Toward Secure Trust and Reputation Systems for Electronic Marketplaces

by

Reid C. Kerr

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2007

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Reid C. Kerr

Abstract

In electronic marketplaces, buying and selling agents may be used to represent buyers and sellers respectively. When these marketplaces are large, repeated transactions between traders may be rare. This makes it difficult for buying agents to judge the reliability of selling agents, discouraging participation in the market. A variety of trust and reputation systems have been proposed to help traders to find trustworthy partners. Unfortunately, as our investigations reveal, there are a number of common vulnerabilities present in such models—security problems that may be exploited by 'attackers' to cheat without detection/repercussions. Inspired by these findings, we set out to develop a model of trust with more robust security properties than existing proposals.

Our Trunits model represents a fundamental re-conception of the notion of trust. Instead of viewing trust as a measure of predictability, Trunits considers trust to be a quality that one possesses. Trust is represented using abstract trust units, or 'trunits', in much the same way that money represents quantities of value. Trunits flow in the course of transactions (again, similar to money); a trader's trunit balance determines if he is trustworthy for a given transaction. Faithful execution of a transaction results in a larger trunit balance, permitting the trader to engage in more transactions in the future—a built-in economic incentive for honesty. We present two mechanisms (sets of rules that govern the operation of the marketplace) based on this model: Basic Trunits, and an extension known as Commodity Trunits, in which trunits may be bought and sold.

Seeking to precisely characterize the protection provided to market participants by our models, we develop a framework for security analysis of trust and reputation systems. Inspired by work in cryptography, our framework allows security guarantees to be developed for trust/reputation models—provable claims of the degree of protection provided, and the conditions under which such protection holds. We focus in particular on characterizing *buyer security*: the properties that must hold for buyers to feel secure from cheating sellers. Beyond developing security guarantees, this framework is an important research tool, helping to highlight limitations and deficiencies in models so that they may be targeted for future investigation. Application of this framework to Basic Trunits and Commodity Trunits reveals that both are able to deliver provable security to buyers.

Acknowledgements

This work is dedicated to my wife, Victoria Larke, who has enriched my life beyond measure or description. I would not be the person that I am today without her, and this work would never have been possible without her support.

I owe deep gratitude to my supervisor, Robin Cohen. I consider myself incredibly fortunate to have benefited from Professor Cohen's experience and extraordinary talents as a coach, mentor, and researcher. Her dedication to her students goes far beyond the call of duty, and far beyond my expectations.

I would like to thank Kate Larson. In addition to her expertise, her advice and encouragement has been greatly appreciated. I also wish to thank Peter Forsythe for sharing his experience and providing valuable insights.

Life in any institutional environment entails coping with administrative issues and procedures. Margaret Towell and Wendy Rush were always there to help, and working with them was a true pleasure.

While it is not possible to name everyone, there are a number of people that particularly enriched my graduate experience, both within and beyond academics: Laurent Charlin, Matthew Enss, Greg Hines, Fred Kroon, Kevin Regan, Tyrel Russell, Martin Talbot, Rob Warren, John Whissell, Dana Wilkinson, and Jie Zhang.

My accomplishments to date, including this work, are built upon the foundation of unwavering love and support provided by my parents, Boyd and Laura Kerr. In addition, John and Lorraine Larke prove that the stereotypes of in-laws do not always hold true: I deeply appreciate their caring, support and generosity of spirit.

Finally, I would like to acknowledge the generous support of the Natural Sciences and Engineering Research Council of Canada (NSERC).

Contents

List of Figures

Chapter 1

Introduction

In the field of artificial intelligence, the term *agent* has come to represent a software program acting on behalf of a user. An agent may act autonomously and pro-actively to determine appropriate actions for a user, according to that user's perceived preferences [33].

Multiagent systems consist of multiple intelligent software agents acting independently [33]. In co-operative multiagent system environments, each agent tries to select the most effective partners in order for them to collectively achieve a task on behalf of a user. Modeling the trust and reputation of each agent then becomes important.

In other environments, self-interested agents acting on behalf of users compete with other agents for existing resources. At times, these agents must engage in activities that involve other agents. Under these circumstances, an agent's success may also depend on its ability to wisely choose partners; a bad choice might have grave implications. For example, a partner might actively seek to take advantage of the agent in pursuit of its own goals. As such, the topic of modeling trust and reputation has begun to receive significant attention within the context of competitive multiagent systems.

The design of electronic marketplaces is one area where multiagent systems approaches have become prevalent. Here, trading agents (acting on behalf of buyers and sellers) evaluate other trading agents, seeking to choose the most appropriate trading partners for their users. The issue of trust is especially important in these marketplace scenarios, where transactions are essentially economic in nature, and where an agent may benefit directly from the act of cheating a trading partner. One might argue that issues of trust are magnified in electronic marketplaces, compared to conventional ones, because in electronic marketplaces traders often do not have the opportunity to inspect goods or to confirm identities prior to committing to a purchase. Typically, electronic marketplaces are also very large, making repeated transactions between pairs of traders infrequent; this, coupled with the remote execution of transactions, inhibits the formation of trusting relationships.

As we explore how best to model the trustworthiness of agents in electronic marketplaces, we begin by looking for insights from several researchers designing trust and reputation models for multiagent systems. For the remainder of the thesis, we use the term *trust* to reflect an agent's view of another agent's degree of trustworthiness. Trustworthiness may be represented as a value in a numeric range (e.g., [0, 1] as in [12], or (-1, 1) as in [31]) with values at the top of the range reflecting a high degree of trust, and values near the bottom, distrust. *Reputation* commonly refers to the trustworthiness of an agent as perceived by a community of agents, and is also typically represented by values in such a numeric range (e.g., [26]). In the discussion that follows, we acknowledge that an agent's view of another's trustworthiness may be determined by direct observation or by reports from other agents, and in both cases we simply refer to this as trust.

Predominant amongst current proposals for modeling trust is an approach we characterize as 'predictive'. Essentially, a predictive model attempts to determine the likelihood that any given agent will be trustworthy, or that he will cheat. While there are many such proposals, there are two main themes. In the first, sophisticated learning techniques are employed to estimate a potential partner's trustworthiness; as experience is gained with partners, the agent learns over time which partners are likely to be honest. In the second, recommendations are sought from other agents in order to evaluate the trustworthiness of a potential partner; the agent must evaluate the credibility of the recommendations, and integrate them into a single estimate. In both cases, ongoing work attempts to increase the predictive accuracy of models, to improve the chances for agents to achieve their goals.

While this work constitutes substantial progress, it leaves important issues largely unaddressed. In our view, the most important omission is explicit consideration of system security. The goal of a predictive model is to predict the actions of an agent; if strategies are available to an agent that defy the predictive power of the system, then we might consider such a system to be insecure. More generally, if there are cheating strategies that a model is helpless to prevent, then that model might be considered insecure. Our work began with an investigation of existing proposals for modeling trust, seeking to determine if vulnerabilities exist within them. In fact, such vulnerabilities are common—every model in our survey suffered from multiple serious weaknesses. We present a catalogue of vulnerabilities in Chapter 2, and an examination of existing proposals in Chapter 3. Ultimately, while these models may be accurate when agents are not actively attempting to circumvent them, in the face of educated attacks the protection fails.

Inspired by these findings, we sought to develop a model that would be more robust in the face of such attacks. Our findings armed us with a catalogue of known vulnerabilities, and more importantly, recommended techniques to counter these problems. Moreover, ongoing work on predictive models appeared to be making incremental improvements over older systems, without recognizing or addressing the fundamental security issues. This motivated us to consider the problem from a fresh perspective, one informed by work in an area known as *mechanism design* [19]. Mechanism design, a field closely related to game theory, concerns itself with setting up the rules of a 'game' so that self-interested players will exhibit the behaviour that the system designer desires—the rules make the desired behaviour the best self-serving behaviour for players as well. Mechanism design depends on the *rationality* of agents. A rational agent is one that "acts in its own best interests" [32] (or when acting on behalf of a user, in that user's interests). The notion of 'best interests' typically is specified as a set of preferences; the agent acts so as to achieve the best outcome based on these preferences [32].

If we consider the primary goal of a trust system for electronic marketplaces to be the protection of the interests of market participants, rather than the prediction of behaviour, then non-predictive approaches may offer that fresh perspective. In the spirit of mechanism design, we endeavour to structure the marketplace to provide the required protection to participants, by providing financial incentives for honesty. If agents are rational and self-interested, then we might ensure trustworthiness by making honesty the most profitable (and hence, best self-serving) strategy.

Our model, known as ‘Trunits’, is described in Chapter 4. Trunits is a fundamental re-conception of the nature of trust. Rather than trust being seen as a measure of predictability, it is viewed as a quality one possesses, which is gained over time through honest behaviour. In the Trunits model, abstract units are used to represent trust in much the same way that units of money represent value. In a manner similar to money, *trunits* flow during transactions. A trader’s trunit balance determines if she is trustworthy enough to engage in a given transaction. Faithful execution of a transaction results in a larger trunit balance, permitting the trader to engage in more transactions in the future—a built-in economic incentive for honesty. Dishonest behaviour results in lost trunits (i.e., lost trust), which both punishes for the dishonest action, and protects other agents by inhibiting the cheater’s future transactions. While Trunits is a model of trust, it has characteristics of a mechanism, since its rules align the goals of agents with the desired behaviour—honesty. Based on this model, we introduce two specific mechanisms. Basic Trunits, described in Chapter 4, is a direct implementation of the Trunits model. In Chapter 5, we introduce an extension to Basic Trunits known as Commodity Trunits, in which units of trust may also be bought and sold by traders, and we show that this feature provides enhanced security.

Trunits has a number of important strengths; we list a few here. First, it directly addresses many of the important vulnerabilities common in existing models, because the rules of the system make these forms of cheating less attractive (i.e., less profitable) than honesty. Second, because protection is provided by the structure of the market, every buyer is equally protected. This is in contrast to predictive models, where protection may be limited by individual agents’ information and capacity for prediction. Third, in one important way Trunits is ‘fairer’ than existing models, in that every seller with the required number of trunits is considered trustworthy for a given transaction—recent entrants are not discriminated against in favour of established sellers. This is in contrast to certain predictive models, in which the most trustworthy seller is selected and new entrants may begin with a very low trust rating. Fourth, Trunits has extremely low computational and storage requirements. As will be shown, long transaction histories need not be stored under Trunits, and a limited number of simple mathematical operations are required after each transaction. Finally, Trunits provides an interesting possibility:

marketplaces in which buyers and sellers can be anonymous to one another, but can still trust one another.

A list of strengths, however, does not directly address our overriding concern: security. While Trunits combats many of the vulnerabilities common in existing systems, other unknown weaknesses might be present, perhaps even introduced by our very efforts to combat vulnerabilities. This raises a key question: how can one be certain of a system's security? In our view, if 'security' requires the absence of vulnerabilities, it is not sufficient simply to show that a system does not suffer from any known vulnerabilities—unless an exhaustive list of possible vulnerabilities were to be developed, one could never trust in the security of the system. Instead, we borrow ideas from two fields as the basis for our notion of security. First, from cryptography we adopt the view that for a system to be considered secure, it should be provably secure; further, if a system cannot be shown to be absolutely secure, we might specify and prove the precise degree of security it does provide [20]. Second, from the field of formal methods we make use of the idea of *safety properties*—to verify that our system will not allow harm to occur, we formally define a system specification consisting of a set of properties that must hold for participants to be 'safe' [25].

Ultimately, this is the key goal: to assure users that the trust system itself is trustworthy, so that they can feel safe in adopting or participating in the system. To that end, we have developed a framework for evaluating the security provided by trust and reputation systems, presented in Chapter 6. The application of this framework results in a precise characterization of the security properties of a model; in particular, it enumerates a set of conditions that must hold in a marketplace for the system to guarantee the security of participants. This set of conditions is an important research tool for the designers of trust and reputation models in multiagent systems. Application of this framework to both Trunits mechanisms reveals that they do offer provable protection under specific conditions, and highlights areas for further development.

Chapter 2

Motivation

Our consideration of trust and reputation models begins with an examination of the system used by eBay. Using eBay as a example, we are able to identify a number of important vulnerabilities in trust models—weaknesses that might allow unscrupulous participants to damage others and/or to profit unfairly at others' expense. In Chapter 3, we examine a number of existing proposals, identifying multiple vulnerabilities present in each. The presence of these vulnerabilities, i.e., the insecurity of the models, is a key motivation for our own work.

2.1 The eBay Reputation System

While eBay's reputation system [10] is not itself derived from (published) research, it has been widely discussed and deeply studied in current research in the field. eBay typifies the type of 'new market' with which many investigators are concerned, and has demonstrated considerable success in implementing a reputation system that allows traders to have confidence in one another. A discussion of eBay's system is a valuable starting point, for two main reasons:

1. It provides a convenient framework within which to discuss the issues faced by trust/reputation systems for marketplaces, with real world context, and

2. It provides a benchmark against which other systems might be measured. Given

that eBay is an established, implemented system, any proposal that specifically targets markets of this type must improve upon eBay's system in some way to be interesting. (It should be noted that 'improve upon' does not mean only that a system must outperform eBay in terms of effectiveness; a system might be important in other ways, e.g., by being applicable outside the marketplace scenario, by being implementable in intelligent agents, etc.)

The eBay system is simple in operation, compared to many proposed models. In the eBay marketplace, sellers advertise goods that they wish to sell. Buyers seeking to purchase goods search through those offered by sellers, choosing goods (and sellers) that they believe will meet their needs. Before engaging in a transaction, both buyer and seller have the opportunity to view each other's 'feedback profiles' (discussed below). After a transaction has been completed, both buyer and seller have the opportunity to give feedback on one another's performance, although they are not obliged to do so. Giving feedback consists of choosing a rating of 'positive', 'neutral', or 'negative', for the trading partner, and optionally leaving a single line, free-form text comment. Once feedback has been given, it is added to the rated user's feedback profile, and is visible to all future potential trading partners.

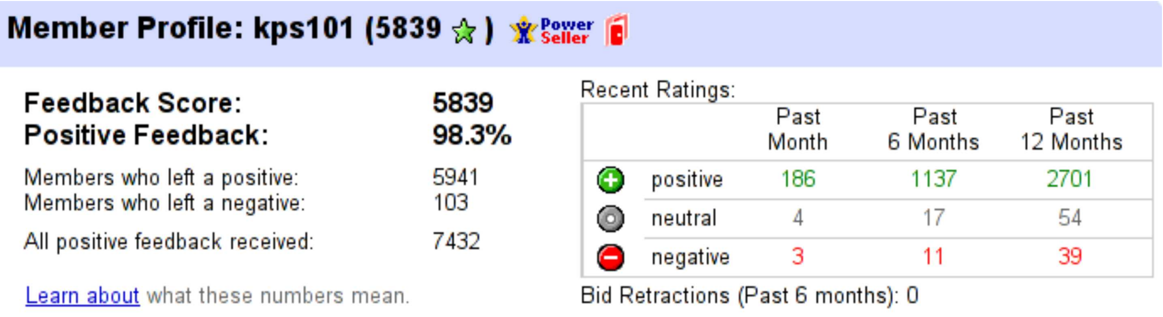
A view of a feedback profile is shown in Figure 2.1. The profile presents two key pieces of information about the user:

1. The feedback score, which is calculated as the number of unique users giving positive ratings, less the number of unique users giving negative ratings. This score seems intended to give a rough idea of the volume of trustworthy transactions in which the user has engaged.

2. The positive feedback percentage, calculated as the number of unique users giving positive ratings, divided by the total number of unique users providing ratings. This score seems intended to give a rough idea of the likelihood that the user will be trustworthy in a given transaction.

Based on this information, the user may decide if she considers the potential partner trustworthy enough to trade with him. For example, if a seller has too many negative ratings in his profile, a buyer may decide that he cannot trust in that seller's reliability.

If a seller has too few ratings in his profile, a buyer may decide that there is not enough evidence of the seller's reliability to trust in him.



Member Profile: kps101 (5839 ☆)

Feedback Score:	**5839**
Positive Feedback:	**98.3%**
Members who left a positive:	5941
Members who left a negative:	103
All positive feedback received:	7432

Learn about what these numbers mean.

Recent Ratings:

		Past Month	Past 6 Months	Past 12 Months
●	positive	186	1137	2701
○	neutral	4	17	54
●	negative	3	11	39

Bid Retractions (Past 6 months): 0

Figure 2.1: A feedback profile

While the system is quite simple in operation, there is no question that eBay has achieved a substantial degree of success in convincing traders that it is adequate for them to participate in the market. Superficially, the mechanism is extremely successful in ensuring the satisfaction of traders, with 99.1% of feedback being positive [8]. A deeper examination, however, suggests that the system may not be as effective as it first appears.

2.1.1 Possible problems with the eBay System

While feedback scores indicate a high degree of trader satisfaction, there is reason to believe that the data does not accurately reflect real satisfaction levels. In [2], it is noted that in systems such as eBay's, an agent who receives a bad rating may retaliate by likewise leaving (undeserved) bad feedback for the trading partner. Fear of receiving such a retaliatory rating discourages users from leaving negative feedback, artificially inflating rankings. A study cited in [35] found that there is a high correlation between the ratings of buyers and sellers in transactions, implying that feedback may say more about the smoothness of the bilateral execution of the transaction, than about the reliability of a single participant. The authors of [24] find that there is often pressure on traders to give positive ratings, causing traders to be more lenient in their feedback. (Having personally experienced this, we can anecdotally testify that such pressure exists.)

Aside from the validity of rankings, there are also questions about whether users can effectively and accurately make use of the ranking data. One problem, discussed in [8], is that users are required to interpret feedback profiles themselves, without any context or information about the users who have given this feedback. Reputation impacts both the likelihood and price of sale, but not consistently so, indicating that individual buyers interpret the reputation data very differently. Thus, it is hard to measure the value of reputation, and hard for users to make decisions based on such value.

2.1.2 Vulnerabilities in the eBay System

Beyond these more abstract problems, there are specific vulnerabilities within the eBay reputation system that dishonest sellers might use to take advantage of buyers. That such vulnerabilities exist should not be surprising, even given the high levels of satisfaction noted above. According to Battacharjee and Goel [2], studies have shown a substantial amount of fraud in electronic marketplaces; if trust/reputation systems were perfect, there would be no such fraud.

Several key vulnerabilities are identified and illustrated below. (Most of the names assigned to the problems are our own, and are not yet standard terminology.)

The *Reputation Lag* problem

In many marketplaces, buyers are required to pay for goods before the seller delivers them. After payment, there is usually some delay before the buyer actually receives the good (due to processing, shipping, etc.), and has the opportunity to evaluate the transaction and/or give feedback. For example, on eBay the lag between payment by the buyer and registering the buyer's feedback is typically one to three weeks.

This lag opens a window of opportunity for a seller to engage in unlimited cheating. Consider a seller who decides to cheat a buyer on a transaction; the situation is depicted in Figure 2.2. The seller may know that he intends to cheat from the time of sale, but the buyer will not know until some time later (after he receives an inferior good, or gives up waiting for a good that never arrives). Due to the lag in the buyer's negative feedback being posted, the seller's reputation will not deteriorate immediately, and other buyers will not be alerted to the cheater's dishonest behaviour until some time after it begins—for the duration of the lag, only the cheater knows that he is cheating. Thus, for the entire period of the lag, the seller can make use of his good reputation to cheat a virtually unlimited number of buyers.

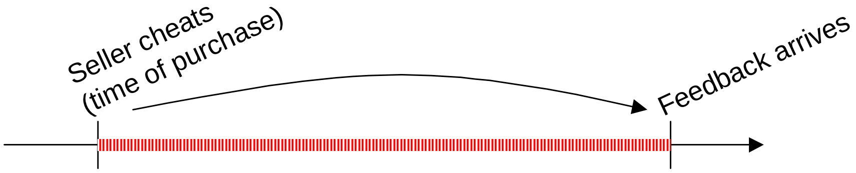


Figure 2.2: The Reputation lag problem

The *Value Imbalance* problem

In [7], Dellarocas identifies the value imbalance vulnerability. This vulnerability stems from the fact that in many systems (including eBay), the impact of each piece of feedback is not related to the value of the transaction—feedback on very small transactions is weighted equally with feedback from very large transactions. A dishonest seller can take advantage of this property to build up reputation by honestly executing a number of small-value trades, and then using the accumulated reputation to cheat a seller in a very high-value transaction.

For example, if the seller were to engage in 5 honest transactions, each of $1 in value, his feedback score would be 5, with a feedback percentage of 100%. To potential buyers, he appears to be quite honest. Using this honest appearance, he might lure a buyer into making a $1,000 purchase, and cheat by not delivering the product. In this example, the seller was able to cheat a buyer out of $1,000, based on the trust earned in $5 worth of transactions. Worse still, after receiving the negative feedback, the seller would still have a feedback score of 4, and feedback percentage of 83.3%—his reputation is far from being irreparably damaged.

The *Ballot-Stuffing* problem

Trust/reputation has value—good reputation is likely to result in greater sales volume, and may allow higher prices to be charged. Further, good reputation can be used to lure buyers into transactions in which they can be cheated.

Many authors ([2, 8, 36]) identify the opportunity for buyers to engage in 'ballot-stuffing'. Ballot-stuffing is collusive behaviour in which parties engage in fake transactions in order to artificially inflate their reputations.

Some systems (such as eBay) have attempted to limit the ability of users to ballot-stuff by only counting feedback from unique users. However, the vulnerability remains, due to the ease with which new identities can be created; it is trivial to create many new accounts, allowing ballot-stuffing transactions to be generated by 'unique users'.

The *Bad-Mouthing* problem

Closely related to ballot-stuffing is a problem known as 'bad mouthing'. Here, agents give unfairly negative reviews of the victim, in an effort to damage the victim's reputation. This is often an attack launched against a seller—if a competitor's reputation is damaged, the attackers may be able to win additional sales.

The *Re-entry* problem

The ease of creation of new accounts introduces another vulnerability, identified by numerous authors ([2, 8, 36]). In many systems (such as eBay), it is preferable to have no reputation rather than a bad reputation; while users may be hesitant to deal with a trader with no history, they are more hesitant to deal with a disreputable user. If a seller has engaged in a significant amount of cheating, he will develop a bad reputation, which should warn users not to deal with him. However, he can start fresh by simply creating a new account, freeing himself of the restrictive reputation. In essence, the seller has improved his reputation without engaging in even a single honest transaction. The seller can repeat this cycle as many times as desired, cheating until his reputation is destroyed, and then beginning again with a new identity.

The *Exit* problem

The exit problem occurs when a seller who has gained some positive reputation decides to leave the market. The accumulated reputation can induce buyers to engage in transactions with the seller. The seller has no further need of her good reputation, since she does not plan to make further sales in the market. Thus, the entire accumulated reputation can be used to cheat buyers, until it deteriorates to the point that no buyer will deal with the seller; at this point, the seller completes her exit. This is an extremely difficult problem to address, and one that afflicts most trust/reputation systems (including eBay).

2.1.3 A catalogue of vulnerabilities

The problems outlined in this chapter may be seen as a catalogue of known vulnerabilities in trust and reputation systems. We make no claim, however, that it is an *exhaustive* catalogue. In fact, it is very likely that other potential attacks, and potential vulnerabilities, exist. Thus, it is important to understand the role of this catalogue. It may be useful in establishing the insecurity of a model (by citing vulnerabilities present in it), and in informing researchers as they seek to develop more secure methods. It cannot be used to declare a model secure, on the basis that it does not suffer from the known vulnerabilities.

In the next chapter, we consider a number of existing proposals, using this catalogue to evaluate the security of each.

Chapter 3

Related Work

As discussed in Chapter 2, several important vulnerabilities exist in the eBay reputation system. Such vulnerabilities are of great importance to a trust/reputation system: they can allow an attacker to cheat without repercussions. Specifically, weaknesses of this nature can allow cheating actions that are not predictable or preventable by the system, or that have no negative consequences for the cheater.

Ideally, a ‘secure’ trust/reputation system would prevent each of these problems, without introducing substantial new ones. Unfortunately, as our survey below will reveal, these vulnerabilities also affect many of the other trust/reputation systems that have been proposed. In our examination of related work, we use the vulnerabilities catalogued in Section 2.1.2 as important criteria with which to appraise and compare proposals. In later chapters, we use this same catalogue to evaluate our own proposed mechanisms.

An enormous amount of work has been conducted in this field, and it is well beyond the scope of this thesis to examine all of it. Instead, we examine a sample of systems that we believe to be representative of the breadth of work in the field.

3.1 Predictive models

First, we examine models that we consider to be primarily predictive in nature. These systems attempt to help agents to find better partners by predicting the behaviour of potential partners. Candidates that are seen as likely to be dishonest can be avoided by the agent, while those that have a high likelihood of being honest can be favoured. Predictive models employ a variety of techniques; we discuss a number of such models in this section.

In the survey of trust/reputation systems conducted in [27], the authors characterize systems in a number of dimensions. We use one of these dimensions, that of 'information sources', to group systems for convenience of analysis.

3.1.1 Direct Experience models

Direct experience models are those in which an agent evaluates the trustworthiness of a potential partner based on the agent's own experience with that partner. Models frequently make use of multiple information sources, but here we consider only models that exclusively use direct information.

General/Situational/Multi-Dimensional Trust

So-called 'general trust' models like those described by Marsh [18] and Griffiths [12] can be applied to marketplaces—Griffiths specifically discusses this scenario in his paper. Under these models, an agent's trust of another is based solely on its own experience with the other agent.

Marsh's work represents some of the earliest in the formal consideration of trust in computational scenarios. In his model, agent x's Basic Trust (T_x) constitutes x's *disposition*, her general tendency to trust others. General Trust refers to the agent's degree of trust of other specific individuals, but outside any specific consideration of situation: agent x's trust of agent y is denoted $T_x(y)$. The author also notes that trust may be situation specific; I may trust you to pass along a message to another, for example, but I might not trust you with my money. He thus defines Situational Trust: $T_x(y, \alpha)$ denotes

x's trust of y in situation α. All three types of trust have values in the range [-1, 1), with 1 representing maximum trust, -1 representing maximum distrust, and 0 representing 'no trust', or unknown trustworthiness. An agent will choose to cooperate with another if the appropriate trust value exceeds a certain threshold.

Griffiths' Multi-Dimensional Trust builds on Marsh's General Trust. In this model, the view is taken that overall trust may have different aspects, each of which may be considered separately. For example, I may have a high degree of trust that you will deliver a product on time, but a low degree of trust that the product will work properly. Thus, trust is decomposed into a number of dimensions. In each dimension d, trust of agent α (T_α^d) falls in the range [0, 1], with 0 representing complete distrust and 1 complete trust. To evaluate the trustworthiness of agent α in a particular situation, the evaluating agent assigns a weight u_i to each dimension i reflecting its relative importance. A *performance value* for the situation can then be calculated by taking a weighted product of the trust value in each dimension (f_{α_i}):

$$PV(\alpha) = \prod_{i=1}^{n} (f_{\alpha_i})^{u_i} \tag{3.1}$$

To choose between potential partners, the agent chooses the one with the highest performance value. After the transaction is complete, the agent updates his trust values for α, in each dimension d, using the following formulas:

$$\begin{aligned} update_{success}(T_\alpha^d) &= T_\alpha^d + ((1 - T_\alpha^d) \times (\omega_s \times T_\alpha^d)) & (3.2)\\ update_{failure}(T_\alpha^d) &= T_\alpha^d - ((1 - T_\alpha^d) \times (\omega_f \times T_\alpha^d)) & (3.3)\end{aligned}$$

where ω_s and ω_f are weighting factors for success and failure respectively, specifying the agent's disposition.

Direct experience models like these have limited applicability to a market of size comparable to eBay, since they rely on direct experience with agents. Because of the immense size of the market, a buyer would not be able to gain experience with more than a tiny fraction of the sellers in the marketplace. Further, the high frequency of one-time transactions (which occur because any given seller's offered goods are not likely to match a buyer's needs repeatedly) compounds the difficulty in gaining experience with sellers.

A major disadvantage of these models is that a buyer must learn which sellers to trust and distrust. This means that buyers are vulnerable to unscrupulous sellers until they have learned to avoid those sellers, providing the dishonest sellers with an *initial window of opportunity* to cheat. The models address this problem to a degree by favouring known reputable sellers, but this is only possible when such a seller is offering the needed good at the moment of purchase—this occurs infrequently in large markets with a diverse range of goods, such as eBay.

With reference to the problems identified in Section 2.1.2, these systems tend to be resistant to the reputation lag problem (since they rely only on direct experience, not on receiving the (delayed) feedback of others) and the ballot-stuffing/bad-mouthing problems (for the same reason). However, in the form presented, they are vulnerable to the value imbalance problem (since the value of the purchase is not taken into account when considering the trustworthiness of the user), the re-entry problem (since there is an incentive for disreputable users to re-enter the market, because unknown sellers are treated differently than disreputable ones), and the exit problem. Further, the re-entry problem can combine with the initial window of opportunity in a dangerous way: a seller may re-enter the market each time that sellers learn of his dishonesty, in order to take advantage of the initial window repeatedly.

Tran and Cohen

In the work of Tran and Cohen [30, 31], both buyers and sellers are learning agents—buyers learn to avoid purchasing low quality goods, while sellers learn to maximize profit. In this model, buyers and sellers make decisions based only on their own experiences.

In this system, each buyer b maintains ratings $r^b(s)$ for each seller s; ratings fall in the range (-1, 1), with a beginning value of 0. Each buyer b also maintains an *expected value function* $f^b(g, p, s)$ for each seller s, the expected value that the buyer will derive from buying good g at price p from the seller.

A buyer maintains sets of known reputable sellers (i.e., those with ratings above a *reputable threshold* Θ) and known disreputable sellers (those with ratings below a *disreputable threshold* θ). Sellers who fall into neither category are considered to be at least non-disreputable. A buyer first attempts to choose from the reputable set the seller who provides the maximum expected value. If there is no reputable seller offering the good, the buyer chooses a non-disreputable seller who again provides the maximum expected value for the good. In addition, with a small probability the buyer chooses to *explore* (rather than *exploit*) the marketplace by randomly choosing a non-disreputable seller. This allows a buyer to discover new sellers.

After a purchase, the buyer adapts using reinforcement learning, updating both the reputation rating and the expected value for the seller. Expected value is updated using

$$f^b(g,p,s) \leftarrow f^b(g,p,s) + \alpha\Delta \tag{3.4}$$

where Δ is the difference between the actual value received and the expected value, and α is the learning rate. If the seller provided a good of at least the value demanded by the buyer, his reputation is updated using:

$$r^b(s) \leftarrow \begin{cases} r^b(s) + \mu(1 - r^b(s)) & \text{if } r^b(s) \geqslant 0, \\ r^b(s) + \mu(1 + r^b(s)) & \text{if } r^b(s) < 0 \end{cases} \tag{3.5}$$

where $\mu > 0$ is called the *cooperation factor*. If the value provided is less than that demanded by the buyer,

$$r^b(s) \leftarrow \begin{cases} r^b(s) + \nu(1 - r^b(s)) & \text{if } r^b(s) \geqslant 0, \\ r^b(s) + \nu(1 + r^b(s)) & \text{if } r^b(s) < 0 \end{cases} \tag{3.6}$$

where $\nu < 0$ is called the *non-cooperation factor*. The values for μ and ν can be varied so as to give greater weight to transactions of higher value, providing protection again value imbalance. Additionally, these parameters can be set so as to make trust easier to lose than it is to earn.

In this system, sellers also employ reinforcement learning; the learning simply focuses on how to maximize profit by choosing the parameters of the goods offered.

As a direct experience model, the work of Tran and Cohen shares many characteristics with the Marsh/Griffiths models discussed above. Again, this model is of limited applicability to very large markets, if repeated transactions between traders are rare, due to the lack of direct experience gained with individual sellers. Further, although the system explicitly favours known sellers, it is still subject to the initial window of opportunity when buyers use sellers with whom they have no experience.

Due to its reliance on directly acquired information, this system is resistant to the reputation lag, ballot-stuffing, and bad-mouthing problems. Further, by considering the value provided by the seller, the model provides protection against value imbalance. The model is vulnerable, however, to the re-entry problem (since unknown and disreputable sellers are treated differently) and the exit problem. Further, as with the previously-discussed models, this system suffers from the potentially dangerous combination of re-entry with the initial window problem.

3.1.2 Witness Information models

Witness information models are those in which the agent makes use of information supplied by other agents (instead of, or in addition to direct experience) in evaluating the trustworthiness of a potential partner.

Sporas and Histos

In [36], Zacharia et al. introduce two systems, Sporas and Histos.

Under the Sporas system, each agent has a single, global reputation score, which is updated after each transaction. According to [27], Sporas is an ‘evolved version’ of an eBay-like system, making use of a related mechanism. However, it incorporates many interesting features:

- In order to eliminate the re-entry problem, new users do not begin with better reputations than disreputable users.

- If an agent rates another more than once, only the most recent rating is considered, to minimize the impact of ballot stuffing.

- To combat the creation of new accounts for the purpose of ballot stuffing, more weight is given to ratings from users with established reputations.

- A ‘memory’ factor is used to de-emphasize older ratings.

The Histos system allows for personalized ratings by making use of a ‘web of trust’ model. A digraph is maintained in which nodes represent agents and edges represent ratings. When an agent seeks information about another, unknown agent, all paths (of less than a specified length) from the first agent to the second are found; reputation scores are determined by composing the scores along each of these paths, reflecting the trustworthiness of recommending agents along the path. Histos relies on a large number of rankings, since paths must be found between nodes. To bootstrap the system, or in the absence of paths between two nodes, the system relies on the Sporas system.

As discussed above, Sporas is resistant to ballot-stuffing and provides no incentive for re-entry. However, both models are vulnerable to the reputation lag problem (since they rely on the information provided by others), the value imbalance problem (since reputation changes are not linked to the value of transactions), and the exit problem.

REGRET

Like Griffith’s model [12], the REGRET system [26] also takes a multi-dimensional perspective on the modeling of reputation.

In this model, each agent maintains a database of ‘impressions’ of previous transactions, where impressions record the agents involved in the transaction, the particular dimension of the transaction being rated (for example, delivery time), and a rating. The ratings fall in the range [-1, 1], where 1, 0, and -1 represent maximally positive, neutral, and maximally negative ratings, respectively. An agent’s individual, subjective view of another agent’s reputation is the average of the ratings from impressions that match the desired transaction parameters (e.g., where required delivery time was less than 5 days), weighted to place greater emphasis on more recent transactions. The number and variability of the ratings are used as measures of the reliability of the estimate.

REGRET takes an ontological view of the multi-dimensionality of reputation—the opinions in multiple dimensions can be combined into a rating for a higher-level dimension (e.g., opinions on price and product quality might be combined into a general opinion of a seller’s quality.) The REGRET system also allows an agent to incorporate the views of other agents. Based in the idea that members of a group ‘share a common way of thinking’, agent A’s view of another agent B can be combined (using weighted averages) with the opinions of members of A’s group about B, opinions of A about members of B’s group, and opinions of members of A’s group about members of B’s group.

In [26], the authors discuss the use of this system in a marketplace scenario. In this environment, no ‘groups’ exist—all agents are considered to belong to the same group. Here, the model devolves into a combination of direct experience and a global reputation value. Under these circumstances, REGRET does not diverge from the eBay system in any way that substantially addresses the problems identified in Section 2.1.2, and therefore faces the same vulnerabilities.

Yu and Singh

In the model presented in [35] by Yu and Singh, a trader estimates the trustworthiness of a potential partner by using both its own experience, and advice from other agents. Their model makes use of referral networks, in which each agent has a set of ‘acquaintances’ (other agents for which the agent maintains reliability models). The agent also keeps a subset of acquaintances as a ‘neighbourhood’, a set of agents from whom the agent would ask for recommendations, and whom the agent would recommend to others. When asked for a recommendation, an agent may provide one from its own experience, or may forward the query to its neighbours—hence the term, ‘referral network’. An agent modifies its model of an acquaintance based on its own experience, ratings of the acquaintance provided by others, and the validity of recommendations provided by the acquaintance.

While many systems combine recommendations from agents, the authors have chosen to do so using the Dempster-Shafer theory of evidence [16]. They believe this method is superior to others due to its ability both to consider the trustworthiness of the sources of recommendations, and to distinguish the case where an agent believes another to be unreliable from the case where it does not believe another to be reliable (based on lack of evidence). This is in contrast to the Bayesian approach, which ‘cannot distinguish between lack of belief and disbelief’—if an agent is not known to be reputable, it is considered disreputable.

Only to the degree that an agent relies on its own experience rather than witness information, this system is resistant to the ballot-stuffing and bad-mouthing problems. It appears to be vulnerable to the re-entry problem, since it explicitly treats those with no reputation and poor reputation differently. The model is vulnerable to the reputation lag problem (since it relies on the information of others), the value imbalance problem (since reputation changes are not linked to the value of transactions), and the exit problem.

The Beta Reputation System

In [15], Jøsang and Ismail introduce the Beta Reputation System (BRS), a model based in statistical theory. BRS makes use of the well known beta probability distribution, which allows the estimation of the probability of a binary event occurring, given the observed number of successes and number of failures in the past. Based on these two input parameters, a probability density function (PDF) can be derived—this function allows the determination of the likelihood (based on the observations) that the actual probability of the event occurring on any given trial falls within any given range.

In BRS, an agent uses the beta distribution to estimate the likelihood that another agent will fulfill his commitment, based on past experience with that agent. An agent X will combine its own evaluation of another agent T (based on experience) with ratings for T provided by other agents in the system.

The authors acknowledge that recommending agents may not be trustworthy; for this reason, the authors introduce ‘reputation discounting’, a process by which an agent Y’s recommendations are modified before they are added to the sum of all other recommendations received, based on the evaluating agent X’s view of Y’s reliability.

In a later paper [34], an alternative method for dealing with unfair ratings is introduced. Here, the authors take an *endogenous* perspective, the view that unfair ratings might be detected and excluded based solely on their statistical properties. The essential idea is that extremely high or low ratings (relative to the bulk of ratings) may be of dubious validity, so removing such recommendations from consideration may yield better predictions. In particular, the removal of extreme ratings may limit the ability of agents to manipulate the results by intentionally giving inaccurate reviews (e.g., ballot stuffing and bad mouthing.)[1]

[1] This is not unlike the ratings systems used in sports such as diving and figure skating. In these sports,

By allowing different ratings to have different weights (e.g., more expensive purchases might be weighted more heavily than less expensive ones), BRS can provide protection against the value imbalance problem, and the filtering algorithm provides some protection against ballot-stuffing and bad-mouthing. Unfortunately, however, if many agents are providing unrealistically high or low ratings (as in a coordinated ballot-stuffing or bad-mouthing attack), these ratings will not appear to be so extreme in the context of all collected ratings, undermining the effectiveness of the algorithm.

The authors acknowledge the vulnerability of BRS to re-entry. The system provides no protection from reputation lag (since agents rely on the recommendations of others), nor from the exit problem.

TRAVOS

The TRAVOS system, described in [29], has much in common with BRS. It, too, makes use of beta distributions to predict the likelihood of honesty of an agent, based on the total numbers of observed successes and failures; the expected value of the resulting beta distribution is the evaluating agent's level of trust. (Unlike BRS, it limits these ratings to binary events: complete success, or complete failure.)

As with BRS, TRAVOS allows the evaluating agent to combine its own experience with recommendations of other agents, essentially by adding the number of successes and failures reported by recommenders to its own counts. TRAVOS provides a specific decision rule, however, to determine when an agent should seek out recommendations from others to supplement its own experience: if the agent's confidence is below a predefined threshold, then the agent seeks out these recommendations.

In contrast to BRS, TRAVOS takes an *exogenous* approach to cope with the issue of unfair ratings by recommenders, making use of information beyond just the statistical properties of the ratings. When evaluating the credibility of a recommendation, the evaluating agent considers the degree of accuracy of ratings previously provided to the agent by the recommender. Essentially, TRAVOS reduces the weight of recommendations

competitors' performance is rated by a number of judges. A competitor's score is calculated by accumulating the scores of all judges, except the highest and lowest scores, which are discarded—this is an effort to prevent a single judge from manipulating the scores.

from unreliable sources by ‘pulling’ them towards the uniform distribution (i.e., the no-evidence case)—the lower the level of confidence in the recommender, the further the values are ‘pulled’.

Unlike BRS, TRAVOS provides no protection against value imbalance, because it does not allow ratings to be weighted based on transaction value. TRAVOS may provide greater protection than BRS against ballot-stuffing and bad-mouthing, however: the credibility of ratings is determined based on the actual accuracy of past recommendations, rather than against current recommendations from other agents, which may have been manipulated. A key problem with TRAVOS, acknowledged by the authors, is that an agent’s behaviour is assumed to be consistent over time. This assumption may undermine the protection against ballot-stuffing and bad-mouthing discussed above. An agent who is part of a coalition may provide unfairly high ratings for coalition partners, but fair ratings for agents outside the coalition. If the agent provides a large number of recommendations for non-coalition agents, he may be considered credible.

As with BRS, TRAVOS is vulnerable to re-entry, reputation lag (since agents rely on the recommendations of others in some cases), and the exit problem. It should be noted that TRAVOS is also vulnerable to the initial window problem. If the agent has no experience with a seller, he will seek recommendations, but if he has received no recommendations in the past either, the current recommendations cannot be properly evaluated.

3.2 Transactional models

The Trunits model is difficult to classify into the information source categories identified in [27], because it is not a predictive model. Under Trunits, a buyer does not form a belief of a seller’s trustworthiness based on personal experience, nor does he consult with others to gain their opinions. Instead, Trunits make use of ‘units’ of trust that flow in the course of transactions, with a seller’s quantity of trunits determining her established level of trustworthiness. Here, we consider models that are also ‘transactional’, making use of an accounting system, or relying on units that flow during transactions.

Peer-to-peer systems, like those used for sharing files, work well only if there are sufficient numbers of users providing content. Unfortunately, these systems can be plagued with 'free-riders' who download large quantities of files without serving any content. Further, attacks have been made on such systems by users serving useless content disguised as more desirable material. In such attacks, the goals are to waste bandwidth and make it more difficult for users to find the files they seek. For these reasons, there has been significant interest in trust/reputation in the peer-to-peer community, with some proposals being transactional in nature.

Gupta et al.

One such proposal is that of Gupta, Judge, and Ammar [14]. Under their system, reputation scores are maintained using one of two mechanisms. Under the debit-credit policy, reputation is increased when a user contributes resources, and decreased when he consumes resources. The credit-only policy handles contributions in the same way, but ignores consumption. (The authors note the vulnerability of the credit-only policy to ballot-stuffing.)

While this proposal is described as a reputation system, it essentially models value, not trustworthiness: a user's ability to download is directly tied to the value of its contributions, meaning that uploads are effectively being exchanged for downloads. Thus, the reputation score is serving as currency, and is a proxy for value rather than trust.

Mojo Nation

At the suggestion of a reviewer of one of our papers, we examined a system employed by a now-defunct company called Mojo Nation[21]. Mojo Nation was a "peer-to-peer content distribution technology", in which agents are both consumers and providers of network resources such as bandwidth and storage space. A 'digital currency' known as *Mojo* is employed to execute transactions, using a system of micro-payments. Essentially, when an agent consumes a resource provided by another, it makes a micro-payment of some quantity of Mojo—this Mojo may be used by the resource provider to consume resources later. As the author claims, this mechanism makes the system resistant to attacks such as denial-of-service. As with the previous system, however, resources provided are essentially being exchanged for access to resources—Mojo is taking the role of currency. As such, Mojo simply models value, addressing none of the issues of trust in a marketplace any more than the exchange of money does on its own.

Grothoff

[13] discusses the economic system underlying GNUnet, another peer-to-peer file sharing system. The author states that GNUnet uses trust, rather than money, as its currency. Further, GNUnet makes use of the concept of risk, an idea central to Trunits.

Essentially, the GNUnet economic system operates as follows:

- When an agent wishes to request a resource, it offers some quantity of trust to the potential service provider. This trust is risked by the requester.
- If the load on the service provider is low enough that it can easily service all requests, it does not actually collect the trust risked by requesters. If resource demand exceeds what can be accommodated, however, it prioritizes requests by the quantity of trust risked, and charges each serviced node the quantity of trust offered.

This system is designed to promote the fair allocation of resources. Specifically, it is designed to allow free access to plentiful resources, while charging a ‘fee’ for scarce ones. It is clear, however, that the ‘trust’ used in GNUnet is, again, really a unit of value, since it is exchanged for access to scarce resources. As such, it does not address the issues of trust in a marketplace any more than the exchange of money does on its own.

3.3 Mechanism design approaches to trust

The Trunits model has a strong connection to the area of mechanism design [19]—Trunits attempts to eliminate dishonest behaviour by providing a strong incentive for sellers to behave honestly. Buyers, in turn, ultimately trust the system, beyond simply trusting individual sellers.

In this section, we consider some mechanism design approaches to the issue of trust. Researchers in mechanism design have largely taken a different perspective than many trust and reputation researchers: rather than providing a model to be used in a wide variety of circumstances (and leaving issues/vulnerabilities outstanding), they have instead attempted to provide provably correct solutions to smaller problems. As such, the proposals described below do not constitute 'complete' trust/reputation models. We examine them to understand the strengths and limitations of current work in this area, and to discover techniques that may be applied in our own work. In the interest of brevity, the systems are not examined in great detail. Instead, we focus on the following key questions, where applicable:

- What is the overall rationale/operation of the system?
- To what degree is the system subject to the vulnerabilities we have already identified?
- What new problems/issues, which we need to address, are revealed in the work?
- What can be learned from the work, and applied to strengthen/extend our method?

3.3.1 Incentive Compatible Mechanisms for Trust Revelation

Braynov and Sandholm

Braynov and Sandholm have investigated *incentive-compatible* mechanisms for trust and reputation. An incentive-compatible mechanism [19] is one in which the each agent's best strategy is to tell the truth—in this case, the mechanism would be designed so that an agent's profit is maximized when it honestly declares its trustworthiness. Such a mechanism would be a powerful tool. In [3] a proof of the existence of such a mechanism is provided, without a specific implementation. The existential proof is based on some assumptions, as follows:

- The seller has a certain probability of completing any transaction honestly. This probability, β, is known to the seller. The expected utility of the seller is dependent on β (since its cost depends on whether it fulfills the sale honestly or not), as well other transaction parameters (good purchased, price, quality, etc., composed into a single variable q).

- There is a specific value for q that maximizes the seller's utility.

The mechanism proceeds as follows. First, the seller declares its β value. Based on this declaration, the buyer chooses the remaining transaction parameters q. Because the seller's utility is a function of both β, and q, the buyer can choose a value for q that maximizes profit at the declared β. If the seller lied about its β, then it will earn a profit less than it would if it had declared honestly.

Other required assumptions impose important limitations, however:

- It is assumed that β is known to the seller, but whether or not the transaction is actually completed honestly is required to be random, even from the seller's perspective. In reality, a seller often knows with certainty whether it will complete a transaction honestly.

- More importantly, the proof of such a mechanism's existence is based on the seller's honest declaration of β, followed by the buyer's choice of q. However, the seller is extremely unlikely to have a fixed value of β for all transactions—likelihood of completion almost certainly varies depending on transaction parameters. This means that β is dependent on q, and undermines the mechanism whose existence has been proven.

In [4], a specific mechanism of this nature is presented, where the seller's utility is dependent on the quantity of good sold. Given price P and quantity q, the utility function of the seller is defined to be $U_S(q) = Pq - \beta C(q)$, where β is the seller's actual trustworthiness (i.e., the probability of delivering the promised good), and $C(q)$ is the cost function of the seller in providing quantity q of the good. The cost function is assumed to be increasing and convex.

A seller makes a declaration of its trustworthiness, denoted as δ. The buyer then chooses the quantity to purchase based on the seller's declaration, using a 'quantity function'. The authors prove that there is exactly one quantity that makes truthful declaration the seller's utility-maximizing strategy:

$$q(\delta) = C'^{-1}\left(\frac{P}{\delta}\right) \tag{3.7}$$

where C' is the first derivative of the seller's cost function, and the cost function is twice differentiable.

Note that this system does not prevent dishonesty, but rather allows the buyer to receive truthful declarations of trustworthiness from each agent, which might be used to choose between traders. While subject to the problems discussed above, this mechanism suffers from another: it is based on the assumption that the buyer knows the seller's cost function with certainty. While the author asserts that this information is relatively easy to secure, we find this claim dubious. It is often the case that a business will not even know its own cost function with certainty—establishing that of another is extremely problematic. Further, the assumption that the cost function is convex is contrary to the notion of economies of scale, which apply in many circumstances.

3.3.2 Mechanisms based on correcting cheating behaviour

Goodwill Hunting

In [7], the Goodwill Hunting system is proposed. Under this system the marketplace contains not only buying and selling agents, but also a market operator who acts to ensure that any excessive profits earned by a cheating agent are subsequently 'clawed back' from that agent.

Under this system, a product's utility for the buyer is seen as a direct function of product quality. To initiate a sale, the seller declares the quality of his product. However, this declaration is made to the market operator, and not to the buying agent; instead, the market operator publishes a quality value to buyers that may differ from the seller's declaration (as detailed below). The price that the buyer pays for a product is based on its utility, given the published quality. After the transaction is completed, the buyer reports its perceived quality of the good received.

Initially, the operator will publish the seller's declared quality without modification. If the buyer's reported quality is the same as that published, a fair transaction is deemed to have occurred. Consider the case, however, where the buyer's perceived quality is less than that reported by the agent. Since the price is determined by quality, the buyer will have paid too much—a price based on a higher expected quality. The difference between what was actually paid, and what ‘should’ have been paid is seen as excess profit for the seller. To rectify the situation, the market operator will devalue future quality declarations by that buyer. Reducing the published quality will reduce profit on those future sales—the reduced future profits will eventually offset the unfairly earned profit, until the situation is rectified. (The reverse is also true—if the seller under-declares quality, future declarations will be increased, so increased profits will compensate for money ‘left on the table’. This raises a potential issue—can a system be seen as providing protection to buyers, when it intentionally overstates product quality to those buyers?) In the long term, a seller fares no better by misrepresenting product quality.

This system is noteworthy in the active participation of the market operator in the execution of transactions; rather than simply providing information, the market operator intervenes to ensure that honest behaviour is most profitable. Our Trunits model takes a similar approach. However, Goodwill Hunting is based on several assumptions that drastically limit its applicability. One important such assumption is that sellers cannot control the order in which products are sold. The author acknowledges that control over the sequence allows the seller to cheat profitably via the value imbalance vulnerability; to cope with this, he assumes that inventory levels do not exceed a single unit. Even under his assumptions, the system suffers from the exit problem, the re-entry problem, and the ballot-stuffing problem.

Ba et al.

In [1], an economic incentive mechanism is proposed to encourage trustworthy behaviour in markets. The authors model individual transactions as instances of the prisoner’s dilemma. Noting that defection by both parties is the Nash equilibrium in any single transaction (i.e., each player’s action is an optimal response to the other’s action), they present a scheme in which defectors might be punished in the repeated game, making cooperation the more profitable strategy. Their proposed system imposes this punishment even if each sale in the repeated game is with a different partner—they make use of a trusted third party (TTP) and a cryptographic system to allow this. As in [7], this scheme makes the TTP an active participant in the execution of transactions.

TTPs are currently used to establish identity on the Internet via digital certificates.

The authors suggest the augmentation of this scheme by having the TTP track trustworthiness as well as identity. When engaging in a transaction, the agent offers its certificate to the other, who then verifies it with the TTP. If the TTP verifies the certificate, it confirms not only the identify, but the trustworthiness of the agent. Essentially, the scheme proceeds as follows:

- At the beginning of the transaction, the agents exchange certificates. Each verifies the trustworthiness of the partner by verifying the certificate with the TTP. (Transactions are also permitted without the use of a certificate, but these provide no protection against cheating.)
- After successful verification, the game proceeds. If an agent defects, the TTP imposes a fine (commensurate with the transaction value) on that agent.
- If the agent pays the fine, it continues to be considered trustworthy. If it does not, then the TTP 'revokes' the certificate, refusing to validate it in the future.

The authors show that, with appropriate parameter values, this mechanism makes honest play the most profitable strategy.

To eliminate re-entry, the authors depend on the ability of the TTP to research and firmly establish the identity of every trader, and to investigate and render judgement in every instance where a buyer is unsatisfied. This seems impractical, if not impossible, in markets of even modest size, let alone ones of scale comparable to eBay. Removing these flawed assumptions, the system is vulnerable to re-entry—with no fine imposed until after a cheating transaction has occurred, this vulnerability renders the system impotent. Even with the assumptions in place, the mechanism suffers from the reputation lag and exit problems.

3.3.3 Mechanisms for securing honest reviews

Jurca and Faltings

In [17], an incentive compatible mechanism is presented for eliciting honest reviews from buyers. The authors acknowledge the difficulties in obtaining (honest) feedback, and propose overcoming this difficulty by paying buyers for reviews.

Under this proposal, a separate currency is used within the reputation system, one that cannot be exchanged for money. Before a transaction occurs, the buyer can 'buy' a rating of the seller from an 'R-agent', a reputation broker. After the transaction has completed, the agent has the opportunity to provide a rating of the seller to the R-agent. The seller may or may not receive a payment for this rating: if the next rating which comes in matches that provided by the current agent, then she is paid; otherwise, she is not. If the agent rates the seller honestly, there is a greater likelihood that her rating will match the next one that is received, and she will be paid; this provides a financial incentive for honesty.

This paper is particularly relevant as a possible complement to our Trunits model, since we address only the trustworthiness of sellers. The operation of Trunits is dependent on receiving feedback from buyers, while this mechanism gives financial incentives for buyers to provide this feedback; the two systems might be used in parallel.

Unfortunately, this model suffers a key problem that must be addressed: it is vulnerable to ballot-stuffing and bad-mouthing. Under normal circumstances, an agent's review is most likely to match the subsequent one if she is honest. However, if we are 'stuffing the box' with our own ratings, then we skew the probabilities such that honesty may no longer be the best policy.

3.3.4 Incorporating trust into established mechanisms

Dash et al.

In [6], the incorporation of trust into the Vickrey-Clarke-Groves (VCG) mechanism is considered; specifically, they target a scenario where multiple suppliers are competing for a buyer's business. In the VCG mechanism, all bidders submit their (single) bids for the good (or for the buyer's business, in this case) privately to the auctioneer. It has been established that in this well-known mechanism, each bidder's optimal (utility-maximizing) strategy is to honestly declare its true valuation of the good [19]. Under the standard VCG mechanism, however, bidders know with certainty the value that they will derive from the good—an assumption that breaks down when faced with possibly untrustworthy suppliers. Essentially, under the proposal of Dash et al. the trustworthiness of competing providers is addressed by modifying the standard VCG allocation and pricing functions, incorporating the probability of success (POS).

This paper is interesting as an example of the incorporation of trust into an existing, established mechanism. However, it provides no system for actually determining trustworthiness. Instead, it relies on the use of a separate system that provides trustworthiness estimates as POSes. Further, it is focused on the task-allocation scenario, in which the trust of each bidder must be considered by a single buyer. In the standard market scenario that we address, it is the single seller whose trust must be evaluated by many buyers.

3.4 Summary of Related Work

In summary, there are a wide variety of approaches being considered to ensure agents encounter trustworthy partners, but none yet offers complete protection. Many of the models discussed in this chapter suffer from security vulnerabilities; others do not directly attack the issue of trust, or are very constrained in the situations/assumptions under which they apply. Convinced of the importance of security to trust and reputation systems, and inspired by the apparent lack of secure systems, we set out to develop a system that tackled these issues directly. The resulting model is described in the next chapter.

Chapter 4

Trunits

In this chapter, we present a novel model for trust in multiagent systems, Trunits. Trunits differs from most existing trust models, in that it attempts not to predict behaviour, but rather to ensure good behaviour by making honesty the most profitable strategy.

Trunits 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:

- Buyers and sellers had to interact directly, in order to exchange goods.
- For two parties to trade, each had to possess something the other needed. The goods exchanged had to be of comparable value.
- Storing value for later use was difficult, requiring that the value be in an imperishable form.

These limitations represent severe bottlenecks on both the speed and quantity of trade. A primary role of money is to overcome these limitations [23].

Money is an abstract 'substance', representing quantities of value. Many forms of money (e.g., paper money) have no intrinsic value—its only worth is derived from what someone will give you for it. 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 traders from the requirement for a direct two-way relationship where goods move in both directions—value gained from one trader can be 'spent' with another [23].

Many researchers working on trust and reputation models for multiagent systems are concerned with marketplaces, particularly the 'new' marketplaces that have recently become prominent. Such marketplaces may be very large, and buyers and sellers rarely meet one another directly; eBay is a typical example. The absence of direct relationships between traders in these scenarios prevents trust from forming 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.

4.1 The Trunits Marketplace

Electronic marketplaces require protocols to enable buying and selling agents to enter into transactions. Trunits was developed with a focus on *advertised price* marketplaces. In such a marketplace, sellers offer products for sale, specifying the nature of each good and the price at which it is offered. Buyers select goods that they wish to purchase, accepting sellers' offers. (Our focus on this scenario does not mean that Trunits need be inapplicable to other scenarios, however.)

When a seller's offer is accepted by a buyer, the seller is obliged to fulfill the terms of the offer. (This may be viewed as a promise, or more formally as a contract.) We consider a seller's failure to fulfill his obligation as *cheating*, or *dishonesty*. Cheating would include both the case where a delivered good does not conform to the seller's promise, and the case where the seller fails to deliver a good at all. This perspective follows that taken by other authors (e.g., [29]).

4.2 The Trunits Model

The movement of money in a transaction mirrors that of the flow of value in a direct bartering situation. Similarly, the flow of trunits should mirror that of trust in a direct trading relationship. The two 'flows', however, are fundamentally different in nature. The flow of value in a transaction is an exchange process, wherein each trader receives something of value, in exchange for providing the other with something of value. In contrast, we see the ‘flow’ of trust in a transaction as a *risk* process. We outline this process below, focusing on the buyer’s 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, relative to her expectations.
  - 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, and in which trust of the 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 purchasing something 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 (i.e., to the claims of the advertisement).
  - 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.

4.3 The Basic Trunits Mechanism

With this general model as a foundation, we formalize a basic mechanism that builds on the model. This mechanism uses the notion of a *market operator*, who is responsible for administering the marketplace. The operator maintains accounts and trunit balances for each trader, holds trunits in escrow during the course of a transaction, collects feedback from buyers and updates trunit balances, etc. We focus on marketplaces where the market operator is an identifiable entity (e.g., eBay Inc. operating the eBay marketplace); where such an entity is identifiable, it is considered to be a trusted third party. Trunits might be employed, however, in situations where there is no such identifiable operator. An example of such a scenario would be a peer-to-peer system. Under such circumstances, the activities of the market operator must still be performed; we may view the 'market operator' as the implementer of the system, or simply as the enforcement of rules by the system.

4.3.1 Mechanism overview

When an agent wishes to make a sale, we require him to put up a quantity of trunits to 'cover' the sale. 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:

$$V = r\tau \tag{4.1}$$

where V is the value (selling price) of the transaction, τ is the number of trunits, and r is the required *risk ratio*, 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:

$$(1 + p)\tau = (1 + p)V/r \tag{4.2}$$

where p is a *premium* or *reward* of additional trust for acting in an honest manner. p is a (positive) parameter set by the market operator. (It is suggested that p be less than 1, in order for trust to be harder to earn than it is to lose (as suggested in [30, 31]), but this is not a strict requirement.) In the basic mechanism presented here, the same values of r and p are used for all traders and all 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.

4.3.2 An Example

To illustrate the mechanics of the system, consider the following example:

- A seller has 20 trunits.
- The required ratio $r = 5$.
- Since value and trust are related by $V = r\tau$, the maximum transaction value that the seller can cover is $20 \times 5 = \$100$.

Suppose now that the seller engages in a sale for a price of $50:

- Since $V = r\tau$, the seller must place 50/5 = 10 trunits in escrow.
- If the buyer is unsatisfied, the seller loses the 10 trunits in escrow. She now has 10 trunits remaining, meaning that she can now cover transactions with a maximum value of $50.

- Suppose the reward ratio $p = 0.2$. If the seller is satisfied, then the seller's trunits are returned to her, and she receives an additional $p\tau = 0.2 \times 10 = 2$ trunits. She now holds a total of 22 trunits, and can cover transactions with a maximum value of $22 \times 5 = \$110$.

4.3.3 Important properties of the mechanism

The intention, illustrated in this example, is that trunits themselves will be valued, deriving from the fact that they enable profitable transactions. Under this mechanism, the more trustworthy a seller becomes, the greater volume/value of sales she can execute.

One possible downside of this policy is that it restricts the total volume of trade, since sellers may not possess enough trunits to sell their desired volumes of goods. It has the beneficial consequence, however, that trustworthy traders tend to dominate the market due to the larger sales volumes they can achieve. In many scenarios, this is a favourable trade-off in the eyes of buyers and market developers/operators. (In turn, having willing buyers makes a market attractive to sellers.)

Discussion of 'market dominance' can raise concerns of monopolistic behaviour. It should be noted that where the same risk-ratio r applies to all traders, a seller cannot use a large trunit balance to make its offering more attractive than that of other sellers; any seller with sufficient trunits to cover a transaction can compete. In fact, unlike many models, Trunits is quite egalitarian: a new and an established seller competing for a sale will be treated equally if they have enough trunits to cover the sale, regardless of their total accumulated trunit balances. In contrast, models such as BRS [15] and TRAVOS [29] treat buyers with longer histories differently than those with shorter histories, even if they have been honest with the same relative frequency. (Price-based monopolistic behaviour is an orthogonal issue to that of trust, and falls outside the scope of our model.)

Beyond the clear advantage of 'egalitarianism' for new sellers, there may be an advantage for buyers as well. When different sellers have different degrees of trustworthiness, a buyer may feel compelled to chose the 'most trustworthy' seller, even if that seller's good does not match his preferences as well as one offered by a less trustworthy seller. Under trunits, however, each agent with sufficient trunits can be considered equally trustworthy—the buyer is then free to choose products that best match his needs.

An obvious question is, if trunits have measurable value, how does this system differ from one in which the seller puts up a cash bond for each transaction, which is lost in the event of dishonesty? There are several key issues that highlight the differences between the two mechanisms:

- If the seller puts up a cash bond, and is determined to have been untrustworthy in executing the transaction, what happens to the bond? If it is paid out to the buyer, then the buyer has a financial incentive to rate the seller as untrustworthy even when the seller has been honest. If the market operator keeps it, then the market operator has an incentive to structure the system to encourage unfairly negative ratings, to the detriment of sellers.

- How large does the bond have to be for a transaction? If the bond is not at least as large as the seller's cost to furnish the purchased good, then the seller realizes greater profit from simply cheating the buyer (by keeping the money without providing the good) than from honestly completing the transaction. For example, if the price of the item is $100, the cost of the item to the seller is $60, and the value of the bond is $50, then the seller realizes a profit of $50 from cheating the buyer and conceding the bond, but only $40 from honestly executing the transaction. Thus, the amount of capital used to cover transactions must be of the same order as the revenue realized from the transactions. In the case of a high-volume seller, this financial requirement is likely to be unworkable, due both to the enormous amount of capital that must be devoted to bonds rather than to operations, and the financing cost of this capital.

- A bond scheme cannot easily accommodate the growth of trust that Trunits employs. If the value of a bond is to increase after each successful transaction, that money must come from somewhere—from the market operator (in which case the mechanism may not be financially sustainable), from the seller (in which case it constitutes no real gain for the seller), etc. In contrast, the market operator creates trunits without cost, so there is no obstacle to trust growth: trunits are closer to the notion of licenses than to cash bonds.

- Consider a transaction in which a buyer unfairly rates a seller as untrustworthy. In the case of a cash bond, the seller loses actual money. In the case of the Trunits model, the seller incurs an opportunity cost (i.e., he forgoes the opportunity to engage in a certain quantity of profitable business *in this particular marketplace* in the future), but does not lose actual money. One might argue that the unfair penalty in the case of the cash bond is more severe. Further, the loss of actual money could undermine the financial solvency of the seller, while the loss of Trunits will only impact his ability to operate in this particular market.

4.3.4 Why a buyer can trust in the system: The incentive for honesty

As noted by Dellarocas [8], "sellers care about buyer feedback primarily to the extent that they believe it might affect their future profits."[1]. As discussed above, honesty increases a seller's trunit balance, and hence her possible future sales, while dishonesty reduces both. This provides *some* degree of incentive for honesty; the question must be asked, is it a large enough incentive? Will honesty be the most profitable policy for sellers, motivating them to make trustworthy choices, and thus allow buyers to trust in them? Here, we present a simple analysis of the operation of trunits, to establish that the incentive does in fact exist. Moreover, the analysis is intended to further illustrate the operation of the system, to provide an intuitive understanding of the incentive for honesty provided by Trunits. In Chapter 6, we perform more rigorous analysis to establish the safety and security of the system with precision.

Here, we consider the expected profits of the seller when executing transactions over a period of time. The unit of 'time' employed is that required for the completion of a transaction, from the initial sale until the buyer ultimately provides feedback. Note that a seller may be harmed by unfair feedback from a buyer—the seller might lose trunits despite having honestly provided the good. Trunits makes no effort to prevent buyer dishonesty, focusing on regulating the behaviour of sellers.[2] For the purposes of this

[1] It can be argued that predictive models also have an incentive effect—if the seller knows he is being modeled, he may act honestly in order to improve his prospects for future sales. We discuss this issue in Chapter 7.

[2] It should be noted, however, that while Trunits does not discourage unfair feedback from buyers, it

analysis, we assume that buyers provide fair feedback; we address this issue in detail in Chapter 6.

At the beginning of each unit of time, the value of goods that can be sold is limited by the number of trunits available. We assume here that the seller engages in transactions of the maximum allowed value, since doing otherwise would incur opportunity costs in terms of both profit, and trunits if the sale is an honest one. (We relax this assumption in Section 4.3.5.) The seller might execute a single large transaction, or split the trunits in order to execute several smaller transactions. Where the same ratio r is used for every transaction, and assuming that the cost of the goods to the seller is the same in both cases, these scenarios have equivalent outcomes under our model: the total value of the transactions, the total profit, and the total trunits gained are the same in both cases. For example, consider two alternatives, a single transaction with a price of \$10, or two transactions at prices of \$5 each. If $r = 5$, the single transaction requires $10/5 = 2$ trunits, while the pair of transactions requires $2 \times (5/5) = 2$ trunits. If $p = 0.2$ and $c = 0.5$, the single transaction yields $2 \times 1.2 = 2.4$ trunits and a profit of $(1 - 0.5) \times 10 = 5$ when executed honestly, while the pair of transactions yields $2 \times (1 \times 1.2) = 2.4$ trunits and a profit of $2 \times ((1 - 0.5) \times 5) = 5$. For this reason, we simplify the analysis by considering only the case where a single transaction is made in each time period. Under these assumptions, in a period of time consisting of h units, our seller engages in a sequence of h transactions.

Profit on a transaction is the difference between the selling price and the cost incurred in selling the item. We express cost, c, as a fraction of selling price. c would include both the actual cost incurred in the creation/provision of the good, as well as expenses such as the commission charged by the market operator. Given that the value of a transaction $V = r\tau$, profit on a honest transaction is

$$P = (1 - c)r\tau \tag{4.3}$$

Let τ_0 represent the seller's available quantity of trust before the first transaction in a sequence. If all transactions from the first to the i-th are executed honestly, then after

does not provide an incentive for unfairness either; an individual buyer receives no compensation if he reports that the seller was dishonest.

the i-th transaction the seller's quantity of trust is:

$$\tau_i = (1+p)^i \tau_0 \tag{4.4}$$

This quantity of trust available after the i-th transaction will allow the next transaction to be of value $r(1+p)^i\tau_0$. The profit earned from executing transaction $i+1$ will then be

$$P_{i+1} = (1-c)r(1+p)^i \tau_0 \tag{4.5}$$

The total profit over a sequence of h honest transactions, beginning with trust τ_0, then, is:

$$\begin{aligned} P_{S_h} &= \sum_{i=0}^{h-1}(1-c)r(1+p)^i\tau_0 \\ &= (1-c)r\tau_0\sum_{i=0}^{h-1}(1+p)^i \end{aligned} \tag{4.6}$$

The summation above is a geometric series, so it can be represented in closed form as:

$$P_{S_h} = (1-c)r\tau_0\left(\frac{(1+p)^h - 1}{p}\right) \tag{4.7}$$

Note that we have established, with precision, the value of a quantity of trunits (if used honestly and maximally) in terms of the future sales it allows. This valuation can serve as the basis for rational decision making by agents.

Now, consider a seller who has engaged in a transaction with value $r\tau_0$. She has two choices: either fulfill the transaction honestly, or cheat the buyer. In either case, the seller receives revenue $r\tau_0$. When cheating, she loses all (τ_0) of her trunits and realizes a maximum profit on the transaction of $r\tau_0$ (in the case where she fails to supply the purchased good at all). If she is honest, she will gain $p\tau_0$ trunits while earning a profit of $(1-c)r\tau_0$. While cheating results in a larger immediate profit, honesty may be more profitable in the long run, since the trunits will allow her to engage in additional transactions in the future. Let h be the seller's *horizon*, the number of sales she can foresee making in the future, including the current one. For honesty to be economically advantageous, her total expected profit over the transactions in her horizon must be greater than that realized by cheating on this first transaction. Setting this inequality, and then solving for h:

$$\begin{aligned}
P_{S_h} &> r\tau_0 \\
(1-c)r\tau_0\left(\frac{(1+p)^h-1}{p}\right) &> r\tau_0 \\
(1-c)\left(\frac{(1+p)^h-1}{p}\right) &> 1 \\
\frac{(1+p)^h-1}{p} &> \frac{1}{(1-c)} \\
(1+p)^h &> \frac{p}{(1-c)}+1 \\
h &> \log_{p+1}\left(\frac{p}{(1-c)}+1\right) \qquad (4.8)
\end{aligned}$$

In the inequality above, note that τ_0 disappears—the existence of the incentive for honesty is not dependent on the value of the transaction. Charting this inequality for several values of p yields the graph displayed in Figure 4.1. Points above the curves indicate combinations of cost ratio c and horizon h for which the honesty incentive exists. It is evident from the chart that, unless sellers have extremely low profit margins, honesty is economically advantageous with even very short horizons. For example where $p = 0.5$ and $c = 0.6$, the minimum such horizon is 2 transactions.

Further, it should be noted that the above analysis assumes no cost to the seller in the event he decides to cheat. This is a very conservative assumption—given possible costs incurred outside the mechanism itself (penalties imposed by the market operator, remedies within the legal system, etc.), the incentive for honesty is likely to be even larger than stated above.

It is unlikely that a seller's cost structure will be known. Since the incentive mechanism is sensitive to the value of c, it is valuable to understand the impact c has on the incentive. Solving the same inequality for c, we obtain:

$$c < 1 - \frac{p}{(1+p)^h - 1} \qquad (4.9)$$

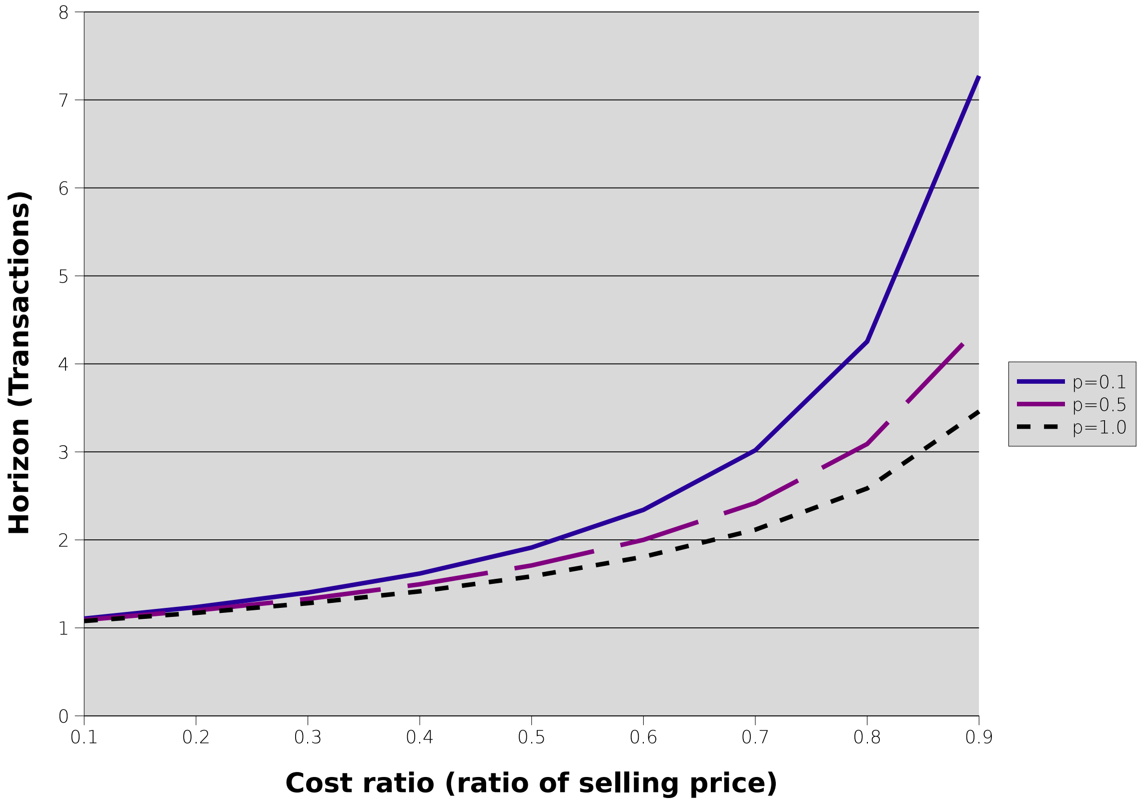


Figure 4.1: Minimum Horizon for which the incentive for honesty exists.

This inequality yields a chart, shown in Figure 4.2, which is a reflection of the previous one; points under the curves indicate combinations of c and h for which the incentive is evident. It is clear in this graph that the incentive exists even for very high cost ratios. For example, with $p = 0.5$ and a horizon of only three transactions, the incentive exists for cost ratios up to almost 80%.

One potential problem is obvious: if the seller's desired number of future transactions is below this threshold, then the economic incentive for honesty is no longer clear. This is a problem common to most trust/reputation systems—if the seller intends to make only a small number of sales and then exit the market, the impact of dishonesty on his repu-

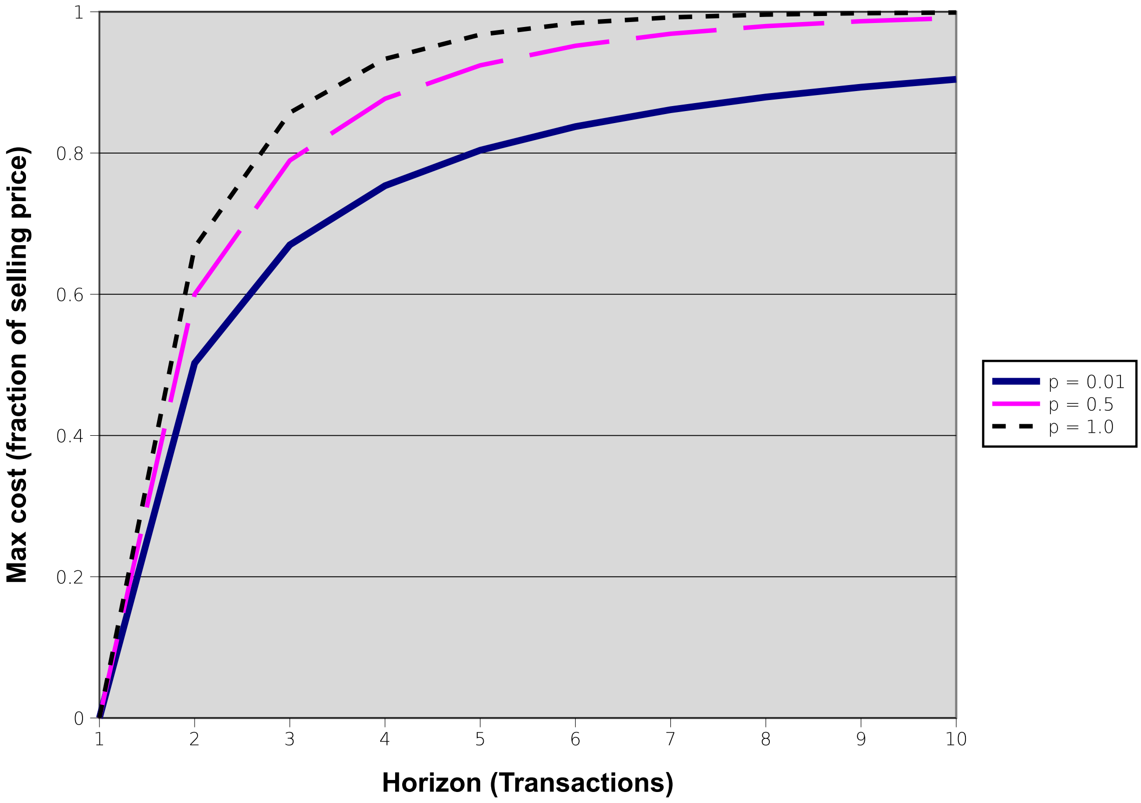


Figure 4.2: Maximum cost ratio for which the incentive for honesty exists.

tation is of little consequence. This is an instance of the exit problem that was identified previously, and will be discussed throughout this chapter and the following one.

While this analysis highlights the existence of an incentive for honesty, it gives the impression that the incentive is dependent on market parameters. In fact, the incentive exists essentially regardless of the values of p, r, and c, as discussed in the next section.

It is important to note that the analysis above depends on certain assumptions: the seller knows his minimum horizon, the seller will continue to find buyers, buyers will rate the seller fairly, etc. To eliminate these assumptions, our analysis might introduce probabilistic elements. For example, instead of having a known horizon, after each transaction an agent might engage in a subsequent transaction with some probability δ. Under these circumstances, an agent's decision would be made based on expected revenues, rather than on the 'guaranteed' revenues assumed above.

While these issues of uncertainty and risk are important to the incentive for honesty, we do not incorporate them into our analysis at this point, for several reasons. First, while certain of these issues may appear important here, analysis conducted later in this thesis reveals them to be of lesser relevance. For example, as explained in the next section, the incentive for honesty exists regardless of an agent's horizon, obviating the need to introduce probabilistic handling of this issue. Second, certain issues point to limitations in the Basic Trunits mechanism. For example, the ability to find buyers in the future essentially speaks to the exit problem: if a seller cannot find any further buyers, the seller essentially exits the market. Basic Trunits provides only limited protection against the exit problem (as detailed in Section 6.4.1); rather than incorporating probabilistic handling of this issue into the current analysis, we instead develop a mechanism in Chapter 5 that directly addresses the problem.

These issues are discussed in further detail in Section 8.2.

4.3.5 The incentive for honesty in the general case

The analysis above illustrates the operation of the mechanism, and establishes the presence of an incentive for honesty under certain circumstances. Moreover, it gives an intuitive feeling of the the interplay between market parameters and the nature of the exit problem, and importantly, provides a basis for establishing the value of trunits. It considers only a simplified scenario, however: a sequence of non-overlapping honest transactions of maximum value. In this section, we remove these simplifying assumptions. Instead, we consider any arbitrary transaction, which may coincide with or overlap other sales by the same seller. If, on any given transaction, the rational seller will be honest, then *every* sale should be conducted honestly.[3]

[3] Note that this proof, considering any arbitrary *single* transaction, necessarily ignores issues of collusion—the seller is assumed to be acting alone. The issue of collusion is discussed more directly in

We consider any seller engaged in a single transaction (i.e., a buyer has agreed to purchase a good for price V, where $V = r\tau$). The τ trunits used to cover the sale are not necessarily the seller's entire balance, but any arbitrary portion of his balance. The seller now has a choice to make: cheat, or be honest? We assume that the seller makes his choice based solely on his own economic motivation—in particular, he is acting alone. Further, we assume that if there is one buyer interested in purchasing the good, there is at least one additional buyer interested in purchasing it at the same price. (Since an honest sale results in trunits, which have value based on their ability to be used in future sales, at least one subsequent buyer is required to establish this value. Note, however, that this subsequent buyer need not be interested in strictly the same product: any combination of sales that uses the received trunits is sufficient.) We feel that this is a realistic assumption, given the assumed large market size, and that we are operating under traditional advertised price conditions.

If the seller cheats, the maximum profit he can realize from the sale is $r\tau$—where he fails to ship the good at all, so he incurs no cost. By cheating, he also loses his τ trunits. If the seller is honest, he ships the product and realizes a smaller profit, $(1-c)r\tau$, where c is the cost incurred in selling the item, as a fraction of the sale price. However, after the sale he has $(1+p)\tau$ trunits returned to him. Since there is at least one more buyer who wishes to purchase the same item, the seller can use the trunits to engage in another sale with this new buyer. Since the seller decides independently whether to cheat or be honest, he can cheat this second buyer—in fact, he need not even have one of the items in inventory. Cheating this second buyer returns a profit of $r\tau$, and causes the buyer to lose τ trunits, leaving him with $p\tau$ remaining. The total profit earned (so far) is $(1-c)r\tau + r\tau = (2-c)r\tau > r\tau$ (since $c < 1$).

Thus, a rational seller will be honest on the current transaction, since he can make more money by waiting to cheat (and earning honest profits in the interim) than by cheating immediately and destroying the trunits used to cover the transaction.

This is not meant to imply that the rational seller will cheat on the subsequent transaction. Rather, on any given transaction, the seller can make more money in total by delaying his cheating than by cheating immediately. It follows that the rational seller

the rigorous security examination presented in Chapter 6.

will never cheat, since it always makes sense to engage in 'just one more' honest transaction. Thus, the value that one might earn by cheating on the subsequent transaction is actually a lower bound on the profit from future honest transactions using those trunits. We note this lower bound here because it is direct, and because it would be obvious to any rational seller, even those whose computational capacity is limited.

It must be noted that this analysis is valid only when the seller actually has inventory. If the seller has no items to sell (or more precisely, will *never* have more items to sell), then he cannot defer cheating until the next transaction—for a transaction to occur, it must be a dishonest one. This is an instance of the exit problem, which afflicts most models of trust/reputation. We discuss a solution to this problem in Chapter 5, with an extension to Basic Trunits known as Commodity Trunits.

4.3.6 The Start-up problem

The trunits mechanism, as described to this point, results in a 'chicken-or-egg' problem: you earn trunits only by engaging in honest sales, but you need trunits in order to engage in sales. There are a number of possible ways to allow sellers to enter the market without an existing balance of trunits:

- New sellers might simply be provided with an initial quantity of trunits upon entry to the market. In the general case, this is not an advisable option—it opens the door to the re-entry problem, since new users would be in a better position than maximally disreputable ones. This option may be applicable, however, in scenarios where sellers' true identities can be established with certainty, or some means external to the mechanism can prevent re-entry.

- One possibility is to allow a new seller to put up a cash bond in order to be fronted with a loan of some trunits. Once sufficient trunits had been earned, the seller could repay this loan, reclaiming her bond. This system incurs some of the disadvantages of the cash bond system described earlier, but such bonds are in use only for short periods of time. (As a generalization of the above, trunits could be loaned to bonded sellers any time they had insufficient quantities to engage in desired transactions; note that this would address the potential restraint of trade cited as a possible problem in Subsection 4.3.3.) We examine this option in detail in Chapter 5.

- New sellers might gain entry to the market by using a trusted broker. Such brokers exist on eBay today—they maintain high reputations, and sell products on behalf of other people in exchange for a fee. If the brokerage transaction were executed inside the system, then the new seller could gain trunits from the transaction.

- One possible extension to this system, detailed in Chapter 5, is to adopt a 'free-market' approach, eliminating the required ratio r, and allowing sellers to offer any product/trunit combination they see fit. In this scenario, market forces would determine the value of trunits. Under this system, a new entrant could sell products using zero trunits, if they were discounted enough to motivate buyers to take the risk. In doing so, sellers could earn trunits for later use.

The appropriate choice of these methods depends on the scenario to which the mechanism is being applied.

4.3.7 The Basic Trunit mechanism's handling of key problems

As discussed earlier, a system ideally would prevent each of the problems identified in Chapter 2, without introducing substantial new ones. We provide a rigorous examination of the protection provided by Trunits in Chapter 6. Here, we briefly discuss Trunits' handling of each key vulnerability to further illustrate the operation of the system, and to highlight its value.

The Reputation Lag problem

The Trunits mechanism deals directly with this vulnerability by compelling the seller to place trunits in escrow to cover transactions, forcing him to wait until the trunits have been returned before he can use them in another transaction. The Trunits mechanism regulates the rate at which transactions can occur: if the seller holds τ trunits, then the maximum value of transactions he can engage in during one unit of time is $r\tau$, regardless of timing or circumstances. In effect, the Trunits mechanism prevents the use of the 'same trust' to support multiple simultaneous transactions.

The Value Imbalance problem

The Trunits mechanism deals directly with this vulnerability by basing both the quantity of trunits required to cover a transaction, and the size of the reward, on the value of the transaction.

Revisiting the example from Section 2.1.2, assume that the seller executes five honest transactions at $1 each. Using $r = 5$ and $p = 0.2$, $V/r = 5/5 = 1$ trunit required in total to cover the five transactions. Executing these transactions honestly would result in a new trunit balance of $(1 + p)\tau = (1 + 0.2) \times 1 = 1.2$ trunits, or enough to cover a $6 transaction. The seller is unable to use the reputation gained from the five smaller transactions to cheat a buyer out of $1,000, as he might be able to do under the eBay system.

The Ballot-Stuffing and Bad-Mouthing problems

Collusion is a notoriously difficult problem to combat, one which has seen little progress to date from trust and reputation researchers. As presented, Trunits provides no obvious impediment to bad-mouthing. It is possible, however, to configure a Trunits marketplace so as to make ballot stuffing unattractive. The key lies in an idea suggested in [2]: if transaction costs (e.g., commissions) are larger than the expected future gain from a ballot stuffing transaction, then there is an economic disincentive to engage in such transactions. While this technique can be employed in Trunits, it places a strict limit on the allowable premium p. This approach is discussed in more detail in Chapter 5.

Other possibilities for coping with these collusive attacks are discussed in Chapter 7.

The Re-entry problem

In [11], Friedman and Resnick suggest a solution to dealing with the re-entry problem: new entrants to the market should incur a cost, such that the cost of re-entry exceeds the benefit. In [9], it is established that in marketplaces with binary feedback mechanisms, the policy of ‘optimal social efficiency’ is one where new users begin with the worst possible reputation (i.e., the same as very disreputable users).

In the Trunits mechanism, new and maximally disreputable users are treated the same, providing no incentive for re-entry, and consistent with the optimal policy. Further, this system is fully compatible with charging fees for new accounts.

The Exit problem

The Trunits mechanism, as described above, is vulnerable to the exit problem. In fact, while the exit problem is common to most trust/reputation systems, it could potentially be magnified under the Trunits mechanism.

Consider a user who has accumulated a large quantity of trust/reputation, and who has decided to leave the market. This user may decide to use her reputation to cheat users before exiting the marketplace. Under the eBay system, the user might engage in numerous dishonest transactions, but there would be a practical limit on the number that could be completed. Buyers would have some warning—seeing a string of negative transactions in the feedback profile, it would be obvious that the seller’s behaviour had deteriorated, and buyers would stop trading with her despite her remaining positive reputation score. In contrast, under the Basic Trunits mechanism the seller would be free to use every available trunit for the purposes of deception, until her supply was exhausted.

Note that, while the vulnerability might be magnified *if* the seller actually decides to execute a ‘cheating exit’ from the market, Trunits provides protection against such an event. First, as has been discussed, trunits have value, in terms of the future profitable transactions they make possible. This value can be determined with some precision, and may be strong incentive for the seller to stay in the market. Alternatively, this asset might be profitably transferred to another seller (e.g., sale of the company), rather than used for dishonest purposes. Additionally, we might allow the user to sell the trunits themselves, providing a more attractive (i.e., profitable) alternative to the cheating exit. While allowing the sale of ‘trust’ is counter-intuitive, it has has many beneficial properties, and does not undermine the incentive for honesty. These issues are examined in Chapter 5.

A new problem: *Surplus trust*

The Trunits mechanism, as described here, suffers from another potential problem, not yet noted in another model: that of *surplus trust*. A seller may accumulate trunits beyond what is required to cover his regular transactions. Such surplus trunits could be used to cheat buyers, without having a negative impact on the regular transactions.

For example, consider a seller who has a fixed production capacity, so he can only sell five items per week. Assume that the seller has enough trunits to cover sales at this rate. Also assume that the trust reward $p = 0.2$. For each honest transaction, the seller will get back the trunits used to cover that transaction, plus a 20% reward. The trunits that were returned are all that is required to cover his future sales as well, since his production capacity is fixed; this means that the 20% reward is strictly in excess of his needs. The seller may now use these surplus trunits to cheat buyers; as long as he does not spend his original trunits on dishonest transactions, he will continue to be able to sell his entire production without impediment.

A solution to this problem was discussed in reference to the exit problem, above: allowing the sale of trunits. Such a policy would be strong incentive for sellers to cash in unneeded trunits rather than using them to deceive users, if trunit sales were more profitable than cheating. This issue is explored in Chapter 5.

4.3.8 Other noteworthy properties

In closing our overview of the Basic Trunits mechanism, it is worthwhile to highlight some other important properties of the system.

Potential Advantages:

- The basic Trunits mechanism is very simple computationally, particularly for buyers, who do not need to perform any calculations to determine a seller's trustworthiness.
- The storage requirements for the mechanism are minimal, since a single trunit balance must be kept for each seller.

- In many systems, it is clear that trust has an impact on profitability (since being trustworthy allows more products to be sold, and/or fetches higher prices for products), but it is very difficult to measure that impact. This complicates many issues, including both strategic decisions by sellers, and buyers' interpretation of sellers' histories/reputations. The basic Trunits mechanism allows the value of trust to be determined with some precision, serving as the basis for rational decision making.

Potential Disadvantages:

- While it is possible that an encryption-based scheme might allow this system to be implemented in a decentralized form (perhaps similar to that used in [14]), the model strongly favours a centralized implementation, with some sort of trust management infrastructure required. In this way, it resembles the electronic payment systems (where payments are transfers from one account to another, managed by institutions or firms) more than that of cash payments (where individuals hold their own funds). It is easy to envision the market maker providing such an infrastructure, in the same way in which eBay has provided a funds transfer system (PayPal).

- In this form, Trunits is not suitable for use in auctions, with the possible exception of descending price auctions (i.e., 'Dutch' auctions). Since the ultimate selling price (or even the maximum possible price) is unknown at the beginning of an ascending price or sealed-bid auction, the number of trunits required to cover the selling price would also be unknown. If the auction is one where the price proceeds downward, the seller might be required to have enough trunits in his possession to cover the maximum possible selling price; however, if the prices proceed upward, there is no way to set a maximum requirement beforehand.

- Unlike many systems, this model requires feedback from sellers on every sale. Under a system such as eBay's, if a buyer does not rate a seller, the seller's profile simply includes less information about his behaviour. In contrast, if a buyer doesn't rate a seller under Trunits, the seller's trunits are not returned from escrow. Implementations of Trunits, then, must require that buyers furnish feedback in every case. (Alternatively, given that it is an established aspect of consumer behaviour that dissatisfied customers tend to be much more vocal than satisfied ones, the system could consider the lack of feedback to constitute a positive response, after some period of time has passed.)

- Under Basic Trunits, sellers build up trunit balances over time. Moreover, the incentive for honesty is dependent on sellers engaging in transactions beyond the current one. These two properties both mean that Basic Trunits may be less suited for marketplaces with substantial numbers of small sellers (e.g., individuals) who are very transient. The extended mechanism proposed in Chapter 5 overcomes this limitation.

Basic Trunits has many desirable properties, but also presents certain limitations. In the next chapter, we present an extended model that directly addresses some of these limitations.

Chapter 5

Extensions to Trunits

The Basic Trunits mechanism, as described in Chapter 4, has valuable properties, for example: it provides a strong incentive for honest behaviour; it provides protection from many of the attacks to which many existing models are vulnerable; it has minimal computation and storage requirements; it allows the value of trust to be measured with precision, allowing rational decision-making by agents. The Basic Trunits model also faces limitations. As noted in Chapter 4, there are a number of possible extensions or modifications of Basic Trunits that have the potential to address these limitations. In this chapter, we discuss one such direction, which we term *Commodity Trunits*.

Commodity Trunits directly addresses the common and difficult exit problem vulnerability, as well as the start-up and surplus trust difficulties particular to the Trunits system. The direction taken here is mechanistic in nature—we seek to create an environment in which rational sellers behave honestly. It might be argued that the approach described here diminishes the claim that the result is a *model* of trust—aspects of what we propose (particularly, allowing trust to be traded as a commodity) are difficult to reconcile with any ‘real-world’ notion of trust. We discuss this issue in Chapter 7.

5.1 Revisiting the Start-up problem

In Chapter 4, we noted that the Basic Trunits model faces a ‘chicken-or-egg’ problem—one needs to make sales to earn trunits, but one needs trunits to make sales. To allow new sellers to get started in the market, we suggested several possible approaches for investigation. One such proposal was the use of loans to provide an initial quantity of trunits to the seller. This loan would be secured by a cash bond; when the seller repays the trunit loan, the bond is refunded. More generally, we discussed the possibility of providing trunit loans at any time, alleviating unnecessary restraint of trade by allowing traders to sell goods even when their trunit balances are insufficient. In fact, in this chapter we explore a more general concept, one that encompasses these ideas: the treatment of trust as a tradable commodity. Allowing traders to buy and sell ‘trust’ may be counter-intuitive; as shown in the sections that follow, it is an approach that is both safe and beneficial. Trunit loans can be seen as a special case of allowing the sale of trunits: buying trunits, and then reselling them at the same price is equivalent (from the borrower’s point of view) to taking out a secured loan. It is clear that, if sellers can buy trunits, the start-up problem disappears. In the following sections, we verify the safety of this approach, and then discuss its implications for the system, including a discussion of key outstanding problems from Basic Trunits.

5.2 Safety requirements for trunit sales

To ensure that allowing sales of trunits is safe, some conditions must be established. We draw our notion of safety from the field of formal methods, where system safety is defined in informal terms as, ‘nothing bad happens’ [25]. Here, we mean specifically that the introduction of trunit sales does not undermine the incentive for honesty in any case for which it existed under Basic Trunits. This, in turn, would mean that it is safe for buyers to participate in the market, since they won’t be cheated. (We address the concept of ‘safety’ more formally in Chapter 6, in defining what it means for a system to be secure.)

We denote the selling price of a trunit as b. In this section, we assume that all trunits are bought and sold at this price; we relax this constraint in Section 5.5.

For a seller to wish to buy a trunit, he must expect to profit from the purchase (i.e., to increase his profits from sales). For the seller to make a rational purchase decision, he must understand both the purchase price, and the expected change in profit. In Chapter 4, we established that (given an initial quantity of trunits τ_0 and a horizon of h transactions), the profit on a series of transactions can be determined using formula 4.7. This formula understates the profit to be earned from the quantity of trunits, however. At the end of the sequence of h transactions, the seller still possesses trunits, which might be used to engage in further profitable transactions, to cheat sellers, or (as discussed below) sold for a profit. Thus, we modify formula 4.7 to account for the trunit balance at the end of the sequence. To determine a lower bound on the value of the trunits, we base our calculation on the assumption that the seller uses all left-over trunits to cheat at the end of the sequence. This serves as a lower bound, because the mechanism permits the seller to cheat at will (if he is willing to sacrifice the trunits), so he can assuredly achieve at least this level of profit. The profit to be earned from a quantity of trunits τ_o, then, is

$$P_{SC_h} = (1-c)r\tau_0\left(\frac{(1+p)^h-1}{p}\right) + r(1+p)^h\tau_0 \tag{5.1}$$

where again, c is the cost of goods to the seller (as a fraction of selling price), r is the required ratio of trunits to selling price, and p is the premium for honesty. The profit to be earned, then, from any individual trunit can be determined by setting $\tau_0 = 1$, yielding:

$$P_{\tau_h} = (1-c)r\left(\frac{(1+p)^h-1}{p}\right) + r(1+p)^h \tag{5.2}$$

For a rational agent to purchase a trunit, then, his expected profit from the trunit must exceed the purchase price, i.e., $b < P_{\tau_h}$.

A seller can use any sum of trunits τ immediately to cheat, yielding a profit of $r\tau$. For the sale of trunits to be safe, it must be the case that it costs more to buy them than can be earned by cheating with them; otherwise, the seller could profit by buying trunits, and then cheating buyers with them immediately. This requires that $b\tau > r\tau$, or $b > r$.

Together, this requires that $r < b < P_{\tau_h}$. This introduces no conflict since, given the allowable ranges of the parameters, formula 5.2 yields $P_{\tau_h} > r$.

For example, consider a marketplace in which the required ratio $r = 5$, the reward for honesty $p = 0.2$, the seller's cost to furnish goods $c = 0.5$, and the seller's horizon $h = 3$ transactions. Suppose the seller has the opportunity to engage in a transaction requiring 10 trunits (i.e., with a value of $V = r\tau = 5(10) = 50$), but has no trunits available. If the seller acquired enough trunits to execute the sale, she might do so honestly or dishonestly.

If the seller were to cheat (maximally), her profit from the sale (excluding the cost of acquiring the trunits) would be $V = r\tau = 50$. For the same 10 trunits, using formula 5.1 we can determine the expected profit to the seller if she were to purchase them and engage in honest sales to her horizon:

$$\begin{aligned}
P_{SC_h} &= (1-c)r\tau_0\left(\frac{(1+p)^h-1}{p}\right)+r(1+p)^h\\
&= (1-0.5)5(10)\left(\frac{(1+0.2)^3-1}{0.2}\right)+5(1+0.2)^3\\
&= 25\left(\frac{0.728}{0.2}\right)+5(1.728)\\
&= 177.4
\end{aligned}$$

Now, consider b, the price paid to purchase trunits. If $b < r$, say $b = 4$, then the seller can purchase the required trunits for $b\tau = 4(10) = 40$. This would allow the seller to earn a profit of 10 by purchasing trunits for 40, then immediately cheating on a transaction and receiving 50. By comparison, if $b > r$, say 6, then the seller can purchase the required trunits for $b\tau = 6(10) = 60$. In this case, the seller loses money by purchasing trunits to cheat, paying 60 but only taking in 50. When $b > r$, the seller will purchase trunits only if they can be used to engage in an honest transaction.

But will the seller purchase the trunits at all? She will only be interested in purchasing the trunits if they cost less than her expected profit over her horizon—177.4 in this case, as calculated above. Thus, when $b\tau < P_{SC_h}$, i.e., $b(10) < 177.4$, or $b < 17.74 = P_{\tau_h}$, purchasing trunits is attractive to the seller.

Having established that, with b in the proper range, it makes no sense to buy trunits and cheat immediately, we revisit our proof of the mechanism's incentive for honesty.

5.3 Safety in a single transaction

In Chapter 4, we demonstrated the presence of the incentive for honesty under Basic Trunits. Here, we revisit this proof under the extended Commodity Trunits scenario.

In Section 4.3.5, we showed that for any single transaction, the seller makes more money by being honest than by being dishonest. Note that this is true for any trunit balance—the source of the trunits was not discussed, and in fact, is not relevant. (Certainly, the trunits must come from somewhere, but the honesty incentive exists for any of the possible sources.) Thus, we consider only the seller's choice of how to use the quantity of trunits.

With the trunits used to cover the sale, there are now three choices:

1. Cheat, and realize profit of $r\tau$, while losing the τ trunits.

2. Sell the trunits, and realize a profit of $b\tau$, while relinquishing the τ trunits.

3. Act honestly, earning $(1-c)r\tau$, keeping the τ trunits and gaining an additional $p\tau$ trunits, for a total of $(1+p)\tau$. These trunits can then be used in one of three ways: cheat on future transaction(s), sell the trunits, or engage in future honest transaction(s). We need only consider one choice, though, to clarify the situation. Since the trunits can be sold at price b, the $(1+p)\tau$ trunits will fetch $b(1+p)\tau$ on the market. The total profit in this case, then, is $(1-c)r\tau + b(1+p)\tau$.

Since, $b > r$, $b\tau > r\tau$; selling the trunits is more profitable than cheating. Since $c \leqslant 1$ and p is positive, $(1-c)r\tau + b(1+p)\tau \geqslant b(1+p)\tau \geqslant b\tau > r\tau$, so engaging in an honest sale is also more profitable than cheating. Cheating, then, is the profit minimizing choice; there is an economic incentive to be honest even with the introduction of trunit sales. For any single transaction then, a rational seller will choose to be honest, since this is the profit maximizing strategy.

Note, too, that unless c and p have their maximum and minimum values (1 and 0) respectively, $(1-c)r\tau + b(1+p)\tau > b\tau$, meaning that engaging in honest sales is more profitable than simply selling trunits—agents are encouraged to engage in honest sales.

One critical point must be noted. As discussed earlier, if the seller has no further goods to sell, engaging in honest trades is not possible. Without the sale of trunits, his only (profitable) option was to cheat with the remaining trunits. Now, there is another choice; moreover, it is more profitable to sell the trunits than to cheat.

5.4 Addressing key problems

As discussed earlier, the initial motivation for considering trunit sales/purchases was the start-up problem. The Commodity Trunits mechanism successfully deals with this issue. However, several other key problems are solved as well.

5.4.1 The Exit problem

The exit problem is a weakness that plagues most other systems with which we are familiar: if a seller intends to leave the market, there is no remaining incentive for her to maintain her good reputation. She is free to cheat as many buyers as she can before completing her exit.

Under Basic Trunits, the seller who wished to extract value from her trunits had two choices—cheat eventually, or engage in trades to infinity. Under Commodity Trunits, a third choice has been added to this set: sell the trunits.

Under this mechanism, the exit problem disappears—for any quantity of trunits, the seller earns more money by honestly selling the trunits than by using them to engage in cheating transactions. This is an extremely important property, and a key result: we are unaware of any other system that solves the exit problem.

5.4.2 The Surplus trust problem

To our knowledge, the surplus trust problem was first identified in our consideration of the Basic Trunits mechanism. It occurs when a seller has a fixed production capacity, and hence, a fixed quantity of trunits required to sell the goods he produces. Due to trunit growth via honest transactions, the seller continually accumulates more trunits than required to maintain his regular business (since he is reusing the same trunits to honestly sell his goods). These surplus trunits can be used for cheating, without consequence.

Commodity Trunits changes this situation dramatically. Again, any given trunit can be used to cheat (for a profit of r) or can be sold (for a profit of b). Since $b > r$, selling the trunits is more profitable than cheating. A rational seller, then, will not cheat, but will sell his trunits instead; the problem disappears.

5.4.3 The Ballot-Stuffing problem

Under Commodity Trunits, a solution to the ballot stuffing problem is available. The motivation for ballot stuffing is the expectation of higher future revenue, based on the artificial increase in reputation. In [2], the idea is introduced that if transaction costs (e.g., commissions) are larger than the expected future gain from a ballot stuffing transaction, then there is an economic disincentive to engage in such transactions. (If no such fees are used, this technique cannot provide protection.) Under Commodity Trunits, ballot stuffing is unattractive if the cost of a fake transaction is greater than the value of the trunits obtained.

Consider a coalition (including both buying and selling agents) that is choosing between two methods of obtaining trunits: ballot-stuffing, or purchasing the trunits honestly. Here, we decompose the selling agent's cost c into two components: operating expense c_{oe}, including all costs in furnishing the good for sale, and selling expense c_{se} (specifically, commission charged by the market operator). All cost components are expressed as fractions of selling price, and $c = c_{oe} + c_{se}$.

In a ballot stuffing transaction, the coalition incurs no operating cost to furnish the good since the sale is a false one, but it will pay the commission imposed by the market operator. Let τ represent the quantity of trunits covering the sale, so $V = r\tau$ is the price of the sale. An honest transaction receives a premium of $p\tau$ trunits. The commission charged for the sale is $Vc_{se} = r\tau c_{se}$. To simply purchase the same quantity of trunits on the open market at price b would cost $bp\tau$.

Ballot stuffing is unattractive, then, if

$$\begin{aligned} r\tau c_{se} &> bp\tau \\ rc_{se} &> bp \end{aligned}$$

It must be the case that $r < b$ in a well-functioning trunits marketplace. Let $r = kb$ for some k, where $0 < k < 1$. (As outlined in Section 5.5.4, the market operator can exert control over r, effectively keeping k constant if desired.) Then:

$$\begin{aligned} kbc_{se} &> bp \\ kc_{se} &> p \end{aligned} \tag{5.3}$$

Since $0 < k < 1$, it is possible to maintain to this relationship: the market operator controls both p and the commission rate, so they can be set to make ballot-stuffing unprofitable. It does, however, require p to be set lower than the commission rate changed by the market operator, resulting in very slow trunit growth (or very high commissions!). Slow growth may or may not be acceptable to marketplace participants. Additionally, it is worth noting that even if transaction costs cannot be set high enough to guarantee that ballot-stuffing is unprofitable, they can still substantially reduce the gain realized from engaging in the activity, and hence reduce the motivation to engage in it.

Note that the ability to make ballot-stuffing unprofitable derives directly from the explicit nature of the incentive mechanism. The practicality of such an approach may be impossible to prove under systems that do not offer an explicit, measurable incentive.

5.5 New issues raised by Commodity Trunits

The Commodity Trunits mechanism provides individual agents with the desired incentive for honest behaviour, while eliminating certain key problems. That said, there are issues, particularly at the system-wide level, that must be considered. These issues are examined in this section.

5.5.1 Limitations of the market operator

The proposed mechanism is dependent on sellers being able to buy and sell trunits whenever they want. If a seller cannot buy trunits when needed, sales will be constrained. If a seller cannot sell trunits when he wishes to do so, the possibility exists that he will cheat instead. A natural question is, who will buy/sell the trunits to/from the agents?

The market operator is the natural choice for such a role, since all trunits would reside in accounts within its system, and since it is in a position to enforce the pricing required for system safety. Unfortunately, in general it does not appear to be financially viable for the market operator to serve in this role.

Since the market operator can create trunits, it can accommodate the sale of trunits without problem. Consider, however, both the initial sale to a customer and the eventual purchase from that customer. For example, an agent may buy 100 trunits from the seller, at a cost of $100b$. It may then engage in numerous honest transactions, with its trunit balance increasing each time. Subsequently, it may seek to 'cash-out' its balance of trunits, which has grown to, say, 1000. This will cost the market operator $1000b$, much greater than the revenue taken in by the initial sale of the trunits ($1000b - 100b = 900b$). Unless the market operator has taken in more than $900b$ in revenue on the agents' sales (commissions, etc.), the operator will lose money in buying back the trunits. We cannot assume that the market operator is willing or able to sustain continual losses of funds due to the operation of the system; this approach is not budget-balanced, making it unlikely to be sustainable in many applications.[1] In fact, there may be no identifiable market operator entity at all; for example, a decentralized implementation might be developed where trunit balance are stored locally and updated in a secure manner. In this case, a mechanism that runs at a financial deficit is not practicable.

5.5.2 Trunit movement between traders

Since the market operator cannot afford to purchase all of the trunits, we look to the other obvious choice—allowing traders to sell trunits to one another. (Allowing the sale of trunits between traders does not necessarily preclude the market operator buying and selling as well, when the operator feels it is appropriate. In fact, it is inevitable that the market operator will be involved in the 'trunit economy' in any case—the operator

[1] If some agents were cheating, and thus trunits were being destroyed, the market operator's total revenue on sales of trunits to all agents might be larger than the total expenditure on repurchasing trunits; in effect, the overall rate of loss due to cheating might exceed the overall natural growth of trunits due to honesty. However, we cannot rely on this: if rates of cheating are large enough to offset the growth in trunits, then the mechanism has failed in its very purpose of preventing cheating!

is required to create/destroy trunits after the honest/dishonest execution of transactions by sellers.)

We have illustrated above that is safe to allow agents to buy and to sell trunits, as long as $b > r$. (We develop more rigorous proofs of this safety in Chapter 6). There is no reason to believe that allowing agents to sell to one another introduces new vulnerabilities—the incentive for honesty demonstrated above is not dependent on the source/destination of the trunits.[2]

It is worth noting that this proposal is consistent with our key requirement for safety. If an agent could buy trunits for price $b < r$, then he would be able to profit by buying trunits and cheating with them immediately. This won't happen, however. If it were the case that $b < r$, the agent with trunits won't want to sell them: instead, he would keep them and use them himself to cheat, realizing a larger profit than by selling them. Thus, trunits sales between agents will only take place under the very conditions that make such sales safe.

5.5.3 A potential problem: Surplus trunits in the market

Although it is safe for agents to buy and sell trunits where $b > r$, unfortunately, this condition is not guaranteed when we allow free buying and selling of trunits among agents. The problem is essentially an economic one, lying in the 'natural' growth of trunits.

Each time an honest transaction is completed, trunits are created, based on the reward ratio p. Trunits are destroyed when an agent cheats. Given the characteristics of the system, however, the rate of cheating should be very low (ideally, zero). It is likely that the growth through trunit creation will exceed the loss from cheating. This suggests a growing trunit supply.

Trunits are required to cover each sale made. Over time, as the volume of sales increases (via entry of new traders, or increased activity per trader), the number of

[2] Note further that the direct sale of a quantity of τ trunits from one agent to another at the price $b\tau$ is numerically equivalent to the sale at $b\tau$ from the first agent to the operator, followed by a purchase at $b\tau$ by the second agent from the operator. If we allow any agent to buy and sell trunits from/to the market operator at will, allowing agents to buy and sell trunits directly from each other is equally safe.

trunits required to support this activity will rise. This suggests increasing demand.

It is a fundamental principle of basic microeconomics that the balance between supply in demand will determine the price of a good in a competitive market. Consider the supply-demand graph in Figure 5.1, representing 'safe' conditions.

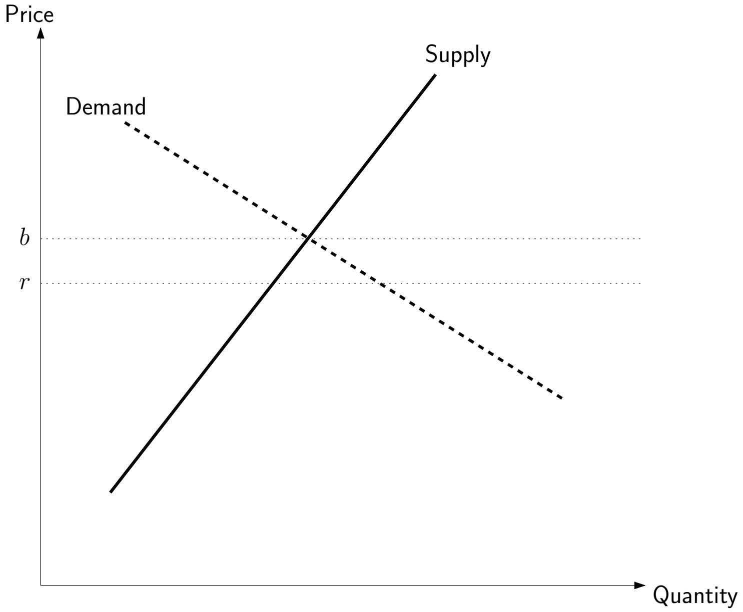


Figure 5.1: A 'safe' market

Supply and demand are balanced in this market, such that the equilibrium price b is greater than r. As discussed above, there are forces that might increase both supply and demand. If trunit growth outstrips sales volume growth, the situation depicted in Figure 5.2 will occur.

The increase in supply relative to demand causes the equilibrium price to drop. If it drops too far, it will fall below r, negating the incentive to sell, rather than cheat. This has two effects:

1. The problems associated with our earlier mechanism return (meaning that, if they choose to leave the market, cheating with their trunits is the most profitable option), and

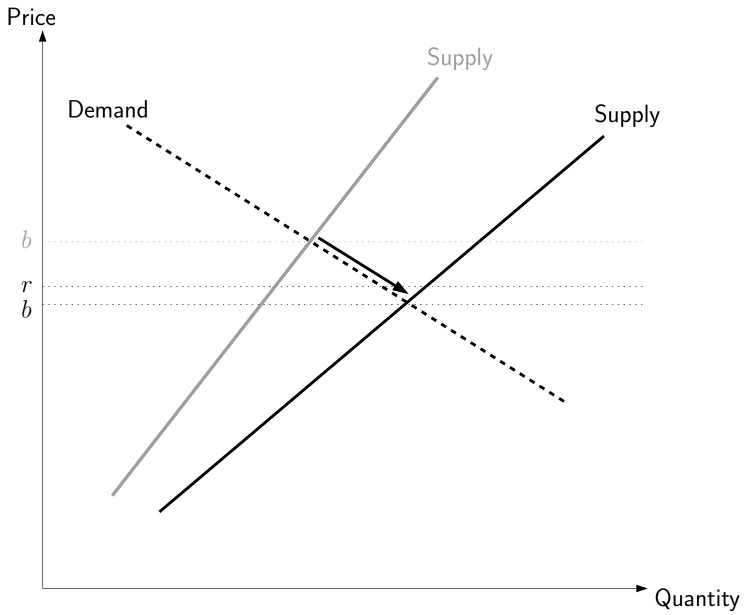


Figure 5.2: Increase in supply

2. It becomes profitable for agents to buy trunits and then use them immediately to cheat.

Is it possible to prevent this from happening? We considered several techniques that the market operator might employ:

- Since all trunit sales take place via the market operator, the operator might simply try to enforce a price. This is doomed to failure, however. If a price is enforced that is above the market equilibrium price, then supply will exceed demand—some agents will not be able to find buyers for their trunits, so their only recourse may be to turn to cheating.

- The market operator might step in to buy only those trunits that aren't purchased by other buyers, in effect 'soaking up' excess supply. In this way, the operator acts similarly to a government enacting monetary policy—buying or selling currency to control the money supply, and manage exchange rates. The operator's ability to do so is again limited by its revenue stream, however. This approach was investigated in our experiments, discussed in Section 5.6.

- The operator may decide not to enforce a fixed required ratio r, and let the market determine the value of trust. This option is discussed in the next section.

5.5.4 A free market for trust

In Chapter 4, we discussed the possibility of allowing a 'free market for trust', in which there is no set required ratio r. Instead, sellers would be free to offer any price/trunit combination they desired, and buyers could decide how many trunits would be required for them to consider a transaction safe. Previously, this discussion centred around the start-up problem and flexibility—for example, a seller could offer an item with few (or no) trunits in order to get started, while the seller could decide if he was getting an attractive enough deal to make it worth the risk. Under our extended model, however, the free market approach would serve a more important purpose.

Consider again Figure 5.2. With b set below r, we are likely to see a surge in cheating in the marketplace. How will buyers react to this? A buyer who has been cheated, or who learns of high cheating rates, is likely to increase the number of trunits she demands to cover transactions, in order to increase confidence in the transaction. This might be termed 'trunit inflation', and will have two effects:

- Since $V = r\tau$, increasing the trunits demanded for a transaction (i.e., τ) effectively lowers r. As many buyers do this, the market rate will decrease, moving the market towards safe conditions—r will decrease until it falls below b, at which point cheating is no longer profitable.

- As more trunits are demanded, the number required to support the volume of sales will increase. This will effectively increase demand, as shown in Figure 5.3. This will increase b, pushing it towards safe territory (i.e., when it exceeds r.)

When $b < r$, these two forces will push towards rectifying the situation. Thus, one might expect that $b > r$ will be an equilibrium condition in the marketplace.

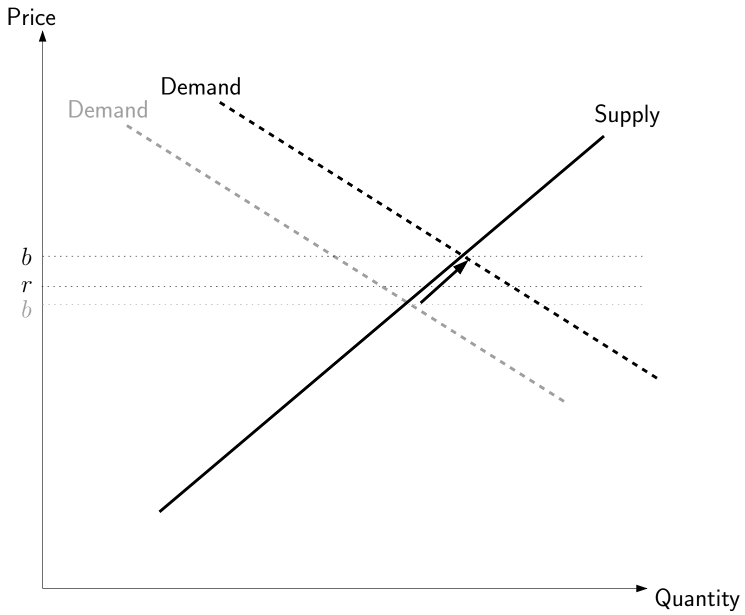


Figure 5.3: Increase in demand, restoring safety to market

It is possible, however, that a destructive phenomenon might occur instead. A seller who has been victimized might not increase his future trunit demands—he may become disheartened and leave the market instead. This will have the effect of reducing the demand for trunits, as sales levels fall. If levels of cheating are high, this might occur with high frequency, overwhelming the beneficial forces described above.

There are clear steps that the market operator can take to encourage the beneficial forces, and minimize the destructive one. For example, one might expect that buyers who are actually cheated might be more likely to leave the market, but buyers who are warned of cheating are more likely to protect themselves by increasing their trunit demands. The market operator can help to maintain a safe equilibrium by providing agents with information: a 'recommended safe ratio' that buyers should follow, data on cheating levels, etc. This will allow the market operator to influence r and b significantly, without the need to actually buy or sell trunits (and without the need for high cheating levels to move market rates). Furthermore, this activity will help agents; in addition to providing the information they need to protect themselves, it also relieves them of the burden of acquiring and processing the information to make their own decisions.

To illustrate, consider such a market, in which the operator has observed that in most transactions, buyers are requiring a ratio in the range of 4 to 6 (i.e., 4 to 6 dollars of sale value per trunit used to secure the sale). Trunits are bought and sold freely in this market; the operator has noted that trunits are currently trading at a price $b = 6.5$, but that as the trunit supply is growing, this price is dropping. The danger exists that, if b drops below 6, then sellers can profit by buying trunits and using them to cheat buyers who are requiring ratios of 6 or higher.

To combat this, the operator can inform buyers of the ratios they should use to ensure their safety. In this case, the operator might publish a maximum safe ratio of 5—since $V = r\tau$, lowering the ratio increases the number of trunits required to cover a sale, offsetting the effects of the surplus. If buyers follow this advice, then they will be safe from cheating, since the profit to be earned from cheating them at a ratio of 5 will be less than the price of the trunits, even if it drops to 6 as projected. Buyers are free to ignore this advice, but they do so at their own risk.

5.6 Simulation: Validation of Commodity Trunits

Safety of Commodity Trunits relies on the price of trunits b never dropping below the required ratio r. The use of simulation to investigate economic markets has a long history (e.g., [5, 22]). To investigate the feasibility of maintaining this condition in a marketplace, we ran extensive simulations under a well-defined scenario. Our simulation method, and our findings, are detailed in this section.

5.6.1 Approach and goals of the simulation

The policies adopted by the market operator play a major role in the operation of that market. In a Commodity Trunits marketplace, the market operator might be passive or active: he might simply allow market forces to determine b and r, and hope for the best, or he might try to actively participate in the market in an effort to maintain a favourable environment. In our simulations, we model a marketplace that starts from ‘day zero’, in which sellers begin with no trunits, and must purchase them to begin trading. Since all sellers are in the same situation, there is no natural trunit supply—the market operator is forced to create new trunits and sell them to agents. (There is no technical impediment to the operator creating trunits at will. In practical terms, however, trunit creation must be undertaken with care in order to maintain confidence in the operator's integrity, and to ensure that excessive supply does not push prices down to unsafe levels.) Thus, our market operator has already been pushed toward an active role. With this in mind, and with the suspicion that an active market operator would improve the possibility of success, we proceeded with an operator that intervenes when necessary to maintain desired market conditions.[3]

The goal of our simulation was not to perfectly model agents, and all of the possible complexity of their strategies. Instead, we sought only to verify that safety can be maintained. As such, we focused on the interplay of supply and demand in determining the price of trunits. Instead of introducing the additional complexity of market-determination of r, we adopted a method discussed in Section 5.5.3: having the market operator determine r at any given point. In an actual market, this might take the form of the operator actually enforcing a required ratio, or notifying buyers of the 'recommended safe ratio', with buyers adhering to it faithfully.

Under these circumstances, the goal of our simulation was simplified. If the market operator can set any required ratio r, he can always set r such that $r < b$, as long as $b > 0$. If b reaches a price of 0, then a seller might earn more by cheating with unneeded trunits than by selling them, and the mechanism has failed to guarantee safety. Thus, we investigated the ability of the mechanism, and the operator, to maintain $b > 0$.

[3] As implemented here, with agents starting with zero trunit balances, Commodity Trunits requires a market operator that is an identifiable entity, one that is capable of holding money in order to execute trunit sales. Further, some of the techniques described below require the operator to make purchases as well. The possibility that Commodity Trunits might be implemented without an identifiable operator entity is discussed in Chapter 8.

5.6.2 Modeling of Agents

Since trunits are only used by sellers, buyers influence the price of trunits only in their impact on sellers. Buyers affect sellers' need of trunits in two important ways:

1. Through the goods that they seek to buy from sellers, which require sellers to own enough trunits to cover the transactions, and

2. Through the quantity of trunits that they demand in order to cover a sale (i.e., by determining r for each sale, in the free market scenario).

To study trunit pricing, then, we need not model buyers explicitly. From a seller's perspective, the sale of goods to buyers can be modeled using frequencies/rates of sales, without even identifying buyers. (We assume, without loss of generality, that each seller sells only one type of good.) In our scenario, the operator sets r, so we do not need to consider the second point.

Agent properties

Each agent has a set of properties, including the following noteworthy values:

- The price for which it sells its goods (V). This is a fixed, advertised price, which is randomly chosen from a uniform distribution with range [1, 100].

- The cost incurred by the agent in selling its goods (c), as a fraction of selling price. This is drawn from a normal distribution, with mean of 0.5.

- The initial rate of sale, expressed in sales per day. This is drawn from a uniform distribution, with range [1, 10].

- The rate of change in the agents' sales volume. This is expressed as an annual percentage change; it is drawn from a normal distribution, with the mean specified by an input parameter to the simulation. The rate of change may be positive or negative.

- The probability that the agent will decide to exit the market. This is a fixed probability for all agents, specified as an input parameter and expressed as the fraction of sellers who will leave the market per year.

- The agent's conservatism, drawn from a uniform distribution with range [0.5, 1]. This reflects differing temperaments that agents might have, in terms of the degree of certainty they require before taking a 'risky' action. For example, an agent with a conservatism of 0.7 will sell surplus trunits today unless there is at least a 70% chance that the price will be higher on a future day within his horizon.

- The agent's horizon, the number of days he looks into the future when deciding whether or not to buy/sell trunits today. Horizons are drawn from a uniform distribution with range [1,10].

Agent behaviour

As noted above, we were primarily concerned with understanding trunit pricing in a free marketplace, and not with realizing sophisticated trading agents. As such, our goal was to keep agents as simple as possible, while achieving results that were valid.

Initially, we considered modeling agents using trivial strategies: they would buy trunits when needed to execute a sale (if it was profitable to do so at the going price) and sell them when no longer needed. This approach is intrinsically flawed, however. An agent will buy trunits to make a sale if it is profitable to do so, but calculation of the profit must consider not only the acquisition cost of the trunits, but also the value of the trunits after the sale, since this is an asset now owned by the agent. This value is reflected by the market price of trunits after the completion of the transaction, since the trunits can be sold at that price. The basic profit formula is

$$profit_{net} = profit_{gross} - b_{t_1}\tau + b_{t_2}\tau(1+p) \tag{5.4}$$

where t_1 is the time at which the sale is initiated, and t_2 is the time at which the sale is completed and the trunits are sold. In order to determine whether a trunit purchase is profitable, then, an agent must have some idea of what the price of trunits will be at time t_2. This requires the agent to forecast future trunit prices, ruling out the simplicity of strategy we had originally considered.

If an agent is forecasting future trunit prices, one cannot reasonably exclude more sophisticated behaviour. For example, consider an agent who has sufficient trunits for his current sales, but who expects to need more trunits in the future. The trivial strategy discussed above would have him wait until the trunits are needed before purchasing them. But what if his forecast reveals that prices are going up? While we want our agents to be simple, we also want them to be rational, and it would seem that a rational agent would take this into account. Here, we would likely expect an agent to buy ahead of time, to take advantage of the lower prices.

This perspective formed the basis of our agents' strategies. Agents forecast future trunit prices based on previous prices, using simple regression analysis. For each predicted future price, there is uncertainty; this uncertainty is incorporated by using prediction intervals based on each agent's level of conservatism. Our agents' behaviour, then, consists of the four following rules:

1. **Buy for current need:** If trunits are needed for the current day to execute sales, buy these trunits if it is profitable to do so, considering both the acquisition cost and forecast selling price of the trunits at completion of the transaction.

2. **Buy for future need:** Beyond any trunits bought for the current day, if the agent expects to need additional trunits within its horizon, purchase those trunits today unless the agent expects (at a confidence level equal to the agent's conservatism) that the price of trunits will be cheaper within its horizon.[4] Since the rate of sales for an agent changes slowly, the expected number of required trunits is estimated using a moving average of past requirements.

3. **Sell surplus:** If an agent has surplus trunits beyond its expected needs, sell those

[4] If prices are rising, an agent might speculatively purchase trunits *beyond* its expected needs, simply as an investment. Attempting to model such behaviour raises real difficulties, however: one would need to consider an agent's available investment capital, returns on alternative investments, the agent's risk preference and portfolio composition, etc. For this reason, our agents do not buy beyond their anticipated needs.

trunits unless the agent expects (at the required confidence level) that the price will be higher within its horizon.

4. **Speculative sale:** If an agent has trunits that it does not need today, but might need in future, sell these trunits if it is expected (at the required confidence level) that they can be repurchased for less within the horizon.

While it is possible to envision more sophisticated strategies, we believe that these strategies constitute a reasonable model of agent behaviour—agents prefer to hold trunits as price increases, and not to hold them as price decreases—a model that is sufficient to establish the viability of a trunit marketplace.

5.6.3 Simulation execution

Each round of the simulation consists of one day. The following is an outline of the sequence of events that occur during each day:

1. Each seller's trunit balance is updated to reflect transactions that completed on that day, returning trunits to the seller for honestly-executed sales.

2. The set of sellers who elect to exit the market that day is determined. Sellers who decide to exit the market make no further sales of goods, but continue to participate in the trunit marketplace until they have sold their remaining supply of trunits (both those held now, and those still to be returned to them from honest sales). Once all in-process sales are completed (i.e., no trunits remain to be returned to the seller), the seller has no further participation.

3. New sellers join the market. The number of new sellers is determined using a Poisson distribution. The average rate of growth in the number of sellers is an input parameter to the simulation, expressed as the annual percentage increase in the number of sellers.

4. For each seller, the number of incoming sales is determined; a Poisson distribution is used, with the seller's current sale rate as the mean.

5. The price of trunits for that day is determined. Essentially, this is a process of finding an equilibrium price where trunit supply and demand are balanced. This process, and the role of the market operator in influencing the process, are discussed in detail below.

    - If the price of trunits has hit zero, then cheating may be a profitable strategy. This is considered a failure of the system, and the simulation terminates.

6. Trunit sales and purchases are executed.

7. Sales of goods to buyers are initiated. Sellers choose whether to execute sales honestly or dishonestly; honest sales are tracked so that trunits can be returned to the seller upon finalization of the transaction (i.e., feedback from the buyer).

8. The required ratio r is updated by the market operator if the price of trunits has dropped below what the operator considers to be a 'safe' margin. Here, the operator ensured that r was 80% of b, at most.

9. Sellers' rates of sale are updated, based on each seller's rate of change.

5.6.4 Market operator intervention: Capping price increases

In initial tests, the market operator allowed the price of trunits to float freely without interference. However, periods of rising prices resulted in a problem, related to the simple strategies of our agents discussed above. When an agent forecasts rising prices in the future, he opts to buy now in advance of his need, rather than waiting to purchase at a higher price. Similarly, agents with surplus trunits do not want to sell immediately; instead, they would rather wait and sell at a higher price. These two effects increase demand and reduce supply, further increasing price, and magnifying the effect. The result is that agents buy trunits up to their maximum foreseen needs. Beyond this point, since our agents were designed to buy only for need, and not for purely speculative investments, they buy no further trunits. The result: sharply increasing prices for trunits, than a nearly instantaneous drop once all sellers have provisioned for their future needs. This drop results in a market crash, and a failure of the system.

We recognized that this effect results from the simple regression-based price forecasting that our agents use. In real world marketplaces, prices do not rise indefinitely based simply on a pattern of increasing price in the past. Real-world traders may ‘sense’ that prices can’t rise much further; more formally, in many markets there are indicators and benchmarks to help traders estimate likely trading ranges and peak prices. For example, in stock markets where prices are (to a degree) determined by the profitability of the company, measures such as price-to-earnings ratio can reveal if a stock is over- or under-priced.

With this in mind, we considered enhancing our agents’ strategies in an attempt to allow them to forecast peaking prices. Unfortunately, the difficulty in devising such a system is complicated by the fact that with trunits, there are no obvious measurements that individual agents might use for such predictions. Moreover, while we realized that the observed behaviour might be unlikely with real buyers and sellers, it *is* a *possible* behaviour, and as such, we would like our model to be robust enough to deal with it. We did not want to devise agent behaviour specifically to suit our model.

As was noted in Section 5.5.3, the seller cannot buy unlimited quantities of trunits for financial reasons, so he cannot completely prevent price drops by buying up excess trunit supply. The market operator is free, however, to create as many trunits as desired, so he *can* completely control price increases—if he does not want price to exceed a certain level, he can supplement the natural supply with newly created trunits until the equilibrium price is lowered to the desired level.

Periods of falling prices raise none of the issues encountered by rising prices. When prices are dropping, agents will tend to buy trunits when needed (if it is profitable to do so), and to sell excess trunits quickly. (If prices are dropping fast enough, agents might even sell trunits that they expect to need later, in the hopes that they can repurchase them later at a lower price. Such a situation might result in a different sort of self-reinforcing cycle, where prices drop at an accelerating rate. This effect might be prevented by using transaction costs (e.g., fees/commissions on the sale of trunits) to make such speculative sales unattractive. That said, although the ‘speculative sale’ strategy was implemented in our agents, it did not pose a significant problem—no special steps needed to be taken for us to realize stable, well-functioning marketplaces.)

With this in mind, we experimented with two possible market operator policies:

1. Cap the maximum trunit price at the initial price, or
2. Never allow the price to exceed the previous day's price.

When capping at the initial price, we saw dramatic price fluctuations within the allowable range, again attributable to the simple agent behaviour described above. Such price fluctuations are not particularly good for agents; it is difficult enough for businesses to cope with the economic forces at play in the markets for their actual goods, without introducing unpredictable trunit marketplaces. Moreover, in some cases this instability resulted in trunit market failures. In contrast, the non-increasing price policy led to very stable, predictable trunit marketplaces. This policy was used throughout the remainder of our simulation.

5.6.5 Market operator intervention: Buffering price decreases

While dropping prices are preferred to rising prices, we do not want prices to drop too quickly, for several reasons. First, if prices are dropping dramatically, it might not be profitable for agents to engage in sales at all: the gross profit from the sale might be less than the loss incurred by buying trunits and then selling them at a dramatically lower price. Moreover, the velocity of a quickly dropping price increases the danger that prices may crash entirely, reaching zero.

Ensuring that prices do not drop at an inappropriate rate is partly a matter of setting market parameters properly. If trunit growth does not greatly exceed actual sales growth in the marketplace, prices should not drop precipitously. (The interplay of market parameters is examined in our simulation results, presented in Section 5.6.7.) Beyond this, the market operator can also take a key role in moderating price decreases.

As noted in Section 5.5.3, the seller cannot buy unlimited numbers of trunits to control price drops, for financial reasons. The operator *can*, however, buy *some* trunits in an attempt to buffer price drops. In the process of creating and selling new trunits (e.g., at start-up, or when capping upward price increases) the market operator earns revenue. This revenue can be spent to purchase trunits in times of decreasing prices, 'mopping up' excess supply and moderating price increases.

We experimented with two policies. The first, referred to as *limit price drop* is quite simple: (arbitrarily) setting a maximum desired rate of price decrease, the operator buys trunits any time a drop would exceed this rate. If he does not have enough accumulated revenue to do so, then prices drop as far as they will naturally. To explain the second policy, we must first delve deeper into how equilibrium prices are determined in our simulation.

Imperfectly balanced equilibria

In each day of our simulation, we find an equilibrium price where supply and demand are balanced; all trades that day are then executed at that price. (This method is a simplification, based on microeconomic theory, used instead of developing a full bid-ask marketplace.)

It must be noted that in our simulation, supply and demand are usually imperfectly balanced. This is a result of the nature of trunits, and the strategies of our agents. Consider an agent seeking to buy trunits. Based on market parameters, his forecast of future prices, and his confidence level, an agent determines the maximum price that he would be willing to pay for trunits. Below this price, he expects to earn profit, while above this price, he expects to incur a loss. The agent's individual demand, then, is a sharp threshold function where he either buys all of the trunits he needs, or none. (In fact, there are two thresholds, one below which he will buy trunits for immediate use, and another below which he will also buy trunits for future use.) For this reason, the overall demand curve for the trunit market consists of 'steps', rather than a smooth curve. For exactly the same reasons, supply behaves in the same manner.

Given the stepwise nature of the supply and demand curves, it is often the case that no price exists where supply and demand are perfectly balanced. Thus, we have a dilemma: we must choose between one price, where supply exceeds demand, or a slightly lower price, where demand exceeds supply. Based in our understanding that the market operator cannot buy unlimited quantities of trunits, our original policy was to always choose the slightly lower price. Using this strategy, demand will always exceed supply; the market operator can create trunits to meet the unsatisfied demand, resulting in an artificially-balanced equilibrium. While this policy works, it has two undesirable consequences:

- The continual creation and insertion of additional trunits into the marketplace ultimately increases overall supply, increasing the rate of price decrease.
- The market operator realizes extraordinary revenues from trunit sales, which might undermine his credibility and traders' confidence in the market.

Buffering price drops while balancing equilibria

To eliminate these problems, we arrived at a second market operator intervention strategy, referred to as *buy at equilibrium*. When determining the equilibrium price, the following method is used when a perfect balance of supply and demand cannot be achieved:

- At the lowest price where supply exceeds demand, if the market operator can afford to buy up the excess trunit supply, he sets this price and does so.
- If he cannot afford to purchase the excess supply, he chooses the slightly lower price, where demand exceeds supply, and creates additional trunits as necessary.

This policy ensures that the operator never incurs a deficit in running the marketplace. It avoids the problem of unnecessarily contributing to increasing supply; in fact, it means that the seller is regularly reducing the overall trunit supply, buffering price drops.

It should be noted that both policies have another, critical benefit: they can foster traders' confidence in the market. If the market operator incurs sizable profits by creating and selling trunits, this may call into question the integrity of the operator. For example, the operator might manipulate market parameters, encourage unfair reviews, etc., in an effort to increase demand and sell more trunits. In contrast, if the operator spends all such revenue in buying back trunits, and manages this function transparently, then it will be clear that the operator has no financial incentive to 'cheat'.

These two policies are compared in the simulation results, below.

5.6.6 Market parameters

Beyond those values described above, other important market parameters, and their settings in our simulation, include:

- The duration of transactions in our simulation (i.e., the length of time between the sale, and the arrival of feedback and return of trunits to the seller) is fixed at 14 days.
- The initial number of sellers was set to 100. In some trials with 1000 sellers, very similar results were obtained to those with 100 sellers, but the simulations were too time-consuming to permit the full investigation presented below.

5.6.7 Simulation results

Existence of sustainable marketplaces

Initially, we sought to answer two questions: a) Is it possible to sustain a long-term market for trunits? b) How will prices change over time in such a marketplace? To answer these questions, we experimented with parameter values in ranges that we intuitively considered to be reasonable. The results of one such representative run are depicted in Figure 5.4. In this simulation, the market was in operation for 10,000 days (more than 27 years). The rate of growth in the number of agents was set at 20% per year, the average change in agents' sales volumes was 10% per year, and the rate at which agents exit the market was 10% per year.

The market price of trunits b never reached zero over the span of 10,000 days, and b never fell below the required ratio r—cheating was never more profitable than honesty. This confirms that it is, in fact, possible to sustain such a trunit marketplace. Prices dropped to levels where their behaviour is difficult to discern in this figure; Figure 5.5 depicts the same result, using a logarithmic scale for the y-axis.

Several points should be noted here:

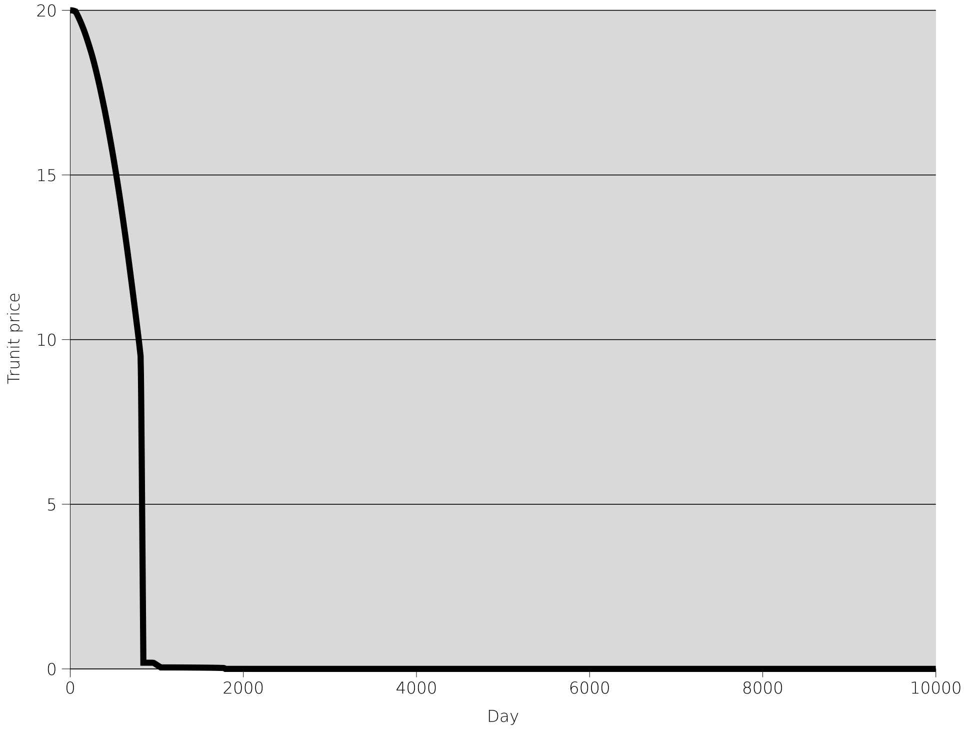


Figure 5.4: A single simulation run

- Price drops in a stable, predictable manner. This is desirable, since it makes it easier for agents to predict future prices. The ability to make accurate predictions makes the market safer, and hence more attractive, to sellers.

- As this market would continue to run beyond 10,000 days, the pattern suggests that prices would continue to drop but remain above zero.

- Prices appear to drop very quickly, to very low levels. This might initially seem to be troubling: agents may not want to invest in trunits, since quickly dropping prices mean lost value for those holding the trunits. It is worth reiterating, however, that this chart depicts more than 27 years; price decreases are not as fast as they may appear. More importantly, these dropping prices are factored into the agents' decision making behaviour—they did not consider the rate of decrease in prices a

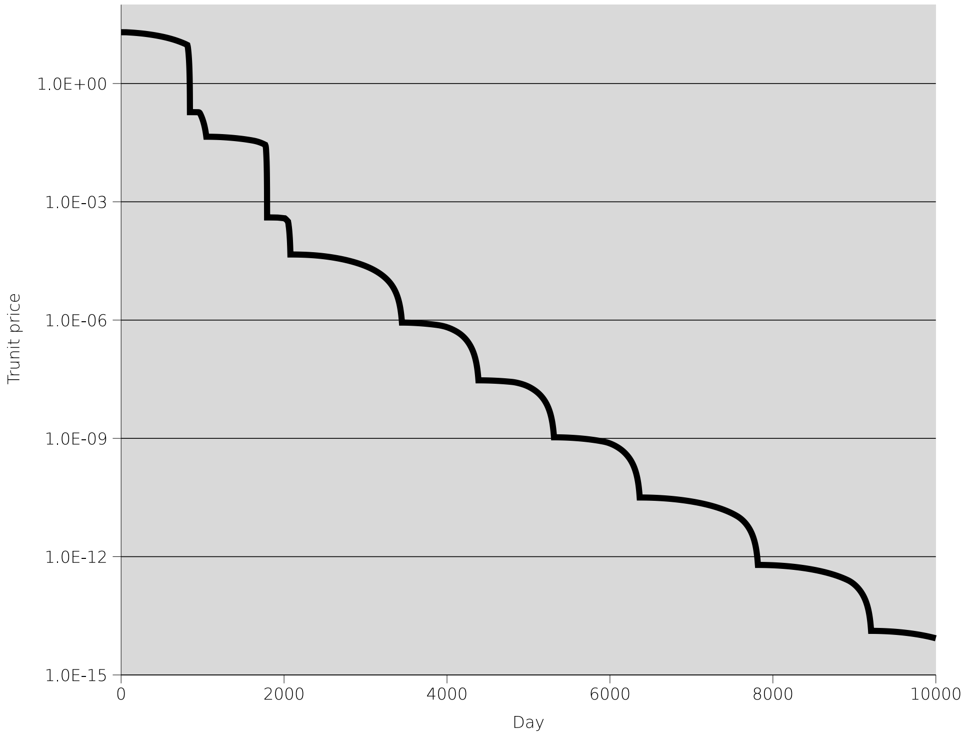


Figure 5.5: A single run, with logarithmic price axis

deterrent, and continued to buy trunits and execute sales. This means that they still found it profitable to operate in the market.

- As market prices drop, the numbers involved become so small as to be cumbersome. (Similarly, since r is lowered as well, the number of trunits required for sales becomes extremely large.) To remedy this, it is possible to reduce the scale of the numbers by *consolidating* trunits. We experimented with this approach—any time prices would drop below a certain threshold, we would execute a 10-for-1 *merge* of trunits. Under such a merge, for every 10 trunits currently held, the agent would receive 1 new trunit, while r was similarly modified; prices responded accordingly. In our experiments, this policy was found to be sound, having no impact on market operation beyond maintaining values in convenient numerical ranges. This form of merger is an established practice in real-world stock markets, and used for essentially the same reason.

Comparing price buffering methods

As discussed above, we investigated two methods for the market operator to buffer the speed at which trunit prices drop: *limit price drop* to a specified maximum rate, where available resources permit, or *buy at equilibrium*, where the agent buys trunits if possible in the process of obtaining a balanced equilibrium price. We ran experiments to compare these techniques; the results of one run using each method are depicted in Figure 5.6.

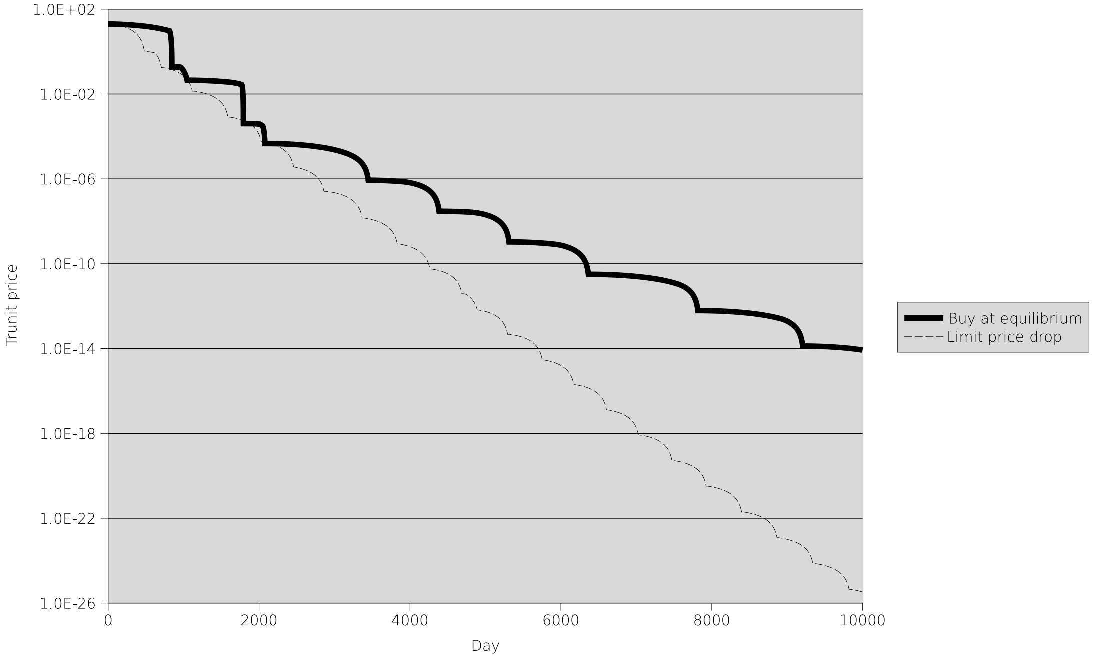


Figure 5.6: Market operator policies compared

Two patterns are evident. First, *buy at equilibrium* results in prices dropping at a slower rate than *limit price drop*. This is predictable, since the former has the market operator buying trunits at every opportunity, reducing supply as soon and as quickly as possible. Second, *limit price drop* results in even more stable/predictable price behaviour than *buy at equilibrium*. Those implementing markets might choose which of these features has higher priority. For the remainder of our simulations, we used *buy at equilibrium*.

Investigating the impact of market parameters

The results above verify that scenarios exist under which it is possible to manage a sustainable marketplace for trunits, one that preserves the incentive for honestly. From here, we investigated the range of market parameters for which we could expect a market to succeed.

Again, our goal is to maintain a non-zero price for trunits. Supply and demand are the key determinants of trunit price. Thus, we focused our study on market parameters that directly impact supply and demand. Supply, driven by trunit availability, is most directly impacted by the premium for honesty p, since new trunits are created after each honest transaction. Demand is driven by the need to 'cover' sales, so parameters relating to sales volume are important: the rate at which new agents enter the market, the rate at which agents leave the market, and the average change in agents' sales.

Intuitively, it seems that trust should grow slowly, as each agent demonstrates its trustworthiness. We sought to investigate this intuition in our simulations: values for p of 0.2, 0.5, and 0.8 were tested.

It could be argued that we need not study the effects of the three sales-related parameters individually; increases in the number of agents by 25% and in agents' sales by 25% are likely to have a similar effect to an increase in agents' sales by 50% (with no change in the number of agents), since the impact on total sales volume will be similar. There may be important differences between seemingly similar sets of parameters, however. For example, the change in total sales volume will be similar under an agent growth rate of 25% with an agent exit rate of 25%, and under growth and exit rates of 0%. The impact on the trunit marketplace might be different, however. In the former situation, agents leaving the market will be selling trunits, while new agents are buying; these transactions don't take place in the latter case.

Thus, we tested a range of values for the rate of growth in the number of agents (0%, 25%, and 50%), the average rate of change in agents' sales (-25%, 0%, 25%, and 50%), and the rate at which agents exit the market (0%, 25%, 50%, and 75%). All values are annual rates. While we believed it was important to include the effects of the separate parameters in our simulation, for presentation we have consolidated our results using 'composite growth rates', reflecting the net effect of the variables. For example, rates of agent growth of 25%, change in sales of 50%, and agents exiting of 25% are reflected in the chart as a approximate composite growth rate of $25\% + 50\% - 25\% = 50\%$.

Five simulation runs were executed for each combination of the four parameters. Simulations ran for 10 years (3650 days).

Before this simulation, we expected that market failures might result from high values of p (since trunit creation enhances supply) and for low rates of market growth (since lower sales volumes result in lower demand). The results of these simulations are reflected in Figure 5.7, showing the number of market failures for each combination of values.

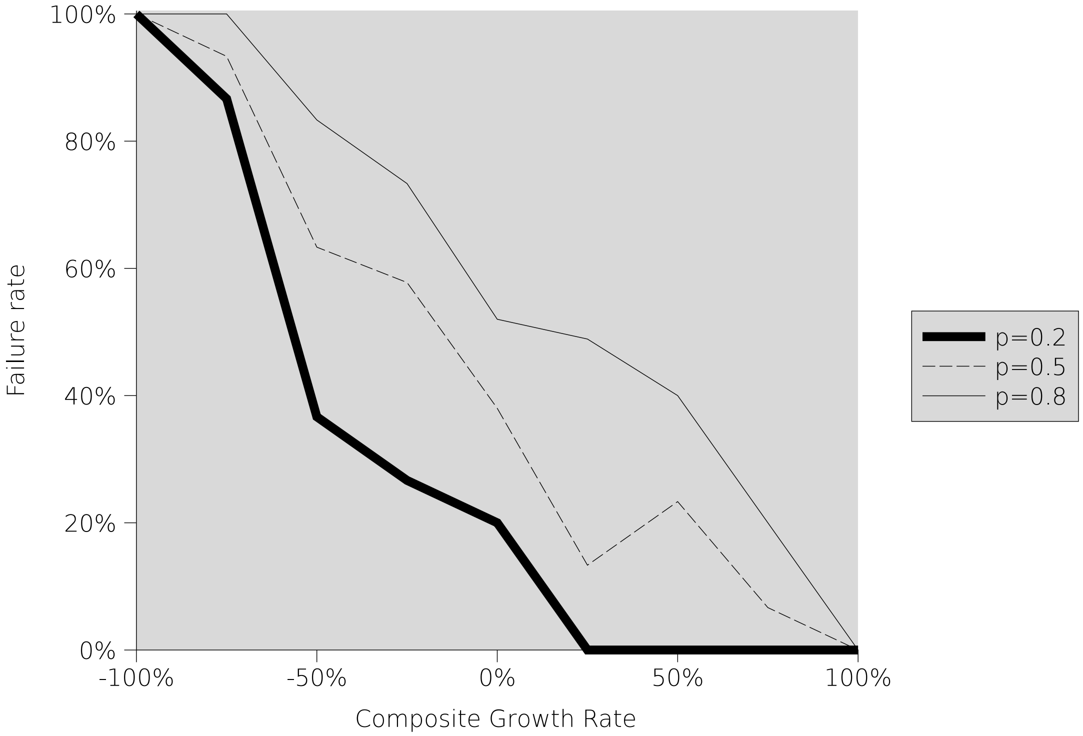


Figure 5.7: Effect of trunit and market growth on market success

Our expectations were confirmed to be correct: relatively high values of p (0.5, 0.8) resulted in high failure rates, as did low (specifically, negative) rates of market growth. In contrast, what we considered to be a reasonable value for p (0.2) resulted in no failures for any positive level of market growth. High rates of growth resulted in lower failure rates, with no failures occurring when growth was 100% per year.

Based on these results, we note the following:

- If overall sales growth is positive, the trunits market is very likely to be sustainable for p values of 0.2 or less.[5] (If overall growth is negative, this often has larger business/economic implications, aside from concerns about the trunit marketplace.)

- Values of p below 0.2 might yield sustainable marketplaces even under conditions of sales decline. Further investigation is required to confirm this.

In short, for a range of realistic market conditions, trunit marketplaces are sustainable.

5.7 Trunits vs. bonds, revisited

In Chapter 4, we contrast the Trunits model with a bond-based scheme, where a cash bond would be placed by the seller to cover each sale. Despite the similarities between the approaches, we showed a number of differences that, while perhaps not obvious at first, are important. However, in the case where we allow trunits to be bought and sold, the distinctions between the two methods are further blurred, and merit further discussion—if a seller puts up money (i.e., buys trunits) to cover his sales, then gets the money back after successful execution (i.e., sells the trunits), aren't we simply talking about bonds? It is worth noting that this similarity with bonds is not necessarily bad. We did not seek to distance ourselves from a bond scheme because it is ineffective at inducing trustworthy behaviour—in fact, it is quite powerful in this regard. Rather, we

[5] This may seem trivial, that a growth rate of 20% in trunits is offset by a positive growth rate in sales. Note, however, that while growth rates are *annual* rates, trunits can potentially grow by 20% every 14 days (i.e., the duration of a transaction.)

acknowledge some key limitations of a bonding system, while noting some of the key features of Trunits:

- A bond is money. A trunit is commodity, one that may be bought and sold. In effect, a trunit is a license that allows the seller to engage in sales of up to r dollars at a time. Cash cannot (or should not!) be created or destroyed by the market operator. Trunits can be freely created or destroyed.

- In the case of cash bonds, something must be done with the cash in the event the seller cheats. The problem here is that whoever receives this cash, whether market operator or buyer, has an incentive to treat the seller unfairly. Since trunits that are 'lost' by the seller are simply destroyed by the market operator, there is no such incentive under the Trunits model.

- Honest transactions allow the seller to grow his trunit balance substantially, something that is not possible using standard bonding, since input of additional funds would be required. This potentially frees the seller from having to put up enormous amounts of capital to cover a volume of sales—significant capital outlay is required only if the seller wishes to 'jump-start' his sales volume, rather than actually building trust.

Seamless handling of short- and long-term traders

This last point highlights an interesting property of the mechanism. While long-term traders can build trust, for traders who wish to conduct small numbers of sales, the system is very similar to bonding. This is quite positive, in fact. Most trust/reputation systems make it difficult for a short-term seller to effectively sell goods (or get full value for them) because his reputation is undeveloped. Using the best known example, under the eBay feedback system, sellers with established histories (obvious from their high feedback scores) are strongly preferred over new sellers. Moreover, if a system does allow such new participants to sell effectively, it is very difficult to ensure they will be trustworthy, given their short-term focus. The method presented here overcomes these obstacles as well as a cash-bond system, but accomplishes this seamlessly within the context of a system that allows traders to build trust/reputation over the long term.

Commodity Trunits provides protection against the exit problem; we are aware of no other system that does so. It can be configured to protect against ballot stuffing. Further, it alleviates the start-up and surplus trunits issues present in Basic Trunits, while retaining the beneficial properties of the simpler model. For Commodity Trunits to be considered safe, it is necessary for the price of trunits to remain above the required ratio for securing sales. Our investigation shows that it is possible for a market operator to maintain this safety property under a range of realistic market conditions.

In the next section, we lay out a framework for rigorously analyzing the security provided by trust and reputation systems. Within this framework, we consider more formally the security of the Basic Trunits and Commodity Trunits models, arriving at provable guarantees of the degree of protection offered, and under what circumstances.

Chapter 6

The Security Framework

6.1 The need for rigorous security analysis

In Chapter 2, we presented a catalogue of vulnerabilities in trust and reputation systems. In Chapter 3, we examined a number of existing models of trust and reputation, and discovered that such vulnerabilities were common. The results of this study served as inspiration for the Trunits model; however, they also highlight the need for greater attention to security, and for more rigorous security analysis during the development of such models. Motivated by work in the field of cryptography, we seek methods for ensuring the provable security of trust and reputation systems. In this chapter, we develop a security framework that may be used in the analysis of such systems. We then apply this framework to analyze the Trunits mechanisms described in this thesis.

6.1.1 What does it mean for a trust/reputation model to be secure?

The Cryptographic Perspective

As noted above, our approach has been inspired by work in the field of cryptography. "Cryptography is about the prevention and detection of cheating and other malicious activities" [20], which parallels the aims of designers of trust and reputation systems.

In Cryptography, for an interaction to be 'secure', "...all parties to a transaction must have confidence that certain objectives associated with information security have been met". The set of primary goals in cryptography consists of *confidentiality* (information is kept from those who are not authorized to view it), *data integrity* (unauthorized alteration can be detected), *authentication* (parties identify each other, and the source of the data can be determined), and *non-repudiation* (previous communications cannot be denied by a party) [20].

In cryptography, the view is taken that "Until a protocol is proven to provide the service intended, the list of possible attacks can never be said to be complete." While it may be possible to establish that a particular cryptographic method offers *unconditional security* or *perfect security*, it is acknowledged that this may not be achievable for all methods; cryptographers thus consider measures of security that are 'less than' unconditional. Once such evaluation model is that of *provable security*, where "...'provable' here means provable subject to assumptions." Provable security "... is considered by some to be as good a practical analysis technique as exists." [20]

Security in trust and reputation systems

If our aim is to ensure that a trust/reputation system for marketplaces is secure, we must first characterize what we mean by security: a set of goals. 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 [25]. As discussed below, it is extremely difficult to deliver unconditional security in any system that might be practically applied; instead, we seek to deliver provable security.

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.[1] Development of required safety properties can be simplified by considering

[1] In some marketplaces, an 'operator' can be identified. For example, in the case of eBay, there is a corporation that operates the market, a corporation that has interests that should be protected from dishonest behaviour. In other marketplaces, however, no identifiable 'operator' exists. For example, in a peer-to-peer system, participants may simply execute trades directly with one another, without a central market oper-

the needs of each stakeholder group separately. In this thesis, we focus primarily on the security of buyers, for two reasons:

1. Protecting buyers from cheating sellers is a predominant focus of current research. This may be because, in the marketplace scenario typically considered, buyers are more vulnerable than sellers in any given transaction, due to information asymmetry: the seller has access to information about the actual good that will be delivered, information that the buyer does not possess [7].

2. The protection of buyers is the explicit goal of our Trunits model, so we are particularly interested in characterizing the level of buyer security provided.

We discuss the security requirements of sellers, and of the market operator, in less detail, leaving a deeper examination for future work, as discussed in Chapter 8.

6.2 The Security Framework

Given the marketplace scenario, individual transactions consisting of a sale from one agent (seller) to another (buyer) are the fundamental unit of activity. 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

ator. In such cases, ‘market security’ might refer to protection of the interests of the implementer of the market, or simply the protection required for the market to continue to function.

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 $honest(t_D)$. (This parallels the perspective take in, for example, [29], where 'successful' transactions are those in which an agent meets his obligations, or fulfills the explicit terms of a contract.) The details of how a good or promise is specified are left to the system designer. For instance any $d_d \leqslant d_p$ may be considered honest. We consider a transaction 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.[2]

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 values of individual transaction parameters. Each such function has the same name as the parameter that it returns.

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 of fails to meet the advertised promise), or deciding not to deliver the good at all.

For any coalition of sellers G, consider a 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

[2] Note that buyers will often also close off a given transaction by computing a rating for the seller. The timing of when ratings are elicited from buyers may vary by system, however. Thus, we refrain from defining a rated transaction for generality.

unto itself, however, since the coalition might also strategically choose the composition of T_D by choosing the transactions into which its members will enter.)

Not all schedules can actually be executed. For example, a seller 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. For 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

$$P(G, T_D) = \sum_{t_D \in T_D | A_s(t_D) \in G \wedge A_b(t_D) \notin G \wedge feasible(t_D, T_D)} P(t_D) \tag{6.1}$$

6.3 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 he pays 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 Chapter 8.

In our framework, for a system to be secure we do not hold the seller responsible for the buyer's complete satisfaction—a buyer may have very unreasonable expectations, ones that cannot reasonably be met by the seller. 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 he will receive. Should he purchase the good, he would expect that it corresponds to each claim made in the offer—this is the essence of an *agreed transaction*, discussed above. This, then, is the basis for our notion of *buyer security*: a buyer will be secure under a trust system if

$$\forall t_D \in T_D : honest(t_D) \tag{6.2}$$

Note that buyer security directly addresses certain issues identified in Chapter 2, such as the value imbalance and reputation lag problems. Other forms of dishonest behaviour may not be directly addressed by the buyer security safety properties, since they may not be directly 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.

6.3.1 Levels of buyer security

We might term the previous property, should it hold, as *full buyer security* (or *unconditional 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 [7].

While it may not be feasible for a system to guarantee this property in every scenario, it may be possible to achieve it under certain conditions, when certain assumptions hold for the marketplace. By limiting the guarantee to only 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:

$$(assumption_1 \wedge \cdots \wedge assumption_m) \Rightarrow \forall t_D \in T_D : honest(t_D) \tag{6.3}$$

The assumptions denote limitations in the system, which prevent it from fully 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:

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.

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 sellers' best interests 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:

$$\text{selling agents are rational} \Rightarrow \forall t_D \in T_D : honest(t_D) \tag{6.4}$$

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

$$[\forall G : rational(G)] \Rightarrow \forall t_D \in T_D : honest(t_D) \tag{6.5}$$

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:

$$[\forall G : \forall T_{D1}, T_{D2}[P(G, T_{D1}) > P(G, T_{D2}) \Rightarrow T_{D1}\text{is selected}]] \Rightarrow \forall t_D \in T_D : honest(t_D) \tag{6.6}$$

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:

$$[\forall G : rational(G) \wedge |G| = 1] \Rightarrow \forall t_D \in T_D : honest(t_D) \tag{6.7}$$

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 would be considered an instance of collusion, where multiple user accounts are acting in concert.)

$$[\forall G : rational(G) \wedge |G| = 1 \wedge \forall t_D \in T_D : A_b(t_D) \notin G] \Rightarrow \forall t_D \in T_D : honest(t_D) \tag{6.8}$$

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. That said, 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.):

$$\begin{aligned}
&[\forall G : rational(G) \\
&\quad \wedge \forall T_D[honest(T_D) \wedge feasible(T_D) \\
&\qquad\quad \Rightarrow \exists t \notin T_D : (honest(T_D \cup \{t\}) \wedge feasible(T_D \cup \{t\}))]] \\
&\hspace{20em} \Rightarrow \forall t_D \in T_D : honest(t_D)
\end{aligned} \tag{6.9}$$

Other security concerns

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, systems are of unknown security at best and (based on the results of our survey in Chapter 3) likely to be insecure.

6.4 Security analysis of Trunits

This framework was designed with the goal of allowing security analysis of any trust/ reputation system targeting a similar marketplace scenario. Having outlined a framework for establishing security guarantees and enumerated a number of important levels of security, we provide no general 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.

Here, we revisit our Trunits mechanisms, applying the framework to precisely characterize the degree of security they provide. The purpose of this is twofold: a) to provide a more rigorous and complete analysis of the security of Trunits than was offered in the preceding chapters, and b) to illustrate the use of the framework, to serve as an example for other system designers.

6.4.1 Buyer Security in Basic Trunits

We seek to verify that the essential buyer security property, $\forall t_D \in T_D : honest(t_D)$ will hold. We understand already that certain conditions must be met for Basic Trunits to work successfully, and that certain limitations exist. We outline these now, as a starting point for our analysis.

As an incentive mechanism, Basic Trunits relies on agent rationality to ensure desirable behaviour: thus, we target rational-agent security. Essentially, under Basic Trunits 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 might actually relax this assumption under Trunits, instead specifying with precision an acceptable range in the degree of control, but this possibility requires further study; it is discussed in Chapter 8.)

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-agent 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). As will be shown below, however, there is a strong incentive not to exit the market; our analysis is conducted under the assumption that the infinite-transaction property holds, where the agent can engage in infinite honest sales if desired.

This analysis is based on the assumption that selling cost is a fixed fraction c of selling price. While we do not believe that this constraint is required for Trunits to be secure, it has been assumed in order to simplify analysis; since it has been assumed, we will include it as an assumption in our guarantee.

Security guarantee of Basic Trunits

What we seek to prove, then, is:

Basic Trunits is in use $\wedge$
selling agents are rational (A) $\wedge$
selling agents act alone (B) $\wedge$
selling agents can engage in infinite honest transactions (C) $\wedge$
buying agents are honest (D) $\wedge$
selling agents can reliably meet commitments if willing (E) $\wedge$
cost c is a constant percentage of selling price (F)

$$\Rightarrow \forall t_D \in T_D : honest(t_D) \quad (6.10)$$

Specified more formally:

[Basic 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$]\wedge$
$\forall G : |G| = 1 \wedge$
$\forall T_D[honest(T_D) \wedge feasible(T_D)$
$\quad \Rightarrow \exists t \notin T_D : (honest(T_D \cup \{t\}) \wedge feasible(T_D \cup \{t\}))]\wedge$
buying agents are honest $\wedge$
selling agents can reliably meet commitments if willing $\wedge$
cost c is a constant percentage of selling price]

$$\Rightarrow \forall t_D \in T_D : honest(t_D)$$

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 (including reward) at completion, the balance of trunits available at any given *time* is:

$$\tau_{bal}(time) = \tau_{init} - \sum_{i \in T_D | start(i) \leqslant time} \tau_i + (1 + p) \times \sum_{i \in T_D \setminus C | end(i) \leqslant time} \tau_i \tag{6.11}$$

Recall that C is the cheating set, the subset of transactions in T_D that are executed dishonestly.

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 all transaction in T_D. 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 i to C never increases τ_{bal}, but will lower it (specifically, after $end(i)$). 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.

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 $C \subseteq T_A$,

$$P(T_A, C) < P(T_A, \varnothing). \tag{6.12}$$

Note that by our assumptions, the seller acts alone. Further, we make use of a Basic Trunits system that does not allow a seller to sell goods to himself in order to increase his trunit balance. This means that all cash and trunit flows come only through transactions with real buyers.

The expected profit function for Basic Trunits requires explanation, 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 expected total profit will be the sum of the profit from honest sales, the profit from cheating sales, and the revenue from the 'final cheat' (using the remaining trunits) after the schedule has completed:

$$\begin{aligned} P(T_A, C) &= (1-c)r \sum_{i \in T_A \backslash C} \tau_i + r \sum_{i \in C} \tau_i + r\tau_{bal}(exit) \\ &= (1-c)r \sum_{i \in T_A \backslash C} \tau_i + r \sum_{i \in C} \tau_i + r \left(\tau_{init} - \sum_{i \in T_A} \tau_i + (1+p) \sum_{i \in T_A \backslash C} \tau_i \right) \qquad (6.13) \end{aligned}$$

(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:

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

$$
\begin{aligned}
& +r(1+p) \sum_{i \in C_2 \backslash C_1} \tau_i \\
= \quad & (1-c)r \sum_{i \in C_2 \backslash C_1} \tau_i - r \sum_{i \in C_2 \backslash C_1} \tau_i + r(1+p) \sum_{i \in C_2 \backslash C_1} \tau_i \\
= \quad & (1-c-1+1+p)r \sum_{i \in C_2 \backslash C_1} \tau_i \\
= \quad & (1-c+p)r \sum_{i \in C_2 \backslash C_1} \tau_i \qquad (6.14)
\end{aligned}
$$

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 $C \subseteq T_A$, $P(T_A, C) < P(T_A, \varnothing)$.

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 6.13 simplifies to:

$$P(T_A) = r(1-c+p) \sum_{i \in T_A} \tau_i + r\tau_{init} \qquad (6.15)$$

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:

$$P(T_A \cup \{t\}) = r(1-c+p) \sum_{i \in T_A \cup \{t\}} \tau_i + r\tau_{init} \qquad (6.16)$$

$$= r(1-c+p) \left(\sum_{i \in T_A} \tau_i + \tau_t \right) + r\tau_{init} \qquad (6.17)$$

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,

$$
\begin{aligned}
&[\text{Basic Trunits is in use} \wedge \\
&\forall G : |G| = 1 \wedge \\
&\forall T_D[honest(T_D) \wedge feasible(T_D) \\
&\qquad \Rightarrow \exists t \notin T_D : (honest(T_D \cup \{t\}) \wedge feasible(T_D \cup \{t\}))] \wedge \\
&\text{buying agents are honest} \wedge \\
&\text{selling agents can reliably meet commitments if willing} \wedge \\
&\text{cost } c \text{ is a constant percentage of selling price}] \\
&\qquad\qquad \Rightarrow \forall T_A : [C \subseteq T_A, |C| > \varnothing \Rightarrow P(G, T_A, \varnothing) > P(G, T_A, C)]
\end{aligned}
$$

This yields:

$$
\begin{aligned}
&[\forall G : \forall T_{D1}, T_{D2}[P(G, T_{D1}) > P(G, T_{D2}) \Rightarrow T_{D1} \text{is selected}] \wedge \\
&\forall T_A : [C \subseteq T_A, |C| > \varnothing \Rightarrow P(G, T_A, \varnothing) > P(G, T_A, C)] \\
&\qquad\qquad \Rightarrow \forall T_A : (T_A, \varnothing) \text{ is selected } \Rightarrow \forall t_D \in T_D : honest(t_D) \qquad (6.18)
\end{aligned}
$$

Essentially this means that since it will be less profitable for a rational seller to cheat, she will execute every transaction honestly. Thus, the Trunits mechanism can provide the user with a guarantee of security, at this level.

The (labeled) list of assumptions given in Formula 6.10 identifies the limitations of the Basic Trunits mechanism. Understanding these, how can we be sure that the mechanism will be secure? We address each assumption below:

- 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. This issue is discussed in Chapter 8.

- The need for infinite transactions (C) can be addressed through enhancements to the Basic Trunits mechanism, such as the Commodity Trunits mechanism. We analyze Commodity Trunits in the following section.

- 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. This is discussed in Chapter 8.

- The assumption that sellers can control quality (E) can actually be refined to specify the degree of control required, as discussed in Chapter 8.

- 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.

This analysis gives us a clear picture of exactly what guarantee Basic Trunits provides, and under what conditions—Basic Trunits has achieved a 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. Indeed, this is one of the benefits of our security framework—it allows designers of trust and reputation systems for marketplaces to clearly identify meaningful limitations or deficiencies in their systems.

6.4.2 Buyer Security in Commodity Trunits

In Section 6.4.1, we evaluated the degree of security provided by Basic Trunits. In this section, we perform the corresponding analysis of Commodity Trunits. The analysis of Commodity Trunits is simplified relative to Basic Trunits, for several reasons:

- Since trunits have a market price, their (minimum) value can be determined at any time without the need to evaluate an entire stream of future possible transactions. This simplifies the evaluation of the decision whether or not to cheat.
- Transactions are never infeasible under Commodity Trunits, since an agent can simply buy additional trunits when needed to cover a transaction.

The analysis is complicated, however, by the fact that r and b change during market execution—our profit function cannot rely on the same summations of trunit values used above, since each sale or trunit transaction will use a different r/b. Thus, we approach our analysis differently.

We begin with the same set of assumptions as for Basic Trunits, but make several modifications. The introduction of buying and selling of trunits was introduced specifically to remedy the exit and surplus trunit problems. Thus, we can eliminate the need for infinite transactions. In addition, the safety of Commodity Trunits depends on the selling price of trunits b remaining above the required ratio r.

Security guarantee of Commodity Trunits

Our goal, then is to prove:

Commodity Trunits is in use $\wedge$
selling agents are rational (A) $\wedge$
selling agents act alone (B) $\wedge$
buying agents are honest (C) $\wedge$
selling agents can reliably meet commitments if willing (D) $\wedge$
cost c is a constant (for each agent) percentage of selling price (E)
price of trunits b exceeds the required ratio r at all times $time$ (F)

$$\Rightarrow \forall t_D \in T_D : honest(t_D) \quad (6.19)$$

Specified more formally:

[Commodity 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]$\wedge$
$\forall G : |G| = 1 \wedge$
buying agents are honest $\wedge$
selling agents can reliably meet commitments if willing $\wedge$
cost c is a constant (for each agent) percentage of selling price
$\forall time : b_{time} > r_{time}]$

$$\Rightarrow \forall t_D \in T_D : honest(t_D) \quad (6.20)$$

Note that while we have shown that Commodity Trunits provides some protection from the ballot-stuffing problem, it has not been established that the mechanism provides protection from other forms of collusive behaviour (e.g., bad-mouthing). For this reason, we have retained the condition that agents act alone.

Proving the guarantee

Given that transactions cannot be infeasible under Commodity Trunits, and that we need not consider an entire set of future transactions to determine the value of trunits, we do not need to consider entire schedules in the same way that we did for Basic Trunits. Instead, we can consider the impact of each added transaction individually. Thus, our proof proceeds by induction on n, the number of transactions in a schedule.

As discussed in Chapter 5, the price of trunits b will vary over time. The precise profit earned by an agent will depend on exactly when he purchases trunits, and when he sells them. This issue should not obscure our consideration of whether honesty or dishonesty is more profitable. With this in mind, we make two important notes:

- If an agent buys trunits at price b, he incurs the same cost whether he intends to use them for cheating or for honest sales. He might save money by timing his purchase to take advantage of lower prices, but such savings are not attributable to his choice of strategy, so we do not consider them here: corresponding trunit purchases (and sales, for the same reason) are made at the same time and price in both the honest and dishonest cases.
- An agent might profit by buying trunits at a low price and selling them at a high price, without even having used them for transactions. Again, such profits (or losses, in the opposite case) are not relevant to the agents' choices of (dis)honesty, so we ignore such activity. (Putting this another way, we assume that the agent buys only as many trunits as required to engage in transactions.)

Base case: Any single transaction, n = 1: Any seller faced with a single transaction (requiring τ trunits) has three choices: execute the sale honestly, cheat, or foresake the sale and instead honestly sell his trunits. We consider the latter two options here, addressing the first afterwards.

The seller may begin with enough trunits to engage in a sale, or may need to buy additional trunits. Let τ_h be the number of trunits already possessed that are used for the transaction, and τ_p the number that are purchased, so $\tau = \tau_h + \tau_p$.

If the agent decides to cheat, he can gain (at most) $r\tau = r(\tau_p + \tau_h)$, and incur a cost of $b\tau_p$ in purchasing the trunits. If he instead decides not to engage in the transaction, and simply sells the trunits he already possesses, he will gain $b\tau_h$. Honestly selling the trunits is more profitable than cheating if:

$$\begin{aligned} r(\tau_p + \tau_h) - b\tau_p &< b\tau_h \\ r(\tau_p + \tau_h) &< b\tau_h + b\tau_p \\ r(\tau_p + \tau_h) &< b(\tau_h + \tau_p) \\ r &< b \end{aligned} \tag{6.21}$$

By assumption, $r < b$. Therefore, for any single transaction, forgoing the sale and selling the trunits honestly is more profitable than cheating.

This result is quite direct and simple. The mechanics of the market ensure that there is a preferable alternative to cheating: sell the trunits (i.e., engage in no transaction). A more interesting question is, does it make sense for an agent to actually engage in honest sales, or should he simply avoid cheating by not engaging in transactions? Consider the case where an agent must purchase τ_t trunits (at time t_{start}) to engage in any single sale t. He can purchase the trunits at price $b_{t_{start}}$; after the sale, any resulting trunits can be sold (at time t_{end}) at price $b_{t_{end}}$. His profit for a single transaction, then is:

$$P(t) = (1 - c)r\tau_t + b_{t_{end}}(1 + p)\tau_t - b_{t_{start}}\tau_t$$

Since a seller would likely only engage in a sale if his expected profit is greater than 0,

$$\begin{aligned} 0 &< (1 - c)r\tau_t + b_{t_{end}}(1 + p)\tau_t - b_{t_{start}}\tau_t \\ b_{t_{start}}\tau_t - b_{t_{end}}(1 + p)\tau_t &< (1 - c)r\tau_t \end{aligned} \tag{6.22}$$

that is, the profit from the sale of the good must be greater than any expected loss buying and selling the trunits. Ideally, we would hope all agents would have motivation to engage in honest sales, so we set $c = 1$ (the maximum ratio):

$$\begin{aligned}
b_{t_{start}}\tau_t - b_{t_{end}}(1+p)\tau_t &< (1-1)r\tau_t \\
b_{t_{start}} - b_{t_{end}}(1+p) &< 0 \\
b_{t_{start}} &< b_{t_{end}}(1+p) \\
\frac{b_{t_{start}}}{b_{t_{end}}} &< (1+p) \qquad (6.23)
\end{aligned}$$

that is, the rate of decrease in trunit prices (per transaction duration, as a fraction of the ending price) cannot exceed the premium for honesty.

In our simulations, using $p = 0.2$, this property always held. (We did not test for this property for other values of p). Thus, it should be expected that in a trunits marketplace, it will be profitable for most if not all sellers to engage in sales.

Induction step: Assume that honesty maximizes expected profit for any schedule of n transactions. Consider a schedule of size $n + 1$. Of the $n + 1$ transactions, we choose one arbitrarily, and denote this as transaction i. In the absence of i, the remaining set of transactions constitutes a schedule of size n.

Now, consider the addition of transaction i to the schedule. The addition of transaction i does not deprive any other transaction of trunits, since trunits can be purchased. For the same reason, trunits returned upon the completion of i are not required in the execution of other transactions. Thus, the n transactions can be executed independently of the execution of i: in isolation from i, their execution is identical to that of any schedule of size n. By the induction hypothesis, expected profit from the n transactions alone is maximized by executing each of them honestly. Now, we need only consider the change in profitability incurred by the addition of i into the schedule.

The execution of i requires trunits at its start time (t_1). If these trunits were to be purchased on the market (i.e., the seller did not use trunits that he already owned and was using to execute the schedule of n), the trunit requirement would have no effect on the schedule of n. Existing trunits may be used, however, trunits that might be required for the execution of some of the transactions in n. Consider a transaction starting at time t_2, that is deprived of some quantity of trunits τ that are instead used to execute

i. This will save us from purchasing the τ trunits at price b_{t_1}, but require us to purchase them at price b_{t_2} to execute the later transaction. The net impact on profit is $\tau(b_{t_1} - b_{t_2})$. Note, however, that this implies that under the schedule of n transactions alone, we have τ trunits available at time t_1 that are not required until time t_2. Thus, in the absence of i, we could simply (speculatively) sell the trunits at time t_1 and repurchase the same quantity at t_2: the net impact on profit is $\tau(b_{t_1} - b_{t_2})$, identical to the case where we use the trunits for transaction i. Even if we use existing trunits for i, then, profit is identical to the case where we simply purchase separate trunits for i independently of the execution of the n transactions.

Similarly, τ trunits returned after the execution of i (at time t_3) might be sold separately at price b_{t_3}, but they might be used instead to cover future sales of a transaction in n, starting at t_4, saving us from buying trunits at price b_{t_4}. The net impact on profit, then is $\tau(b_{t_4} - b_{t_3})$. This impact, however, is identical to that if we purchased trunits separately for the transaction in n, and simply held the trunits from i without using them for any of the n transactions, selling them at time t_4 instead of t_3.

In short, any mixing of trunits between i and the schedule of n is equivalent to executing i completely separately from the n in terms of the profit impact from trunits. Thus, the additional transaction i has no impact on the profitability of the n due to trunits (nor does it impact the profits earned directly via the n honest sales): execution of i can be considered in complete isolation from execution of the n transactions.

i, then, may be treated simply as a single, isolated transaction: an instance of our base case. As shown above, executing the single transaction honestly maximizes expected profits. Thus, for any schedule of $n + 1$ transactions, executing them honestly maximizes profit. By induction, expected profit for any schedule of size greater than 0 transactions is maximized by honestly executing every transaction.

This proof, while extremely simple, highlights an important point: the properties of Commodity Trunits allow us to consider transactions essentially in isolation, to determine their impact on profitability. This eases analysis for us. More importantly, it also allows agents to easily determine the expected outcomes of their actions. This permits even very simple agents to clearly understand the incentive for honesty, which in turn allows us to be confident that they will make the correct choice. Further, it should be noted that whether an agent cheats or not depends on only two parameters: b and r.

In summary, for any schedule, expected profit is higher if the agent executes transactions honestly, or forgoes transactions in favour of selling trunits, than if he cheats. Thus,

$$\begin{aligned}
&[\text{Commodity Trunits is in use} \wedge \\
&\forall G : |G| = 1 \wedge \\
&\text{buying agents are honest} \wedge \\
&\text{selling agents can reliably meet commitments if willing} \wedge \\
&\text{cost } c \text{ is a constant percentage of selling price} \wedge \\
&\forall time : b_{time} > r_{time}] \\
&\qquad\qquad \Rightarrow \forall T_A : [C \subseteq T_A, |C| > \varnothing \Rightarrow P(G, T_A, \varnothing) > P(G, T_A, C)] \qquad (6.24)
\end{aligned}$$

This yields:

$$\begin{aligned}
&[\forall G : \forall T_{D1}, T_{D2}[P(G, T_{D1}) > P(G, T_{D2}) \Rightarrow T_{D1} \text{is selected}] \wedge \\
&\forall T_A : [C \subseteq T_A, |C| > \varnothing \Rightarrow P(G, T_A, \varnothing) > P(G, T_A, C)] \\
&\qquad\qquad \Rightarrow \forall T_A : (T_A, \varnothing) \text{ is selected } \Rightarrow \forall t_D \in T_D : honest(t_D) \qquad (6.25)
\end{aligned}$$

Commodity Trunits delivers provable security. The discussion of the conditions required to make this guarantee is virtually the same as that for Basic Trunits, so it is omitted here.

6.4.3 The value of the framework

We have argued that it is necessary to explicitly consider security in the development of trust and reputation systems, and have proposed a framework for doing so. Application of this framework to Basic Trunits and Commodity Trunits revealed that: a) both mechanisms were capable of delivering provable security; b) to deliver that provable security, certain conditions must hold. We believe that the explicit consideration of these conditions has several benefits:

- It provides a clear understanding of if and where a trust/reputation system might be safely applied in a marketplace;
- It aids in the comparison of different models of trust and reputation;
- It clearly identifies meaningful directions for future research.

We believe that the benefits derived from this framework would apply to a wide variety of trust and reputation systems, not just Trunits. We discuss this issue in the next chapter.

Chapter 7

Discussion

In earlier chapters, we examined the vulnerabilities that commonly afflict trust and reputation models, described the Trunits model and two Trunits mechanisms, presented a framework for analyzing the security of trust and reputation systems, and applied this framework to our own proposals. Throughout these chapters, issues were raised that merit further discussion. In this chapter, we touch on a number of these issues, expanding on ideas raised earlier.

7.1 Is Trunits a model of Trust?

Both Basic Trunits and Commodity Trunits are mechanisms, intended to structure the marketplace so as to elicit the desired behaviour (honesty) from sellers. The question might be asked, however, is Trunits actually a model of trust? Commodity Trunits, in particular, might raise objections: it is difficult to reconcile a notion of real-world trust with the idea of trust units that are bought and sold as a commodity. Despite appearances, our model is quite similar to more conventional trust/reputation models in practice, even predictive models.

7.1.1 The practical equivalence of predictive and incentive approaches

Superficially, predictive- and incentive-based models seem to be entirely dissimilar: the former seek to predict the actions that agents will choose, while the latter seek to influence the choice of action. It is our position, however, that the distinction is a false one in many cases. We contend that predictive models often act as incentive schemes, intentionally or not.

Predictive approaches use some record of an agent's past performance in order to estimate the likelihood that the agent will be trustworthy in a future transaction. Under such a model, we might view a seller's trust rating as reflecting something intrinsic to the agent: how innately trustworthy she is, or how 'good a person' she is. In situations where agents are economically motivated, however, this view may be flawed. As noted in [7], "sellers care about buyer feedback primarily to the extent that they believe it might affect their future profits." 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 her decisions.

If the seller knows that some predictive model is in use, she could reasonably expect that her future sales will be impacted by her current activities. Acting badly may decrease future profitable sales—agents may be less likely to buy from her, lower prices may be commanded, etc.— while conversely, honesty may increase future sales. Regardless of her (dis)honest disposition, a rational agent seeking to maximize profit will behave honestly if she believes that is the more profitable course. In this case, a predictive model might be viewed as a de facto incentive mechanism.

Given this practical equivalence between predictive- and incentive-based approaches, Trunits has a great deal in common with existing predictive models. As such, it may justifiably be considered a 'model' of trust, despite its use of unconventional techniques.

7.2 The importance of security to all models of trust and reputation

As discussed in Chapters 2 and 3, vulnerabilities are a significant problem for existing trust and reputation systems. Given the prevalence of these vulnerabilities, one must question why they are so common. It may simply be the case that researchers are unaware of these issues. We believe that there is some truth to this. Moreover, issues of system-wide security may not be the foremost concern of researchers seeking to develop learning algorithms to allow individual agents to make good decisions in marketplaces. It is our position, however, that security is a critical concern for all developers of trust and reputation systems, including predictive models. We outline the case for this position in this section.

7.2.1 Issues of security in predictive models

As argued above, a predictive model may be viewed as de facto incentive mechanism, albeit with implicit rather than explicit incentives. Unfortunately, the implicit nature of these incentives may be a source of problems.

Consider a seller in a marketplace where a predictive trust model is in use, a model of which the seller is aware. He is attempting to make a decision: cheat or be honest? Understanding that a trust model is in use, he expects that if he is dishonest, his future sales may decrease, while if he is honest, sales may increase. This provides an incentive for honesty. Unfortunately, the implicit nature of the incentive leaves pivotal questions unanswered. How much future business will be lost in the case of dishonesty? How much will be gained though honesty? In short, is the incentive large enough to make honesty the more profitable choice? Without answers to these questions, it is difficult for the seller to make a well-supported decision.

Because the agent faces uncertainty in making this decision, we as market operators or developers cannot be sure which choice he will make. While the agent is attempting to make a decision based on the incentive, *there is no obviously superior choice*: he may decide to be dishonest, believing it to be the best choice under the incentive. Despite the fact that we wish the agent to be honest, we have set up the system in such a way that he may be lead to dishonesty. There are two key implications. First, although the model may be a de facto incentive mechanism, it has failed in this role. Second, because of the uncertainty of the incentive, it is difficult to ascertain whether or not vulnerabilities exist—the agent may, in fact, be correct in his belief that he has found a profit-maximizing cheating strategy.

In contrast, incentive-based approaches typically make the incentive explicit and measurable. This allows agents operating in such an environment to make strategic decisions more easily, since the outcomes of alternative actions can be determined more definitively. Perhaps more importantly, it helps developers to evaluate the incentive provided, to determine the circumstances under which it does or does not hold. This in turn allows easier identification of vulnerabilities, or ideally, establishment that no such vulnerability exists.

One might argue that agents need not be aware that a trust model is in use; to an oblivious agent, no such incentive is visible, and the agent's true disposition may be revealed through its actions. In this case, predictive models do not influence behaviour, and retain their full predictive ability. We do not believe that the security of a model should rest on the assumption of agent obliviousness, however, as it seems unlikely that such a 'secret' might be kept. This is reminiscent of the notion of 'security by obscurity', which does not find favour amongst cryptographers.

We do not mean to suggest that predictive approaches are inferior, or that researchers should only pursue incentive mechanisms. Quite the opposite: we believe that predictive models hold much promise, and that existing proposals have made significant strides in the understanding of trust and the development of trust and reputation models. We mean only to suggest that developers of predictive models should explicitly consider the incentive-like effects of their models, to make progress towards vulnerability-free models. System security is a specific goal, and it is unlikely to be achieved without being acknowledged and explicitly targeted by researchers.

7.2.2 Vulnerabilities and the ‘aware’ agent

That an agent may be aware of the use of a trust model raises darker concerns for all such models, regarding vulnerabilities. Consider an agent in a marketplace, who is aware that a trust model is in use. Given the marketplace scenario, we might expect that such an agent is a rational profit maximizer.

The agent chooses its actions based on its understanding of its environment. When no trust system is in use, the agent chooses actions based on the rules of the marketplace, attempting to maximize profit. If a trust/reputation system is in use, we can likewise expect the agent to choose its actions based on its understanding of the rules—the rules have simply been augmented by those of the trust/reputation system, creating a richer environment. We should expect the agent to try to maximize profit under this new set of rules just as he did under the original set; a profit maximizing strategy might very well involve exploiting a vulnerability. If a vulnerability exists in the trust/reputation system, then, we should expect an agent to exploit it.

This is a key point—not only is a vulnerability a potential problem, but we should *expect* agents (human or otherwise) to seek these out, to strategize with a knowledge of these actions. From this perspective, we believe that all researchers should seek to eliminate vulnerabilities from their models: security is a critical concern for all trust and reputation systems.

This is not to suggest that every system must achieve security to be valuable or of interest, or that security is more important than other design goals. Security should inform design, however, and security analysis should supplement other forms of evaluation.

7.2.3 The key role of security in adoption of trust and reputation systems

While extensive research has been conducted into trust and reputation systems, current proposals have inspired little adoption for real-world marketplaces, highlighting the importance of security. Trust and reputation systems generally focus on trust *between* agents *within* a marketplace. In contrast, the issue of security draws attention to a higher-level issue: trust of the *marketplace itself* by participants.

Marketplace developers are under no obligation to make use of a particular trust model, and traders have no obligation to participate in a given marketplace. For a party to be eager to participate in a marketplace, it may need to be confident that its interests will be protected, i.e., that the market will be secure from its perspective. In contrast, a model that has not been shown to be secure, or worse, contains known vulnerabilities, raises doubts for potential adopters/participants. This points to the key role of security for the adoption of trust and reputation models.

7.2.4 The Role of the Security Framework

The previous discussion highlights the importance of security to trust and reputation models. In this context, our security framework can serve three important purposes:

- It can aid in the precise determination of the degree of security provided by a given model, allowing the model to be evaluated by adopters/participants for use in a particular scenario;
- It can provide a sound and objective foundation for comparison of models, without the bias that may be introduced using other methods (e.g., a simulation may make use of a scenario that favours or penalizes a given model);
- It can help to identify meaningful avenues of future research, by identifying limitations in a model that relate directly to security.

We believe that the framework presented in this thesis offers a new direction for researchers in the area of trust and reputation, one that will help to address the needs and foster the confidence of real users.

7.2.5 The Role of Game-Theoretic Concepts

In the areas of game theory and mechanism design [19], there are well-defined solution concepts and properties (e.g., Nash Equilibria, Incentive Compatibility, etc.) that might be used in considering the choice of action an agent might take in a 'game'. These concepts, however, apply to strategic games where all players have choices to make, and an agent's best choice of action depends on other players' possible actions. As such, they do not necessarily apply to all trust/reputation systems. We therefore felt they were less appropriate to use as the basis for the security framework developed in this thesis. For example, under Trunits, the seller has the luxury of making his choice after the buyer has already committed and revealed her action. The seller's decision of whether or not to be honest is therefore better seen as a decision problem than as a strategic game. For this reason, we specify the concept of security based on properties that should hold regardless of the model used.

Note that developers may find the concepts like Incentive Compatibility useful in proving that our security properties hold, however. Conversely, game theory and mechanism design may provide important insights, useful as we expand our framework to incorporate seller and market security, discussed in Chapter 8.

7.3 Selecting Basic Trunits or Commodity Trunits

In this thesis, we have presented two Trunits variants: Basic Trunits and Commodity Trunits. While some of the characteristics of each model may have been apparent in earlier discussion, here we discuss how and why one might select one model or the other for use in a given marketplace.

Buying and selling of trunits was introduced into Commodity Trunits to deal with three issues: the start-up problem, the exit problem, and the surplus trust issue. Commodity Trunits provides solutions to these issues, while sacrificing none of the security offered by Basic Trunits (if the required pricing property for safe operation is maintained, i.e., the price of trunits always exceeds the required ratio for sales). Thus, Commodity Trunits may be preferable to Basic Trunits in many situations, because of the enhanced protection it provides.

That said, Basic Trunits is attractive in its simplicity, and may be useful in circumstances where either the enhanced protection of Commodity Trunits is not required, or obstacles prevent the implementation of the extended system. An example of the former case would be a scenario in which sellers' identities can be established with certainty; since there is no threat of re-entry, the start-up problem can be negated by simply providing new sellers with some initial quantity of trunits, and the threat of legal action (using the known identity of the seller) might prevent cheating exits from the market. An example of the latter would be the implementation of Trunits in a peer-to-peer system. In such an implementation, there is likely no identifiable market operator. Without an operator who can hold money and engage in trunit sales and purchases (as described in Chapter 5), it may be difficult to maintain the required pricing property for safe operation of Commodity Trunits.

Ultimately, the implementer/operator of a marketplace must weigh her priorities and the nature of the market itself, and select the option that best fits.

7.4 Trunits vs. Existing Models: Important Non-Security Properties

Much of this document has been devoted to investigating issues of security in trust and reputation systems. In Chapter 3, we examined the security properties of many existing models; in Chapters 4 and 5 we explored the Trunits mechanisms from the same perspective, thus contrasting them with existing work. Security is not the only important issue for trust and reputation systems, however. Here, we compare Trunits to two existing models with respect to other key characteristics.

7.4.1 Tran and Cohen

The Tran and Cohen model [30, 31] is an example of a direct experience model, in which an agent relies solely on its own experience in evaluating the trustworthiness of others. (An overview of this model can be found in Section 3.1.1.)

Computation and Storage Requirements

Both Tran and Cohen and the Trunits mechanisms require fairly simple calculations: in the former, after a transaction takes place the buyer performs simple algebraic updates of his reputation and expected value functions for the seller, while in the later, a simple algebraic update of the trunit balance is required. Tran and Cohen requires the storage of values for each seller and each good, while Trunits requires the storage of a single value for each seller. It should be noted, however, that because Tran and Cohen is a direct experience model, every buyer maintains separate values for every (known) seller; if there are n buyers, m sellers, and g goods, $O(mng)$ storage is required in total. In contrast, trunit balances are global values, so only $O(m)$ values needed be stored for m sellers, and no work need be performed by the buyers.

Equal Protection of Buyers

We consider it desirable for every agent in the system to be equally protected (hopefully, well protected). If, for example, new participants are more vulnerable than established ones, it can serve as a disincentive for new agents to enter the market. Because Tran and Cohen is a direct experience model, an agent's ability to choose good partners is directly tied to its experience in that market. New sellers, with no experience, choose essentially at random. Under Trunits, a seller's ability to engage in a sale is determined by that seller's trunit balance, regardless of the identity of the buyer—every buyer receives the same degree of protection.

Equal Opportunity for Sellers

Just as equal protection may foster the entry of new buyers into the market, equal opportunity may foster entrance of new sellers—a seller may be more likely to join a market if he does not feel at a grave disadvantage to established sellers.

Under Tran and Cohen, buyers preferentially choose sellers whom they know to be reputable. While this is reasonable, it also places new sellers at a disadvantage—they will only be chosen if a buyer cannot find a known reputable seller, or if a buyer chooses to explore the market (which occurs with a small probability). In contrast, under Trunits any agent who possesses sufficient trunits is considered to be reputable for that sale; an agent does not become 'more reputable' if he has a longer history. Thus, a new seller is on equal footing with an established seller when competing for any given sale. (An established seller may be able to engage in *more* sales than a new seller, due to a larger trunit balance, but on any given sale, they are equally attractive in terms of trustworthiness.)

7.4.2 The Beta Reputation System

The Beta Reputation System (BRS) [15, 34] is a witness information model, incorporating the recommendations of others as potential partners are evaluated. (An overview of this model can be found in Section 3.1.2.)

Computation and Storage Requirements

Under BRS, each buyer must keep a count of successes and failures for each (known) seller. For m sellers and n buyers, this requires $O(mn)$ storage, as compared to the $O(m)$ global values stored under Trunits.

The computational requirements for BRS are noteworthy. Determining a seller's reputation score (i.e., the expected value of the beta distribution given the recorded number of successes and failures) requires one simple algebraic calculation. Filtering out dubious reviews, however, is an iterative process, consisting of: a) summing the ratings from each recommender in the current set; b) computing a reputation score using the totals from the previous step; c) for each recommender, computing a reputation score using the ratings provided by that individual, and removing the recommender from the current set if the result is too high or too low compared to the result from part b); d) repeating from part a), until no more agents are removed. The exact amount of work required depends on the number of recommendations that are solicited. That said, it is significantly more work than the simple algebraic update that occurs after every sale under Trunits. Note, too, that the process is repeated under BRS each time a seller is to be considered for a sale; many sellers may be considered before one is selected. Under Trunits, to determine if a seller is eligible for a sale requires only a comparison of the required trunits to the seller's actual trunit balance.

Equal Protection of Buyers

Under BRS, a buyer combines its own experience with the reviews provided by recommenders. Since the recommendations are of unknown quality, direct experience may be more credible than the reviews received from others. To this degree, buyers with more experience may be protected more than inexperienced sellers. As noted above, Trunits provides equal protection to all buyers.

Equal Opportunity for Sellers

Under BRS, beta distributions are constructed based on the number of positive and negative experiences with the seller. The shape of the distribution (and hence, its expected value, or 'reputation score') varies depending on the number of experiences with the seller. Specifically, a larger number of ratings results in a narrower distribution with a higher peak, reflecting greater statistical confidence. For example, if a seller had 2 positive ratings and 1 negative rating, its reputation score under BRS would be 0.2. If the seller had 20 positive ratings and 10 negative ratings, its reputation score would be 0.3125.[1] Despite the fact that they have cheated with exactly the same relative frequency, the second seller is favoured, owing to his longer history. In contrast, as noted above, Trunits treats all sellers with sufficient trunits equally.

[1] Under BRS, reputation scores are calculated using Rep$(r, s) = \frac{r-s}{r+s+2}$, where r is the total of positive ratings, and s is total of negative ratings.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

Trust and reputation models for multiagent systems have received significant research attention. Unfortunately, existing proposals suffer from a number of common vulnerabilities; we catalogued these vulnerabilities, and investigated their presence in a number of existing models.

A vulnerability in a trust/reputation model provides methods and opportunities for agents to cheat and 'get away with it'. For an incentive-based approach, this means that the incentive for honesty fails to hold in some cases, making dishonesty the more profitable choice. Predictive systems, by definition, seek to predict the future behaviour of agents. It should come as no surprise that it is possible for agents to engage in behaviour that is not predictable by such a system—they may simply alter their behaviour from previous patterns. Vulnerabilities in predictive models are more troubling, however: they provide a means for agents to manipulate the system to improve their ability to cheat, and/or allow agents to cheat without serious repercussions.

In effect, vulnerabilities allow a dishonest agent to bypass the trust model. An insecure model might fare well in laboratory circumstances where simple/oblivious agents are employed. When faced with more sophisticated agents (or real-world traders) that are aware of such opportunities, these vulnerabilities may render the model ineffective, even irrelevant. For this reason, we believe that security is a critical design criterion for trust and reputation systems.

We argue that security has received insufficient attention from developers of trust and reputation systems. This is validated by the ubiquity of vulnerabilities in existing systems; moreover, issues related to security are rarely discussed in published work, particularly so for predictive proposals. It is possible that the developers of predictive models, concerned with developing learning algorithms and statistical methods to aid individual agents in decision-making, have not focused to the same degree on system-wide security issues. We contend, however, that predictive approaches are de facto incentive mechanisms when an agent knows that its trustworthiness is being modeled—the agent's behaviour may be altered in an effort to appear trustworthy. From this perspective, issues of vulnerabilities and the notion of provable system security are particularly relevant to developers of predictive approaches.

Motivated by these findings, we set out to develop a model of trust with explicit attention to security, one that addressed many of these common vulnerabilities. Our Trunits model is based on the use of abstract units that represent trust, in much the same way that units of money represent value. The quantity of trunits possessed controls an agent's ability to engage in transactions. Agents lose trunits when dishonest, and gain them when honest; honest behaviour allows increased future sales, providing an incentive for honesty. Based on this model, we developed two mechanisms, Basic Trunits and Commodity Trunits.

Basic Trunits is a direct implementation of the model described above. In contrast to Commodity Trunits, trunits are assigned to a specific trader and are not transferable. Basic Trunits has a number of attractive characteristics; important properties include:

- Agents are provided with financial incentives for honesty. In contrast to predictive models, where there is uncertainty regarding an agent's future actions, under Basic Trunits we can expect that a rational seller will be honest (when specific conditions are met, as discussed in Section 6.4.1).

- Basic Trunits provides a solution to the Reputation Lag problem, because trust cannot be used to support multiple simultaneous transactions.

- Basic Trunits provides a solution to the Value Imbalance problem, because the amount of trust earned from a transaction is proportional to the price of the sale.

- Basic Trunits treats new sellers and maximally disreputable sellers the same, the optimal policy in countering the re-entry problem.
- Basic Trunits provides a partial solution to the exit problem: while there is nothing to prevent an agent from cheating when it exits the market, there is an incentive for agents to remain in the market rather than exiting.
- Under Basic Trunits, decision making is extremely simple for buyers (in determining whether to trust a seller) and sellers (in determining whether or not to cheat).
- Because virtually no computation is required to evaluate sellers, buyers are equally protected regardless of their own computational capacities.
- Because the honesty of sellers is ensured by the mechanism, buyers are free to select goods that best meet their own preferences, rather than choosing suppliers based on their relative estimated trustworthiness.
- Computation and data storage requirements are extremely low for administering Basic Trunits.
- Basic Trunits is egalitarian, in that sellers with less accumulated trust/reputation are not at a disadvantage to sellers with longer histories, in any given transaction.

While Basic Trunits is an important step towards secure trust models, it has certain limitations. In particular, it provides no protection against collusion, and is subject to the surplus trust problem. It also provides no single solution to the start-up problem, the question of how an agent acquires an initial quantity of trunits.

Commodity Trunits is an extension of Basic Trunits, in which agents are permitted to buy and sell trunits. Commodity Trunits retains many of the characteristics of Basic Trunits, while providing important enhancements. In particular:

- Commodity Trunits remedies the start-up problem, by allowing agents to purchase trunits at start-up.

- Commodity Trunits alleviates concerns of artificially constrained trading volume—any time an agent has insufficient trunits, it can purchase the required quantity.

- Commodity Trunits provides a solution to the important exit problem, where an agent who is leaving the market can cheat freely. Under Commodity Trunits, an exiting agent finds it more profitable to sell her trunits rather than using them to cheat. To the best of our knowledge, Commodity Trunits is unique in providing a solution to this problem, an important result.

- Similarly, Commodity Trunits provides a remedy to the surplus trust problem, in which an agent can use surplus trust beyond its routine needs to cheat.

The security of Commodity Trunits depends on the ability to maintain a price of trunits higher than the value that can be gained by using them to cheat. Simulation verified that it was possible to maintain this price relationship for a realistic range of market parameters.

We have identified three components of security that must be addressed in order for a trust system to be considered fully secure: buyer security, seller security, and market security. We delved into buyer security in detail, providing a framework that might be used to precisely characterize the degree of security provided by a model. The result of the application of this framework to a model is a specific set of conditions that must be met in order for a model to provide a provable security guarantee. Based on these conditions, the model can be assessed for suitability for use in a given marketplace; further, the set of conditions can provide clear directions for future research.

We applied this framework to both the Basic and Commodity Trunits mechanisms—both were provably able to provide a security guarantee (namely, that rational sellers will not cheat) under specific sets of conditions. While certain of these conditions are likely to hold for a given market scenario, others are key issues for future investigation, and are discussed below.

In summary, the Trunits model and mechanisms represent important progress towards secure trust and reputation models. They are valuable in their own right for their attractive properties. Perhaps more importantly, together with our security framework they provide a promising new direction for research in this area.

8.2 Future Work

While our work addresses important issues regarding the security of trust and reputation systems, it raises other questions and highlights areas that may be fruitful upon further investigation. Some of these avenues have been discussed in Chapter 7, and we do not restate them here. Beyond these ideas, however, there are a number of possibilities that we hope to explore in the future. We briefly record them here.

Addressing seller and market security

At present, our security framework focuses almost exclusively on buyer security. There are two reasons for this: a) protecting buyers from untrustworthy sellers has been the predominant focus of work in trust and reputation systems to date; b) our own trust model shares this perspective. That said, we believe that security must be a primary design goal for trust and reputation systems if they are to have relevance and achieve adoption. A complete framework that incorporates both seller and market security is critical in this quest, and it is our goal to develop this broader framework as future work. We suggest two starting points for investigation here.

Seller security is more difficult to define than buyer security. Buyers wish to receive goods as promised—their vulnerability stems from the fact that they provide payment before receiving and inspecting the good. Buyer protection can thus be specified within the context of the execution of 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 illustrates this point, wherein a coalition artificially inflates the reputation of one of its members; the 'attack' consists of stealing sales from non-coalition sellers. 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 other business issues: quality of marketing, legitimate competitive activity, etc.

We suggest that for a system to be secure for a seller, one promising approach may be to ensure that the dishonest activity of any attacker (dishonestly executed sales, unfair ratings, ‘fake’ transactions between coalition members, etc.) should not reduce a seller’s revenue. Specifically, these activities should not reduce revenue from buyers *outside the coalition*. Revenue from coalition members might be reduced by these activities—for example, cheating by coalition members might remove those agents from consideration for future transactions. This would not constitute a successful attack on the seller, but rather the very sort of protection we would hope a trust model would provide.

To protect the market operator (or to ensure the continuing operation of the market where no operator can be identified), 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.
- Dishonest activity by participants should not cause the market to fail (for example, by driving buyers/sellers out of the market).

Again, these issues require further investigation, as future work.

Incorporating protection from dishonest buyers into Trunits

Just as our framework focuses on buyer security, Trunits focuses solely on ensuring that sellers are honest. We intend to incorporate a system for managing the behaviour of buyers into the model as well. Doing so would constitute progress towards a ‘complete’ model, one that provides seller and market security as well.

It must be noted that Trunits does not preclude the parallel use of another system in order to provide market/seller security. For example, we might consider using a variant of Jurca and Faltings’ system [17] (discussed in Chapter 3) to ensure buyer honesty, basing the system on real money, rather than on the separate ‘reputation money’. The act of the market operator allowing a transaction is considered a ‘recommendation’, and the ‘fee’ is either charged to the buyer, or added to the commission paid by the seller. A buyer is paid out in cash if the subsequent buyer’s rating matches her own.

Unfortunately, as discussed in section 3.3.3, this model suffers one key problem that will need to be addressed: it is vulnerable to collusive attacks such as ballot-stuffing and bad-mouthing.

Addressing collusion

The application of the security framework to Trunits identified several important directions for future development, but dealing with collusion was of special importance. Collusion is a problem to which most (if not all) current trust and reputation systems are vulnerable.

Two well-known collusive attacks are ballot-stuffing and bad-mouthing, described in Chapter 2. One solution to ballot-stuffing was presented in Section 5.4.3; here, we outline some other possible avenues for adjusting Trunits to better cope with collusion.

While the basic Trunits model uses very simple trust ratios to set the risk and reward for transactions, collusion is one reason why more sophisticated functions might be preferable. A few examples of possibilities to explore in future work are as follows:

- Incorporation of a system to ensure the honesty of buyers is likely to provide protection against collusion. The collusive attacks described above require coalition members to act as both buyers and sellers; if buyers have an incentive for honesty, this will make the collusive activity less attractive.

- In many systems (e.g., [10, 36]) only one rating is considered for each buyer, with older ratings being discarded. While we might not simply discard trunits from previous transactions, we can consider reward functions that drastically reduce the marginal reward from each successive transaction with the same user. This fits with our real world model—recommendations from ten different users are likely to inspire more confidence than a single recommendation from a user who has engaged in ten transactions with a given seller.

- Similarly, we might reduce the reward realized from transactions with new users (as done in [36]) in order to deter the creation of new accounts for ballot stuffing.

- One option for implementing the above policies is the use of a probabilistic reward function, where the likelihood of reward is tied to several factors, including past number of transactions between this buyer-seller pair and the ‘age’ of the buyer’s account.

Exploring other techniques for market management

The market operator may attempt to manage market parameters to ensure that growth in supply doesn’t outstrip that of demand. For example, the operator might decrease p to slow trunit growth, or ‘tax’ trunits as was suggested in Chapter 4. The market operator is ideally positioned to do so, having complete access to data on market participation, sales volume, trunit sales, etc. However, the effects of shifting market parameters during its operation have yet to be studied.

Decentralization of the mechanism

As presented, Trunits is a centralized model. There is nothing about the mathematics of the model that necessitates centralization, however. The possibility exists for a decentralized implementation in which trunit balances are stored locally for each seller, and updated as transactions occur. Such an implementation would require secure storage and update of trunit balances; a cryptographic scheme such as that used in [14] might permit this. Under a decentralized implementation, there may be no need of an identifiable market operator, similar to the lack of central nodes in many peer-to-peer systems.

Anonymous marketplaces

Under the trunits model, the buyer does not need any information about the seller to determine her trustworthiness, beyond knowing that she has enough trunits for the transaction—the incentive for honesty dissuades the seller from cheating, rather than the buyer’s ability to connect her action with her identity. This suggests the possibility that the system might support markets in which buyers and sellers are anonymous to one another, yet can trust one another. Such a marketplace may be desirable for many applications. However, anonymity may have other as-yet unforeseen ramifications; further study is required.

Exploring alternative methods for addressing Surplus Trust

The Commodity Trunits mechanism provides a solution to the surplus trust problem: unneeded trunits may be profitably sold. If a market operator did not want to use Commodity Trunits, there are other possible means of dealing with this problem. One such approach is related to a common technique used in other models. Many systems (e.g., [35]) emphasize the most recent rankings by either discarding older rankings, or applying some form of 'decay factor' to de-emphasize them. We might employ a similar idea, with the goal of paring the number of surplus trunits. For example, trunits that have gone unused over a period of time (i.e., the lowest balance in the account over that period of time) are likely to be surplus. A portion of these trunits could be 'taxed'. In a way, this policy has the effect of emphasizing recent history—your level of trustworthiness (i.e., your trunit balance) is tied to your recent activity, even if you were 'more trustworthy' in the past. On a more practical level, the goal of the system is to ensure that, if you are sufficiently trustworthy, you have enough trunits to conduct your business; surplus trunits are, by definition, in excess of the trunits required to conduct business, and paring them is not unreasonable.

Adjusting Trunits by lifting assumptions

The Trunits model, as presented in this thesis, operates under certain assumptions about the marketplace and the agents. Some of these assumptions (e.g., the advertised-price marketplace) simply reflect the scenario for which the model was conceived. Others, as revealed in Chapter 6, reflect required conditions for Trunits to offer a provable level of security. For future work, we can explore the implications of lifting these assumptions.

Lifting the assumption of a human market operator: We have suggested in this thesis that Trunits might be implemented without a 'market operator' that is an identifiable entity. For example, in a peer-to-peer implementation of Trunits, tasks of the operator (holding trunits in escrow, updating trunit balances after honest/dishonest transactions, etc.) might be performed in a decentralized manner by the client software. This possibility requires further study, however. Certain techniques discussed in this thesis seem difficult to employ without an identifiable operator. For example, active buying and selling of trunits to control their price (as done by the market operator in our simulation) would be problematic without an entity that can own money and enter into sales. Other issues, and opportunities, may become apparent with further study.

Lifting the assumption of honest buyers, and of sellers' perfect ability to fulfill commitments: The security analysis conducted in Chapter 6 included the assumptions that buyers are honest, and that sellers are able to fulfill their commitments if they choose to do so. Dishonest buyers and unreliable quality control of sellers can potentially undermine the incentive for honesty under Trunits. If a seller cannot be confident of receiving his reward for honesty, even if he has tried to be honest (and incurred the cost of an honest transaction), then honestly may not clearly be his profit-maximizing strategy.

It is worth noting, however, that a seller need not be *perfectly* confident that he will be rewarded in order for honesty to be the profit-maximizing strategy. Consider the case where buyer honesty and successful execution of the transaction are random, with uniform probability across transactions. Let f be the probability of receiving a reward on an honestly executed sale:

$$f = prob(\text{a seller can fulfill his commitment if he intends to do so}) \times \tag{8.1}$$

$$prob(\text{a seller is rated honestly by a buyer}) \tag{8.2}$$

Under these conditions, we can modify equation 4.2 to reflect the number of trunits that an agent can expect to receive after attempting to honestly execute a sale requiring τ_i trunits:

$$f(1+p)\tau_i \tag{8.3}$$

Here, we revisit the proof of the incentive for honesty developed in Section 4.3.5. The expected profit from executing a transaction requiring τ trunits honestly, then, (since the agent can cheat immediately with any returned trunits) is at least:

$$(1-c)r\tau + f(1+p)r\tau \tag{8.4}$$

For honesty to have the highest expected profit, it must exceed the profit from cheating immediately, $r\tau$:

$$\begin{gathered}(1-c)r\tau + f(1+p)r\tau > r\tau \\ (1-c) + f(1+p) > 1\end{gathered}$$

Since we want this property to apply for all sellers, let c be the maximum cost ratio, 1:

$$\begin{aligned}(1-1) + f(1+p) &> 1 \\ f(1+p) &> 1 \\ f &> \frac{1}{1+p}\end{aligned} \tag{8.5}$$

For a typical reward ratio $p = 0.2$, this requires the probability of receiving the reward to be at least $0.8\bar{3}$.

Note, as well, that as discussed throughout this thesis, Trunits permits the use of a parallel mechanism to ensure the honesty of buyers, or to rectify the effects of dishonest reviews. If such a system were in use, we would expect rates of buyer honesty to be extremely high. Further, the seller might fulfill his obligation even if his quality control is poor, through the use of warranties/etc. to rectify issues. We would expect f to be quite high in practice.

Thus, it seems that these two assumptions might be removed. We note, however, that the assumption that every transaction has the same probability of reward is problematic. For example, understanding that he might not be guaranteed a reward for honesty, a seller might begin to model buyers, attempting to choose those that are most likely to deliver rewards. This introduces another layer of complexity, requiring further study.

Incorporating riskiness into the evaluation of earnings

The discussion in the previous section highlights an important issue. Traders do not always simply choose the option with the highest expected value—they take into account the riskiness of projected earnings as well. In the previous section, we discussed two important forms of risk to the seller. There are others, however: an important example, addressed in our discussion of Basic Trunits, is that the seller may not be able to find buyers in the future.

One might address these issues by incorporating explicit treatment of the riskiness of future cash flows into the evaluation of the flows. This is common practice in finance: future returns are 'discounted' based on risk. In evaluating alternative courses of action, the value of future expected returns are reduced in proportion to their riskiness, reflecting the greater desirability of certainty over risk. One well known model used for this purpose is the Capital Asset Pricing Model [28].

While it is conceivable that such an approach might be useful in the analysis of Trunits, we have taken a different direction here. Given our goal of establishing provable security, we seek to identify the specific conditions that must hold for a trader to be guaranteed protection. Thus, we favour the approach employed in the previous section, where a lower bound (i.e. a specific condition) is established on the amount of risk permitted for a security guarantee to be made. Application of discounting techniques in the analysis of Trunits may provide further insight, however: how agents will behave if risk cannot be expressed as a single probability, how sellers might react to different buyers with different risk profiles, etc.

Moving beyond the marketplace

While it is logical to explore this model completely in the marketplace scenario before looking beyond, the model may have other applications. For example, in multiagent systems where agents are motivated to seek delegated tasks from others, such agents are similar to sellers of a service. This model may be useful under such conditions to ensure trustworthy collaboration between agents.

Bibliography

[1] Sulin Ba, Andrew B. Whinston, and Han Zhang. Building trust in online auction markets through an economic incentive mechanism. *Decis. Support Syst.*, 35(3):273–286, 2003.

[2] Rajat Bhattacharjee and Ashish Goel. Avoiding ballot stuffing in ebay-like reputation systems. In *P2PECON '05: Proceeding of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems*, pages 133–137, New York, NY, USA, 2005. ACM Press.

[3] S. Braynov and T. Sandholm. Trust revelation in multiagent interaction, 2002.

[4] Sviatoslav Braynov and Tuomas Sandholm. Incentive compatible mechanism for trust revelation. In *AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 310–311, New York, NY, USA, 2002. ACM Press.

[5] M. Broadie and P. Glasserman. Pricing american-style securities using simulation. Papers 96-12, Columbia - Graduate School of Business, 1996. available at http://ideas.repec.org/p/fth/colubu/96-12.html.

[6] Rajdeep K. Dash, Sarvapali D. Ramchurn, and Nicholas R. Jennings. Trust-based mechanism design. In *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 748–755, Washington, DC, USA, 2004. IEEE Computer Society.

[7] Chrysanthos Dellarocas. Goodwill hunting: An economically efficient online feedback mechanism for environments with variable product quality. In *AAMAS '02: Revised Papers from the Workshop on Agent Mediated Electronic Commerce on Agent-Mediated Electronic Commerce IV, Designing Mechanisms and Systems*, pages 238–252, London, UK, 2002. Springer-Verlag.

[8] Chrysanthos Dellarocas. The digitization of word of mouth: Promise and challenges of online feedback mechanisms. *Manage. Sci.*, 49(10):1407–1424, 2003.

[9] Chrysanthos Dellarocas. Efficiency and robustness of binary feedback mechanisms in trading environments with moral hazard. Working papers 4297-03, Massachusetts Institute of Technology (MIT), Sloan School of Management, April 2003. available at http://ideas.repec.org/p/mit/sloanp/1852.html.

[10] eBay, Inc. Web site, 2005. `http://www.ebay.com/` (accessed November 2005).

[11] Eric Friedman and Paul Resnick. The social cost of cheap pseudonyms. *Journal of Economics and Management Strategy*, 10(2):173–199, 2001.

[12] Nathan Griffiths. Task delegation using experience-based multi-dimensional trust. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 489–496, New York, NY, USA, 2005. ACM Press.

[13] C. Grothoff. An excess-based economic model for resource allocation in peer-to-peer networks. *Wirtschaftsinformatik*, June 2003.

[14] Minaxi Gupta, Paul Judge, and Mostafa Ammar. A reputation system for peer-to-peer networks. In *NOSSDAV '03: Proceedings of the 13th international workshop on Network and operating systems support for digital audio and video*, pages 144–152, New York, NY, USA, 2003. ACM Press.

[15] Audun Jøsang and Roslan Ismail. The beta reputation system. 15th Bled Electronic Commerce Conference e-Reality: Constructing the e-Economy, June 2002.

[16] Henry E. Kyburg Jr. Bayesian and non-bayesian evidential updating. *Artif. Intell.*, 31(3):271–293, 1987.

[17] R. Jurca and B. Faltings. An incentive compatible reputation mechanism, 2003.

[18] S. Marsh. Formalising trust as a computational concept, 1994.

[19] Andreu Mas-Colell, Michael D. Whinston, and Jerry R. Green. *Microeconomic Theory*. Oxford University Press, 1995.

[20] Alfred J. Menezes, Scott A. Vanstone, and Paul C. Van Oorschot. *Handbook of Applied Cryptography*. CRC Press, Inc., Boca Raton, FL, USA, 1996.

[21] Mojo Nation. Web site, 2004. `http://www.mojonation.net/` (now defunct; accessed through the Internet Archive, web.archive.org).

[22] Thomas H. Naylor. *Computer Simulation Experiments with Models of Economic Systems*. John Wiley & Sons, 1971.

[23] W. T. Newlyn. *Theory of Money*. Oxford University Press, 1971.

[24] P. Resnick and R. Zeckhauser. Trust among strangers in internet transactions: Empirical analysis of ebay's reputation system. 2001.

[25] John Rushby. Critical system properties: Survey and taxonomy. *Reliability Engineering and System Safety*, 43(2):189–219, 1994.

[26] Jordi Sabater and Carles Sierra. Regret: reputation in gregarious societies. In *AGENTS '01: Proceedings of the fifth international conference on Autonomous agents*, pages 194–195, New York, NY, USA, 2001. ACM Press.

[27] Jordi Sabater and Carles Sierra. Review on computational trust and reputation models. *Artif. Intell. Rev.*, 24(1):33–60, 2005.

[28] William F. Sharpe. Capital Asset Prices: A Theory of Market Equilibrium under Conditions of Risk. *The Journal of Finance*, 19(3):425–442, 1964.

[29] W. T. Teacy, Jigar Patel, Nicholas R. Jennings, and Michael Luck. Travos: Trust and reputation in the context of inaccurate information sources. *Autonomous Agents and Multi-Agent Systems*, 12(2):183–198, 2006.

[30] Thomas Tran and Robin Cohen. A learning algorithm for buying and selling agents in electronic marketplaces. In *AI '02: Proceedings of the 15th Conference of the Canadian Society for Computational Studies of Intelligence on Advances in Artificial Intelligence*, pages 31–43, London, UK, 2002. Springer-Verlag.

[31] Thomas Tran and Robin Cohen. Improving user satisfaction in agent-based electronic marketplaces by reputation modelling and adjustable product quality. In *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 828–835, Washington, DC, USA, 2004. IEEE Computer Society.

[32] Wiebe van der Hoek and Michael Wooldridge. Towards a logic of rational agency. *Logic Journal of the IGPL*, 11(2):135–159, 2003.

[33] Gerhard Weiss, editor. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. MIT Press, Cambridge, MA, USA, 1999.

[34] Andrew Whitby, Audun Josang, and Jadwiga Indulska. Filtering out unfair ratings in bayesian reputation systems. In *Proceedings of the 7th Int Workshop on Trust in Agent Societies*, 2004.

[35] Bin Yu and Munindar P. Singh. Distributed reputation management for electronic commerce. *Computational Intelligence*, 18(4):535–549, 2002.

[36] Giorgos Zacharia, Alexandros Moukas, and Pattie Maes. Collaborative reputation mechanisms in electronic marketplaces. In *HICSS '99: Proceedings of the Thirty-second Annual Hawaii International Conference on System Sciences-Volume 8*, page 8026, Washington, DC, USA, 1999. IEEE Computer Society.