Trust Transfer in Distributed Systems

Changyu Dong, Giovanni Russello and Naranker Dulay

Department of Computing
Imperial College London
180 Queen's Gate, London, SW7 2AZ, UK
{changyu.dong,g.russello,n.dulay}@imperial.ac.uk

Abstract. Trust transfer is a common technique employed in trust management systems to establish relationships between parties that are strangers. It is also well known that trust is not always transferable. That is, given an existing trust relationship, it may or may not be possible to derive new trust from it. In particular, it is not known under which constraints trust is transferable. In this paper we investigate trust transfer and identify when trust is transferable. Our analysis starts with a simple trust model. By using the model, we find that trust transfer is related to trust policy entailment. We then present a modal logic system which captures how trust and beliefs evolve in distributed systems. With the modal logic system we identify the key constraints on trust transfer regarding the communication between the trustor and the recommender and the trustor's belief state.

1 Introduction

The open and dynamic nature of modern distributed systems presents a significant challenge to security management. Traditional security management systems are centralised and operate under a closed world assumption. All participants must have an identity established by the system and share some secret information with the system for authentication purposes. The centralised model is usually infeasible in open distributed systems. Trust management [1, 2, 3, 4, 5, 6] is an alternative approach that utilises some notion of *trust* in order to specify and interpret security policies and make authorisation decisions on security-related actions.

One of the main objectives of trust management is to build up trust between two strangers effectively. Trust can be established by direct experience [7, 8]. Generally, two parties start from interactions requiring little or no trust, the outcome of each interaction with the trustee affects the trustor's trust towards it. A positive outcome increases the trust while a negative outcome decreases the trust. As trust increases, the parties can engage in interactions which require more trust. However, building trust in this way needs time and is inappropriate when both parties require a quick decision, for example, for a one-off interaction. *Trust transfer* (or trust transitivity) is more useful in such cases. Trust transfer is the process of deriving new trust from existing trust. One example of

Please use the following format when citing this chapter:

Dong, C., Russello, G. and Dulay, N., 2007, in IFIP International Federation for Information Processing, Volume 238, Trust Management, eds. Etalle, S., Marsh, S., (Boston: Springer), pp. 17–29.

utilising trust transfer is *recommendations.* A recommendation is a statement regarding the trustworthiness of the potential trustee from another party, the recommender. The trustor makes its decision based on the recommendation. For example, Alice may trust Bob to be a good car mechanic if her friend Carol says so. This kind of scenario is common in the real-world and seems to work well. But when we try to capture it in computational trust models, we encounter difficulties.

A key problem is that trust is not always transferable [9, 10, 11, 12]. That is, given an existing trust relationship, it may or may not be possible to derive new trust from it. In particular, it is not known under which constraints trust is transferable. Without solving this problem, systems based on trust transfer can be unreliable. Trust may be misplaced when it is not transferable, which may consequently lead to bad decisions.

In the remainder of this paper, we first present a basic trust model and use it to analyse the trust transfer problem. We then develop a modal logic system which captures how trust and beliefs evolve in distributed systems and derive the constraints for trust transfer. We believe that the constraints and the modal logic provide a foundation for constructing more reliable trust management systems.

2 A Basic Trust Model

Our basic trust model is similar to the one presented by Castelfranchi *et al* [13]. It is simple but captures the most important properties of trust. The model is described as follows:

- Trust is a binary relation between two subjects: the trustor and the trustee.
- Trust is a binary decision: trust or distrust.
- Trust is bound to a goal. A goal is what the trustor wants to achieve by relying on the trustee or how the trustee is expected to behave. For example, "be a good car mechanic" or "to read my document".
- Trust is subjective. For the same trustee and goal, different trustors may make a different decision.

In this model, trust is defined as a logic predicate: $Trust(trustor, trustee, goal)$. The predicate is true when the trustor trusts the trustee for the goal, and false otherwise. Each subject has a set of trust policies. A trust policy reflects the trustor's evaluation criteria and sets requirements for certain attributes of the trustee and the environment. A trust policy is modelled as $Trust(trustor, trustee, goal) \leftrightarrow pol$, where the policy body pol is a conjunction of predicates. The trustor trusts the trustee for a goal if and only if the trust policy body is true. Trust policies capture the subjectivity of trust.

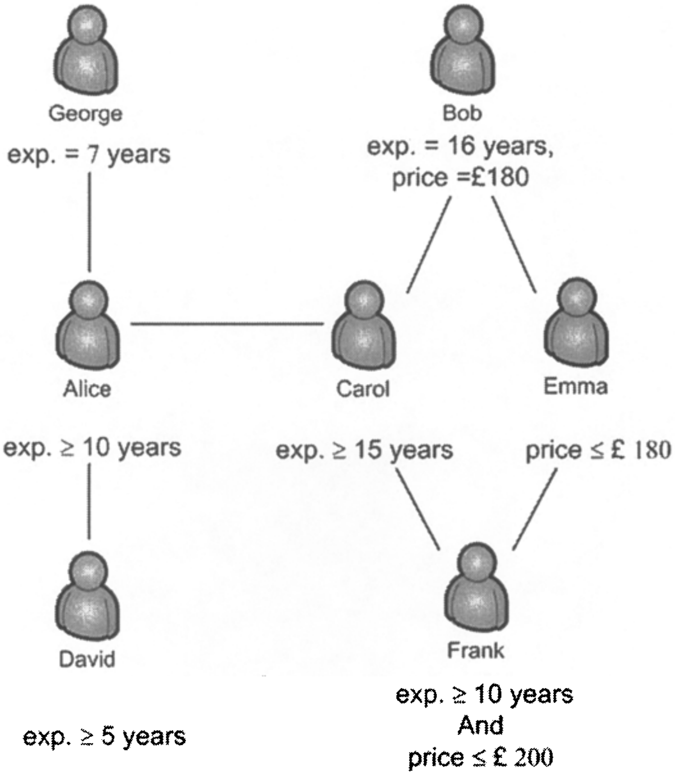

Fig. 1. Trust Transfer Example: subjects and their trust policies

3 Analysis of Trust Transfer

Before we begin our analysis, we need to express the problem more formally. Given $Trust(r, e, g)$ is true if $Trust(t, e, g)$ is also true, i.e. $Trust(r, e, g) \rightarrow Trust(t, e, g)$, then we say trust is transferable from r to t. Our goal is to find the constraints for trust transfer.

It is clear that if the trust policies for subject r and t are $Trust(r, e, g) \leftrightarrow pol$ and $Trust(t, e, g) \leftrightarrow pol'$, then $Trust(r, e, g) \rightarrow Trust(t, e, g)$ if and only if $pol \rightarrow pol'$. Loosely speaking, if pol is more strict than pol', then the trust established by satisfying pol can transfer from r to t. We can explain this using an example (see Fig. 1): Alice will trust anyone to be a good car mechanic if he has at least ten years experience, and Carol will trust anyone to be a good car mechanic if he has at least fifteen years experience. For example, if Carol thinks that Bob is a good car mechanic, Alice can also trust Bob because he satisfies her requirement. In this case, trust is said to transfer from Carol to Alice.

We can derive more rules from the above rule. For example, trust can transfer in a chain. A subject t_1 can derive a trust relationship $Trust(t_1, e, g)$ from $Trust(r, e, g)$, then another subject t_2 derives a new trust relationship

$Trust(t_2, e, g)$ from $Trust(t_1, e, g)$, and so on. According to the above rule, a trust chain $(Trust(r, e, g) \rightarrow Trust(t_1, e, g)) \wedge (Trust(t_1, e, g) \rightarrow Trust(t_2, e, g)) \wedge ... \wedge (Trust(t_{n-1}, e, g) \rightarrow Trust(t_n, e, g))$ is possible if and only if $(pol \rightarrow pol_1) \wedge (pol_1 \rightarrow pol_2) \wedge ... \wedge (pol_{n-1} \rightarrow pol_n)$ where $pol, pol_1, ..., pol_n$ are the corresponding trust policy bodies. In other words, a trust chain can be formed if the trust policies are monotonically relaxed along the chain. Suppose David will trust anyone to be a good car mechanic if he has at least five years experience, then the trust towards Bob can be transferred from Carol to David via Alice.

It is also possible to derive a new trust relationship from a set of existing trust relationships, i.e. $Trust(r_1, e, g) \wedge Trust(r_2, e, g) ... \wedge Trust(r_n, e, g) \rightarrow Trust(t, e, g)$. It can be the case that each recommender's policy only subsumes a subset of the trustor's requirements. For example, Frank will trust anyone to be a good car mechanic if he has at least ten years experience and asks for no more than £200, Carol will trust anyone to be a good car mechanic if he has at least fifteen years experience, and Emma will trust anyone to be a good car mechanic if he asks for no more than £180. Each of Frank's friends cannot convince him, but when both of them think Bob is good, Frank can trust Bob. So if $pol_1 \wedge pol_2 ... \wedge pol_n \rightarrow pol$, then multiple trust relationships can be combined to derive new trust.

If trust is transferable, so is distrust. If two subjects r and t have trust policies $Trust(r, e, g) \leftrightarrow pol$ and $Trust(t, e, g) \leftrightarrow pol'$, where $pol \rightarrow pol'$, then as we have said, trust can transfer from r to t. At the same time, distrust can transfer from t to r, i.e. $\neg Trust(t, e, g) \rightarrow \neg Trust(r, e, g)$. For example, Alice will trust anyone to be a good car mechanic if he has at least ten years experience, and Carol will trust anyone to be a good car mechanic if he has at least fifteen years experience. If Alice thinks that George is not a good car mechanic, Carol should not trust George because if he cannot satisfy Alice's requirement, he will never be able to satisfy her requirement.

4 A Modal Logic for Trust

With the basic model, we revealed the relationship between policy entailment and trust transfer. But this model is not suitable for analyzing trust transfer in distributed systems. One limitation of this model is that the knowledge is global, i.e. every subject knows everything in the system. But in distributed systems, subjects must make decisions based on their local knowledge. For example, if Alice doesn't know Carol's trust attitude towards Bob, she has no legitimate basis to conclude whether to trust Bob or not. In addition, first order logic is too strong for defining trust policies. When evaluating policies in first order logic, a subject must know the logical truth of the predicates, which may not be possible because the subject has only limited knowledge. In many situations, the subjects make decisions not because a predicate is true or false, but rather because they *believe* that it is true or false based on their local knowledge.

In order to overcome the limitations above, we extend the basic trust model to a modal logic system. The logic is built above an idealised model of a distributed system where each subject has its own local state and communicates with others via messages. Communication changes the subjects' local states and in turn results in the evolving of the subjects' beliefs and trust.

4.1 Syntax

First we define the language for the logic. We assume there exists a set $\mathbb{T}$ of primitive terms. $\mathbb{T}$ contains several disjoint sets of constant symbols: a set of primitive propositions, denoted by Φ_0; a set of subjects, denoted by $\mathbb{S}$; a set of goals, denoted by $\mathbb{G}$. Each individual subject is denoted by a natural number, i.e., $1, 2, ..., n$.

The well formed formulae(wff) of the logic is the smallest set that contains:

- The primitive proposition set Φ_0;
- $T_i(j, G)$, read as "subject i trusts subject j for goal G" where $1 \leq i \neq j \leq n$ are subjects and $G \in \mathbb{G}$;

and is closed under the following rules:

- if ϕ is a wff, then so is $\neg\phi$ where $\neg$ is the Boolean connective "not";
- if ϕ is a wff, then so is $B_i\phi$, read as "subject i believes ϕ" where $1 \leq i \leq n$ is a subject;
- if ϕ is a wff, then so is $S_i(j, \phi)$, read as "subject i sees a message from j containing ϕ" where $1 \leq i \neq j \leq n$ is a subject;
- if ϕ and ψ are wffs then so is $\phi \wedge \psi$ where $\wedge$ is the Boolean connective "and".

Other classical Boolean connectives $\vee$ (or), $\rightarrow$ (if), $\leftrightarrow$ (iff), $\top$ (true), and $\perp$ (false) can be defined as abbreviations.

4.2 System Model

Before giving the semantics for the logic, we first sketch our model of the distributed system in which the logic will be used. The system model is similar to those defined in [14, 15].

The basic elements of a system are subjects. For convenience, we use the same notation $\{1, 2, ..., n\}$ as in the syntax to denote the subjects in describing the system model. A subject can be a person, an organisation, a computer process or any other entity. We assume that subjects can be identified uniquely in the system.

The system is modelled using a state-based approach. At any time point, each subject i in the system is associated with a local state ω_i. The local state is determined by the subject's knowledge, e.g. its trust policies, its beliefs and what it has learned from other subjects etc. The system is also associated with a global state ω at the same time, which consists of all the local states of the subjects in the system.

Subjects can communicate with each other via messages. A message contains a conjunction of wffs and must have a sender. The receiver is optional for the message, that means the message can be sent by a point-to-point channel or by broadcast.We require that messages cannot be forged or modified during communication. If a subject forwards a message it received from another subject, e.g. "Alice says that Bob said that X", the original sender can be identified. Each subject maintains a message history, which is a sequence of messages it received. The messages in the history are ordered by the time they were received. When searching the message history, the subject always starts from the latest one and returns when it finds a match. This means that if there is a conflict in two messages, the subject always gets the newer one. We define a function $MESSAGE(\omega_i)$ which returns a set of messages which are the message history in state ω_i. We also define another function $MESSAGE\_CONTAINS(M, \phi, j)$ which returns true if the message M is from subject j and contains a wff ϕ, false otherwise.

Each subject has its own beliefs. The beliefs may come from the subject's preconceptions which are the initial beliefs when it entered the system, or by interacting with other subjects in the system, or come from outside the system, e.g. by perceiving the real world. The beliefs are uniquely determined by the subject's local state.

To make trust decisions, a subject must have a set of trust policies. A trust policy is based on the trustor's beliefs. For a subject i, the trust policy is always in the form of $T_i(j, G) \leftrightarrow B_i\phi$. This means that i, who is the trustor, will trust j, the trustee, for the goal G if and only if he believes ϕ where ϕ is a conjunction of wff.

4.3 Semantics

The most widely accepted modal logic system for beliefs is KD45 [16, 17, 18]. We follow this convention in our logic. Beliefs are interpreted in the *possible worlds semantics* [16] which is a formal semantics for modal logic and has been used intensively in formulating knowledge and beliefs. The intuition behind the possible worlds model is that there are many global states, or "worlds". In a given world, a subject considers a number of worlds to be possible according to its local state. The truth value of a wff depends on these possible worlds. For example, a subject is said to believe ϕ if and only if ϕ is true in all the worlds that the subject considered possible. The set of possible worlds is determined by the accessible relation (or possibility relation).

A Kripke structure [19] is used as a formal model for possible worlds semantics. A model for our logic is a tuple $(W, \pi, (\beta_i)_{1 \leq i \leq n})$, where:

- W is a set of all worlds,
- $\pi : \Phi_0 \rightarrow 2^W$ is a truth assignment mapping each primitive proposition to the set of worlds in which it is true;
- $(\beta_i)_{1 \leq i \leq n} \subseteq W \times W$ is an accessibility relation for the subject i. By convention, β_i is serial ($\forall w \exists u,\ u \in \beta_i(w)$), transitive ($\forall w, u, v,\ u \in \beta_i(w) \wedge v \in$

$\beta_i(u) \rightarrow v \in \beta_i(w))$ and Euclidean $(\forall w, u, v,\ u \in \beta_i(w) \wedge v \in \beta_i(w) \rightarrow v \in \beta_i(u))$.

We are now ready to present a formal definition of the truth of a wff. Given a model $\mathcal{M}$, we define the truth of a wff at a world ω, denoted by $\mathcal{M}, \omega \models \phi$ by induction on the structure of ϕ:

- $\mathcal{M}, \omega \models p$ iff $\omega \in \pi(p)$ for primitive proposition $p \in \Phi_0$;
- $\mathcal{M}, \omega \models \neg\phi$ iff $\mathcal{M}, \omega \not\models \phi$;
- $\mathcal{M}, \omega \models \phi \wedge \psi$ iff $\mathcal{M}, \omega \models \phi$ and $\mathcal{M}, \omega \models \psi$;
- $\mathcal{M}, \omega \models B_i\phi$ iff for all $u \in \beta_i(\omega), \mathcal{M}, u \models \phi$;
- $\mathcal{M}, \omega \models S_i(j, \phi)$ iff in ω, we can find a message $M \in MESSAGE(\omega_i)$ such that $MESSAGE\_CONTAINS(M, \phi, j)$ is true;
- $\mathcal{M}, \omega \models T_i(j, G)$ iff in ω there exists a policy $T_i(j, G) \leftrightarrow B_i\phi$ and $\mathcal{M}, \omega \models B_i\phi$.

Trust and beliefs are interrelated by the trust policies. This means that trust always depends on the subject's belief state. $S_{ij}\phi$ is totally determined by the subject i's local state. In any state, if i can find a message from j containing ϕ in its message history, then $S_{ij}\phi$ is true.

4.4 Axioms and Inference Rules

The axiom schema consists of the following axioms:

P All substitution instances of propositional tautologies
B1 $B_i(\phi \wedge \psi) \leftrightarrow B_i\phi \wedge B_i\psi$
B2 $B_i\phi \wedge B_i(\phi \rightarrow \psi) \rightarrow B_i\psi$
B3 $\neg B_i\bot$
B4 $B_i\phi \leftrightarrow B_iB_i\phi$
B5 $\neg B_i\phi \rightarrow B_i\neg B_i\phi$
S1 $S_i(j, \phi \wedge \psi) \leftrightarrow S_i(j, \phi) \wedge S_i(j, \psi)$
S2 $S_i(j, \phi) \wedge S_i(j, \phi \rightarrow \psi) \rightarrow S_i(j, \psi)$
S3 $S_i(j, S_j(k, \phi)) \rightarrow S_i(k, \phi)$

and the following inference rules

R1 (Modus ponens): from $\vdash \phi$ and $\vdash \phi \rightarrow \psi$ infer $\vdash \psi$
R2 (Generalisation): from $\vdash \phi$ infer $\vdash B_i\phi$

Axioms B1-B5 are standard KD45 axioms which capture the characteristics of beliefs. B1 says that a subject believes the conjunction of two wffs ϕ and ψ, if and only if it believes ϕ and also believes ψ. B2 says that a subject believes all the logical consequences of its beliefs. B3 says that a subject does not believe an obviously false statement. B4 and B5 state that a subject knows what it believes and what it doesn't believe.

S1-S3 are axioms for communication. S1 and S2 are similar to B1 and B2. S3 says that a subject can identify the origin of a message forwarded by another subject. This comes from the requirement of our system model that every message must have a sender and cannot be forged or modified.

5 Constraints for Trust Transfer

We now conduct an in-depth examination of trust transfer. As in section 3, let's first formalize the problem. The difference between the modal logic system and the basic model is that trust is determined by the local state of each subject, and one subject's local state is totally independent of the states of other subjects. The only way that a subject can affect the local state of another subject is through communication. Here we redefine the problem as: given $S_k(i, T_i(j, G))$ is true if $T_k(j, G)$ is also true, i.e. $S_k(i, T_i(j, G)) \rightarrow T_k(j, G)$, then we say that trust is transferred from i to k. This means that a subject must know another subject's trust attitude before deriving a new trust relationship.

From $S_k(i, T_i(j, G))$, we cannot derive $T_k(j, G)$ in our logic system. There are many points to consider. First of all, does this message reflect the real local state of i? If subject i says it trusts j for G, is this the real attitude of i? Also, is the subject k willing to believe what i says? i might be telling the truth, but if k doesn't accept it, it still means nothing.

To make trust transferable, the trustor k must have some beliefs in the recommender i. These can be formalised as:

A1 $B_k(S_k(i, \phi) \rightarrow B_i\phi)$.
A2 $B_k(B_i\phi \rightarrow B_k\phi)$.

The first one says k must believe i is honest, i.e. i only says what it believes. The second one says k must be willing to accept beliefs from i.

With these beliefs, k can begin to derive new trust. Given $S_k(i, T_i(j, G))$, by R2, k has:

$$B_k(S_k(i, T_i(j, G)))$$

Recall A1 says that k believes what i said is what i believes. With the above belief and if we apply B2, k has:

$$B_k B_i(T_i(j, G))$$

Taking the above belief with A2 and applying B2, k has:

$$B_k B_k(T_i(j, G))$$

This can be simplified by applying B4:

$$B_k(T_i(j, G))$$

Now k believes that i trusts j for G. It is quite close, but k still cannot conclude that $T_k(j, G)$ is true. k trusts j for the goal G if and only if the trust policy $T_k(j, G) \leftrightarrow B_k\psi$ is satisfied, i.e. $B_k\psi$ is true. If $B_k(T_i(j, G)) \rightarrow B_k\psi$ is true, then the new trust relationship between k and j can be established.

Recall in section 3, that our analysis showed that policy entailment is an important factor for trust transfer. But in distributed systems, trust policies are in each subject's local state, so k will not believe i has a more strict policy until it sees it and believes this is indeed i's policy. i must show its policy to k, i.e. $S_k(i, T_i(j, G) \leftrightarrow B_i\phi)$. If k thinks i is honest, it can get:

$$B_k(T_i(j,G) \leftrightarrow B_i\phi))$$

The above belief with $B_k(T_i(j,G))$ and A2 can then derive:

$$B_k(\phi)$$

If i's policy is really more strict than k's, i.e. $\phi \rightarrow \psi$, k can generalise it into $B_k(\phi \rightarrow \psi)$ by R2. Then it can finally derive $B_k\psi$, which in consequence, makes $T_k(j,G)$ true.

In summary, our constraints for trust transfer in distributed systems can be stated as follows:

C1 The trustor must know the recommender's trust attitude, i.e. $S_k(i, T_i(j,G))$ is true.
C2 The trustor must believe the recommender is honest, i.e. $B_k(S_k(i,\phi) \rightarrow B_i\phi)$ is true.
C3 The trustor must be willing to acquire beliefs from the recommender, i.e. $B_k(B_i\phi \rightarrow B_k\phi)$ is true.
C4 The trustor must know the recommender's trust policy, i.e. $S_k(i, T_i(j,G) \leftrightarrow B_i\phi)$ is true.
C5 The recommender's trust policy must be more strict than the trustor's, i.e. $\phi \rightarrow \psi$ is true.

Rules for trust transfer chains, trust fusion and distrust transfer as discussed in section 3 can also be derived from the constraints above.

There may be some objections to constraint C4, which says that a trustor must know the recommender's trust policy. Here we make some justification for this. Intuitively, when we seek a recommendation from a friend, we expect the judgement of the recommender is better than ours. But how can we know it is better? We might ask the recommender why does he thinks that it is good or why he thinks that it is not good. In other words, we are trying to figure out his policy and compare it with ours. That is why most online recommendation systems need not only feedback but also comments: comments can provide clues of the reviewer's evaluation standards. It is sometimes possible that we can derive trust without asking for the policy. This usually happens when we already know the recommender very well, so we can infer what his policy is, i.e. we already have $B_k(T_i(j,G) \leftrightarrow B_i\phi))$ and $\phi \rightarrow \psi$.

6 Related Work

Trust transfer has been studied for many years as trust transitivity. Researchers have noticed that trust is not always transitive. Grandison [11] concluded that transitivity cannot be used as an axiom for trust relationships because of the diversity of distributed systems. He also concluded that trust is not transitive in general, but can be in some cases. Christianson *et al* [9] pointed out that modelling trust transitivity requires careful analysis of the beliefs held by principals about each other and the basis upon which these beliefs are held, otherwise using trust transitivity can be harmful.

Abdul-Rahman *et al* [5] studied conditions under which trust transitivity may hold. They came to the conclusion that for transitivity to be held in the simple example "if A trusts B and B trusts C then A trusts C", four conditions must be satisfied:

- B explicitly communicates his trust in C to A, as a 'recommendation'.
- A trusts B as a recommender, i.e. recommender trust exists in the systems.
- A is allowed to make judgements about the 'quality' of B's recommendation (based on A's policies).
- Trust is not absolute, i.e. A may trust C less than B does, based on B's recommendation.

This seems to be a more detailed formulation of trust transitivity, but it can be obscure because the notion of recommender trust does not have clear semantics. They defined it as "closeness of recommender's judgement to the trustor's judgement about trustworthiness", where "closeness" is quite vague. As a result, the computation model for deriving trust value is not concrete.

Jøsang *et al* [20, 21, 12, 4] have done a lot of research on trust transitivity. They argue that for trust to be transitive, trust purpose (scope) must also be considered. Trust purpose expresses the semantic content of an instantiation of trust, i.e. what the trustor wants to achieve through the trust. Trust transitivity can break down because the trust purposes are different and do not fit together for the subject in the chain. So if Alice wants to find a car mechanic and Carol recommends Bob because she trusts him as a good car salesman, this cannot form transitive trust. This result can be explained in our model. Usually with different purposes (goals in our terminology), a subject examines different sets of the trustee's attribute, e.g. for a car mechanic, the subject cares about his experience, and for a car salesman, the subject cares about whether he can offer a good discount. It is hard to form an entailment between policies regarding different attributes, therefore when purposes are different, trust usually is not transferable. In Jøsang's model, trust is expressed as reliability which is the subjective probability by which the trustor expects the trustee to perform a given action. When a transitive trust path is found, the trust value can be propagated from the recommender to the potential trustor, the potential trustor can decide whether to trust the trustee for the trust purpose by calculating a value for the indirect trust. Abstracting trust as a probability makes it easier for computation, but also loses useful information. As a trust value is a subjective probability, it is only meaningful to a particular trustor. When communicated to the other party without justification, this can be misinterpreted.

Modal logic [22] can be used to express modalities such as possibility, necessity, belief, and knowledge etc. It has been used to formalise and analyze trust because trust is closely related to beliefs. Rangan [14] proposed a modal logic for beliefs and presented an axiomatic theory of trust in distributed systems. In his system, trust is modelled as axioms which can provide desirable security properties when added into the logic. The paper discussed how to map certain security requirements into trust axioms and uses the logic to verify the security of distributed systems. Liau [23] presents the BIT logic for belief, information acquisition and trust. In BIT logic, trust is denoted by a modal operator with neighborhood semantics [22] and is used to infer beliefs from acquired information. Liau also discusses trust transfer and gives an axiom to derive new trust when trust is transferable. But he does not address under which conditions trust is transferable. Both works focus on how to use trust as a tool to reason about beliefs, but cover little about how to derive trust from beliefs which is important in the context of building trust management systems.

7 Conclusion and Future Work

In this paper, we considered the trust transfer problem using a simple trust model and then a modal logic system. Our contribution is the identification of the constraints needed for trust transfer in distributed systems, namely that:

- the trustor must know the recommender's trust attitude.
- the trustor must believe the recommender is honest.
- the trustor must be willing to acquire beliefs from the recommender.
- the trustor must know the recommender's trust policy.
- the recommender's trust policy must be more strict than the trustor's.

Besides trust transfer, there are two other mechanisms commonly used to establish indirect trust: credentials and reputation. One area of our future work will be to analyse credential-based and reputation-based trust. For example, a credential is an assertion on the trustee's attributes. It can be viewed in our logic as $S_i(j, B_j\phi)$, where j is the credential issuer. Reputation, on the other hand, can be viewed as the aggregation of trust opinions from a community. We hope to analyse, model and compare these alternatives with each other and with trust transfer.

We plan to apply our results and modal logic system in the implementation of the trust management system for the CareGrid project [24]. CareGrid aims to provide middleware for organising and coordinating trust, privacy and security decisions across collaborating entities using autonomous trust domains and context. The CareGrid trust management system will also be integrated with Imperial's Ponder2 policy management framework [25] and used for developing trust-based distributed, mobile and ubiquitous systems.

Acknowledgments

This research was supported by the UK's EPSRC research grant EP/C537181/1 and forms part of CareGrid, a collaborative project with the University of Cambridge. The authors would like to thank the members of the Policy Research Group at Imperial College for their support and to Marek Sergot for his advice.

References

1. M. Blaze, J. Feigenbaum, and J. Lacy (1996) Decentralized trust management. In: Proceedings of the 1996 IEEE Symposium on Security and Privacy, (Washington, DC, USA), p. 164, IEEE Computer Society.
2. A. Herzberg, Y. Mass, J. Mihaeli, D. Naor, and Y. Ravid (2000) Access control meets public key infrastructure, or: assigning roles to strangers. In: Proceedings of the 2000 IEEE Symposium on Security and Privacy, (Berkeley, CA), pp. 2–14.
3. N. Li, J. C. Mitchell, and W. H. Winsborough (2002) Design of a role-based trust-management framework. In: Proceedings of the 2002 IEEE Symposium on Security and Privacy, (Washington, DC, USA), p. 114, IEEE Computer Society.
4. A. Jøsang, E. Gray, and M. Kinateder (2006) Simplification and analysis of transitive trust networks. Web Intelligence and Agent Systems, 4(2):139–161.
5. A. Abdul-Rahman and S. Hailes (1997) A distributed trust model. In: Proceedings of the 1997 workshop on New security paradigms, (New York, NY, USA), pp. 48–60, ACM Press.
6. B. Yu, M. P. Singh, and K. Sycara (2004) Developing trust in large-scale peer-to-peer systems. in : Proceedings of IEEE First Symposium on Multi-Agent Security and Survivability, pp. 1–10.
7. C. M. Jonker and J. Treur (1999) Formal analysis of models for the dynamics of trust based on experiences. In F. J. Garijo and M. Boman (eds), vol. 1647 of Lecture Notes in Computer Science, pp. 221–231, Springer.
8. A. Birk (2000) Learning to trust. In: R. Falcone, M. P. Singh, and Y.-H. Tan (eds), vol. 2246 of Lecture Notes in Computer Science, pp. 133–144, Springer.
9. B. Christianson and W. S. Harbison (1997) Why isn't trust transitive? In: Proceedings of the International Workshop on Security Protocols, (London, UK), pp. 171–176, Springer-Verlag.
10. E. Gerck (1998) Toward real-world models of trust. http://www.safevote.com/papers/trustdef.htm.
11. T. Grandison (2003) Trust Management for Internet Applications. PhD thesis, Imperial College London.
12. A. Jøsang and S. Pope (2005) Semantic constraints for trust transitivity. In: S. Hartmann and M. Stumptner (eds), vol. 43 of CRPIT, pp. 59–68, Australian Computer Society.
13. C. Castelfranchi and R. Falcone(1998) Principles of trust for mas: Cognitive anatomy, social importance, and quantification. In: ICMAS, pp. 72–79, IEEE Computer Society.
14. P. V. Rangan (1988) An axiomatic basis of trust in distributed systems. In: Proceedings of the 1988 IEEE Symposium on Security and Privacy, pp. 204 – 211, IEEE Computer Society.
15. M. Abadi and M. R. Tuttle (1991) A semantics for a logic of authentication (extended abstract). In: PODC, pp. 201–216.
16. J. Hintikka (1962) Knowledge and Belief. Cornell University Press.
17. W. van der Hoek (1990) Systems for knowledge and beliefs. In:J. van Eijck (ed), vol. 478 of Lecture Notes in Computer Science, pp. 267–281, Springer.
18. N. Friedman and J. Y. Halpern (1994) A knowledge-based framework for belief change, part I: Foundations. In: R. Fagin (ed), TARK, pp. 44–64, Morgan Kaufmann.
19. S. Kripke (1963) Semantical considerations on modal logic. Acta Philosophica Fennica, 16:83–94.

20. A. Jøsang (1999) An algebra for assessing trust in certification chains. In: NDSS 99, The Internet Society.
21. A. Jøsang, E. Gray, and M. Kinateder (2003) Analysing Topologies of Transitive Trust. In: T. Dimitrakos and F. Martinelli (eds) Proceedings of the First International Workshop on Formal Aspects in Security and Trust, (Pisa, Italy), pp. 9–22.
22. B. F. Chellas (1988) Modal logic: an introduction. Cambridge University Press.
23. C.-J. Liau (2003) Belief, information acquisition, and trust in multi-agent systems–a modal logic formulation. In: Artif. Intell., 149(1):31–60.
24. The CareGrid project. www.caregrid.org.
25. The Ponder2 project. www.ponder2.net.