

Guaranteed Security in Trust and Reputation Systems

Reid Kerr and Robin Cohen

David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1
{rkerr, rcohen}@cs.uwaterloo.ca

Abstract

In this paper, we present a framework for evaluating the security of trust and reputation systems for electronic marketplaces populated with buying and selling agents. We argue that current systems to model trust are vulnerable to various attacks; to provide protection from such attacks, systems must be designed not only to predict cheating by agents seeking to deceive one another, but also to cope with agents who are intentionally trying to circumvent the trust/reputation system. Our proposed framework offers a method for researchers to understand the security of their systems, and to provide precise guarantees of the degree of provable security that these systems offer. We focus in particular on characterizing buyer security—the properties that must hold for buyers to feel secure from cheating sellers. We develop a set of security ‘levels’, benchmarks that may be used in the evaluation and comparison of system security. We demonstrate the viability of our proposed framework by presenting a specific monetary-based trust system known as Trunits, along with an analysis that shows that Trunits does provide a guaranteed level of security for buyers.

Introduction

Much research has been conducted in the area of trust and reputation in multiagent systems [2, 3, 5, 6, 8, 9, 10, 11, 12, 13, 14]; a common focus has been modeling the trustworthiness of buying and selling agents in electronic marketplaces. Many existing proposals consist of predictive models, aimed at determining whether agents are untrustworthy and proposing methods for agents to make effective decisions in the face of possible dishonesty. It is our position that trust and reputation systems should be provably secure, making precise and proven claims as to the degree of protection they provide and the circumstances under which such protection holds. The goal of this approach is twofold: a) to provide a basis for users to confidently choose whether or not to adopt or participate in a system; b) to make explicit the security limitations of systems, and the reasons for such limitations, to allow meaningful progress towards meeting the needs of users. A provably secure system allows users to place their trust in a system, removing an obstacle to its use.

We propose a framework for the consideration of security in trust and reputation systems, based on uncompromising guarantees of protection when required conditions are met. Further, we identify a number of ‘levels’ of security, varying in the strength of conditions required for the security guarantee to hold. We then demonstrate how a trust system can be provably secure for a buyer, by presenting the monetary-based Trunits trust model, and discussing the protection that it offers for its buying agents.

A common theme of work in trust and reputation in multiagent systems is to increase the likelihood of selecting a trustworthy business partner. Unfortunately, this may not be strong enough to inspire the confidence of potential users or adopters of such systems—anything less than complete trustworthiness of agents raises doubts about the attractiveness of using the system. Some trust and reputation systems attempt to provide incentives for agents to be honest (e.g., [2]), but one must still ask: Under what circumstances will the incentive hold? Will the incentive always be sufficient?

If systems aren’t provably secure, the potential for vulnerabilities exists; if vulnerabilities exist in trust and reputation systems, self-interested agents will exploit these in order to maximize profit. We have identified a catalogue of vulnerabilities in trust and reputation systems, displayed and briefly described in Table 1. We then provide a chart to record the vulnerabilities to which we believe each of a small sample of trust systems is open, shown in Table 2.

In some cases, we can relate the presence of vulnerabilities to design choices made by system designers. Systems in which agents make use of the recommendations of other agents, and where each opinion may be used as support for multiple transactions simultaneously [4, 9, 13, 14] tend to be vulnerable to Reputation lag. Systems in which the transaction’s impact on an agent’s trustworthiness rating is not tied to the value of the transaction [4, 5, 8, 9, 13, 14] tend to be vulnerable to Value imbalance. The potential for Ballot-stuffing and Bad-mouthing can arise in several ways when agents rely on the advice of others. Opportunities can

occur when a system does not protect against users cooperating to undermine the system, or when users can freely create many new accounts, with feedback from these new accounts being weighted similarly to that from established accounts [4, 9, 13]. Re-entry tends to be a problem in systems that treat unknown users preferentially to disreputable users [4, 5, 8, 9, 11, 12, 13], making it beneficial to create a new user account after one’s reputation is damaged. Systems that rely on direct experience [5, 8, 11, 12] tend to be vulnerable to the Initial window; the presence of the Re-entry problem magnifies this vulnerability, allowing a dishonest agent to repeatedly take advantage of the window. Most systems tend to be vulnerable to the Exit problem, unless they have taken specific steps to provide an attractive alternative to cheating for agents that decide to leave the market.

Table 1. Descriptions of vulnerabilities

Vulnerability	Brief Description
Reputation lag	Agent can engage in virtually unlimited cheating transactions before reputation is updated to reflect his dishonesty
Value imbalance	Agent can build reputation on small transactions, then cheat on large ones
Ballot-stuffing/ Bad-mouthing	Reputation is artificially improved/damaged by registering large numbers of unfair ratings
Re-entry Problem	After his reputation is destroyed, the cheating agent can enter the market under an alias, effectively shedding his history
Initial window	An agent is vulnerable to an unknown agent, until experience is gained with the new agent
Exit problem	If an agent plans to leave the market, he can cheat freely without repercussions

Table 2. Vulnerabilities found in some existing systems

System	Vulnerabilities
eBay feedback system [4]	Reputation lag, Value imbalance, Ballot-stuffing/Bad-mouthing, Re-entry, Exit
General/Situational Trust [8] Multidimensional Trust [5]	Value imbalance, Re-entry, Initial window, Exit
Tran and Cohen [11, 12]	Re-entry, Initial window, Exit
Sporas/Histos [14]	Reputation lag, Value imbalance, Exit
REGRET [9]	(In market scenario) Reputation lag, Value imbalance, Ballot-stuffing/Bad-mouthing, Re-entry, Exit
Yu and Singh [13]	Reputation lag, Value imbalance, Ballot-stuffing/Bad-mouthing, Re-entry, Exit

This analysis highlights the need to consider the issue of system security carefully. Motivated by work in the field of cryptography, which seeks to deliver provable protection to users, we seek methods for ensuring the provable security of trust and reputation systems.

If our aim is to ensure that a marketplace is secure, we must first characterize what we mean by security. It is our position that a secure system for marketplaces is one where participants are protected from harm (at least, harm due to ‘dishonest’ behaviour, rather than from legitimate competition). Thus, we define security in terms of a set of ‘safety properties’: conditions that, if proven to hold for the system, ensure participants within the system will not be harmed by dishonesty. This is distinct from other notions of security, such as the prevention of unauthorized access.

In the marketplace scenario, there are three identifiable ‘stakeholders’ who participate directly in the market, each with their own requirements: buyers, sellers, and the market operator. In this paper, we focus on the security of buyers, since protecting buyers from cheating sellers is a predominant focus of current research. The other groups are discussed only briefly in section 5, to be addressed in more detail in future work.

The Security Framework

In our framework, we distinguish between two transaction states. An *agreed transaction* consists of the terms to which both parties have agreed: $t_A = (g_p, v, d_p, A_b, A_s, \dots)$, where g_p is the good promised, v is the value (agreed price) of the good, d_p is the date/time promised, A_b is the buying agent, and A_s is the selling agent. (The ellipsis indicates that there may be other system- or market-dependent parameters.) This might be viewed as a promise or a contract; it may also be viewed as the transaction at the point both parties have struck a deal, but have not yet acted, so the honesty or dishonesty of the transaction is undetermined.

A *delivered transaction* is one where the selling agent has provided the goods to the buyer, but the buyer has not yet rated the seller: $t_D = (t_A, g_d, c, d_d, \dots)$, where t_A is the agreed transaction, g_d is the good delivered, c is the cost incurred by the seller in providing and delivering the good, and d_d is the date/time of delivery. We consider a delivered transaction t_D to be *honest* if it fulfills the seller’s commitments— g_d satisfies g_p , d_d satisfies d_p , etc.—and denote it by the predicate $honest(t_D)$. (The details of how a good or promise is specified are left to the system designer. For instance any $d_d \leq d_p$ may be considered honest.)¹ We consider a transac-

¹ Note that buyers will often also close off a given transaction by computing a rating for the seller. When ratings are elicited from buyers may vary by system, however. Thus, we refrain from defining a ‘rated transaction’ for generality.

tion where an agent (intentionally) fails to fulfill his commitment (whether by providing a good that does not meet the commitment, or by not providing a good at all) to be an instance of *cheating*, or *dishonesty* on the part of the seller.

It is possible that a buyer could cheat by withholding payment after receipt of the goods. We base our framework, however, on the common policy that a buyer must pay before goods are shipped.

For brevity, we make use of ‘accessor’ functions that return the value of individual transaction parameters. These functions have the same name as the parameter that they return.

An agent attempting to cheat may act alone, or as part of a coalition. We denote such a coalition G ; an agent acting alone is equivalent to the case where $|G| = 1$.

We term a set of transactions a *schedule*. Let T_D represent a schedule of delivered transactions. For any T_D , there is a corresponding T_A consisting of the same transactions with the delivery parameters removed. Note that for any T_A , there are possibly many T_D , since each transaction in T_A might be executed honestly or dishonestly. *Executing* a transaction refers to delivering a good (that either meets or fails to meet the advertised promise), or consciously deciding not to deliver the good at all.

For any coalition of sellers G , consider T_D where $t_D \in T_D \Leftrightarrow A_s(t_D) \in G$. For each transaction in the set, the sellers in G may choose to execute the transaction honestly or dishonestly. We denote $C \subseteq T_D$ as the *cheating set*, the subset that is executed dishonestly. The coalition may have a choice of many different cheating sets for any given schedule; choosing C is a strategic choice. (We stop short of saying that C is a strategy, however, since the coalition might also strategically choose the composition of T_D by choosing the transactions into which they will enter.)

Not all schedules can actually be executed. For example, an agent that cheats repeatedly might not continue to find buyers for its products; although it might be possible to formulate a schedule that includes continued future business, such a schedule may be impossible under the trust system. Continuing the example, if trustworthiness is rated in the interval $[0, 1]$, and an agent’s score has dropped to 0, he may not be able to engage in further transactions, even though he has inventory. We define the predicate *feasible*(t, T) to denote that a transaction t can actually be executed within the schedule T , in the system under consideration. *feasible*(T) denotes that every transaction in T is feasible. We do not define feasibility further, since it will be system- or market-specific.

As we will see, profitability is a key concern when considering the security of trust systems. The profit to the seller on an individual transaction is the selling price minus the cost, or $P(t_D) = v(t_D) - c(t_D)$. The profit to a coalition on the entire set of transactions is

$$P(G, T_D) = \sum_{\{t_D \in T_D \mid A_s(t_D) \in G \wedge A_b(t_D) \notin G \wedge \text{feasible}(t_D, T_D)\}} P(t)$$

Buyer Security

A buyer who engages in no transactions suffers no direct harm from those transactions. A buyer who enters into a transaction (assuming the common pay-before-delivery policy) becomes vulnerable at the moment that they pay for a good. From this point, the seller is in control, and the buyer may be harmed by receiving an inferior good, or no good at all.

A seller may be harmed, for example, by unfair feedback from buyers. For a seller to wish to be honest, he may need confidence that buyers will provide fair reviews. We do not address means to ensure the fairness of buyers here, however, since this is an element of seller security. To establish that a system is buyer secure, then, may require the assumption that buyers are honest. We discuss how to address this problem, and lift this assumption, in the Discussion and Future Work section.

In our framework, for a system to be secure, we do not hold the seller responsible for the buyer’s complete satisfaction. Instead, she need only deliver the good that she was ‘supposed’ to give. When a seller offers a good for sale, she provides information about that good. This information constitutes the basis for the buyer’s understanding of the good they will receive. Should they purchase the good, they would expect that it corresponds to each claim made in the offer. This, then, is the basis for our notion of buyer security: a buyer will be secure under a trust system if

$$\forall t_D \in T_D \text{ honest}(t_D)$$

Note that ‘buyer security’ directly addresses issues such as the value imbalance and reputation lag problems. Other forms of dishonest behaviour may not be relevant to the buyer. For example, ballot stuffing would be precluded by this property if used to lure buyers into cheating transactions, but not if used by a seller to steal sales from another seller; the later is an issue of seller security.

Levels of buyer security

We might term the previous property, should it hold, as *full buyer security*—i.e., it is impossible for a seller to cheat a buyer. Unfortunately, this property would be extremely hard to guarantee in practice. For example, one might envision a trusted third party who receives both payment from the buyer and the good from the seller, and only forwards payment to the seller after inspecting the good to ensure it fulfills the agreement. Such a system might offer great security, but is unlikely to be practical or scalable. [3]

While it may not be feasible for a system to guarantee this property, it may be possible to achieve it under certain con-

ditions, when certain assumptions hold for the marketplace. By limiting the guarantee to those circumstances where the assumptions hold, we effectively weaken the guarantee, allowing us to specify levels of security that are weaker than the ideal. We specify these properties in the form of an implication:

$$(assumption_1 \wedge \dots \wedge assumption_m) \Rightarrow \forall t_D \in T_D \text{ honest}(t_D)$$

The assumptions denote limitations in the system, which prevent it from delivering on the unconditional guarantee. This does not mean that the system is useless, however. For each assumption, there are two primary approaches to dealing with it:

1. External: It may be possible to ensure that an assumption actually holds for the marketplace in question. If the property can be verified to hold for the marketplace, or if some mechanism external to the trust/reputation system can be used to guarantee the property, then the system will function adequately despite the presence of the assumption.
2. Internal: It may be possible to modify the system to remove the assumption as a requirement for safety. Such modification may yield a more robust system, capable of working under a smaller set of assumptions. Thus, the presence of an assumption can provide important guidance for future research, allowing meaningful progress to be made.

Through the use of these techniques, the goal would be to arrive at a system for which every remaining assumption can be ensured to hold in the marketplace—such a system would be secure for that marketplace. It is our contention that clearly stating assumptions aids understanding of the security delivered, and the limitations of this security, as well as easing comparisons between possible models.

Rational-agent secure

While we may not be able to guarantee that every sale is executed honestly, we may be able to design the system so that it is in a seller's best interest to be honest. Such incentive-based approaches depend on agents being rational profit-maximizers—operation of the system depends on agents reliably choosing what is best for them. We believe this to represent an important and high level of security, stated as:

$$\text{selling agents are rational} \Rightarrow \forall t_D \in T_D \text{ honest}(t_D)$$

More formally, denoting a coalition of selling agents as G :

$$[\forall G \text{ rational}(G)] \Rightarrow \forall t_D \in T_D \text{ honest}(t_D)$$

Recall that this entire statement is a specified property of a system. It does not state that the implication holds in all cases; rather, for a system to be considered Rational-agent secure, it must be proved that under the system, if selling agents are rational then all transactions are honest.

Since rational sellers are profit maximizers, the property above can be restated as:

$$[\forall G \forall T_{D1}, T_{D2} [P(G, T_{D1}) > P(G, T_{D2}) \Rightarrow T_{D1} \text{ is selected}]] \\ \Rightarrow \forall t_D \in T_D \text{ honest}(t_D)$$

For a system to be rational-agent secure, sellers must be able to understand that honesty is the most profitable policy. Under some systems this may require considerable computation. For example, determining that honesty maximizes profit may require the computation of an entire tree of possible future outcomes, which may be beyond the capabilities of the agent. Where this may be an issue, the set of assumptions should include the computational capacity required of the agents.

Just as rational-agent security is quite a strong guarantee, it may also be difficult to achieve. We consider several lower levels of security, derived by adding weakening conditions to the rational-agent secure property. The assumptions described below are not mutually exclusive, nor can they be ordered in terms of security. Systems requiring one or more of the following assumptions may be useful for certain scenarios, or may be only of research interest, as a stepping stone to a more secure method.

Rational single-agent secure

Ideally, a system would make the buyer secure regardless of collusion between agents. However, collusion is notoriously difficult to combat. A lower level of security might protect agents only from sellers who are not part of a coalition:

$$[\forall G \text{ rational}(G) \wedge |G| = 1] \Rightarrow \forall t_D \in T_D \text{ honest}(t_D)$$

Rational single-seller-only secure

Under some systems, a seller might be able to execute attacks by acting as a buyer for some transactions, and as a seller for others. As a weaker extension of single-agent security, a system might be secure when sellers cannot act as buyers. (A seller might be able to open another account to use as a buyer, but that is an instance of collusion.)

$$[\forall G \text{ rational}(G) \wedge |G| = 1 \wedge \forall t_D \in T_D A_B(t_D) \notin G] \\ \Rightarrow \forall t_D \in T_D \text{ honest}(t_D)$$

Rational infinite-transaction secure

The exit problem is an extremely difficult one to combat, and it may be difficult to prevent dishonest sales once sellers have exhausted finite inventories. However, a system may make it more attractive for a seller to continue to do business than to exit at any point. Such a system may prevent the exit problem, but requires agents to be able to engage in infinite transactions (e.g., the seller never runs out of inventory, there are always buyers willing to purchase the product, etc.):

$$[\forall G \text{ rational}(G) \wedge \forall T_D [\text{honest}(T_D) \wedge \text{feasible}(T_D) \Rightarrow \\ (\exists t \text{ honest}(T_D \cup t) \wedge \text{feasible}(T_D \cup t))]] \\ \Rightarrow \forall t_D \in T_D \text{ honest}(t_D)$$

Of course, a buyer may require protection in other ways—that the market operator won't take her money, that her personal information won't be sold, etc. However, these issues fall outside the traditional role of a trust/reputation system, and it is difficult to conceive of a trust/reputation system preventing behaviour that occurs outside of the marketplace itself, or that controls the behaviour of its operator.

It may seem very difficult to use these standards in the analysis of many models, particularly those that are predictive in nature. It is worth reiterating, however, that unless proofs of such properties can be rendered, these systems are of unknown security at best; from our survey in section 1, it appears likely they are insecure.

Security analysis of Trunits

Having outlined a framework for establishing security guarantees and enumerated a number of important 'levels' of security, we provide no guidance in the construction of such proofs. The reason is simple—proof methods are likely to vary greatly depending on the nature of the system used. Instead, we provide an analysis of our particular model of trust, Trunits, for two reasons. First, it provides an example of the analysis of a trust model using our proposed framework. Second, we believe that it serves as one example of how trust might be modeled with an eye towards providing provable levels of security to adopters and participants.

The Trunits Model

Trunits is a model of trust—a buyer will engage in sales with a seller in which there is a certain degree of trust. It differs from most trust models, however, in that it attempts not to predict behaviour, but rather to ensure good behaviour by making honesty the most profitable strategy. We provide a brief description of Trunits here; a more detailed treatment can be found in [7].

The 'Trunits' model is inspired by the concept of money. Before the advent of money, goods and services were exchanged by bartering. This placed several limitations on trade; here, the most relevant was the requirement for buyers and sellers to interact directly, to exchange goods of comparable value. A primary function of money is to overcome this requirement.

Money is an abstract 'substance', representing quantities of value. Money flows in a transaction, mirroring the flow of value in a barter transaction: the value of the money stands in for the value of a good. Money frees the traders from the requirement that goods move in both directions—value gained from one trader can be 'spent' with another.

The key problem with trust in our new breed of marketplaces is that buyers and sellers usually do not have direct

relationships, so trust cannot form naturally. Since we seek to overcome the requirement for a direct relationship—to allow trust gained from one trader to be 'spent' with another—it seems natural to consider the use of abstract trust units, or 'trunits', to play the same basic role in which money has been so successful.

As with money, the movement of trunits should mirror that of trust in a direct relationship. This movement, however, is very different from that of value. While the flow of value is an exchange process, we see the 'movement' of trust as a *risk* process, and suggest a model based on this view. We focus on trust of the seller as the primary issue:

- Before a buyer will purchase something from a seller, the buyer must have sufficient trust in the seller. The degree of trust required is dependent on a number of factors; the price of the item is likely a major one.
- After purchasing the good, the buyer will evaluate it.
 - If the good met her expectations (i.e., it was at least as good as was advertised by the seller), then the seller is likely to gain more of her trust.
 - If the good did not meet her expectations, then the seller is likely to lose some of her trust.

Based on this view, we suggest a model that makes use of abstract units of trust, where trust of a seller is not tied to a specific buyer:

- The seller has some quantity of trunits, representing all of the trust gained from all buyers to date. For a buyer to consider buying from a seller, the seller must possess a sufficient degree of trust, i.e., must hold sufficient trunits. The required number of trunits is tied to the price of the good.
- After purchasing the good, the buyer will evaluate it, relative to her expectations.
 - If the good met her expectations, then the seller gains some additional quantity of trunits.
 - If the good did not meet her expectations, then the seller loses some quantity of trunits.

As a seller executes honest transactions, his trunit balance grows, allowing future profitable transactions. In contrast, dishonest sales curtail future transactions. This provides the fundamental incentive for honesty. The number of trunits gained is proportional to the size of the sale. Honest execution of small transactions will allow a seller to continue making small sales, and to grow his sales volume, but will not allow him to immediately jump to disproportionately large sales for which he has not demonstrated trustworthiness.

We propose a 'basic Trunits mechanism', based on this model. When an agent wishes to make a sale, we require him to put up a quantity of trunits to 'cover' the sale².

²Under Trunits as currently described, trunits are kept with a 'market operator' who administers the system. If this market

These trunits represent the trust that the seller is risking by engaging in a transaction. We require that the number of trunits risked be directly tied to the value of the transaction, using the formula:

$$V = r\tau$$

where V is the value (selling price) of the transaction, τ is the number of trunits, and r is the required *risk ratio*³, a (positive) parameter set by the market operator. The trunits are put into escrow with the market operator, pending completion of the transaction. Upon completion, if the buyer rates the transaction as unsatisfactory, then the seller loses the τ trunits placed in escrow. If, on the other hand, the buyer rates the transaction as satisfactory, then the τ trunits are returned to the seller, along with some additional quantity of trunits related to the value of the transaction, for a total of:

$$(1+p)\tau = (1+p)V/r$$

where p is a *premium* or *reward* of additional trust for acting in an honest manner, a (positive) parameter set by the market operator. In the basic mechanism presented here, the same values of r and p are used for all traders and transactions. In this implementation of Trunits, we do not allow a buyer to sell to himself (i.e., to the same user account), in order to gain trunits through ‘honest’ transactions.

From a buyer’s perspective, no evaluation or computation is required prior to purchasing to determine if a seller is trustworthy—if the seller possesses enough trunits for a transaction, then *by definition*, she is trustworthy *for that transaction*. The market operator will not allow a transaction to be executed unless the seller has sufficient trunits. From a seller’s perspective, honesty results in a growing trunit balance and the ability to engage in more sales in the future, while dishonesty will reduce the potential for future sales.

One issue encountered with this model is the ‘start-up’ problem: how does an agent acquire an initial quantity of trunits? In our current work, we investigate several options that appear to be safe, including loaning agents an initial quantity of trunits to be secured by a cash bond, and (where identities can be established with certainty, preventing Re-entry) simply providing new agents with an initial quantity. A third alternative, allowing agents to purchase trunits on an open market, is discussed briefly in section 5.

Note that, while Trunits is a model of trust, it is also a mechanism, designed to encourage honest behaviour.

operator is an identifiable entity (e.g., in a centralized implementation), this market operator is considered to be a trusted third party.

³While this relationship determines the number of trunits required to secure a sale, note that it does not imply an ‘exchange rate’; trunits cannot be directly traded for money in this manner under the mechanism presented.

Moreover, the incentive provided by the mechanism can be calculated with precision (as outlined below), serving as a basis for rational decision making.

Buyer Security in Trunits

We seek to verify that the essential buyer security property, $\forall t_D \text{ honest}(t_D)$ will hold. As an incentive mechanism, Trunits relies on agent rationality to ensure desirable behaviour: we target rational-agent security. The basic function of Trunits is that an agent makes more money if he fulfills his commitment, so he tries to do so. For this incentive to hold, the agent must *actually be able* to fulfill his commitments. If he is unable to do so successfully (e.g., poor quality control) he might find it more profitable to cheat, rather than incurring the cost associated with honestly executing a transaction, and still getting a bad rating. Thus, we assume that agents can control quality in order to meet commitments if they choose to do so. (We can actually relax this assumption under Trunits, instead specifying with precision an acceptable range in the degree of control, but omit these details for brevity.)

The basic Trunits mechanism regulates the behaviour only of sellers, so on its own, it cannot provide provable security in the face of coalitions of both buyers and sellers. Thus, we attempt to prove that Trunits provides rational single-seller-only security. Further, since Trunits is based on buyer feedback controlling future sales, we must assume that buyer honesty is ensured through some parallel mechanism. Finally, basic Trunits provides no direct impediment to a seller cheating as she exits the market, should she exhaust her ability to honestly sell goods (e.g., if she has run out of inventory). However (as will be shown below), there is a strong incentive not to exit the market, so our analysis is conducted under the assumption that the infinite-transaction property holds, where the agent can engage in infinite honest sales if desired. (By ‘infinite’, we mean both that the seller’s activity is unbounded in duration, and that the seller’s capacity for sales at any given moment is unbounded as well.)

This analysis is based on the assumption that selling cost is a fixed fraction c of selling price. This assumption is not unreasonable; many companies determine selling prices by applying percentage markups to cost, and in many industries the markup used is consistent among sellers. While we do not believe that this constraint is required for Trunits to be secure, it has been assumed in order to simplify analysis.

Security guarantee of Trunits

What we seek to prove, then, is:

- Trunits is in use \wedge
- selling agents are rational \wedge (A)
- selling agents act alone \wedge (B)
- selling agents can engage in infinite honest transactions \wedge (C)

- buying agents are honest \wedge (D)
 selling agents can reliably meet commitments if willing \wedge (E)
 cost c is a constant percentage of selling price (F)
 $\Rightarrow \forall t_D \text{ honest}(t_D)$

Specified more formally:

- [Trunits is in use \wedge
 $\forall G \forall T_{D1}, T_{D2} [P(G, T_{D1}) > P(G, T_{D2}) \Rightarrow T_{D1}$ is selected] \wedge
 $\forall G |G| = 1 \wedge$
 $\forall T_D [\text{honest}(T_D) \wedge \text{feasible}(T_D) \Rightarrow$
 $(\exists t \text{ honest}(T_D \cup t) \wedge \text{feasible}(T_D \cup t))] \wedge$
 buying agents are honest \wedge
 selling agents can reliably meet commitments if willing \wedge
 cost c is a constant percentage of selling price]
 $\Rightarrow \forall t_D \in T_D \text{ honest}(t_D)$

A note on feasibility

The feasibility of a schedule is important to its profitability, so impacting the analysis of Trunits. A transaction under Trunits begins when the agreement is made, and ends when the buyer has rated the seller. Let $start(i)$ represent the start time of transaction i , and $end(i)$ its time of completion. Let τ_{init} represent the seller's initial trunit balance, and τ_i the trunits required for transaction i . Since every transaction i requires an outflow of trunits when it begins, but only honest transactions have inflows (plus reward) at completion, the balance of trunits available at any given *time* is:

$$\tau_{bal}(time) = \tau_{init} - \sum_{i \in T_A, start(i) \leq time} \tau_i + (1+p) \times \sum_{i \in T_A \setminus C, end(i) \leq time} \tau_i$$

If, at any *time*, $\tau_{bal} < 0$, then some transaction(s) starting before *time* required more trunits than were available (i.e., the transaction(s) would not have been allowed). A feasible schedule (from the standpoint of the constraints imposed by Trunits), then, is one for which τ_{bal} is never less than 0 for all sellers of transactions in the schedule. Consider any T_D and cheating set C , where $C \subseteq T_D$. Note that the addition of a transaction (that is a member of T_D) to C (i.e., changing an honest transaction to a dishonest one) does not change the number of 'outflow' trunits, but does reduce the number of 'inflow' trunits. Thus, the addition of a transaction to C never increases τ_{bal} , but will lower it (specifically, after *end*). This means that the addition of a transaction to C might result in a previously feasible transaction becoming infeasible. Conversely, the removal of a transaction from C only increases the number of trunits available, so it cannot render a feasible transaction infeasible.⁴

Proving the guarantee

Since rational sellers choose the most profitable option, our goal is to show that, for any arbitrary schedule, profit is

maximized by executing each transaction in the schedule honestly. First, we consider only finite schedules. Consider any honest, feasible schedule T_D , and a schedule T_D' with the same set of agreed transactions T_A . T_D' has the non-empty cheating set C . Since we have assumed that sellers act alone, we omit the G from our profit formula. For each computation below, we denote each T_D by its corresponding schedule of agreed transactions and its cheating set. Thus, we seek to show that for any non-empty $C \subseteq T_A$,

$$P(T_A, C) < P(T_A, \emptyset).$$

Note that by our assumptions, the seller acts alone; further, the mechanism does not allow him to sell to himself. This means that all cash and trunit flows come only through transactions with buyers. Note, too, that the seller can meet his commitments if he chooses to do so, and that buyers are honest—this means that if a seller intends to fulfill a transaction honestly, he will receive the trunit flows due him.

The profit function for Trunits requires further consideration, regarding the value of accumulated trunits. At the end of the schedule, the seller will have earned some profit, and will have some quantity of remaining trunits (denoted $\tau_{bal}(exit)$, where *exit* is the time at which the last transaction is completed). While cheating might increase profit earned during the schedule, it would reduce the number of leftover trunits—since trunits can be used to earn future profits, this is a reduction in value gained by the seller. To measure this value, we introduce one additional transaction that occurs after *exit*. In this transaction, the seller uses all remaining trunits to cheat, as he is free to do. We do not mean to suggest that this is what the seller will or should do. (As we will show below, if he is rational he would continue to make honest trades beyond the end of the schedule.) Instead, we use this to determine the value he can *assuredly* gain from his trunits, and effectively set a lower bound on the future profits that could be earned with them. Thus, for every schedule, the total profit will be the sum of the profit from honest sales, the profit from cheating sales, and the revenue from the 'final cheat' after the schedule has completed:

$$\begin{aligned} P(T_A, C) &= (1-c)r \sum_{i \in T_A \setminus C} \tau_i + r \sum_{i \in C} \tau_i + r \tau_{bal}(exit) \\ &= (1-c)r \sum_{i \in T_A \setminus C} \tau_i + r \sum_{i \in C} \tau_i + r \left(\tau_{init} - \sum_{i \in T_A} \tau_i + (1+p) \sum_{i \in T_A \setminus C} \tau_i \right) \end{aligned}$$

(In fact, if the schedule is infeasible, the profit will be less than this, because some of the transactions will not be permitted to occur. Thus, this represents an upper limit on the profitability of the schedule.) Now, consider the same schedule, but with two different sets of cheating transactions C_1 and C_2 , where $C_1 \subset C_2$, (i.e., C_2 may be thought of as the result of adding cheating transactions to C_1). If the delivered schedule using C_1 is feasible, the one using C_2 may be either feasible or infeasible. To compare profits from each schedule, we subtract the profit of the second from that of the first:

⁴ While feasibility depends on the timing of transactions, we do not specify temporal parameters for the schedules (T_A/T_D); instead, the timing of each transaction can be specified within the transaction itself.

$$\begin{aligned}
P(T_A, C_1) &= (1 - c) r \sum_{i \in T_A, C_1} \tau_i + r \sum_{i \in C_1} \tau_i + r \left(\tau_{init} + \sum_{i \in T_A} \tau_i + (1 - p) \sum_{i \in T_A, C_1} \tau_i \right) \\
P(T_A, C_2) &= \left[(1 - c) r \sum_{i \in T_A, C_2} \tau_i + r \sum_{i \in C_2} \tau_i + r \left(\tau_{init} + \sum_{i \in T_A} \tau_i + (1 - p) \sum_{i \in T_A, C_2} \tau_i \right) \right] \\
&= (1 - c) r \left(\sum_{i \in T_A, C_1} \tau_i + \sum_{i \in T_A, C_2} \tau_i \right) + r \left(\sum_{i \in C_1} \tau_i + \sum_{i \in C_2} \tau_i \right) \\
&\quad - r(1 - p) \left(\sum_{i \in T_A, C_1} \tau_i + \sum_{i \in T_A, C_2} \tau_i \right) \\
&= (1 - c) r \left(\sum_{i \in C_1 \cup C_2} \tau_i \right) + r \left(\sum_{i \in C_1} \tau_i + \sum_{i \in C_2} \tau_i \right) + r(1 - p) \left(\sum_{i \in C_1 \cup C_2} \tau_i \right) \\
&= (1 - c + p) r \sum_{i \in C_1 \cup C_2} \tau_i
\end{aligned}$$

Given that $(1 - c)$, p , and r must all be greater than 0, as must all trunit values in the sets (and hence in the summation), this subtraction yields a positive number. (Further, note that if C_2 yields an infeasible schedule, then its profit will be reduced, increasing the result of the subtraction.) This means that if $C_1 \subset C_2$, the profit using C_1 must be higher than that of C_2 . Given that the empty set is a subset of every set, for any finite T_A and non-empty $C \subset T_A$, $P(T_A, C) < P(T_A, \emptyset)$.

The exit problem

The analysis above shows that for any finite schedule, profit is maximized through honesty, but for the last ‘cheating exit’ transaction. Ideally, the seller will never want to make such an exit; we now relax the finite schedule constraint, consistent with our stated assumption. Consider any arbitrary feasible schedule T_A . A rational seller will maximize profit by executing every transaction honestly, so the profit formula simplifies to:

$$P(T_A) = r(1 - c + p) \sum_{i \in T_A} \tau_i + r \tau_{init}$$

Instead of cheating on exit, the seller might consider executing one more honest transaction t . Assuming that the new transaction yields a feasible schedule (and since every sale in T_A is honest, it must be possible to add a feasible transaction), the new profit is:

$$\begin{aligned}
P(T_A \cup \{t\}) &= r(1 - c + p) \sum_{i \in T_A \cup \{t\}} \tau_i + r \tau_{init} \\
&= r(1 - c + p) \left(\sum_{i \in T_A} \tau_i + \tau_t \right) + r \tau_{init}
\end{aligned}$$

Since all of r , p , $(1 - c)$, and τ_t are positive, $P(T_A \cup \{t\}) > P(T_A)$, meaning that for any given schedule, it is more profitable for the seller to add profitable transactions. (Note that adding dishonest transactions does not increase the profit—cheating within the schedule is no more profitable than during the ‘cheating exit’.)

The result implies that to maximize profit, the seller should never cheat, but should continue to sell items indefinitely.

In summary, for any schedule, profit is maximized by executing every transaction honestly, and continuing to add honest transactions to infinity. Thus,

[Trunits is in use \wedge
 $\forall G |G| = 1 \wedge$
 $\forall T_D [\text{honest}(T_D) \wedge \text{feasible}(T_D) \Rightarrow$
 $(\exists t \text{ honest}(T_D \cup t) \wedge \text{feasible}(T_D \cup t))] \wedge$
buying agents are honest \wedge
selling agents can reliably meet their promised specifications \wedge
cost c is a constant percentage of selling price
 $\Rightarrow \forall T_A [\forall C \subseteq T_A, |C| > 0 \Rightarrow P(G, T_A, \emptyset) > P(G, T_A, C)]$

This yields:

[$\forall G \forall T_{D1}, T_{D2} [P(G, T_{D1}) > P(G, T_{D2}) \Rightarrow T_{D1}$ is selected] \wedge
 $\forall T_A [\forall C \subseteq T_A, |C| > 0 \Rightarrow P(G, T_A, \emptyset) > P(G, T_A, C)]$
 $\Rightarrow \forall T_A (T_A, \emptyset)$ is selected $\Rightarrow \forall t_D \in T_D \text{ honest}(t_D)$

Essentially this means that since it will not be profitable for a rational seller agent to cheat, they will execute every transaction honestly. Thus, the Trunits mechanism can provide the user with a guarantee of security, at this level.

The (labelled) list of assumptions given above identifies the limitations of the basic Trunits mechanism. Understanding these, how can we be sure that the mechanism will be secure? Rationality of agents (A) is a fundamental assumption of most work in mechanism design, and a limitation that we likely must accept; it is not an unreasonable expectation of sellers in a marketplace, however. Sellers acting alone (B) speaks to the issue of collusion, a difficult problem with which the trust and reputation community continues to struggle. Since it is unlikely that we can safely assume that agents won’t collude, we must devote effort to extending the mechanism to make it collusion-proof. The need for infinite transactions (C) can be addressed through enhancements to the Trunits mechanism; this is discussed in more detail in section 5. The requirement for agents to be honest (D) speaks directly to the absence of any system to address the trustworthiness of buyers. Extension of Trunits, or the use of a parallel system to ensure buyer honesty, is required to address this limitation. The assumption that sellers can control quality (E) can actually be refined to specify the degree of control required, as noted. Finally, the assumption that cost is a fixed percentage of selling price (F) is a special case of an internal limitation. This assumption is not likely required for the desired property to hold, but has been added to ease analysis. It might be eliminated with more detailed consideration.

The Trunits mechanism, as presented, provides security against reputation lag (since trunits must be placed in escrow during the course of a transaction, they cannot be used to cheat multiple buyers at once) and value imbalance (since the number of trunits earned in honest sales is proportional to the value of the sale), and some degree of protection against the exit problem (since there is an incentive for sellers to remain in the market). Note too that re-entry does not diminish the security of buyers (since there is no advantage for a seller who has lost his trunits to re-enter

under a different name), and that no initial window of opportunity exists (since buyers are not relying on their own experience to choose partners.) These arguments are outlined in greater detail in [7]. However, success against a large number of catalogued vulnerabilities is not enough for any system to be considered secure—our catalogue is almost certainly not exhaustive. In contrast, our analysis above gives us a clear picture of exactly what guarantee basic Trunits provides, and under what conditions—Trunits has achieved a specific provable level of security. The ability to do so appears to be linked to the fact that the incentive for honesty is explicit, and thus measurable. Based on this guarantee, informed decisions can be made about whether this mechanism is appropriate for a given scenario. Moreover, clear directions have been identified for future research, in overcoming these restrictions.

Discussion and Future Work

This paper has argued that it is necessary to explicitly consider security in the development of trust and reputation systems, and has proposed a framework for doing so. This approach is contrasted with that of current researchers in the area of trust for multiagent systems in electronic marketplaces, which focus predominantly on developing methods for predicting untrustworthiness.

Existing methods of modeling trust and reputation are subject to vulnerabilities; the gravity of the existence of such vulnerabilities should not be underestimated. To a rational agent who knows that a trust/reputation model is in use, the model constitutes part of the agent's environment as much as the operational rules of the market do. Just as we would expect this agent to maximize profit within the operational rules, we should expect he might do so with this expanded understanding of the environment—a profit maximizing strategy might very well involve exploiting a vulnerability.

To be considered secure, a system with potential vulnerabilities would need to demonstrate that it is secure against an exhaustive list of such vulnerabilities (which is difficult to obtain). Moreover, in the process of adjusting the system to remove vulnerabilities, the potential for introducing new vulnerabilities exists. In contrast, proving a system to be secure according to our framework enumerates a complete set of assumptions to be addressed.

It can be argued that when a predictive model is being used, if an agent is rational and knows that trustworthiness is being modeled, it will affect his decisions—he might be inclined to try to act honestly, to maximize future profit. In this case, a predictive model might also be viewed as a de facto incentive mechanism. This view makes our framework especially relevant for those working on predictive approaches.

While we argue that security is critical to the adoption of trust and reputation systems, we do not mean to suggest that other goals are unimportant. Indeed, improving predictive accuracy, maximizing social welfare, etc., are all worthwhile objectives. It is our contention, however, that a complementary consideration of security needs to be included as well.

We focused in this paper on characterizing security for the buyer, but to be complete a system should also offer security to sellers, and to the market operator. We present a very brief outline of seller and market security here, to be expanded in future work.

Seller security is more difficult to define than buyer security. Buyers wish to receive goods as promised, and the required protection can be specified within individual transactions. By comparison, the primary goal of sellers is profit, which can be attacked in a variety of indirect ways. The example of ballot-stuffing [1] illustrates this point, wherein a coalition artificially inflates the attacker's rating in order to steal sales from the victim. Such activity can certainly cause damage, but this damage is more difficult to isolate than an unfulfilled commitment, and hence safety properties are more difficult to formulate. This is complicated by the fact that a trust system cannot guarantee certain levels of profit or revenue, since these will be affected by quality of marketing, legitimate competitive activity, etc. We suggest that for a system to be secure for the seller, the dishonest activity of any attacker should not reduce the seller's revenue (from agents other than the attacker(s)). Other approaches will also be considered.

To protect the market operator, or to ensure the continuing operation of the market where no operator can be identified (e.g., in some peer-to-peer systems), a proposed set of security requirements might consist of the following. Dishonest activity should not cause costs to be incurred by the 'market', because violation of this property would allow attacks to render the 'market' insolvent. Operation of the trust system should be budget balanced, or profitable. Finally, dishonest activity by participants should not cause the market to fail.

Our analysis revealed that Trunits is buyer secure, under specific conditions. Future work will pursue both the removal of some of these conditions through enhancements to the system, and providing protection for other market participants. First, since Trunits regulates only the behaviour of sellers, it does not provide seller security; further, it depends on the honesty of buyers to deliver buyer security, rather than providing a means to ensure buyers are honest. Trunits might incorporate protection from dishonest buyers directly. Alternatively, it is important to note that the model does not preclude the parallel use of another system for this purpose. For example, a mechanism like that of Jurca and Faltings [6] might be considered.

Trunits provides weak protection against the exit problem, by providing an incentive for sellers to stay in the market. This protection requires the assumption of unlimited future transactions for sellers, however, which is unrealistic. In our current work we are studying the treatment of trust as a tradable commodity (i.e., allowing agents to buy and sell trust). This notion, while counterintuitive, seems to allow us to cope with the exit problem without sacrificing the other beneficial properties of the model. Under this system, selling unneeded trunits can be made more profitable than using them to cheat, providing a strong incentive for honesty whether the seller exits the market or not. This approach also provides a natural solution to the ‘start-up problem’ noted above, since agents can purchase initial quantities of trunits.

In summary, the security framework presented in this paper offers a new direction for researchers in the area of trust and reputation to promote confidence in their models for real users. The Trunits model also provides a promising direction for designing electronic marketplaces in a way that offers guarantees of security to buyers.

References

- [1] Bhattacharjee, R. and Goel, A. 2005. Avoiding ballot stuffing in eBay-like reputation systems. In *Proceeding of the 2005 ACM SIGCOMM Workshop on Economics of Peer-To-Peer Systems* (Philadelphia, Pennsylvania, USA, August 22 - 22, 2005). P2PECON '05. ACM Press, New York, NY, 133-137.
- [2] Braynov, S. and Sandholm, T. 2002. Incentive compatible mechanism for trust revelation. In *Proceedings of the First international Joint Conference on Autonomous Agents and Multiagent Systems: Part 1* (Bologna, Italy, July 15 - 19, 2002). AAMAS '02. ACM Press, New York, NY, 310-311.
- [3] Dellarocas, C. 2002. Goodwill Hunting: An Economically Efficient Online Feedback Mechanism for Environments with Variable Product Quality, *Lecture Notes in Computer Science*, Volume 2531, 238 – 252.
- [4] eBay. <http://pages.ebay.com/help/feedback>
- [5] Griffiths, N. 2005. Task delegation using experience-based multi-dimensional trust. In *Proceedings of the Fourth international Joint Conference on Autonomous Agents and Multiagent Systems* (The Netherlands, July 25 - 29, 2005). AAMAS '05. ACM Press, New York, NY, 489-496.
- [6] Jurca, R. and Faltings, B. 2003. An incentive compatible reputation mechanism. In *Proceedings of the Second international Joint Conference on Autonomous Agents and Multiagent Systems* (Melbourne, Australia, July 14 - 18, 2003). AAMAS '03. ACM Press, New York, NY, 1026-1027.
- [7] Kerr, R. and Cohen, R. 2006. Modeling Trust Using Transactional, Numerical Units. In *Proceedings of the Conference on Privacy, Security and Trust* (Markham, Ontario, Canada). PST '06.
- [8] Marsh, S. *Formalising Trust as a Computational Concept*. PhD thesis. University of Stirling, 1994.
- [9] Sabater, J. and Sierra, C. 2001. REGRET: reputation in gregarious societies. In *Proceedings of the Fifth international Conference on Autonomous Agents* (Montreal, Quebec, Canada). AGENTS '01. ACM Press, New York, NY, 194-195.
- [10] Sabater, J. and Sierra, C. 2005. Review on Computational Trust and Reputation Models. *Artif. Intell. Rev.* 24, 1 (Sep. 2005), 33-60.
- [11] Tran, T. and Cohen, R. 2002. A Learning Algorithm for Buying and Selling Agents in Electronic Marketplaces. In *Proceedings of the 15th Conference of the Canadian Society For Computational Studies of intelligence on Advances in Artificial intelligence* (May 27 - 29, 2002). R. Cohen and B. Spencer, Eds. Lecture Notes In Computer Science, vol. 2338. Springer-Verlag, London, 31-43.
- [12] Tran, T. and Cohen, R. 2004. Improving User Satisfaction in Agent-Based Electronic Marketplaces by Reputation Modelling and Adjustable Product Quality. In *Proceedings of the Third international Joint Conference on Autonomous Agents and Multiagent Systems - Volume 2* (New York, New York, July 19 - 23, 2004). International Conference on Autonomous Agents. IEEE Computer Society, Washington, DC, 828-835.
- [13] Yu, B. and Singh, M.P. 2002. Distributed Reputation Management for Electronic Commerce. In *Computational Intelligence*, 18(4): 535-549, 2002.
- [14] Zacharia, G., Moukas, A., and Maes, P. 1999. Collaborative Reputation Mechanisms in Electronic Marketplaces. In *Proceedings of the Thirty-Second Annual Hawaii international Conference on System Sciences - Volume 8 - Volume 8* (January 05 - 08, 1999). HICSS. IEEE Computer Society, Washington, DC, 8026.