FORMALIZING END-TO-END CONTEXT-AWARE TRUST RELATIONSHIPS IN COLLABORATIVE ACTIVITIES

Ioanna Dionysiou
Department of Computer Science, School of Sciences, University of Nicosia, Nicosia, Cyprus

Dave Bakken, Carl Hauser
Department of Computer Science, School of Electrical Engineering and Computer Science
Washington State University, Pullman, WA, U.S.A.

Deborah Frincke
CyberSecurity Group, Pacific Northwest National Laboratory, Richland, WA, U.S.A.

Keywords: Trust relation, dynamic trust, composable trust, activity-oriented trust.

Abstract: The diversity of the kinds of interactions between principals in distributed computing systems, including critical infrastructures, has expanded rapidly in recent years. However, such applications and their users are vulnerable with respect to both the diversity of the principals providing these services or data and the interactions between them. This paper introduces formalisms for a new trust model that addresses these limitations. The novelty of the new model is its ability to specify and reason about trust dynamically and when composed beyond pairwise relationships for a specific interaction. An intuitive and practical way is presented to manage end-to-end trust assessment for a particular activity, where multiple trust relationships are examined in order to derive the overall trust for the activity.

1 INTRODUCTION

Distributed computing has evolved greatly in the last decade or so. Interactions between entities used to be mainly client-server or database oriented. Recent years have seen a great increase in the kind and number of entities interacting, the patterns in which they interact, and the kinds of distributed services supporting these interactions. This ubiquitous use of distributed applications provides increased convenience, safety, and enjoyment for society. However, such applications and their users are vulnerable with respect to both the diversity of the principals providing these services or data and the interactions between them.

Consider the North American electric power grid, for instance, with nearly 3500 utility organizations (Force, 2004). These individually owned utility systems have been connected together to form interconnected power grids, which must be operated in a coordinated manner. There are many points of interactions among a variety of participants and a local change can have immediate impact everywhere. In order to detect disturbances that could escalate into cascading outages and take corrective actions, real-time information about the grid dynamics must be obtained to enhance the wide-area system observability, efficiency, and reliability. Power utilities are reluctant to disclose information in order to protect themselves financially and legally. Sharing of data might jeopardize their business due to their inability to quantify the risk regarding interactions with other grid participants. For example, unrestricted access to a utility's data that are market-sensitive indicators could give a competitor an unfair advantage in adjusting its own contracts and prices. Similarly, a utility could distribute inaccurate data to mislead the other market participants.

The "no sharing" policy could be relaxed under normal operating conditions if the risk of sharing were systematically contained (Dionysiou et al., 2007). In order to do that, private, public and national entities must collaborate in a way that sensitive information is shared without compromising it. This collaborative environment is nontrivial to establish and operate because its participants do not have the nec-

Dionysiou I., Bakken D., Hauser C. and Frincke D. (2008).
FORMALIZING END-TO-END CONTEXT-AWARE TRUST RELATIONSHIPS IN COLLABORATIVE ACTIVITIES.
In *Proceedings of the International Conference on Security and Cryptography*, pages 546-553
DOI: 10.5220/0001926905460553
Copyright © SciTePress

essary knowledge and tools to assess the quality of the received data and the risk of compromising that data. Trust management is a service that, when used properly, has the potential to enable the entities that operate within critical infrastructures to share confidential, proprietary, and business sensitive information with reliable partners.

In this paper, we introduce a conceptual trust model that allows entities to reason about the following issues: how to specify and adapt the degree of trust that they place in an entity (*dynamic trust*) and how much trust to place in data they receive that comes through nontrivial chains of processing or services without using traditional transitivity (*composable trust*). To be more specific, the paper presents the following contributions:

- A notation for specifying trust relationships that are tied not only to a narrow context but to a broader activity
- An intuitive and practical way to manage end-to-end trust assessment for a particular activity, where multiple trust relationships are examined in order to derive trust for the activity, including explicit consideration of expectations and their violations in analyzing end-to-end trust

The remainder of the paper is organized as follows. Section 2 discusses trust among collaborators for a specific activity. Section 3 presents our conceptual trust model, with emphasis on the trust ontology involved in deriving end-to-end trust assessments. Related work is discussed in Section 4 and Section 5 concludes.

2 ACTIVITY-ORIENTED TRUST RELATIONSHIPS

Creating a universally acceptable set of rules and mechanisms to specify trust is a difficult process because of the variety in trust interpretations. Researchers have defined trust concepts for many perspectives, with the result that trust definitions overlap or contradict each other (Presti et al., 2003). Nevertheless, trust is an abstraction of individual beliefs that an entity has for specific situations and interactions. An entity's beliefs are not static but they change as time progresses and new information is processed into knowledge. Trust must evolve in a consistent manner so that it still abstracts the entity's beliefs accurately. In this way, an entity continuously makes informed decisions based on its current beliefs.

In a typical trust setting, there is a *trustor* and a *trustee*. A trustor is the entity that makes a trust

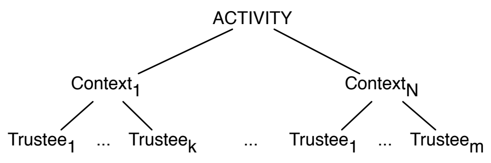

Figure 1: Context-Aware Activity.

assessment for an entity, which is the trustee. The scope of the trust relationship between a trustor and a trustee is narrowed to a specific action called a *context*. A trust relationship may be one-to-many to cover a group of trustees, which are trusted similarly within the same context (Grandison and Sloman, 2000). Such approach cannot encompass the complexity of trust in an *activity* that involves different contexts and trustees. An *activity* is an interaction that involves multiple trustees that may assume different roles (Figure 1); in other words, the successful outcome of an activity requires the collaboration of trustees performing specific functions, which are not necessarily the same.

Consider Figure 2. In this setting, there is a data stream d between entity P and entity C. It could appear that entity C may assess the risk of using data d by making a trust assessment regarding entity P's ability to produce reliable data d. However, this is not sufficient. The presence of intermediate entities S_1 and S_2 that forward this data to C affect the quality of received data d. As a result, trustor C has to make trust assessments for all interacting trustees that collaboratively execute a task and combine them in order to derive an end-to-end trust assessment about the quality of the data stream. Thus, if C were to make a trust assessment concerning the *activity of the information flow between P and itself*, then the following trust relationships had to be examined:

- relationship $\tau(C,P,...)$[1] between P and C regarding P's ability to *produce* data d
- relationship $\tau(C,S_1,...)$ between S_1 and C regarding S_1's ability to *forward* data d
- relationship $\tau(C,S_2,...)$ between S_2 and C regarding S_2's ability to *forward* data d

3 TRUST MODEL ONTOLOGY

The theory of sets and relations is used to represent trust between trustors and trustees (Dionysiou, 2006). This section formally defines *trust between trustors*

[1]For simplicity reasons, we omit for now the remaining trust attributes

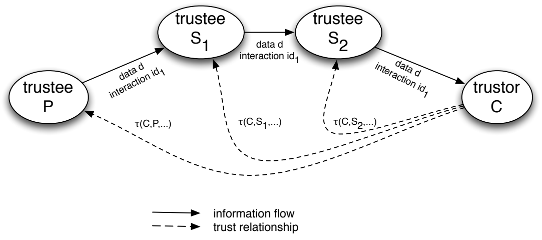

Figure 2: Trust Relationships in an Activity.

and trustees as a relation τ and examines the relation's attributes, properties, and operations. The notation $\tau(\gamma,\delta,c,\lambda,\iota,\varepsilon,id,s)$ represents a trust relationship between two entities and it is interpreted as "*trustor* γ*, based on* γ*'s trusting attitude, believes that the extent to which trustee* δ *will act as expected for context c during time interval* ι *is* λ*, and this belief is subject to the satisfaction of expectation set* ε*. This relationship is valid for a specific interaction id and its status is indicated by s.*"

3.1 Trust Relation Attributes

The attributes of the trust relation τ are trustor γ, trustee δ, context c, levels λ, time interval ι, expectation set ε, interaction identifier id, and status s. These are explained below.

3.1.1 Trustors and Trustees

The first two attributes γ,δ of the trust relation τ represent the trustor and trustee respectively. These are entities with unique identifiers.

3.1.2 Context

The scope of the trust relationship is narrowed to a specific function called context. Without loss of generality, context c is portrayed as an action performed on data. In this case, action could be *producing*, *forwarding*, and *consuming* data. Data is classified based on its type, which includes *sensitive* data, *public* data, *recommendation* data, etc. Given that A is the set of actions performed on data and DT the set of data types, then context $c=(A,D)$ with $D\subset DT$.

3.1.3 Trust Levels

Trust is subjective because a trustor's requirements are not met by trustees at the same degree or a trustor's expectations for trustees varies. There is a number of ways to denote how much an entity is worthy of trust. Trust values (Abdul-Rahman and Hailes, 1997), degrees (Abdul-Rahman and Hailes, 2000) and levels (Grandison and Sloman, 2000) are all used by the trustor to categorize trustees based on their perceived trustworthiness.

Our model adopts the term *trust level*. Trust levels are closely related to two important concepts: trustfulness of the trustor and trustworthiness of the trustee (Buskens, 2002). Trustfulness is defined as the extent to which the trustor is willing to take the risk of trust being abused by the trustee. On the other hand, trustworthiness is the extent to which the trustee honors trust, if trust is placed. The trustfulness of the trustor varies and it depends on the trustor's willingness to trust. In our model, the trustfulness of the trustor is a synonym of trusting attitude.

Trust levels λ is an ordered pair (λ_{im},λ_t) with $\lambda_{im}\in\Lambda_{im}$, $\lambda_t\in\Lambda_t$, and its coordinates represent trusting attitude and trustworthiness extent respectively. Λ_{im} is the set of values for trusting attitude whereas Λ_t is the set of values for trustworthiness extent.

3.1.4 Time Interval

A trust relation consists of trust relationships that are valid for a period of time. The temporal database interpretation of time is chosen for modeling time. In temporal databases, time domain T is considered to be an ordered sequence of points t_i in some application-dependent granularity (Elmasri and Navathe, 2000). The granularity of time does not affect the logic of the trust relation if it is consistent throughout the model.

There are two types of intervals: *expected* interval and *actual* interval. The expected interval $\iota=(t_s,t_e)$ indicates the anchored time duration in which a trust relationship is predicted to be valid. The actual interval is the one that the trust relationship was observed to be valid in the real world. The latter is not part of the relation, but it's recorded as experience upon termination of trust relationship.

3.1.5 Expectations

An expectation is defined as a requirement and its allowed values that a trustor has for a particular interaction with a trustee. An expectation is a tuple (π,o,ν_o,ν_a,ev) , where π is a trust requirement, o is a standard relational operator, ν_o is the observed or actual value for the requirement, ν_a is the allowed value for that requirement and ev represents the evaluation criteria for the specific requirement. The observed value ν_o is the aggregated value of multiple observations (also called evidence) over time. ev contains the aggregation algorithm that is used to derive the actual value.

The first expectation attribute is the trust requirement π. Trust requirements are grouped in behavioral

Table 1: Aggregation Algorithms for Single Evidence Type.

Average	$v^j_{all} = \frac{1}{n}\sum_{i=1}^{n}(v^j_i)$, where n is the number of external sources
Weighted Average	$v^j_{all} = \frac{1}{n}\sum_{i=1}^{n} f(\tau_i, v^j_i)$, where f is a function
Weighted Average Combination	$v^j_{all} = \frac{u}{x}\sum_{i=1}^{x} f(\tau_i, v^j_i) + \frac{v}{z}\sum_{i=1}^{z} f(\tau_i, v^j_i)$, where $x+z=n$, $u+v=1$
Majority	v^j_{all} is the value that more than $\frac{n}{2}$ external sources agree on (allow for a δ deviation)
m-out-of-n	v^j_{all} is the value that at least m out of n external sources agree on
Plurality	v^j_{all} is the value that most external sources agree on
User-defined	Any customized aggregation method such as median value, etc

(β), security (σ), and QoS (ϕ) categories. Trust requirement $\pi \in \Pi$, with $\Pi = \Pi_\beta \cup \Pi_\sigma \cup \Pi_\phi$.

The second expectation attribute is the equality or inequality operator o. Operator o takes values from the set O that consists of the relational operators $=$, $<$, $\leqslant$, $>$, $\geqslant$, and $\neq$.

The third and fourth attributes are the observed and allowed values respectively. Value $\nu \in V$, with set V consisting of values that are assigned to requirements. Set V is defined as $\cup V_i$, with V_i be the set of values that requirement π_i takes. Without loss of generality, assume that every V_i is totally ordered by $\leqslant$ and all operators $o \in O$ are defined for all $\nu \in V$.

Finally, the fifth attribute ev represents the evaluation parameters for the particular requirement. At time t, a requirement can only have one actual value, which is the result of aggregating many observed values. An element ev is a tuple $(covering, triggering, aggregation)$ that describes the covering method, triggering rule and aggregation scheme for π. The covering attribute provides the conditions under which an expectation is considered valid and the remaining two characterize the "when" and "how" the observed value is updated.

Incoming evidence must be evaluated in the context of existing trust relationships. When triggering conditions are activated, the evidence is aggregated. Aggregating evidence can be seen as a voting mechanism where instances of evidence types are combined by a voting scheme into a single output. Aggregation algorithms can be triggered either at predefined intervals or when a number of instances arrive at the evaluator. The aggregated result is an observed value for a trust requirement. It is important to note that an observed value is not necessarily related directly to a single evidence type. A user could specify functions that map multiple evidence types to trust requirement values.

There is a spectrum of aggregation algorithms that could target the aggregation of evidence from multiple instances of a single evidence type and evidence from different evidence types. The aggregation algorithms that could be supported include *average*, *weighted average*, *majority*, *m-out-of-n*, *plurality*, and any user-defined aggregation. Consider v^j_{all} to be the aggregated observed value for entity j, v^j_i to be evidence value by external source i for entity j, and τ_i to be the trust relationship between the evaluating trustor k and external entity i. The aggregation algorithms that trustor k can use to derive a single observed value for j are shown in Table 1.

The covering techniques mentioned in this paper are *strict* and *relaxed*, where in the former case the observed value satisfies the allowed value under operation o and in the latter case a deviation d is allowed. The expectation semantics dictate that an expectation is valid if and only if the relationship between observed value ν_o and allowed value ν_a under operation o and covering method *covering* is also valid. Otherwise, a *violation* occurs. Table 2 illustrates examples of valid expectations and violations.

Table 2: Valid Expectations and Violations.

Op. o	ν_o	ν_a	Covering	Valid or Violation
$=$	3	5	strict	violation
$=$	5	3	d=2	valid
$\leqslant$	6	5	strict	violation
$\leqslant$	6	5	d=4	valid

An expectation set ε describes the requirements that a trustor has for a trustee. Expectation set ε is the subset of the cartesian product $\Pi \times O \times V \times V \times EV$. Each member of the expectation set is an expectation as defined above. There is a unique tuple $(\pi, o, \nu_o, \nu_a, ev) \in \varepsilon$ that corresponds to any given π.

By itself an expectation set is not interesting unless operations are performed on its elements. However, prior to defining these operations we must first define the primitive comparison relationships between its elements. The relationships between expectation tuples determine the relationships between expectation sets. These binary relationships include the standard *equality* ($=$) and *less than or equal* ($\leqslant$) relationships as well as the redefinition of *not equal* $\neq$. In addition, a new relationship called *relaxed equal* is defined. Based on the binary relationships, expectation sets when compared fall in one of the four categories: strictly-equal, relaxed-equal, covered/covering and unrelated (Dionysiou, 2006).

Starting with the relationships between expectation tuples, the equality relationship is presented first. Two expectation tuples are equal if their respective trust properties π_1, π_2, observed values ν_{o_1}, ν_{o_2}, allowed values ν_{a_1}, ν_{a_2}, and covering methods $covering_1, covering_2$ are the same. The triggering and aggregation methods do not need to be the same since they don't affect the semantics of the expectation tuples. Those attributes merely affect the when and how the observed value changes.

Equal Expectations (=). Expectation $(\pi_1, o_1, \nu_{o_1}, \nu_{a_1}, ev_1)$ is equal with expectation $(\pi_2, o_2, \nu_{o_2}, \nu_{a_2}, ev_2)$ if and only if $\pi_1 = \pi_2 \wedge o_1 = o_2 \wedge \nu_{o_1} = \nu_{o_2} \wedge \nu_{a_1} = \nu_{a_2} \wedge covering_1 \in ev_1 = covering_2 \in ev_2$.

The *relaxed equal* relationship is a new relationship that relates two expectations that refer to the same property mapped to different values.

Relaxed Equal Expectations (≈). An expectation $(\pi_1, o_1, \nu_{o_1}, \nu_{a_1}, ev_1)$ is relaxed equal with another expectation $(\pi_2, o_2, \nu_{o_2}, \nu_{a_2}, ev_2)$ if and only if $(\pi_1 = \pi_2 \wedge o_1 = o_2 \wedge \nu_{o_1} \neq \nu_{o_2} \wedge \nu_{a_1} \neq \nu_{a_2} \wedge covering_1 = covering_2)$ or $(\pi_1 = \pi_2 \wedge o_1 = o_2 \wedge \nu_{o_1} \neq \nu_{o_2} \wedge \nu_{a_1} = \nu_{a_2} \wedge covering_1 = covering_2)$.

Two expectation sets are strictly equal if they contain the same elements. Expectation sets $\varepsilon_1 = \{(cooperation, =, 1, 1, ev), (reliability, >, 0.98, 0.97, ev2)\}$ and $\varepsilon_2 = \{(cooperation, =, 1, 1, ev), (reliability, >, 0.98, 0.97, ev2)\}$ are strictly equal.

Strictly Equal Expectation Sets. Expectation set ε_1 is strictly equal to expectation set ε_2 if and only if ε_1 is an improper set of ε_2, under the equality definition.

The relaxed equal comparison is a generalization of the strictly equal comparison between two expectation sets. Consider expectation sets $\varepsilon_1 = \{(cooperation, >, 3, 1, ev), (reliability, >, 0.98, 0.97, ev2)\}$ and $\varepsilon_2 = \{(cooperation, >, 2, 1, ev), (reliability, >, 0.98, 0.97, ev2)\}$. These two sets don't have the same elements, but they both contain values for the same properties: cooperation and reliability. These two sets are called relaxed equal.

Relaxed Equal Expectation Sets. Expectation set ε_1 is relaxed-equal to expectation set ε_2 if and only if for all tuples i=$(\pi_1, o_1, \nu_{o_1}, \nu_{a_1}, ev_1) \in \varepsilon_1$ there is tuple j = $(\pi_2, o_2, \nu_{o_2}, \nu_{a_2}, ev_2) \in \varepsilon_2$ such as $|\varepsilon_1| = |\varepsilon_2|$ and $(i \approx j$ or $i = j)$.

In order to address the issue of trust composability, one must provide answer to queries like "What is the expectation set for a path that starts from X and terminates at Y?" . Merging is an operation that accomplishes that by applying a function f_π on the values of a property. The function f_π essentially aggregates the observed and allowed values into single values respectively; average, maximum, minimum, weighted average are all candidate f_π.

Merging of Expectation Sets. Consider expectation sets ε_1 and ε_2. The merging of the two expectation sets results in a new expectation set ε_{merge} that is constructed as follows:

1. Initialize $\varepsilon_{merge} = \emptyset$
2. If $\varepsilon_1 = \varepsilon_2$, then $\varepsilon_{merge} \longleftarrow \varepsilon_{merge} \cup \varepsilon_1$.
3. if $\varepsilon_1 \approx \varepsilon_2$, then $\forall$i:$(\pi_1, o_1, \nu_{o_1}, \nu_{a_1}, ev_1) \in \varepsilon_1$, j:$(\pi_2, o_2, \nu_{o_2}, \nu_{a_2}, ev_2) \in \varepsilon_2$ such that $i \approx j$ and $\varepsilon_{merge} \longleftarrow \varepsilon_{merge} \cup \{((\pi_1, o_1, f_\pi(\nu_{o1}, \nu_{o2}), f_\pi(\nu_{a1}, \nu_{a2}), ev_1))\}$.

3.1.6 Interaction id

Another trust relation attribute is the interaction identifier *id*. There is a unique identifier for each activity, the interaction id. There are at least two trust relationships for any activity.

3.1.7 Status

The last attribute is the status of a trust relationship. Status $s \in S = \{$OK, WARNING, ALERT$\}$.

3.2 Trust Relation Properties and Operations

The next step in formalizing trust is the definition of its properties. The standard properties of any n-ary relation (reflexive, irreflexive, symmetric, antisymmetric, transitive, and equivalence) do not hold due to non-absolute characteristics of trust relationships. Thus, new properties must be investigated.

One of the characteristics of trust relation τ is its dynamic nature, meaning that $\tau(\gamma, \delta, c, \lambda, \iota, \varepsilon, id, s)$ which is valid in time t_1 may become invalid in t_2, with $t_2 > t_1$, and vice versa. Another characteristic of trust is its composable nature, meaning that existing trust relationships can be aggregated to derive an end-to-end trust assessment for a particular activity at time t. Thus, operations are categorized in two groups: the ones that affect and change the current state of the trust relation and the ones that use the existing state of the trust relation to make trust assessments.

3.2.1 Operations Changing Trust Relation State

Trust relation τ is affected by time, arrival of new evidence, and violation of expectations, to just name a few. There are other events that change the trust relation state such as the change of trusting attitude level, but these are not described here.

Expiration of Valid Time. A trust relationship $(\gamma,\delta,c,\lambda,\iota,\varepsilon,id,s)$ does not hold in relation τ if its valid interval time expires. Thus, a trust relationship $\tau(\gamma,\delta,c,\lambda,\iota,\varepsilon,id,s)$ is not valid in τ if the current time $t > t_e$, $t_e \in \iota$.

Arrival of New Evidence. Suppose that new evidence is available for a particular trustee and context. The new value will be applied to the appropriate trust requirement according to the trustee evaluation information for the trust requirement. Suppose that new evidence arrives at trustor γ for trustee δ regarding context c. The new evidence includes the trust requirement π_r and the recommended value ν_r. All trust relationships $(\gamma,\delta,c,\lambda_i,\iota_i,\varepsilon_i,id_i,s_i)$ are updated to reflect the application of the new evidence on ν.

Expectation Violation. Whenever new evidence arrives, the observed value changes according to the aggregation scheme for the specific requirement. An update in the observed value may lead into expectation violation. In this case, the respective trust relationship's status is set to ALERT. The relationship does not necessarily become false in τ; according to policies it might be the case that a trustor wants to monitor the relationship before terminating it. However, all other trust relationships that are associated with the alerted relationship's interaction identifier have their status set to WARNING. In a case that an expectation $(\pi,o,\nu_o,\nu_a,ev) \in \varepsilon$ is not valid for $\tau(\gamma,\delta,c,\lambda,\iota,\varepsilon,id,s)$, the respective relationship's status s becomes ALERT and all tuples associated with the same interaction identifier id have their status set to WARNING. However, a relationship's status is restored whenever the violation gets corrected during the monitoring interval. In this case, the ALERT gets replaced by OK, and all WARNING are replaced with OK. Note that if there are multiple violations for a specific interaction, then WARNING(s) remain as they are until all ALERT(s) become OK.

3.2.2 Operations using Trust Relation State

The current state of the trust relation can be used to make nontrivial trust assessments for activity-specific relationships with the same context and with different contexts; the later gives the end-to-end trust assessment for the particular activity.

Let's consider first the *aggregate trust assessment for context c in interaction id.* In particular, consider tuples $(\gamma_1,\delta_1,c_1,\lambda_1,\iota_1,\varepsilon_1,id_1,s_1)$ and $(\gamma_1,\delta_2,c_1,\lambda_1,\iota_2,\varepsilon_2,id_1,s_1)$ in τ. Trustor γ_1 may synthesize the two tuples to derive an aggregated trust assessment for context c during interval ι_i (the intersection of ι_1 and ι_2) by applying expectation set operations on the expectation sets ε_1 and ε_2 to derive the aggregated expectation set ε_i. Expectation set ε_i has to be checked against the various trust level specifications in order to assign the trustworthiness level λ_i for the new tuple $(\gamma_1,\delta_{1,2},c_i,\lambda_i,\iota_i,\varepsilon_i,id_1,s_1)$.

Next, consider the *end-to-end trust assessment for interaction id.* Suppose there are aggregated trust assessments for contexts c_1 and c_2, which are the only contexts belonging to interaction id_1: these are tuples $(\gamma_1,\delta_1,c_1,\lambda_1,\iota_1,\varepsilon_1,id_1,s_1)$ and $(\gamma_1,\delta_2,c_2,\lambda_1,\iota_2,\varepsilon_2,id_1,s_1)$. Trustor γ_1 may compose the two tuples to derive an end-to-end trust assessment for interaction id_1 during interval ι_i (the intersection of ι_1 and ι_2) by applying expectation set operations on the expectation sets ε_1 and ε_2 to derive the aggregated expectation set ε_i. Expectation set ε_i has to be checked against the various level specifications in order to assign the trustworthiness level λ_i for the new tuple $(\gamma_1,\delta_{1,2},c_{1,2},\lambda_i,\iota_i,\varepsilon_i,id_1,s_1)$.

We will demonstrate the two operations above with an illustrative example. Assume that Figure 3 is a graph for trust relation τ of the network depicted in Figure 2. A node represents either a trustor or a trustee. Nodes are connected by directed edges that start at a trustor node and end at a trustee node. Each edge carries the tuple that describes the relationships between the two connecting nodes. An edge may carry multiple tuples. Suppose that trustor C would like to derive the end-to-end trust assessment regarding the information flow for data d. Then, as explained in section 2, the following trust relationships must be considered:

- relationship between P and C regarding P's ability to *produce* data d: $\tau(\gamma_C,\delta_P,c_2,\lambda_1,\iota_3,\varepsilon_3,id_1,s_{OK})$
- relationship between S_1 and C regarding S_1's ability to *forward* data d: $\tau(\gamma_C,\delta_{S1},c_1,\lambda_1,\iota_1,\varepsilon_1,id_1,s_{OK})$
- relationship between S_2 and C regarding S_2's ability to *forward* data d: $\tau(\gamma_C,\delta_{S2},c_1,\lambda_1,\iota_2,\varepsilon_2,id_1,s_{OK})$

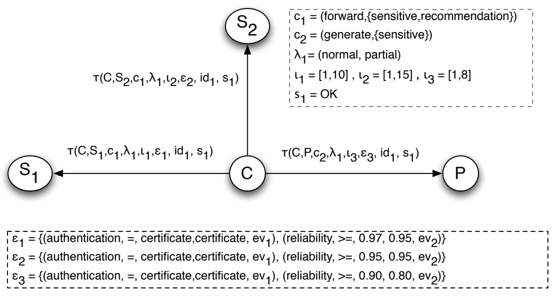
S_2
c_1 = (forward,{sensitive,recommendation})
c_2 = (generate,{sensitive})
λ_1= (normal, partial)
ι_1 = [1,10] , ι_2 = [1,15] , ι_3 = [1,8]
s_1 = OK
$\tau(C,S_2,c_1,\lambda_1,\iota_2,\varepsilon_2, id_1, s_1)$
$\tau(C,S_1,c_1,\lambda_1,\iota_1,\varepsilon_1, id_1, s_1)$
$\tau(C,P,c_2,\lambda_1,\iota_3,\varepsilon_3, id_1, s_1)$
S_1
C
P
ε_1 = {(authentication, =, certificate,certificate, ev_1), (reliability, >=, 0.97, 0.95, ev_2)}
ε_2 = {(authentication, =, certificate,certificate, ev_1), (reliability, >=, 0.95, 0.95, ev_2)}
ε_3 = {(authentication, =, certificate,certificate, ev_1), (reliability, >=, 0.90, 0.80, ev_2)}

Figure 3: Trust Relation Graph.

First, the trust aggregation for the same context c_1 will take place for trustees $S1$ and $S2$. The result will be $\tau(\gamma_C,\delta_{S1,S2},c_1,\lambda_1,\iota_k,\varepsilon_k,id_1,s_{OK})$, with i_k=(1,10) and ε_k={(authentication, =, certificate,certificate, ev1), (reliability, $\geqslant$, average(0.95,0.97), average(0.95,0.95), ev2)}. Then, the end-to-end trust assessment will take place between the new trust relationship $\tau(\gamma_C,\delta_{S1,S2},c_1,\lambda_1,\iota_k,\varepsilon_k,id_1,s_{OK})$ and $\tau(\gamma_C,\delta_P,c_2,\lambda_1,\iota_3,\varepsilon_3,id_1,s_{OK})$. In this case, $\tau(\gamma_C,\delta_{P,S1,S2},c_{1,2},\lambda_1,\iota_m,\varepsilon_m,id_1,s_{OK})$, with i_m=(1,8) and ε_m={(authentication, =, certificate,certificate, ev1), (reliability, $\geqslant$, average(0.90,0.96), average(0.80,0.95), ev2)}.

4 RELATED WORK

In recent years, researchers have investigated various definitions of *trust*, modeling trust and its management (Vacca, 2004; Winslett et al., 2002; Herzberg et al., 2000; Group, 2004; Grandison, 2001; Blaze et al., 1996; Chu et al., 1997; Blaze et al., 1998; Zimmermann, 1995; Marsh, 1994; Josang, 1997; Josang et al., 2006; Abdul-Rahman and Hailes, 2000). Starting with one of the early and widely known trust models, the PGP trust model (Zimmermann, 1995) creates an informal web of trust which is used for authentication purposes. A recent survey of contemporary trust management systems is compiled by Grandison and Sloman (Grandison and Sloman, 2000), who point out the limitations of those solutions as they mostly address access control issues rather than the more general analysis of trust. KeyNote (Blaze et al., 1998) and PolicyMaker (Blaze et al., 1996), for instance, are primarily concerned with security issues (authentication and access control). A solution to more general trust relationships is proposed by SULTAN (Grandison and Sloman, 2000), a trust management model that uses a logic-oriented language to specify trust. In addition to the practical approaches to trust, there are formal trust models that describe more general trust factors. The Marsh (Marsh, 1994) logic-based framework uses formal representation to capture the semantics of the social paradigms of trust whereas Josang's subjective logic (Josang, 1997; Josang et al., 2006) is another formal model that uses beliefs as the basis for trust.

Unlike other approaches, our model derives end-to-end trust assessments without using transitive indirect trust explicitly in the derivations. The semantics of indirect trust are captured in the form of fine-grained recommendations that are considered to be an evidence type; the weight of this evidence on the overall trust assessment, like any other evidence type, is based on the trustor-recommender existing trust relationship. In this case, the trust relationship context could be "recommending" whereas the activity identifier could be the interaction identifier of the particular recommendation information flow. In addition, our model extends the traditional concept of trust conditions into more expressive expectations, which include not only expected values for particular properties but also covering, aggregating, and triggering mechanisms that manipulate the observed value.

5 CONCLUSIONS

This paper presents a new trust paradigm and associated formalisms, devised to support dynamic and composable trust suitable for collaborative activities. Dynamic and composable trust is essential for topologies where interactions are dynamic and they almost always involve the collaboration of multiple entities to disseminate data from its source to its destination. In this setting, dynamic trust enables the specification and management of trust relationships to change over the operational lifecycle of the activity as relevant conditions that affect trust change. Composable trust allows end-to-end trust assessment for the entire activity, where multiple trust relationships are examined in order to derive trust for the activity. We also presented an intuitive and practical way to manage end-to-end trust assessment for a particular activity, including explicit consideration of expectations and their violations.

REFERENCES

Abdul-Rahman, A. and Hailes, S. (1997). A distributed trust model. In *Proceedings of the ACM New Security Paradigms Workshop*, pages 48–60.

Abdul-Rahman, A. and Hailes, S. (2000). Supporting trust in virtual communities. In *Proceedings of the 33th Hawaii International Conference on System Sciences (HICSS)*, pages 1769–1777, Maui, Hawaii.

Blaze, M., Feigenbaum, J., and Keromytis, A. D. (1998). Keynote: Trust management for public key infrastructures. In *Proceedings of the 6th International Workshop on Security Protocols*, Cambridge, UK.

Blaze, M., Feigenbaum, J., and Lacy, J. (1996). Decentralized trust management. In *SP '96: Proceedings of the 1996 IEEE Symposium on Security and Privacy*, page 164, Washington, DC, USA. IEEE Computer Society.

Buskens, V. W. (2002). *Social Networks and Trust*, volume 30 of *Theory and Decision Library. Series C, Game Theory, Mathematical Programming, and Operations Research*. Boston, London Kluwer Academic Publishers.

Chu, Y.-H., Feigenbaum, J., LaMacchia, B., Resnick, P., and Strauss, M. (1997). Referee: trust management for web applications. *Comput. Netw. ISDN Syst.*, 29(8-13):953–964.

DHS (2006). Protected critical infrastructure information (pcii) program. www.dhs.gov.

Dionysiou, I. (2006). *Dynamic and Composable Trust for Indirect Interactions, Ph.D. Dissertation*. Department of Electrical Engineering and Computer Science, Washington State University.

Dionysiou, I., Frincke, D., Bakken, D., and Hauser, C. (2007). An approach to trust management challenges for critical infrastructures. In *Proceedings of the 2nd International Workshop on Critical Information Infrastructures Security (CRITIS07), to appear in Lecture Notes in Computer Science Series, Springer Berlin, 2007*, Malaga, Spain.

Elmasri, R. and Navathe, S. (2000). *Fundamentals of Database Systems*. Addison-Wesley Longman, Inc.

Force, U. C. P. S. O. T. (2004). *Final report on the August 14, 2003 Blackout in the United States and Canada: Causes and Recommendations*. https://reports.energy.gov/BlackoutFinal-Web.pdf.

Grandison, T. (2001). Trust specification and analysis for internet applications. Technical report, Ph.D. Thesis, Imperial College of Science Technology and Medicine, Department of Computing, London.

Grandison, T. and Sloman, M. (2000). A survey of trust in internet applications. *IEEE Communications Surveys and Tutorials*, 3(4):2–16.

Group, T. C. (2004). *TCG Specification Architecture Overview*. TCG.

Herzberg, A., Mass, Y., Michaeli, J., Ravid, Y., and Naor, D. (2000). Access control meets public key infrastructure, or: Assigning roles to strangers. In *SP '00: Proceedings of the 2000 IEEE Symposium on Security and Privacy*, page 2, Washington, DC, USA. IEEE Computer Society.

Josang, A. (1997). Prospectives of modeling trust in information security. In *Proceedings of the 2nd Australasian Conference on Information Security and Privacy*, Sydney, Australia.

Josang, A., Gray, E., and Kinateder, M. (2006). Simplification and analysis of transitive trust networks. *Web Intelligence and Agent Systems*, 4(2):139 – 161.

Marsh, S. (1994). *Formalizing Trust as a Computational Concept*. Department of Computer Science, University of Sterling.

Presti, S. L., Cusack, M., Booth, C., Allsopp, D., Kirton, M., Exon, N., Beautement, P., Butler, M., Leuschel, M., and Turner, P. (2003). Trust issues in pervasive environments, deliverable wp2-01. Technical report, University of Southampton and QinetiQ.

Vacca, J. (2004). *Public Key Infrastructure: Building Trusted Applications and Web Services*. AUERBACH.

Winslett, M., Yu, T., Seamons, K., Hess, A., Jacobson, J., Jarvis, R., Smith, B., and Yu, L. (2002). The trust-builder architecture for trust negotiation. *IEEE Internet Computing*, 6(6):30 – 37.

Zimmermann, P. R. (1995). *The official PGP User's Guide*. MIT Press.