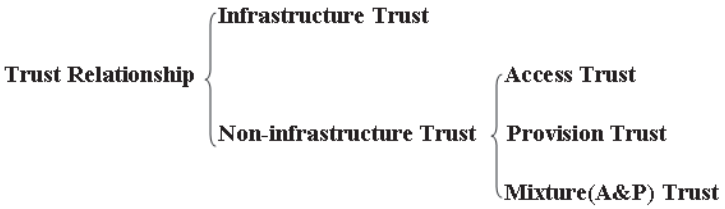

Fig. 2. Trust Hierarchy

properties are owned by trusters, then the trust relationship belongs to mixture (A&P) trust. The hierarchy of trust relationships at layer two is illustrated in Figure 2. In such a classification, delegation trust and certification trust are not independent types. As we have discussed, the delegation trust is a special form of provision trust, trustees are the providers of delegated decisions on behalves of trusters. A certification trust can be any subtype of non-infrastructure trust based on the nature of its property set.

5 Direction and Symmetry of Trust

In this section, we will provide a set of definitions for the properties of trust direction and trust symmetry. The properties of trust direction and trust symmetry play an important role in the analysis and modelling of trust in distributed information systems. These definitions provide general descriptions about the properties of trust direction and trust symmetry. A scenario example is provided to illustrate these definitions and their usage. We hope that these definitions can cover most situations in the real world and can be used as standard scenarios for analyzing and modelling trust about properties of direction and symmetry. In real systems, one or multiple kinds of trust direction and trust symmetry can be chosen based on the specified requirements of the information systems.

The properties of trust direction and symmetry are related to each other and they should be cooperatively used to analyze and model the properties of direction and symmetry of trust in distributed environments. For the properties of trust direction, one-way trust relationship, two-way trust relationship and reflexive trust relationship are defined. For the properties of trust symmetry, symmetric trust relationships, symmetric two-way trust relationship, and the whole set of trust relationships are defined. The details of the definitions are described as follows.

Definition 7. *One-way trust relationship is the trust relationship with a unique trust direction from the trusters to trustees.*

One-way is the default feature of a trust relationship if there is no further description.

Two-way trust relationship can be defined and used in information systems such as Microsoft's domain trust. Actually, two-way trust relationship is the result of binding two one-way trust relationships together. We define two-way trust relationship as follows:

Definition 8. *Two-way trust relationship TT' is the binding of two one-way trust relationships $T =< R,\ E,\ C,\ P >$ and $T' =< R',\ E',\ C',\ P' >$ with $R' = E$ and $E' = R$. T and T' are the reflective trust relationships with each other in the two-way trust relationship.*

In the above definition, "binding" is the key word. If there are two one-way trust relationships between R and E but they are not bound with each other, then they are only two one-way trust relationships and there is no two-way trust relationship. When two one-way trust relationships are bound together, there is a two-way trust relationship and these two one-way trust relationships can be called reflective trust relationships with each other.

If the trusters and the trustees are the same, the trust relationship is reflexive. The reflexive trust relationship is defined as follows:

Definition 9. *Trust relationships $T =< R,\ E,\ C,\ P >$ is an reflexive trust relationship when $R = E$.*

The symmetry of two trust relationships could be an important concern in the analysis or modelling of trust relationships in distributed information systems. The symmetry of two trust relationships is defined as the follows:

Definition 10. *If there is trust relationship $T' =< R',\ E',\ C',\ P' >$ which is the result of swapping trusters and trustees in another trust relationship $T =< R,\ E,\ C,\ P >$ (the swapping includes all possible ownerships in condition set and property set), there is symmetry between T and T', T and T' are symmetric trust relationships with each other.*

In the above definition, the swapping of trusters and trustees includes all possible ownerships in condition set and property set. The two trust relationships have the same condition set and property set except the possible ownerships in them. The symmetric/asymmetric two-way trust relationship is defined as follows:

Definition 11. *A two-way trust relationship TT' is symmetric two-way trust relationship if there is symmetry between T and T'; otherwise TT' is an asymmetric two-way trust relationship.*

Sometimes it is necessary to discuss the symmetry of all trust relationships between a truster set and a trustee set, we have the following definition:

Definition 12. *WTR(R,E) is the whole set of trust relationships with same truster set R and trustee set E.*

Definition 13. *If every trust relationship in WTR(R,E) has a symmetric trust relationship in WTR(E,R) and every trust relationship in WTR(E,R) has a symmetric trust relationship in WTR(R,E), the trust between R and E are symmetric.*

Scenario Example: Here we use Microsoft's domain trust as a regressive scenario example to discuss the properties of trust direction and trust symmetry

defined in this section. Domain trust allows users to authenticate to resources in another domain. Also, an administrator is able to administer user rights for users in the other domain. Our general definitions for the properties of direction and symmetry of trust relationships have general expressive power and can cover broad range of commonly used notations. The related concepts in domain trust can be viewed as specific cases of these general definitions. In the following, we will use our terms defined in this paper to review some concepts in domain trust.

- Based on **definition 1** in section 2, the domain trust can be expressed as "entities in domain A trust entities in domain B without any condition that entities in domain B have the right to get access of the set of resources in domain A".
- Microsoft's domain trust includes both one-way trust and two-way trust. In Microsoft's domain trust, one-way trust is defined as a unidirectional authentication path created between two domains. This means that in a one-way trust between domain A and domain B, users in domain A can access resources in domain B. However, users in domain B cannot access resources in domain A. Microsoft's one-way trust is an example of one-way trust relationship in **definition 7**. In a two-way domain trust, authentication requests can be passed between the two domains in both directions. Two-way trust is an example of two-way trust relationship in **definition 8**.
- The entities in same domain trust each other without any condition that entities have the right to get access of the set of resources in the same domain. This is an example of reflexive trust relationship in **definition 9**.
- There is symmetry in the two-way domain trust. The two one-way trust relationships bound in the two-way trust relationship are "entities in domain A trust entities in domain B without any condition that entities in domain B have the right to get access of the set of resources in domain A" and "entities in domain B trust entities in domain A without any condition that entities in domain A have the right to get access of the set of resources in domain B". These two one-way trust relationships are symmetric trust relationships with each other in **definition 10**. Microsoft's two-way trust is symmetric two-way trust relationship in **definition 11**.
- In domain trust, the $WTR(A, B)$ based on **definition 12** has only one trust relationship from truster domain A to trustee domain B. For two-way domain trust, the trust between domain A and domain B is symmetric based on **definition 13**.

The above definitions about the properties of trust direction and trust symmetry are new elements of the taxonomy framework about trust. We believe that they can cover most situations related with direction and symmetry of trust relationship in the real world. These definitions can provide suitable terms and can be used as scenario examples in the analysis and modelling of trust in distributed information systems.

6 Scope and Diversity of Trust Relationship

In this section, we will discuss the scope and diversity of trust relationship in distributed information systems. The diversity of trust has been discussed by Jøsang [14] who expresses trust in three diversity dimensions. The first dimension represents trusters or trust originators, the second represents the trust purpose, and the third represents trustees. Jøsang uses the term trust purpose based on the observation that trust is relative to a domain of actions. In our formal definition of trust relationship, trusters and trustees are two tuples and they are similar to the terms of Jøsang. The origin diversity about trusters and target diversity about trustees are straightforward and have been described clearly by Jøsang [14]. Jøsang's term of trust purpose is related to a domain of actions. Under our taxonomy framework, we will define trust scope label to take the place of the trust purpose. There are multiple benefits of trust scope label other than the trust purpose and they will be discussed later in this section. The trust scope label is the binding of the condition set and property set based on the formal definition of trust relationship. The trust scope label is a new element of our taxonomy framework. The definition of trust scope label is expressed as follows:

Definition 14. *A trust scope label is a two-tuple* $TSL =< C,\ P >$ *where* C *is a set of conditions and* P *is a set of properties.*

The details of condition set C and property set P can be found in the formal definition of trust relationship in section 2. Actually, trust scope label provides a new layer of abstraction under the trust relationship and it defines the properties of the trust and its associated conditions. To compare two trust scope labels $TSL_1 =< C_1,\ P_1 >$ and $TSL_2 =< C_2,\ P_2 >$, we have the following rules:

1. $C_1 \subseteq C_2$ and $P_1 \supseteq P_2 \iff TSL_1 \geq TSL_2$;
2. $C_1 = C_2$ and $P_1 = P_2 \iff TSL_1 = TSL_2$;
3. $C_1 \supseteq C_2$ and $P_1 \subseteq P_2 \iff TSL_1 \leq TSL_2$.
4. In other cases, TSL_1 and TSL_2 can not be compared with each other.

The trust scope label is beyond the trust purpose in several aspects. Trust scope label composes of a subspace of trust relationships (two tuples out of four tuples) and describes the characteristics of the combination of condition set C and property set P. Trust scope labels could be treated as an independent subspace of trust relationships in the analysis and design of overall information systems. The property set in trust scope label covers not only actions but also attributes of trustees. Two trust scope labels could be compared with each other based on the rules given above.

Scenario Example: Consider an online software shop. We assume that anybody who wants to enter the online shop must register as a member of the online shop first. For describing the condition set and property set in possible trust relationships between the shop and possible customers, we use the following notations:

- $p1$ stands for that customers can read the documentation of the software.
- $p2$ stands for that customers can download the software.
- $c1$ stands for certificate of membership.
- $c2$ stands for the commitment of the payment for the software.
- $c3$ stands for the payment for the software.

We have the following trust scope labels:

1. $TSL1 =< \{c1\}, \{p1\} >$
2. $TSL2 =< \{c1, c2\}, \{p1, p2\} >$
3. $TSL3 =< \{c1, c2, c3\}, \{p1, p2\} >$

Based on the rules to compare two trust scope labels, we have

- $TSL1$ cannot be compared with $TSL2$ (or $TSL3$). There is no obvious relationship between $TSL1$ and $TSL2$ (or $TSL3$).
- $TSL2 > TSL3$. It means that the trust scope of $TSL2$ is less strict than that of $TSL3$.

The scope and diversity of trust is another aspect to be considered in the analysis and modelling of trust in distributed information systems. The trust scope label may be quite complicated and the above comparison rules provide helpful tools in making judgements. The scope and diversity of trust may be coupled with other trust properties such as trust direction and trust symmetry.

7 Life Cycle of Trust Relationships

We are currently working on a methodology for life cycle of trust relationships using the definition of trust relationship, operations of trust relationships and the properties of trust relationship. Trust relationships between possible entities play crucial roles in the collaborative interactions in distributed environments. The analysis and design of trust relationship must be integrated with other requirements of the whole distributed information system. The modelling, implementing and maintaining of trust relationships is an incremental, iterative process. The whole life cycle of trust relationships includes several stages such as extracting trust requirements in system, identifying possible trust relationships from trust requirements, choosing and refining the whole set of trust relationships from possible trust relationships, implementing trust relationships in systems and maintaining trust relationships in systems. The initial trust relationships will be refined in multiple life cycles. There are two ways to accommodate new business requirements. One way is to introduce new trust relationships and another way is to modify existing trust relationships. When new trust relationships are introduced, several things need to be considered such as the scope and diversity of these trust relationships, the properties of direction and symmetry of these trust relationships and the relations between them and existing trust relationships. In section 3, 4,5 and 6, we have proposed a set of operations and definitions to enable the analysis of the above. We have also given some example scenarios.

We believe that they are helpful in the analysis and design of trust relationships for collaborative interactions in the distributed environments and they are part of our overall methodology of life cycle of trust relationships. When trust relationships are modified, the change management of trust relationships must be considered. We are in the process of developing change management schemes for trust relationships, which will become part of our overall methodology as well.

8 Concluding Remarks

In this paper, we have focused on the analysis and modelling of trust in distributed information systems. We have reviewed the definition of trust relationship and operations and definitions about relations of trust relationships. We have discussed the classification of trust under our taxonomy framework. We have discussed different forms of trust and put authentication at layer one of trust and other trust types on layer two. Authentication plays a foundation role for other trust types on layer two. We have proposed a hierarchy of layer two trust relationships based on the nature of four tuples of a trust relationship. This hierarchy is helpful to understand the purposes of trust relationships in the real world. We provide multiple definitions about the properties of trust direction and trust symmetry. We have defined trust scope label to model the properties of scope and diversity of trust. All the definitions proposed in this paper are new elements of our taxonomy framework and they can be used as enabling tools in the analysis and modelling of trust in distributed information systems.

In real implementations, properties of trust discussed in this paper will be customized and configured based on the specific requirements. We are currently working on an overall methodology of life cycle of trust relationships in distributed information systems. This research focuses on the properties of trust relationships and taxonomy framework. The definition of trust relationship provides a starting point and it is the cornerstone of this research. The relations of relationships can provide useful tools for enabling the analysis, design and implementation of trust in distributed information systems. The classification of trust are helpful for better understanding of trust and is helpful in the analysis of trust. The definitions about trust direction, trust symmetry and trust scope can provide suitable terms for the related properties and they can be used as tools for enabling the analysis and modelling of trust in distributed information systems. In the web services paradigm, we hope that the proposed properties of trust and tools for analysis and modelling trust can provide solid foundation for trust related issues in WS-Trust, WS-Security, WS-Policy and WS-Federation.

References

1. TCSEC. Trusted computer system evaluation criteria. Technical report, U.S.A National Computer Security Council, 1985. DOD standard 5200.28-STD.
2. J. Landauer, T. Redmond, and T. Benzel. Formal policies for trusted processes. In *Proceedings of the Computer Security Foundations Workshop II, 1989*, pages 31–40. 1989.

3. S. Marsh. *Formalising trust as a computational concept.* Phd thesis, University of Sterling, 1994.
4. Y. Wang and J. Vassileva. Trust and reputation model in peer-to-peer networks. In *Proceedings of Third International Conference on Peer-to-Peer Computing*, pages 150–157, 2003.
5. L. Xiong and L. Liu. A reputation-based trust model for peer-to-peer e commerce communities. In *IEEE International Conference on E-Commerce*, pages 275–284, 2003.
6. M. N. Huhns and D. A. Buell. Trusted autonomy. *Internet Computing, IEEE*, 6(3):92–95, 2002.
7. W. H. Winsborough, K. E. Seamons, and et al. Automated trust negotiation. In *Proceedings of DARPA Information Survivability Conference and Exposition*, 2000.
8. M. Winslett, T. Yu, and et al. Negotiating trust in the web. *IEEE Internet Computing*, 6(6):30–37, 2002.
9. M. Blaze, J. Feigenbaum, and J. Lacy. Decentralized trust management. In *Proceedings of IEEE Symposium on Security and Privacy*, pages 164–173. 1996.
10. M. Blaze, J. Feigenbaum, and A.D. Keromytis. KeyNote: Trust management for public-key infrastructures (position paper). *Lecture Notes in Computer Science*, 1550:59–63, 1999.
11. Y. H. Chu, J. Feigenbaum, B. LaMacchia, P. Resnick, and M. Strauss. REFEREE: Trust management for Web applications. *Computer Networks and ISDN Systems*, 29(8–13):953–964, 1997.
12. W. Zhao, V. Varadharajan, and G. Bryan. Modelling trust relationships in distributed environments. In *Lecture Notes in Computer Science*, volume 3184, pages 40–49. Springer-Verlag, 2004.
13. T. Grandison and M. Sloman. A survey of trust in internet application. *IEEE Communications Surveys*, pages 2–16, Fourth Quarter, 2000.
14. A. Jøsang. The right type of trust for distributed systems. In *Proceedings of the 1996 New Security Paradigms Workshop*, pages 119–131. ACM, 1996.