Design and Analysis of Electronic Feedback Mechanisms

Qin Li

Thesis submitted to the University of London
for the degree of Doctor of Philosophy

Information Security Group
Department of Mathematics
Royal Holloway, University of London

2012

Declaration

These doctoral studies were conducted under the supervision of Professor Keith M. Martin.

The work presented in this thesis is the result of original research carried out by myself, in collaboration with others, whilst enrolled in the Department of Mathematics as a candidate for the degree of Doctor of Philosophy. This work has not been submitted for any other degree or award in any other university or educational establishment.

Qin Li
December, 2011

To my parents.

Acknowledgements

First and foremost I would like to express my sincerest gratitude to my supervisor, Professor Keith Martin, who has supported me throughout my PhD study in Royal Holloway with his patience and knowledge whilst allowing me the room to work in my own way. He has become not only my supervisor, but also my mentor and friend.

Many individuals have influenced my academic career over the years. Jie Zhang, Siaw-Lynn Ng, Liqun Chen, Jason Crampton, Carlos Cid, Lizzie Coles-Kemp, Kenny Paterson and Allan Tomlinson deserve special mention. Administration and technical staff, among them Pauline Stoner, Jenny Lee, Lisa Nixon, Jon Hart and Tristan Findley, have been very helpful when I was seeking help from them.

In my daily life in Egham, I have been blessed with a group of friends and a friendly group of fellow students. Yi, Ling, Wei, Shuxian, Jialin, Graham, Chunhua and Zhiyi have been like my brothers and sisters, providing me with support. Many regular visitors to my flat have provided endless entertainment. Magic Wok has become the most regular restaurant where I comfort my Chinese stomach, along with many friends. Wei, Amy, Anastasia, Ciaran, Gary and Nadhem generously provided their house garden and table tennis table for amateur players like me to enjoy the moment. Zhiyi has been a wonderful table tennis coach who has helped me to rapidly improve my table tennis skills. The delicious tarts and soup that were occasionally cooked by Amy, particularly for us who played table tennis at her house garden, were alway a wonderful surprise.

Sriram and Yenn, a wonderful couple, have shown me their deep empathy and benevolence towards those who suffer from poverty, war and unfairness. The party organisers, among them Elizabeth and Anastasia, provided me with a good opportunity to meet new people. The films enjoyed at Saif's comfortable and home-theatre-like living room with our friends will always remind me of his kindness and warm hospitality. Many fellow students from the Information Security Group at Royal Holloway, both those who have completed and those who are still on their journeys towards their PhDs, have made me proud of having been their colleague.

Thank you all. You have made my life in Egham unforgettable.

Abstract

With the advent and development of modern information technology, such as the Internet, the difficulty in transmitting data has been reduced significantly. This makes it easier for entities to share their experience to a larger extent than before. In this thesis, we study the design and analysis of *feedback mechanisms*, which are the information systems that enable entities to learn information from others' experience.

We provide a framework for feedback mechanisms. We first provide an abstract model of a feedback mechanism which defines the scope of our concept and identifies the necessary components of a feedback mechanism. We then provide a framework for feedback mechanisms. This provides a global and systematic view of feedback mechanisms. We also use our model and framework to decompose and analyse several existing feedback mechanisms.

We propose an electronic marketplace which can be used for trading online services such as computational resources and digital storage. This marketplace incorporates a dispute prevention and resolution mechanism that is explicitly designed to encourage the good conduct of marketplace users, as well as providing important security features and being cost-effective. We also show how to incorporate the marketplace into Grid computing for exchanging computational resources.

We propose a novel feedback mechanism for electronic marketplaces. In this setting, the role of feedback is no longer a "shadow of the future", but a "shadow of the present". In other words, feedback directly impacts on the seller's payoff for the current transaction instead of future transactions. This changes the fundamental functionality of feedback, which solves many inherent problems of reputation systems that are commonly applied in electronic marketplaces.

We provide a novel announcement scheme for vehicular ad-hoc networks (VANETs) based on a reputation system in order to evaluate message reliability. This scheme features robustness against adversaries, efficiency and fault tolerance to temporary unavailability of the central server.

Contents

Chapter 1

Introduction

Contents

This chapter gives an overview of the thesis. We provide the motivation for our research, describe the contributions of this thesis, and present the overall structure of the thesis.

1.1 Motivation

We say that an entity's personal experience is its *direct experience.* One approach for an entity to learn information is through direct experience. In this approach, the information is learnt without seeking help from any other entities. For example, if a tourist wants to find a restaurant in an unfamiliar place, they can do so by exploring the place themselves. Another approach for an entity to learn information is through *indirect experience.* In this approach, the information is learnt only through other entities. For example, the tourist may ask people on the street for directions, refer to a map, or call a friend for advice.

Obtaining information through direct experience and indirect experience have their own characteristics. Information obtained through direct experience often has a high level of credibility to the owner of the experience. However, obtaining information through direct experience may have limited applicability, since:

- It is sometimes associated with risk. For example, one can learn whether an unknown fruit is edible or poisonous by taking the risk to try it. This risk is sometimes so significant that it may cause death and is thus only taken under extreme circumstances.

- It is sometimes associated with high cost in terms of time, money or other types of resource. For example, in order to learn the quality of service of a merchant, directly purchasing products or services from the merchant and then testing them is often time consuming and financially costly.
- Sometimes we require information about an experience which we are only likely to engage with once, and thus cannot use prior direct experience to inform judgement. For example, if a patient needs some information to inform the choice of surgeon to perform an operation, obtaining this information by trial and error is not an option.
- Reality sometimes imposes constraints that make direct interaction infeasible. For example, if one wants to learn information about an event that occurred in the past, it is of course infeasible to directly witness the event. (A video of the event is regarded as indirect experience rather than direct experience.)

On the other hand, obtaining information through indirect experience requires that:

- There is a communication channel between the *requester* (the entity that seeks the information) and the *information holder* (the entity that possesses the information).
- The information holder is willing to share the information with the requester.
- The requester has some trust in the information holder in terms of the information; otherwise, the information has little credibility to the requester.

When an entity wants to learn some information about a target, this can be done through direct experience and/or indirect experience. In this thesis we will focus on mechanisms that make it easier to process and rely on indirect experience.

We refer to an information system that facilitates entities to obtain information through indirect experience as a *feedback mechanism.* A feedback mechanism aims to collect, process and disseminate users' direct experience about some *targets*, such as some entities and items of interest. This direct experience is referred to as *feedback* and the information disseminated by the feedback mechanism is referred to as *advice*. This advice is disseminated to other users who are interested in targets and may utilise the advice as indirect experience.

Before the advent and development of modern information technology, the difficulty of establishing an efficient communication channel was one of the main barriers against learning information from others. However, when an entity was able to receive some information from another entity, the receiver usually had some sort of trust relationship with the sender and thus knew the credibility of the information. Appropriate feedback mechanisms often operated in a spontaneous and haphazard manner through personal networks, word of mouth, rumour and mass media [106].

However, this situation has changed dramatically due to the advent and development of modern information technology, such as the Internet. With the assistance of modern information technology, the difficulty in establishing an efficient communication channel has been reduced significantly. The range of entities with whom an entity is able to establish an efficient communication channel has broadened dramatically. This facilitates the development of more efficient feedback mechanisms.

However, there are still some significant issues, as follows:

- There is an information explosion, which makes it much more difficult for entities to identify beneficial information and avoid harmful and irrelevant information.
- The entity at one end of a modern communication channel often does not have a strong trust relationship with an entity at the other end.

These limitations motivate the need to study the design and use of effective feedback mechanisms.

Many feedback mechanisms have been proposed in the literature. For example [2, 36, 52, 58, 80, 93] provide surveys into various feedback mechanisms. Some feedback mechanisms are successful and widely used, such as the search engine Google [49], the online encyclopedia Wikipedia [120], the feedback forum used by eBay [41] and the recommender system used by Amazon [3]. However, there is still a great deal of research required. The following aspects are of particular importance:

- Many previous studies focus on narrow application environments. Studies in different application environments are carried out in isolation from the wider body of work. Design lessons learnt from one application scenario are sometimes not transferred to research in another application scenario.
- Many studies lack a systemic view. Some research focuses on a particular aspect while ignoring other aspects, and many designs are incomplete. For example, many proposals pay more attention to techniques for processing and aggregating feedback than on design of sound architectures for feedback mechanisms.
- There is a great diversity of application scenarios in which feedback mechanisms can be deployed. Although there are a large number of proposed feedback mechanisms, there may be application scenarios for which suitable feedback mechanisms have yet to be developed.

In addition, there are aspects of feedback mechanisms that have not been well explored. One aspect that we will consider in this thesis relates to the rationality of adversaries. In many information security studies, such as cryptography, the security of an information system is often evaluated in the worst-case scenario, in which adversaries with the most resources are assumed to attack the system (we use the term "security" here instead of "robustness" in order to adhere to the conventions of the cryptography community, but they are interchangeable in this discussion). If the system is secure in the worst-case scenario then it can be claimed secure in any possible application scenario. This approach has some advantages. For example, it provides an absolute assurance about the security of the system to the users. Practitioners who adopt the system do not have to evaluate the applicability of the system in terms of the adversary capability.

On the other hand, designing an information system that is secure in the worst-case scenario has some limitations. For example:

- It is usually much more technically challenging to design a system that is secure in the worst-case scenario than in a comparatively relaxed adversarial scenario.
- In an application scenario which only features more modest adversaries, the adoption of a system that is secure in the worst-case scenario may be unnecessary.

With respect to design of some feedback mechanisms, these limitations seem fairly prominent because:

- For many feedback mechanisms, insider adversaries are the main threat to robustness (see Section 2.5.6.2). Designing a feedback mechanism that is robust against insider adversaries in the worst-case scenario seems very technically challenging.
- In some application scenarios, such as electronic marketplaces, a feedback mechanism that is robust in the worst-case scenario may not be necessary. For example, some existing feedback mechanisms with robustness assurance in a more relaxed adversarial scenario still work well, such as eBay Feedback Forum [41] (see Section 2.7.5.6).

As a result, it seems interesting to study the design and analysis of feedback mechanisms that are only robust in a relaxed but realistic adversarial scenario. In Chapters 3 and 4, we propose two separate feedback mechanisms that are *rationally robust*, which is a term used in this thesis to describe the robustness of a feedback mechanism in one adversarial scenario that is more relaxed than the worst-case scenario. In the rational adversarial scenario, we assume that all participants, including adversaries, are rational decision makers, i.e. they choose the best actions to maximise their payoff and benefit. This implies that the adversaries have no willingness to carry out an attack when the cost of the attack outweighs the benefit from the attack, even if it is technically feasible. This approach provides a different perspective to the design of feedback mechanisms for some applications.

1.2 Contributions and Organisation of Thesis

The contributions of this thesis can be summarised as follows.

- In Chapter 2 we provide a framework for feedback mechanisms. We first provide an abstract model which defines the scope of our concept of a feedback mechanism and identify the necessary components. We then provide a framework which provides a global and systematic view of feedback mechanisms. Subsequently, we use our model and framework to analyse several existing feedback mechanisms.
- In Chapter 3 we propose an electronic marketplace which can be used for trading online services such as computational resources and digital storage. We observe that a robust reputation system can be designed for this application scenario by applying a solution identified by the framework. This reputation system serves the electronic marketplace as a dispute prevention mechanism. This marketplace features induction of good conduct, cost-effectiveness, transaction security, reputation robustness and penalty balance. Some of this work has been published in [72], [73] and [74].
- In Chapter 4 we propose a novel feedback mechanism for electronic marketplaces. In this scheme we no longer adopt a reputation system, which is a common approach to induce good conduct. Instead, we design a feedback mechanism that does not rely on the concept of reputation. We change the fundamental role of feedback from a "shadow of the future", which is used in reputation systems, into a "shadow of the present". In other words, feedback directly impacts on the seller's payoff for the current transaction. By changing the fundamental functionality of feedback, many inherent problems of reputation systems are overcome.
- In Chapter 5 we propose an announcement scheme for vehicular ad-hoc networks (VANETs) based on a reputation system that allows evaluation of message reliability. Unlike many reputation-based approaches, which adopt a decentralised architecture, we adopt a centralised architecture, which seems counter-intuitive, by taking advantage of the already existing centralised infrastructure in VANETs and the highly mobile feature of vehicles. This allows

us to design a robust, efficient and practical announcement scheme.

- In the final chapter of this thesis, Chapter 6, we provide concluding remarks about our proposals in Chapters 2, 3, 4 and 5. These include a summary of our research findings and suggestions for future work.

Chapter 2

A Framework for Feedback Mechanism

Contents

This chapter provides a framework for feedback mechanisms. We first provide an abstract model of a feedback mechanism which defines the scope of our concept of *feedback mechanisms and identifies the necessary components of a feedback mechanism. We then provide a framework for feedback mechanisms. This provides a global and systematic view of feedback mechanisms.*

2.1 Introduction

There are a huge variety of feedback mechanisms. Many information systems can be regarded as feedback mechanisms. They range from small-scale opinion polls to systems designed for presidential elections, from one-time customer surveys to long-established Better Business Bureaus [14], from spontaneously-formed word-of-mouth networks to purposely-designed Pretty Good Privacy [132], from rumour and gossip to online discussion fora, from general-purpose internet search engines such as of Google [49] to specialised academic citation networks, from human-user-oriented Wikipedia [120] to an automated-device-oriented reputation systems for ad-hoc networks, and from systems which evaluate entity reputation to systems which provides item recommendation.

This huge variety is the result of the great diversity in application environments in which feedback mechanisms can operate. This diversity of application environments has many facets. For example:

- There is a wide range of communication infrastructures within which feedback mechanisms may operate.
- There is a wide range of different types of entity which might participate in a feedback mechanism.
- There is a wide range of adversarial situations existing within an application environment that feedback mechanisms need to provide robustness against.
- There may also be different goals that a particular feedback mechanism needs to achieve.

Such diversity in the application environment provides a wide space within which feedback mechanisms can be studied. Extensive research has been conducted into feedback mechanisms used in various application environments, for example [2, 19, 28, 35, 36, 52, 58, 63, 80, 88, 93, 111, 114].

We observe that many studies of feedback mechanisms focus on some particular application environment. For example, many studies, such as [19, 52, 58, 63, 80, 93, 111, 114], focus on systems that evaluate entity reputation while others, such as [2, 36, 88], are devoted to systems that provide item recommendation. Many studies, such as [28, 79, 80], focus on feedback mechanisms that are suitable for peer-to-peer networks. However, studies in different application environments are often carried out in isolation from the wider body of work. Although this is sometimes understandable because there may be a weak association between different application environments, design lessons learnt from one application environment are sometimes not transferred to research in another application environment.

Another observation is that many studies of feedback mechanisms lack a systematic view. Firstly, many designs of feedback mechanisms start from scratch. As a result, existing research results are sometimes not utilised well by new feedback mechanisms. Besides, some research focuses on a particular aspect of feedback mechanisms while ignoring other aspects. Some research only focuses on one type of solution, while ignoring other possible solutions. Some research also focuses on a particular aspect of a feedback mechanism while ignoring the impact of the proposed solution on other aspects. For example, some studies attempt to address one adversarial threat while ignoring the impact of the proposed solution on other adversarial threats. This sometimes results in the threat of interest being addressed at the price of some other threats becoming more difficult to tackle. Lastly, many designs of feedback mechanisms are incomplete. Some important aspects of feedback mechanisms are neglected.

In this chapter, we aim to contribute to research in feedback mechanisms in the following ways:

- We provide an abstract model of a feedback mechanism. This defines the scope of our wider concept of a feedback mechanism. It also identifies the necessary components of a feedback mechanism.

- We provide a framework for feedback mechanisms. This provides a global view of feedback mechanisms that attempts to include a wide range of application environments. Thus feedback mechanisms designed for different application environments can be considered within one framework. It also provides a systematic view of feedback mechanisms, which organises the factors that influence the robustness and efficiency of feedback mechanisms and shows the impact of these factors on the design of feedback mechanisms. This framework allows the approaches and solutions proposed in the literature to be compared, and can be used to demonstrate the impact of one type of solution on another.

This chapter contributes towards a systematic design approach, which takes into consideration a comprehensive range of relevant factors and then develops a suitable feedback mechanism for an application scenario, which is recognised as a key challenge [34, 45]. This work also contributes towards a systematic analysis approach. By decomposing an existing feedback mechanism according to our model and framework, the robustness and performance of the feedback mechanism can be better understood.

2.2 Related Work

As far as the author is aware, there is no existing research providing a framework for feedback mechanisms with a global and systematic view. But there are several studies aiming to organise and categorise feedback mechanisms used in much narrower application scenarios. Some popular application scenarios are reputation systems, recommender systems and peer-to-peer feedback mechanisms. Although existing studies focus on narrower application scenarios, some of them are helpful to establish our framework.

Several studies focus on reputation systems. For example, Noorian and Ulieru [93], Hoffman et al. [52], Ruohomaa et al. [111], Jøsang et al. [58], and Chadwick [19] provide classifications for reputation systems. Some of these also summarise and compare several concrete reputation systems according to their frameworks. Jøsang et al. [58] provide a classification of feedback aggregation algorithms for reputation systems. Sonnek and Weissman [114] provide an experimental comparison of some existing feedback aggregation algorithms used by reputation systems in Grid environments. Jøsang et al. [58] and Dellarocas et al. [35] discuss some issues and solutions related to the design of feedback mechanisms used for electronic marketplaces.

Some studies focus on recommender systems. For example, Desrosiers and Karypis [36] provide a comprehensive survey of neighbourhood-based recommendation methods used by recommender systems. They provide an overview, classification and comparison of neighbourhood-based recommendation methods. They also identify and discuss some necessary components of neighbourhood-based recommendation methods. Lastly, they discuss some problems of neighbourhood-based recommendation methods and provide an overview of possible solutions to these problems. Adomavicius and Tuzhilin [2] provide an overview of recommender systems. They provide a classification of the algorithms used to generate recommendations. They also show various limitations of each type of recommendation method and discuss some possible solutions. Montaner et al. [88] provide a taxonomy of online recommender systems. They also classify a number of online recommender systems according to this taxonomy.

Some studies focus on peer-to-peer feedback mechanisms. For example, Marti and Garcia-Molina [80] provide a taxonomy of peer-to-peer reputation systems. They identify some constraints imposed by the application scenario. They also decompose peer-to-peer reputation systems into several components and sub-components. Marti and Garcia-Molina [79] propose an economic model of peer behaviour in a decentralised resource exchange environments. By applying the model, they discuss some issues related to the design of reputation systems for such environments. They also show some desirable properties of these reputation systems. Daswani et al. [28] provide a model for evaluating reputation systems targeted at mitigating the document authenticity problem in peer-to-peer networks. They use this model to abstract several high-level reputation systems. They also raise several questions related to designing a reputation system.

2.3 A Model of a Feedback Mechanism

In this section we introduce an abstract model of a feedback mechanism. A *feedback mechanism* is an information system that collects, processes and disseminates information about some previous interactions. The most fundamental motivation for feedback mechanisms is that information about previous interactions may reveal some information about future interactions. This abstract model of a feedback mechanism is depicted in Figure 2.1.

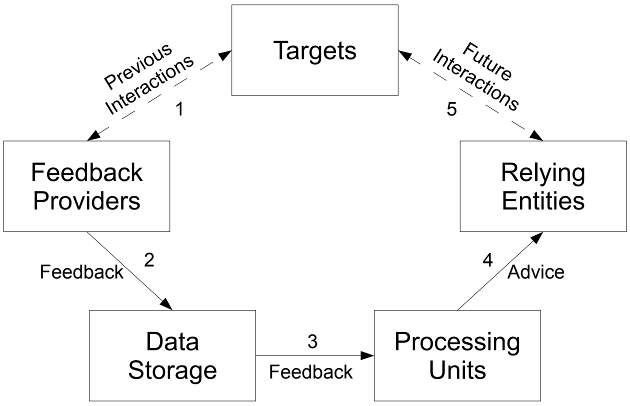

Figure 2.1: An abstract model of a feedback mechanism.

A feedback mechanism comprises the following elements:

- *Entities.* These are entities involved in the operation of the feedback mechanism. There are two types of entity:
  - *Active entities.* These are able to perform some tasks, e.g. computers, human individuals and organisations.
  - *Passive entities.* These are not able to perform tasks, e.g. books, music and films.

  According to the roles played in the feedback mechanism, entities are classified as:
  - *Targets.* These are entities whose *attribute of interest* is evaluated by the feedback mechanism. An attribute of interest is an attribute of a target

that is of interest to the feedback mechanism. Other attributes of targets are not evaluated by the feedback mechanism. Targets can be active or passive entities.

  - *Contributing entities.* These are entities who support the operation of the feedback mechanism. They must be active entities. According to the different tasks that they undertake, they are classified as:
    - *Feedback providers.* These generate and provide *feedback*, which we define shortly.
    - *Data storage units.* These provide storage for data relating to the feedback mechanism.
    - *Processing units.* These generate and provide *advice*, which we define shortly.
  - *Relying entities.* These utilise advice to make decisions about future interactions.

- *Interactions.* These are interactions between feedback providers and targets with respect to the attribute of interest of targets. Interactions are classified as:
  - *Previous interactions.* These have already occurred. Feedback providers rely on their previous interactions to generate their feedback.
  - *Future interactions.* These have not yet occurred. Relying entities rely on advice to make a decision about future interactions.

- *Data.* This is data processed by the feedback mechanism. Data includes:
  - *Feedback.* This contains the feedback providers' evaluation about the attribute of interest of targets. This is the input to the feedback mechanism.
  - *Advice.* This contains evaluation of the attribute of interest of targets aggregated from feedback by processing units.

  Both feedback and advice must contain *evaluation data*, which relates to the attribute of interest of targets. The content of feedback and advice can have the following forms:

  - *Basic form*, if it contains only evaluation data.

- *Extended form*, if it contains additional data other than the evaluation data.

We can also categorise all data related to a feedback mechanism, including feedback and advice, according to its importance:

- *Primary data.* This is essential to and must exist in every feedback mechanism. Evaluation data contained in feedback and advice is primary data.
- *Auxiliary data.* This is all other data, for example, the identity of feedback providers and targets, and time information. Auxiliary data can be provided along with feedback, which expands feedback from basic form to extended form. They can also be provided independently of feedback. Although auxiliary data does not necessarily exist in every feedback mechanism, it is often useful. For example, a digital signature on feedback can be regarded as auxiliary data that provides resilience to forgery and alteration.

The typical information flow among the different roles of a feedback mechanism is as follows:

1. Given that a feedback provider has carried out a previous interaction with a target and obtained some information regarding the attribute of interest of the target, it generates feedback regarding the attribute of interest of the target based on its individual evaluation.
2. The feedback provider makes the feedback available to some data storage units.
3. The data storage units store feedback collected from feedback providers and make it available to some processing units.
4. The processing units generate advice with respect to the attribute of interest of a target based on feedback made available by some data storage units.
5. The advice is disseminated to some relying entities.
6. The relying entities make a decision about a possible future interaction with a target.

2.4 An Overview of Our Framework

In the remaining sections, we propose a framework aiming to identify the main factors that influence the design of feedback mechanisms. We also classify these factors into two main categories:

- *Environmental constraints.* This broad category covers the factors due to the variety of application scenarios for which feedback mechanisms may need to be constructed. Some factors facilitate the design of a robust and efficient feedback mechanism, while others impose constraints on it. More often these factors facilitate one aspect of the design but constrain others. This category includes:

    - *target stability*;
    - *capability of contributing entities*;
    - *motivation of contributing entities*;
    - *availability of contributing entities*;
    - *trust relationships*; and
    - *adversarial models.*

  These will be discussed in Section 2.5.

- *Design choices.* This broad category covers the mainstream design approaches to feedback mechanisms. Many design choices are not suitable for every application scenario but are only suitable for some particular application scenarios. This category includes:

    - *architectural choices*;
    - *data processing choices*; and
    - *robustness solutions.*

  These will be discussed in Section 2.6.

2.5 Environmental Constraints

2.5.1 Target Stability

As defined in Section 2.3, targets are active or passive entities whose attribute of interest is evaluated by a feedback mechanism. The attribute of interest of a target may change over time. We refer to the degree of the change of the attribute of interest of a target as the *stability* of the target. Target stability may vary for different targets and the attribute of interest. For example, if targets are human individuals and the attribute of interest is their date of birth, then the targets are stable. However, if targets are human individuals and the attribute of interest is their reputation, then the targets are often unstable.

The stability of targets influences the design of a feedback mechanism as follows.

- The more unstable a target is, the more challenging it is to predict a future interaction accurately based on previous interactions with the target. In extreme cases, if a target is completely unstable, for example it presents its attribute of interest in a completely random manner, then it is impossible for any approach, including a feedback mechanism, to reveal any information about a future interaction based on previous interactions with the target. Stable targets are relatively easier to predict.
- Target instability results in some data mining techniques which rely on distinguishing conflict feedback used for robustness protection (see Section 2.6.3.3) becoming inapplicable. This is because two feedbacks with different evaluation regarding the same unstable target can both be genuine.

2.5.2 Capability of Contributing Entities

Capability of contributing entities reflects how capably contributing entities perform their roles. We consider the following:

- *Communication capability.* This reflects how capably a contributing entity communicates with other entities. We consider two factors:

  - *Data transmission capability.* This shows the data transmission capability of a contributing entity. For example, an entity in a feedback mechanism operated over the Internet may have a greater data transmission capability than an entity in a feedback mechanism operated over a sensor network.
  - *Connectivity.* This shows how well a contributing entity is connected directly with other entities. For example, if a feedback mechanism is operated over the Internet, then a contributing entity often has very good connectivity. On the other hand, if a feedback mechanism is operated over an ad-hoc network, then a contributing entity often has poor connectivity, as it only has a direct connection with its neighbouring entities.

- *Computational capability.* This reflects how capably a processing unit can perform feedback aggregation. For example, a mainframe computer is able to process a considerably greater amount of data than a portable computing device.

- *Data storage capability.* This reflects how capably a data storage unit can store feedback. For example, a mass storage system is able to store a greater amount of data than a personal computer.

- *Feedback provision capability.* This shows how well a feedback provider can evaluate the attribute of interest of targets. For example, the evaluation of the attribute of interest of target by human feedback providers may be more subject to fluctuation than automated machine feedback providers.

2.5.3 Motivation of Contributing Entities

Motivation of contributing entities reflects their willingness to contribute to a feedback mechanism. It is important to understand such motivation, as contributing requires time, effort and resources, yet the contribution may only benefit other entities.

The source of the motivation of contributing entities is often complex, including aspects of economics, sociology and psychology, etc. For example, in many reputation systems used in electronic marketplaces, a feedback provider does not directly benefit from providing feedback, yet active participation from feedback providers can often be observed. For example, on eBay [41] 52% of buyers and 60% of sellers leave feedback after a transaction [109]. This indicates that many participants have sufficient motivation for leaving feedback.

2.5.4 Availability of Contributing Entities

Availability of a contributing entity reflects how often it is available to a feedback mechanism. Some contributing entities are constantly available while others are intermittently available. For example, in many centralised feedback mechanisms (see Section 2.6.1.2), the centralised data storage unit and processing unit are often constantly available, while feedback providers are intermittently available.

2.5.5 Trust Relationships

Trust relationships reflect the degree to which contributing entities are trusted within a feedback mechanism. Trust itself is often the overall goal of a feedback mechanism: more precisely, to suggest what level of trust a relying entity may reasonably place in a target. Trust is also an important factor in determining feedback mechanism design. For example, suppose all participants, including feedback providers, processing units and data storage units, are trusted. In this case it is not difficult to construct a good feedback mechanism. On the other hand, if there is no trust relationship amongst these participants then it is very challenging to design a satisfactory feedback mechanism.

We classify trust relationships into two groups:

- *Global.* A contributing entity receives global trust if it is trusted by all other entities. For example, in many centralised feedback mechanisms (see Section 2.6.1.2), the data storage unit and processing unit receive global trust.
- *Local.* A contributing entity receives local trust if it is only trusted by some

other entities. For example, in PGP [132] users receive local trust.

We further divide trust relationships into two groups:

- *Static.* A trust relationship is static if it stays unchanged throughout the entire life span of a feedback mechanism. For example, in many centralised feedback mechanisms (see Section 2.6.1.2), the data storage unit and processing unit also receive static trust.
- *Dynamic.* A trust relationship is dynamic if it may change over time. Dynamic trust relationships are often quantitatively represented and updated. For example, in referral networks [123] users receive dynamic trust.

We also divide trust relationships into two categories:

- *Individual.* An individual trust relationship exists when the bearer of the trust is an individual entity. For example, in PGP [132] users receive individual trust.
- *Group.* A group trust relationship exists when the bearer of the trust is a group of entities as a whole and the trust is not placed in individual entities. For example, in Advogato [70] all nodes receive group trust.

2.5.6 Adversarial Models

It is very important to take into consideration adversarial threats when designing feedback mechanisms. In this section, we examine different potential adversaries of a feedback mechanism.

2.5.6.1 Rationality

We firstly classify adversaries into two broad categories:

- *Rational.* A rational adversary attacks a feedback mechanism in order to maximise its own utility.
- *Irrational.* An irrational adversary attacks a feedback mechanism without cost being the primary factor.

While it may be desirable to design a feedback mechanism that is protected against irrational adversaries, it is almost impossible to design a irrational-adversary-proof feedback mechanism for every possible application scenario. This is because feedback mechanisms used in many application scenarios are open systems with little participation barrier. A irrational adversary can participate in the system and inject a great number of malicious feedbacks in order to manipulate the output. Further, it is often sufficient to have a rational-adversary-proof feedback mechanism for many application scenarios. This is because the main motivation of adversaries in many application scenarios is to make some profit. If the cost of the attack outweighs its benefit, then these profit-driven adversaries are discouraged from attacking the feedback mechanism.

2.5.6.2 Location

We further classify adversaries into the following categories:

- *Insiders.* Insiders attack a feedback mechanism by joining the feedback mechanism as legitimate participants and then behaving maliciously. For example, an insider can join a feedback mechanism as a feedback provider and report untruthful feedback in order to influence the advice.
- *Outsiders.* Outsiders attack a feedback mechanism without joining the system as legitimate participants. For example, they can impersonate legitimate participants or manipulate the data maintained by, or exchanged between, legitimate participants.

Insiders and outsiders often attack different components of a feedback mechanism, and employ different attack strategies. The defence techniques to protect against insiders and outsiders are also very different. For example, discouraging insiders from attacking a feedback mechanism and detecting their malicious behaviour are both important defences against insiders. Preventing legitimate entities from being impersonated and preventing legitimate data from being manipulated are both important defences against outsiders.

Outsiders are often less troublesome to feedback mechanisms than insiders. They are also usually easier to address by, for example, the use of cryptographic techniques. In this framework we focus on the more problematic insider adversaries.

2.5.6.3 Strategy Space

We abstract some basic strategies that insiders can adopt in order to attack a feedback mechanism. These form the *strategy space* of insiders, as follows:

- *Sybil identity.* Sybil identity is a strategy of using multiple identities. An insider can re-join a feedback mechanism with a new identity or use multiple identities at the same time. This makes it difficult for the feedback mechanism to link together attacking behaviour conducted under different identities.

- *Fabrication.* Fabrication is a strategy of intentionally sending untruthful information to fool other entities. For example, adversaries can act as feedback providers in order to provide untruthful feedback. They can also act as data storage units in order to send processing units untruthful feedback for aggregation. Similarly, they can act as processing units in order to send relying entities untruthful advice.

- *Non-participation.* Non-participation is a strategy of intentionally not participating. For example, adversaries acting as data storage units intentionally do not provide processing units with stored feedback for aggregation. Adversaries acting as processing units intentionally do not provide relying entities with advice.

- *Collusion.* Collusion is a strategy where multiple adversaries attack a feedback mechanism in a coordinated way. For example, multiple adversaries acting as feedback providers can collude together to intensify the impact of their attacks.

Adversaries can employ basic strategies in arbitrary ways in order to attack a feedback mechanism. For example:

- Adversaries may employ the basic strategy of fabrication to act as feedback providers and report feedback for non-existent interactions, which is often referred to as a *phantom feedback attack* [45].
- Adversaries may employ the basic strategy of fabrication to act as feedback providers and report dishonest feedback for existent interactions, which is often referred to as a *false feedback*, *dishonest feedback* or an *unfair rating attack* [18, 31, 58].
- Adversaries may employ the strategy of sybil identity to act as targets and discard their old identities and use new identities in order to restore their reputation, which is often referred to as a *whitewashing attack* [18, 45, 52].

Adversaries can also arbitrarily combine basic strategies in order to attack a feedback mechanism. For example:

- Adversaries may combine the basic strategies of sybil identity and fabrication by reporting numerous phantom or false feedbacks under the identities of multiple feedback providers, which is often referred to as a *sybil attack* [40].
- Adversaries may also combine the basic strategies of fabrication and collusion by using multiple adversaries acting as feedback providers to collude together to report phantom or false feedback, which is often referred to as a *collusion attack* [45, 52].

2.6 Design Choices

2.6.1 Architectural Choices

Architectural choices of feedback mechanisms include:

- *role setting*;
- *centrality*; and
- *data flow.*

2.6.1.1 Role Setting

In a feedback mechanism, there are four roles that have to be performed by active entities. These are feedback provider F, relying entity R, processing unit P and data storage unit S. There is also one more role that is possibly performed by active entities: target T. We refer to each of these five roles as *atomic*. In a feedback mechanism, an active entity may perform more than one atomic role. For example, in many feedback mechanisms, an active entity can be both a feedback provider and a relying entity. The roles of processing unit and data storage unit are often performed by one active entity. We say that an entity performs a *compound* role if it performs more than one atomic role. A compound role is denoted by the concatenation of the atomic roles. For example, the compound role PS denotes the combination of the atomic roles processing unit P and data storage unit S.

We refer to the design choice of roles (atomic and compound) that make up all the necessary roles of a feedback mechanism as its *role setting*. By a simple calculation, we can derive 52 possible role settings (if target is an active role) that a feedback mechanism can choose from, as shown in Table 2.1.

Table 2.1: Fifty-two possible role settings

1,1,1,1,1	{F,R,T,P,S}				
2,1,1,1	{FR,T,P,S}	{FT,R,P,S}	{FP,R,T,S}	{FS,R,T,P}	{RT,F,P,S}
	{RP,F,T,S}	{RS,F,T,P}	{TP,F,R,S}	{TS,F,R,P}	{PS,F,R,T}
2,2,1	{FR,TP,S}	{FR,TS,P}	{FR,PS,T}	{FT,RP,S}	{FT,RS,P}
	{FT,PS,R}	{FP,RT,S}	{FP,RS,T}	{FP,TS,R}	{FS,RT,P}
	{FS,RP,T}	{FS,TP,R}	{RT,PS,F}	{RP,TS,F}	{RS,TP,F}
3,1,1	{FRT,P,S}	{FRP,T,S}	{FRS,T,P}	{FTP,R,S}	{FTS,R,P}
	{FPS,R,T}	{RTP,F,S}	{RTS,F,P}	{RPS,F,T}	{TPS,F,R}
3,2	{FRT,PS}	{FRP,TS}	{FRS,TP}	{FTP,RS}	{FTS,RP}
	{FPS,RT}	{RTP,FS}	{RTS,FP}	{RPS,FT}	{TPS,FR}
4,1	{FRTP,S}	{FRTS,P}	{FRPS,T}	{FTPS,R}	{RTPS,F}
5	{FRTPS}				

For example:

- In many online reputation systems for electronic marketplaces, such as eBay [41], buyers and sellers are provided with the reputation scores of others. After a transaction, they are invited to provide feedback for each other. They thus act in the compound role of FRT. The marketplace operator, which collects feedback, produces and disseminates reputation scores, thus acts in the compound role PS. So, the role setting adopted by these online reputation systems is {FRT,PS}.
- In many online recommender systems, such as MovieLens [82], users are provided with recommendation about items and are invited to provide feedback about items. They thus act in the compound role of FR. The system operator, which collects feedback, and produces and disseminates recommendation, thus acts in the compound role of PS. Items are the targets and hence act in the atomic role T. The role setting adopted by these online recommender systems is {FR,PS,T}.
- In many web search engine systems, such as Google [49], web pages are the targets to be evaluated and ranked, thus act in the atomic role T. Users request evaluation and rankings of web pages, and thus act in the atomic role R. The web search engines evaluate and generate rankings for web pages, and thus act in the atomic role P. The web page creators generate web pages, which may contain links to other web pages. Hence web pages can be regarded as feedback and web page creators can be regarded as feedback providers, acting in the atomic role F. Web servers maintain web pages, which are provided by web page creators, and provide them to search engines, thus acting in the atomic role S. Hence the role setting adopted by these web search engine systems is {F,R,T,P,S}.

If multiple feedback mechanisms collaboratively operate then entities simultaneously participating in multiple feedback mechanisms may have different roles in different feedback mechanisms. For example, Slashdot [113], a message board forum for posting news, applies two reputation systems to ensure the quality of news posted on its website. In one reputation system, users act in the atomic role F, providing feedback concerning the quality of news posted on the website. In the other reputation system, these users act in another atomic role of T, being evaluated regarding the quality of feedback that they provider in the first reputation system.

The choice of role setting of a feedback mechanism is influenced by the following aspects:

- *Application requirements.* The application scenario sometimes requires the adoption of some compound roles. For example, in electronic marketplaces buyers provide feedback regarding sellers, thus acting as feedback providers. Buyers also retrieve advice regarding sellers, thus acting as relying entities. In this application scenario, the compound role of FR has to be included in the role setting of the feedback mechanism.
- *Capability and motivation of contributing entities* (see Sections 2.5.2 and 2.5.3). If a compound role is adopted then entities acting in this compound role require all the necessary capabilities for each atomic role. These entities must also have sufficient motivation to act in multiple atomic roles.
- *Trust relationships* (see Section 2.5.5). The role setting may be influenced by the trust relationships. For example, if there is no entity acting as a processing unit trusted by relying entities then relying entities may need to compute their own advice. This solution requires the compound role of RP.
- *Aggregation algorithms* (see Section 2.6.2.2). If the feedback aggregation algorithm requires feedback from relying entities in order to compute advice, as in some personalised aggregation algorithms such as those described in [2, 36], then the role setting has to adopt the compound role FR.

In turn, the role setting also influences the design and properties of feedback mechanisms as follows:

- *Centrality* (see Section 2.6.1.2). If the role of processing unit P or data storage unit S is combined with the role of feedback provider F, targets T or relying entities R, then it is impossible to design a centralised feedback mechanism.
- *Robustness* (see Section 2.6.3). The role setting may also influence the robustness of the feedback mechanism. Some role combinations may compromise the

robustness of the feedback mechanism. For example, if a target is designed to maintain the feedback reported concerning itself then it may have an incentive to manipulate the feedback to favour itself. Some role combinations may eliminate some possible robustness threats. For example, choosing the compound role of PS may eliminate the possible threat of data storage units conducting fabrication attacks against processing units.

2.6.1.2 Centrality

The second aspect of the architectural design of a feedback mechanism is the number of entities that are required to perform each role within a given role setting. It is reasonable to assume that there are multiple, or even a large number of, entities performing the roles containing the atomic roles of feedback provider F, relying entity R and target T. The difference thus lies in the number of entities performing the role containing the atomic role of processing unit P and data storage unit S. We refer to this design choice as the *centrality* of a feedback mechanism. Some popular configurations are as follows:

- *Centralised.* There is only one entity performing each of the atomic roles of P and S respectively. This arrangement is seen in many online reputation and recommender systems. However, the processing unit and data storage unit become single points of failure and potential targets of an attack due to their key role in the functioning of the system. Another disadvantage is a lack of scalability.
- *Decentralised.* There are multiple entities performing each of the atomic roles of P and S, such as in many feedback mechanisms designed for peer-to-peer networks.
- *Semi-centralised.* There are multiple entities performing either the role of P or S, and one entity performing the other role.

The centrality of a feedback mechanism is heavily influenced by some environmental and architectural factors:

- *Communication capability* (see Section 2.5.2). It is difficult to adopt a centralised processing unit if the entity acting in the role of processing unit P has limited data transmission capability or poor connectivity. Similarly, it is difficult to adopt a centralised data storage unit if the entity acting in the role of data storage unit S has limited data transmission capability or poor connectivity.

- *Computational capability* (see Section 2.5.2). It is difficult to adopt a centralised processing unit if the entity acting in the role of processing unit P has limited computational capability.

- *Data storage capability* (see Section 2.5.2). It is difficult to adopt a centralised data storage unit if the entity acting in the role of data storage unit S has limited data storage capability.

- *Motivation and availability* (see Sections 2.5.3 and 2.5.4). It is difficult to adopt a centralised architecture if the entities acting in the roles of processing unit P and data storage unit S lack sufficient motivation or availability.

- *Trust relationships* (see Section 2.5.5). It is difficult to adopt a centralised architecture if the entities acting in the roles of processing unit P and data storage unit S do not receive global and static trust.

- *Role setting* (see Section 2.6.1.1). If the atomic roles of processing unit P or data storage unit S are combined with other roles, such as feedback provider F, relying entity R or target T, then there will be multiple entities performing the roles of P or S. This prevents the feedback mechanism from having a centralised architecture.

In turn, the centrality influences other aspects of a feedback mechanism, such as:

- *Data flow* (see Section 2.6.1.3). A centralised architecture results in a "star"-shaped data flow pattern, where multiple feedback providers send feedback to the central data storage unit and the central processing unit sends advice to multiple relying entities. On the other hand, a decentralised architecture results in a "mesh"-shaped data flow pattern, where feedback providers have multiple choices of data storage units to send their feedback to, and relying entities have multiple choices of data processing units from which to obtain advice.

- *Aggregation algorithm* (see Section 2.6.2.2). A decentralised architecture may require an aggregation algorithm which facilitates multiple processing units collaborating together to produce advice.

- *Robustness* (see Section 2.6.3). An appropriate use of centrality may provide some robustness to the feedback mechanism. For example, if a centralised architecture is possible and practical then choosing a centralised architecture eliminates the possible threats of fabrication conducted by the data storage units and processing unit. In addition, a centralised architecture is more suitable for the adoption of some data mining techniques (see Section 2.6.3.3) than a decentralised architecture, because all data is maintained in one place.

2.6.1.3 Data Flow

The third aspect of the architecture design is the data flow of a feedback mechanism. Figure 2.1 illustrates the conceptual data flow amongst different atomic roles. In a concrete feedback mechanism, the design needs to specify the data flow strategy, i.e. the source and (or) destination of data flow, for every entity. It also needs to specify how data flow occurs.

Data flow strategy includes the following basic strategies:

- *Role-based.* In this strategy, one entity chooses another entity acting in a particular role to exchange data. For example, an entity acting as a feedback provider chooses an entity acting as a data storage unit in order to provide feedback.

- *Connection-based.* In this strategy, one entity chooses another entity with which it has established a direct communication channel in order to exchange data. For example, in a mobile ad hoc network (MANET), a node can only directly communicate with its neighbouring nodes using a wireless communication channel.

- *Trust-based.* In this strategy, one entity chooses another entity to exchange data according to its trust in the chosen entity. For example, in a referral system, such as [124], a user acting as a relying entity chooses to ask questions to another user that it trusts.
- *Content-based.* In this strategy, one entity chooses another entity to exchange data based on the content of the data. For example in [100], an entity acting as a feedback provider chooses to report feedback to an entity acting as a data storage unit based on the hash value of the target identity associated with the feedback.

A feedback mechanism can choose any combination of the above-mentioned data flow strategies to form its *overall data flow strategy*.

With respect to the characteristics of data flow, we categorise data flow as follows:

- *Receiver-active.* The receiver sends the sender a request and then the sender responds with some data.
- *Receiver-passive.* The receiver passively receives data sent by the sender.

The design of the overall data flow strategy is influenced by many aspects, such as:

- *Connectivity* (see Section 2.5.2). If entities have limited connectivity then the overall strategy may need to include a connection-based strategy.
- *Trust relationships* (see Section 2.5.5). If there is differentiation in the degree of trust of entities then the overall strategy may need to include a trust-based strategy.
- *Role setting* (see Section 2.6.1.1). If the role setting is not FRTPS, i.e. the role acted by some active entities differs from that acted by other entities, then the overall strategy may need to include a role-based strategy.

In turn, the design of the overall data flow strategy influences some aspects of a feedback mechanism, for example:

- *Robustness* (see Section 2.6.3). A trust-based data flow strategy itself provides a degree of robustness. Also, a receiver-active mode of data flow is more robust than a receiver-passive mode, as the receiver has a choice of sender in receiver-active mode while it has no such choice in receiver-passive mode.

2.6.2 Data Processing Choices

In this section, we examine the design choices related to data processing.

2.6.2.1 Representation of the evaluation data

Evaluation data can be represented in different ways, such as:

- *Numerical score.* This type of representation uses numerical scores to evaluate the attribute of interest of targets. Popular options include:
  - *Discrete value.* This type of representation converts the attribute of interest into a discrete value. For example, eBay [41] allows buyers and sellers to rate each other as positive, neutral and negative (i.e. 1, 0, -1). Scrivener [90] and XRep [26] also adopt a discrete approach to representing evaluation of targets. This type of representation offers a very simple measure of targets. A special case of this type of representation is a binary value. For example, [50] and [122] uses binary representation of advice.
  - *Continuous value.* This type representation converts the attribute of interest into a continuous value. For example, [4], [71] and [118] adopt continuous values to represent advice. Continuous representation is often convenient to represent advice as many aggregation algorithms (see Section 2.6.2.2) output a real number.

  Sometimes evaluation data contains multiple numerical scores.
- *Text.* This type of representation uses text to describe the evaluation of the attribute of interest of targets. For example, eBay [41] enables buyers and

sellers to leave some text comments for each other in addition to numerical scores. The reputation of a trader also includes a list of text comments left for this trader. Many product review websites also adopt text to represent users' evaluation. For example, Epinions [42] and Amazon [3] allow users to write product reviews.

- *Instantiation.* This type of representation only flags up targets whose attribute of interest has a particular property. For example, many email spam filtering schemes allow users to report emails which are considered as spam. In addition, Hoffman et al. [52] summarised a list of peer-to-peer reputation systems, like PGP [132], that adopt this representation. This arrangement is often useful when only a specific property of the target attributes is of interest to a feedback mechanism. An important feature of this design is its simplicity. As a result, the burden on feedback providers, processing units, data storage units and the underlying infrastructure is reduced.
- *Ranking.* This type of representation provides a ranking for multiple targets. For example, many internet search engines, like Google [49], output search results in a sequence. Many recommender systems [2, 36] also output recommendation in a sequence. The advantage of this representation is that it presents a convenient interpretation of the resulting advice. Besides, it is often easier for human users to show their evaluation of multiple targets by using a rank, rather than numerical scores.

We categorise evaluation data according to its subjectivity as follows:

- *Subjective.* Evaluation data describes the attribute of interest according to the subjective taste of an entity.
- *Objective.* Evaluation data describes the attribute of interest according to the objective characteristics of the attribute.

The representation of evaluation data is influenced by the following factors:

- *Data transmission capability* (see Section 2.5.2). If the contributing entities have limited capability for data transmission then a simple representation of evaluation data, such as a numerical score, may be suitable, as it is usually very compact.

- *Characteristics of feedback providers and relying entities.* The representation of evaluation data should be suitable for the characteristics of feedback providers and relying entities. For example, if feedback providers and relying entities are humans then text may be a suitable option. On the other hand, if they are automated machines then numerical scores can be more easily interpreted by automated machines. As another example, humans are often better able to evaluate discrete levels rather than continuous measures [58]. In addition, it is often easier for humans to make their decisions based on a metric with predefined intervals. This is especially true if the continuous metric is not a linear representation [52, 62].

- *Attribute of interest of targets* (see Section 2.3). The representation of evaluation data is also influenced by the characteristics of the the attribute of interest of a target. For example, the choice between single or multiple scores can be made according to the nature of the attribute of interest [35]. When the attributes are complex, such as when the evaluation has to be made from multiple facets, multiple scores may be necessary in order to separate different aspects. On the other hand, if the target attributes do not require being evaluated from multiple aspects, a single score may suffice. If the attribute of interest contains arbitrarily many aspects then text may be more suitable for describing the evaluation of the target.

- *Aggregation algorithms* (see Section 2.6.2.2). The choice of the representation of evaluation data should be compatible with the adopted aggregation algorithm. For example, a Bayesian aggregation algorithm [58] takes binary values as input and produces a continuous value. The algorithms described in [24, 44] take rankings as input and produce a ranking as an output.

2.6.2.2 Aggregation Algorithms

In this section, we examine the algorithms that aggregate feedback to produce advice. We discuss them from the following aspects:

- *Personalisation.* This reflects whether an aggregation algorithm produces personalised or non-personalised advice. We categorise aggregation algorithms into the following groups:
  - *Personalised.* These produce personalised advice. For example, many aggregation algorithms used in recommender systems, such as [12, 30, 89, 105], are personalised.
  - *Non-personalised.* These produce non-personalised advice. For example, many aggregation algorithms used in reputation systems, such as summation and average, median [32], Bayesian [58], belief models [58], PageRank [95], HITs [66] and Eigentrust [62], are non-personalised.
- *Collaboration awareness.* This reflects whether an aggregation algorithm facilitates multiple processing units to aggregate feedback by collaboration. We categorise aggregation algorithms into the following groups:
  - *Collaboration-aware.* These facilitate multiple processing units collaborating together to generate advice. This type is often adopted when there are multiple processing units, and each can only access a share of available feedback. When producing advice, a processing unit often requires input from some other processing units. An example of this category is Eigentrust [62]. The drawbacks of collaborative aggregation algorithms are their reduced efficiency and greater complexity compared with independent aggregation algorithms. However, in some application scenarios, such as in decentralised systems, collaborative aggregation algorithms are more suitable than independent aggregation algorithms.
  - *Collaboration-unaware.* These do not facilitate multiple processing units collaborating together to generate advice. This is often adopted when a processing unit is able to access all necessary feedback. Obviously, aggregation algorithms adopted in centralised feedback mechanisms belong to this category. Advantages of this type of aggregation algorithm are their efficiency and simplicity. However, they are not suitable for application scenarios where collaboration among processing units is needed, such as in decentralised systems.
- *Manipulation resistance.* This reflects how well an aggregation algorithm is immune to adversarial manipulation. We categorise aggregation algorithms

into two broad groups, as follows:

  - *Manipulation-resistant.* These provide resistance against some adversarial manipulations. Adversarial manipulations have little or no impact on the output of a manipulation-resistant aggregation algorithm. For example, hedged-transitive protocols [107], DSybil [129], SybilLimit [127], Pathrank [45], SybilGuard [128] and asymmetric sybilproof reputation functions [22] provide some resilience against sybil attacks, which are a combination of the attacking strategies of sybil identity and fabrication (see Section 2.5.6).
  - *Manipulation-vulnerable.* These are vulnerable to adversarial manipulation. Many aggregation algorithms, such as summation and average [58], are manipulation-vulnerable.

The design choice of aggregation algorithm is influenced by factors such as:

- *Centrality* (see Section 2.6.1.2). Centralised and decentralised feedback mechanisms often require very different aggregation algorithms. Decentralised and semi-decentralised feedback mechanisms with multiple processing units may require a collaboration-aware aggregation algorithm. Centralised and semi-decentralised feedback mechanisms with one processing unit do not require a collaboration-aware aggregation algorithm.
- *Adversarial model* (see Section 2.5.6). Under presence of adversarial attacks, a feedback mechanism may require a manipulation-resistance aggregation algorithm in order to protect against adversarial attacks.
- *Representation of evaluation data* (see Section 2.6.2.1). The choice of the aggregation algorithm should be compatible with the adopted representation of evaluation data.

In turn, the choice of aggregation algorithm may influence a feedback mechanism as follows:

- *Role setting* (see Section 2.6.1.1). If the aggregation algorithm requires feedback from relying entities in order to compute advice, such as some personalised aggregation algorithms described in [2, 36], then the feedback mechanism has to adopt the compound role FR.

- *Data flow* (see Section 2.6.1.3). If the aggregation algorithm is collaboration-aware then the data flow strategy between multiple processing units in order to generate advice needs to be specified.

2.6.3 Robustness Solutions

We first classify robustness solutions into three main categories:

- *Architecture-based.* An architecture-based solution provides robustness by adopting appropriate architectural choices, including role setting, centrality and data flow.
- *Identity-based.* An identity-based solution provides robustness by adopting appropriate rules relating to identity management in a feedback mechanism.
- *Data-processing-based.* A data-processing-based solution provides robustness by adopting appropriate data process choices.

2.6.3.1 Architecture-based Solutions

By adopting some appropriate architectural choices, architecture-based robustness solutions directly provide robustness or facilitate the adoption of identity- and data-processing-based robustness solutions. Architecture-based robustness solutions focus on:

- *Role combination* (see Section 2.6.1.1). This type of approach may eliminate some possible robustness threats by choosing some compound roles. For example, choosing the compound role of PS may eliminate the possible threat of data storage units conducting fabrication attacks against processing units.
- *Centrality* (see Section 2.6.1.2). This type of approach chooses appropriate centrality in order to eliminate some possible threats or enable the adoption

of some other robustness measures. For example, given that there is a trusted entity which is able and willing to perform the compound role of PS in an application scenario, choosing a centralised architecture effectively eliminates the possible threats of fabrication conducted by the data storage units and processing unit.

In addition, choosing an appropriate centrality may also facilitate the adoption of some identity- and data-processing-based robustness solutions, which will be discussed in Sections 2.6.3.2 and 2.6.3.3 respectively. For example, some data mining techniques (see Section 2.6.3.3) are more suitable for a centralised architecture than a decentralised architecture, as all data is collectively maintained by one entity in the centralised architecture while it is separately maintained by multiple entities in the decentralised architecture. Data mining techniques usually perform better when all data is available.

- *Data flow* (see Section 2.6.1.3). This type of approach chooses appropriate choices for data flow in order to eliminate some robustness threats. A common approach is using a trust-based strategy, specifying that an entity exchanges data only with entities that its trusts.

  When the trust relationship (see Section 2.5.5) is *individual* this strategy seems easy to implement: an entity only chooses to exchange data with entities it trusts. For example, in many centralised feedback mechanisms relying entities retrieve advice from, and feedback providers provide feedback to, trusted central servers acting as data storage units and processing units. In a social network, individuals often consult those they trust for advice.

  When the trust relationship (see Section 2.5.5) is *group*, another decision needs to be made about which members among a trusted group an entity should exchange data with. Some approaches include:

  - *Random sampling.* In this case an entity randomly chooses a member among the trusted group.
  - *Redundancy.* In this case an entity chooses multiple members among the trusted group. The same data is exchanged with the multiple entities and then the result is checked for consistency [45].

2.6.3.2 Identity-based Solutions

One key issue in electronic feedback mechanisms is the ease of obtaining a new identity in many digital environments. Identity-based solutions provide robustness by adopting appropriate rules relating to identity management of a feedback mechanism. These include:

- *Proof of identity.* This type of approach takes advantage of a trusted identity service. It requires participants to use identities issued by the trusted identity issuer. For example, participants may be required to use their real world identities or those issued by a trusted authority. For example, Friedman and Resnick [46] proposed a scheme in which an entity is required to use a "once-in-a-lifetime identifier" issued by a trusted authority. An obvious disadvantage of this approach is its impracticality in many application scenarios.

- *Binding attributes to identity.* This type of approach requires participants to bind some attributes to identities. For example, a participant can be required to read a CAPTCHA, so that upon receipt of the correct answer, the participant can be reasonably bound to the attribute of "a human entity" [78]. Another example is that some approaches, such as [40], require participants to bind their IP addresses to identities. However, a disadvantage of this approach is the difficulty in finding an appropriate attribute which is unforgeable and verifiable. For example, the attribute of IP addresses is forgeable.

- *Identity disguise.* This type of approach disguises identities of participants so that an adversary is not able to identify the target that it intends to attack. For example, [31] proposes controlled anonymity for reputation systems applied in online trading scenarios. The identities of buyers and sellers are randomised for every single transaction.

- *Entry barrier.* This type of approach imposes some form of entry barrier on new identities in order to discourage an adversary from acquiring multiple identities. This can be achieved by, for example, charging some monetary payment or requiring completion of some computational task [46, 78]. Although this type of approach may discourage adversaries from obtaining multiple identities, it also indiscriminately discourages participation from non-adversarial entities.

- *Mistrust in new identity.* This type of approach places mistrust in new identities [45], or adaptively assigns trust to newcomers according to the frequency of good behaviour of recent newcomers [8]. A drawback of this type of approach is that it introduces a new problem of *bootstrapping*, where a new participant with a new identity has difficulty in establishing its trust or reputation [76].

Identity-based solutions may be influenced by other aspects of a feedback mechanism as follows:

- *Motivation of contributing entities* (see Section 2.5.3). While some identity-based solutions, such as entry barrier and mistrust in new identity, improve the robustness of a feedback mechanism, they reduce the motivation of non-adversarial contributing entities. If there is insufficient motivation of contributing entities in an application environment then great care should be taken when adopting these identity-based solutions.
- *Adversarial rationality* (see Section 2.5.6.1). If the adversaries in an application environment are irrational then the identity-based solutions that discourage adversarial behaviour, such as entry barrier and mistrust in new identity, will not be effective. Besides, if the adversaries are rational and some identity-based solution that discourages an adversarial behaviour is adopted, then the cost of the adversarial behaviour should be greater than its benefit in order to discourage the adversarial behaviour.

In turn, identity-based solutions may influence some data-processing-based solutions, which will be discussed in Section 2.6.3.3, as follows:

- *Data filtering and weight assigning* (see Section 2.6.3.3). Binding attribute to identity and mistrust in new identity may provide some criteria for data filtering and weight assignment. For example, data provided by identities with a particular attribute may be assigned a greater weight than data provided by those with another attribute. Data provided by entities with a new identity may be assigned a lower weight than data provided by those with an existing identity.

- *Data mining techniques* (see Section 2.6.3.3). Some identity-based solutions, such as proof of identity, entry barrier and mistrust in new identity, enforce or encourage a participant to use a stable identity, which may enhance the performance of data mining techniques. On the other hand, identity disguise may decrease the performance of many data mining techniques, as some potential link amongst data resulting from an identity associated with the data is diminished.

2.6.3.3 Data-processing-based Solutions

Data-processing-based solutions provide robustness by adopting appropriate data process choices. These include:

- *Requiring auxiliary data.* This type of approach requires entities to provide auxiliary data in addition to primary data. Some auxiliary data useful to the robustness of a feedback mechanism is as follows:
  - *Proof of interaction.* This is some evidence relating to an interaction. It allows a feedback provider to justify its feedback. It includes:
    - *Proof of occurrence of interaction.* This is evidence showing that an interaction has indeed occurred. It enables detection of feedback reported for a non-existent interaction. For example, the occurrence of interaction can often be obtained in online reputation systems proprietarily operated by electronic marketplaces, such as the Feedback Forum maintained by eBay [41]. Many electronic marketplaces can verify, to a large extent, the occurrence of a business transaction through the occurrence of a payment transaction. Another form of proof of occurrence of interaction is a target's consensus for being evaluated [115]. This consensus should be unforgeable and obtained ahead of interaction. But this method does not prevent the situation where a target is malicious and colludes with a feedback provider.

    * *Proof of the truthfulness of feedback.* This is evidence showing that a feedback is truthful. It enables detection of untruthful feedback.

  However, proof of interaction is not applicable in many application scenarios. This is because the occurrence of an interaction, or the truthfulness of feedback, often leaves no verifiable evidence, since this is often private knowledge of the involved parties. This results in difficulties for an external party to verify the occurrence of an interaction or the truthfulness of feedback. In some special cases, such as those above-mentioned examples, proof of interaction can be obtained.

- *Identity association.* This allows evaluation data contained in feedback to be associated with identities. For example, a common approach is to include the identity of the feedback provider and target within feedback.
- *Time information.* This allows time information to be contained in feedback or to be associated with feedback. For example, feedback may contain the time of the occurrence of the corresponding interaction. Feedback may also be associated with the time when it is reported.
- *Cryptographic data.* This allows data that is generated by some cryptographic algorithms to be contained in feedback. For example, a digital signature may be included in feedback in order to prevent feedback from being modified maliciously.

Auxiliary data may facilitate other types of data-processing-based solution, as follows:

- *Data filtering or weight assigning.* Some auxiliary data can be used as a criterion for data filtering or weight assigning. For example, proof of interaction may be used as a criterion to conduct data filtering. Data with an invalid proof of interaction may be discarded. Time information may be used as a criterion to assign different weights to data associated with different times, such as in the use of time decay functions [58, 56].
- *Cryptography.* Many cryptographic solutions require use of some cryptographic data, such as a digital signature, hash value or MAC value, in order to fulfil their functionality.
- *Incentive mechanisms* and *trust or reputation update.* These solutions often require that data is associated with the identity of the provider

  of the data. This data thus needs to be provided if these solutions are adopted.

  - *Data mining techniques.* Some auxiliary data, such as data of identity association and time information, often provides helpful information for data mining techniques that are used to detect (ab)normality amongst data.

- *Data filtering or weight assigning.* This type of approach selects a subset of data from, or assigns different weight to, available data before processing. This type of approach includes:

  - *Time-based approaches.* A time-based approach selects, or assigns different weight to, data according to the time information associated with the data. For example, a time decay function [58, 56] assigns less weight to old feedback than recent feedback.
  - *Trust-based approaches.* A trust-based approach selects, or assigns different weight to, data according to the trust or reputation that is associated with the data. For example, in PGP [132], a user only selects a digital certificate certified by users whom he trusts.
  - *Collaborative filtering.* Collaborative filtering selects feedback reported by like-minded feedback providers, which are feedback providers who share a high similarity with a relying entity in terms of commonly reported feedback [47, 108].

- *Cryptography.* Cryptography can be used to prevent data from being manipulated maliciously. It may detect data from being altered maliciously during transmission and storage. Besides, it may prevent an adversary from successfully impersonating a legitimate participant or forging data appearing to be legitimate. It may also enforce data being processed in an intended manner.

- *Manipulation-resistant aggregation algorithms.* These are aggregation algorithms having some immunity to adversarial manipulation. For example, some studies [22, 23, 107, 127, 128, 129] investigate aggregation algorithms which are resilient to sybil attacks. The techniques in [31] use cluster filtering to prevent positive discrimination in the case where controlled anonymity is adopted.

- *Incentive mechanisms.* An incentive mechanism is a set of reward and punishment rules imposed on feedback providers according to their feedback, in order to induce them to report feedback honestly. Some examples of incentive mechanisms are peer-prediction methods [83], side-payment schemes [59, 60], credibility mechanisms [97], trust revelation mechanisms [11, 27], personalised approaches and trust-based incentive mechanisms [130].

- *Trust or reputation update.* This type of approach updates the trust or reputation of participants based on available data, in order to reflect participant trustworthiness more closely. One strategy is to make trust harder to earn than to lose [116, 117]. This type of approach may provide robustness for a feedback mechanism from the following aspects:
  - *Data flow.* This type of approach may result in change of the data flow pattern of an entity. The entity may no longer exchange data with another entity with degraded trust or reputation.
  - *Data filtering or weight assigning.* It may also result in change in the weight of data, according to the trust or reputation of providers of the data.
- *Data mining techniques.* Data mining techniques are methods investigating the logical relationships among the data of a feedback mechanism, in order to infer (ab)normality amongst data. The results can be used in other types of data-processing-based solution as follows:
  - *Data filtering or weight assigning.* The results can be used to ignore or assign insignificant weight to abnormal data.
  - *Incentive mechanisms.* The results can be used by some incentive mechanisms to impose some award or punishment, accordingly, to the providers of data.
  - *Trust or reputation update.* The results also can be used to update the trust or reputation of the providers of data.

2.7 Examples of Feedback Mechanisms

In this section, we use our model and framework to analyse several existing feedback mechanisms. We select a variety of diverse feedback mechanisms in order to demonstrate our model and framework. We will first identify the components of a feedback mechanism according to the model described in Section 2.3. We then decompose the environmental assumptions according to Section 2.5 and the design choices of the scheme according to Section 2.6. Lastly, we perform a brief analysis according to the decomposition of the scheme.

2.7.1 TrustNet

TrustNet [125, 126] is a referral-based reputation system that evaluates the trustworthiness of software agents. If agent A wishes to learn the trustworthiness of agent B, A consults other agents who have directly interacted with B. Agent A finds these agents by seeking and following referrals from its neighbours, a group of agents trusted by A. We outline TrustNet by decomposing it according to our model and framework as follows.

2.7.1.1 Components

We use our model to decompose the components of TrustNet as follows:

- *Entities*. *Feedback providers*, *targets*, *relying entities*, *data storage units* and *processing units* are all represented by software agents. Each software agent acts in a compound role consisting of all five roles.
- *Attributes of interest*: trustworthiness of software agents.
- *Interactions*:
  - *Previous interactions*: previous direct interactions occurred among agents.
  - *Future interactions*: future potential interactions among agents.
- *Data*:
  - *Feedback*: an agent's personal evaluation of its previous interaction with another agent.
  - *Advice*: aggregated evaluation of an agent's trustworthiness.

2.7.1.2 Environmental assumptions

The environmental assumptions of TrustNet are summarised as follows:

- *Target stability*: *unstable.* Targets are active entities and their attribute of interest, i.e. expertise, sociability and cooperativeness, may change over time.
- *Capability of contributing entities*:
    - *Communication capability*:
        - *Data transmission capability*: unspecified.
        - *Connectivity.* An agent has a direct connection with every other agent.
    - *Computational capability*: unspecified.
    - *Data storage capability*: unspecified.
    - *Feedback provision capability*: each agent is able to evaluate accurately its direct interaction with another agent.
- *Motivation of contributing entities*: unspecified.
- *Availability of contributing entities*: unspecified.
- *Trust relationships*: An agent has a group of neighbours, who are trusted to provide the agent with honest feedback or quality referrals. The trust relationship that the agent has with its neighbours fits into the following categories:
    - *Local.* An agent is a neighbour of only some agents.
    - *Individual.* The bearer of trust is always an individual agent.
    - *Dynamic.* The trust is updated over time.
- *Adversarial models*: Possible adversarial models are as follows.
    - *Rationality*: both *irrational* and *rational.* Adversaries may or may not consider the benefit and the cost of their attacks as the primary factor.
    - *Location*: both *insiders* and *outsiders.* Adversaries may attack the system by, or without, joining the system.
    - *Strategy space.* Possible insider adversarial strategy space includes:

- $*$ *Sybil identity.* An adversary can control multiple agents with different identities.
- $*$ *Fabrication.* An adversary acting as an agent can provide a false and misleading response to a query.
- $*$ *Non-participation.* An adversary acting as an agent can provide no response to a query.
- $*$ *Collusion.* Multiple adversaries acting as agents can collude together to attack an agent.

2.7.1.3 Architectural choices

The architectural choices are summarised as follows:

- *Role setting*: $\{FRTPS\}$. Each agent acts as a feedback provider, a relying entity, a target, a processing unit and data storage unit.
- *Centrality*: *decentralised.* There are multiple processing units and data storage units, as every agent is a processing unit and data storage.
- *Data flow.* The overall data flow strategy includes the following basic strategies:
  - *Trust-based.* An agent initiates a query to its neighbours.
  - *Content-based.* An agent follows up a referral by sending a query to the agent specified by the referral.
  - *Receiver-active.* An agent provides a response only if it is requested.

2.7.1.4 Data processing choices

The data processing choices are summarised as follows:

- *Representation of the evaluation data*:
  - In feedback: a *discrete numerical score* from $\{0.0, 0.1, \cdots, 1.0\}$.

  - In advice: a *continuous numerical score* between 0 and 1.

- *Aggregation algorithm*:
  - *Personalisation*: *personalised.* The aggregation algorithm uses the feedback obtained from a referral network of an agent to compute the advice for the agent.
  - *Collaboration awareness*: *collaboration-unaware.* The aggregation algorithm does not require multiple agents collaborating together to compute advice. Advice is computed by individual agents.
  - *Manipulation resistance*: *manipulation-vulnerable.* The aggregation algorithm is not immune to adversarial manipulation of feedback.

2.7.1.5 Robustness solutions

The robustness solution of TrustNet is summarised as follows:

- *Architecture-based solution*: *trust-based data flow strategy.* An agent only initiates a query to its neighbours.
- *Identity-based solution*: unspecified.
- *Data-processing-based solution*:
  - *Trust-based weight assigning.* An agent assigns a higher weight to feedback provided by agents with higher trust level.
  - *Trust update.* An agent update its trust in neighbours and feedback providers according to its personal experience with the target agent.

2.7.1.6 Discussion

One observation of this scheme is that it does not specify the capability, motivation and availability of contributing entities.

A highlight of this scheme is that its robustness solutions relate to trust, as shown in Section 2.7.1.5. As adversaries act as agents and conduct insider attacks, whose strategies are specified in Section 2.7.1.2, the trust in agents controlled by the adversaries decreases and the impact of the attacks decreases.

On the other hand, a disadvantage of this scheme is that an agent may not obtain feedback about a target agent from the referrals of its neighbours. This affects the coverage of the scheme.

2.7.2 PGP

PGP [132] can be considered as a feedback mechanism that facilitates users to verify the correctness of the "ownership" of a public key described in a self-issued public key certificate (hereinafter referred to as a certificate). A user accepts the ownership of the public key only if the certificate is certified by other users whom he trusts. We outline PGP by decomposing the scheme according to our model and framework.

2.7.2.1 Components

We use our model to decompose the components of PGP as follows:

- *Entities*:
    - *Feedback providers*: users of PGP.
    - *Relying entities*: users of PGP.
    - *Processing units*: users of PGP.
    - *Targets*: certificates.
    - *Data storage units*: unspecified. They can be performed by the users or some dedicated data storage services.
- *Attributes of interest*: the correctness of the ownership of the public key described in a certificate.
- *Interactions*:
    - *Previous interactions*: users' personal knowledge about the ownership of public keys described in the certificates.

  - *Future interactions*: potential future use of public keys.

- *Data*:
  - *Feedback*: certification (digital signature) from users on certificates.
  - *Advice*: verification result on the correctness of the ownership of public keys described in certificates.

2.7.2.2 Environmental assumptions

The environmental assumptions of PGP are summarised as follows:

- *Target stability*: *stable.* Targets are passive entities and their attribute of interest, i.e. the correctness of the ownership of the public key described in a public key certificate, can be assumed not to change over time.
- *Capability of contributing entities*:
  - *Communication capability*:
    - *Data transmission capability*: unspecified.
    - *Connectivity*: unspecified.
  - *Computational capability*: unspecified.
  - *Data storage capability*: unspecified.
  - *Feedback provision capability*: Each user is able to evaluate accurately the correctness of the ownership of the public key described in a certificate.
- *Motivation of contributing entities*: unspecified.
- *Availability of contributing entities*: unspecified.
- *Trust relationships*: A user may trust or partially trust in another user's certification of a certificate. The trust relationships in PGP fit into the following categories:
  - *Local.* Trust in a user is only perceived by some users.
  - *Individual.* The bearer of trust is always an individual user.

Note that whether the trust relationship is *static* or *dynamic* is unspecified.

- *Adversarial models*: Possible adversarial models are as follows.
  - *Rationality*: both *irrational* and *rational.* Adversaries may or may not consider the benefit and the cost of their attacks as the primary factor.
  - *Location*: both *insiders* and *outsiders.* Adversaries may attack the system by, or without, joining the system.
  - *Strategy space.* Possible insider adversarial strategy space includes:
    - *Sybil identity.* An adversary can act as users with different identities.
    - *Fabrication.* An adversary acting as a user can provide false certification on a certificate.
    - *Collusion.* Multiple adversaries acting as users can collude together to conduct an attack.

2.7.2.3 Architectural choices

The architectural choices are summarised as follows:

- *Role setting*: {*FRPS,T*} or {*FRP,S,T*}. If the role of data storage unit is performed by users themselves then the role setting is {FRPS,T}. If it is performed by some dedicated data storage services then the role setting becomes {FRP,S,T}.
- *Centrality*: *decentralised* or *semi-decentralised.* If the role of data storage unit is not performed by one entity then the scheme is decentralised; otherwise, it is semi-decentralised.
- *Data flow*: unspecified.

2.7.2.4 Data processing choices

The data processing choices are summarised as follows:

- *Representation of the evaluation data*: Representation of the evaluation data in feedback and advice is as follows:
  - Feedback: *instantiation.* Feedback is represented by a certificate in which a user certifies the ownership of the described public key.
  - Advice: *binary value.* Advice shows whether or not a user should believe in the ownership of a public key described in a certificate.
- *Aggregation algorithm*:
  - *Personalisation*: *personalised.* In order to verify a public key certificate for a user, only the certification provided by the trusted or partially trusted users of the user is selected for aggregation.
  - *Collaboration awareness*: *collaboration-unaware.* The aggregation algorithm does not require multiple users collaborating together to verify a public key certificate. It is conducted by individual users.
  - *Manipulation resistance*: *manipulation-vulnerable.* The aggregation algorithm is not immune to adversarial manipulation of feedback.

2.7.2.5 Robustness solutions

The robustness solutions are summarised as follows:

- *Architecture-based solution*: unspecified.
- *Identity-based solution*: unspecified.
- *Data-processing-based solutions.* These are as follows:
  - *Requiring identity association.* The certification of a certificate has to be associated with the identity of the user providing the certification.
  - *Requiring cryptographic data.* The certification of a certificate has to be represented by a cryptographic digital signature.
  - *Data filtering.* In order to verify a certificate for a user, only the certification provided by the trusted and partially-trusted users are selected for aggregation. The rest of the certifications are ignored.

  - *Cryptography.* Users apply a cryptographic digital signature to protect the certification of a public key certificate from modification. Digital signatures also realise the above-mentioned identity association.

2.7.2.6 Discussion

A highlight of PGP is that it aims to provide a public key certifying mechanism that does not rely on a trusted third party. PGP provides an alternative to popular public key certifying mechanisms, often referred to as public key infrastructure (PKI), which require a trusted third party to act as an issuer of certificates. Another highlight of PGP is its use of cryptography. User certification of the ownership of a public key is by means of a digital signature, which protects the certification from being modified by a malicious entity.

A drawback of PGP is that it neglects some aspects of a feedback mechanism:

- The entities who perform the role of data storage units are not specified. Hence there are two potential role settings, as shown in Section 2.7.2.3.
- The data flow is not specified.
- The capability, motivation and availability of contributing entities are not fully specified.

With respect to the robustness solutions, PGP focuses solely on data-processing-based solutions. A potential improvement is to additionally apply some architecture-based and identity-based solutions.

With respect to the performance of the scheme, a valid certificate may be falsely rejected by a user, due to the lack of certification from its trusted or partially trusted users.

2.7.3 Google

Google [49] is a typical example of a web search engine, an important element of an information system that aims to help internet users to find useful information. This information system can be regarded as a feedback mechanism. We take the information system in which Google is operating as an example and decompose it according to our model and framework. For simplicity, we assume that Google uses only PageRank [95] as its aggregation algorithm.

2.7.3.1 Components

We use our model to decompose the components of the information system in which Google is operating as follows:

- *Entities*:
  - *Feedback providers*: Internet users inserting hyperlinks into web pages, which we hereafter call hyperlink contributors.
  - *Relying entities*: Internet users seeking some useful information.
  - *Processing units*: the web search engine Google.
  - *Data storage units*: web servers hosting web pages.
  - *Targets*: web pages (passive entities).
- *Attributes of interest*: the relative importance [95] of web pages.
- *Interactions*:
  - *Previous interactions*: hyperlink contributors' previous interactions with web pages.
  - *Future interactions*: Internet users' potential future browsing of web pages.
- *Data*:
  - *Feedback*: hyperlinks contained in web pages.
  - *Advice*: a number of web pages ranked according to their relative importance.

2.7.3.2 Environmental assumptions

The environmental assumptions of Google are summarised as follows:

- *Target stability*: *unstable.* The relative importance of web pages may change over time due to potential change and update in the content of web pages.
- *Capability of contributing entities*:
    - *Communication capability*:
        - *Data transmission capability*: *sufficient.* We assume that the Internet, the underlying communication infrastructure, provides sufficient data transmission capability.
        - *Connectivity*: *fully-connected.* We assume that the Internet provides a fully-connected communication infrastructure.
    - *Computational capability*: *sufficient.* We assume that Google has sufficient computational capability to analyse hyperlinks.
    - *Data storage capability*: *sufficient.* We assume that web servers have sufficient data storage capability to host web pages.
    - *Feedback provision capability*: We assume that users inserting hyperlinks into web pages are competent in evaluating the target web pages that the hyperlinks point to.
- *Motivation of contributing entities*:
    - Feedback providers: *sufficient.* Inserting hyperlinks in web pages is a part of the creation or update of the web page.
    - Data storage units: *sufficient.* Hosting web pages is the main functionality of web servers.
    - Processing unit: *sufficient.* Providing web searching is the main business interest of Google.
- *Availability of contributing entities*:
    - Feedback providers: *intermittent.* Users inserting hyperlinks into web pages do not have to be constantly available.

  - Data storage units: *constant.* Web servers can be assumed to be constantly available.
  - Processing unit: *constant.* The search engine can be assumed to be constantly available.

- *Trust relationships*:
  - Feedback providers: users inserting hyperlinks into web pages are implicitly trusted to insert "quality" hyperlinks, i.e. the relative importance of web pages can be derived from the collection of all hyperlinks provided. Trust in feedback providers fits into the following categories:
    - *Global*: It is perceived system-wide.
    - *Static*: It does not change over the life span of the system.
    - *Group*: It is not borne by individuals but, rather, all feedback providers as a whole.
  - Data storage units: web servers are trusted implicitly that they, in general, "honestly" store the web pages provided by feedback providers. Trust in data storage units fits into the following categories:
    - *Global*: It is perceived system-wide.
    - *Static*: It does not change over the life span of the system.
    - *Group*: It is not borne by individuals but, rather, all web servers as a whole.
  - Processing unit: the search engine Google is trusted to "honestly" analyse all provided hyperlinks. Trust in Google fits in the following categories:
    - *Global*: It is perceived system-wide.
    - *Static*: It does not change over the life span of the system, given that Google has a good reputation.
    - *Individual*: It is borne by the individual entity of Google.

- *Adversarial models.* Possible adversarial models are as follows:
  - *Rationality*: both *irrational* and *rational.* Adversaries may or may not consider the cost of their attacks as the primary factor when they try to influence the rank of web pages.
  - *Location*: both *insiders* and *outsiders.* Adversaries may attack the system by, or without, joining the system.

  - *Strategy space*. Possible adversarial strategy space includes:
    - ∗ *Sybil identity*: An adversary may act as multiple feedback providers and data storage units with different identities.
    - ∗ *Fabrication*: An adversary acting as a feedback provider may provide a malicious hyperlink.
    - ∗ *Collusion*: Multiple adversaries acting as feedback providers and data storage units may collude together to carry out an attack.

2.7.3.3 Architectural choices

The architectural choices are summarised as follows:

- *Role setting*: $\{F,R,P,S,T\}$. Each role is acted by a different type of entity.
- *Centrality*: *semi-decentralised*. The role of data storage unit is performed by multiple web servers, while the role of processing unit is performed by one search engine.
- *Data flow*. The overall data flow between entities acting in different roles is shown as follows:
  - Between feedback providers and data storage units:
    - ∗ *Role-based* and *connection-based* strategy: A hyperlink contributor provides its hyperlinks to a web server with which it can establish a connection in terms of updating the hosted web pages.
    - ∗ *Receiver-passive* data flow: Hyperlink contributors directly write hyperlinks into the web pages maintained by web servers without a request from web servers.
  - Between data storage units and the processing unit:
    - ∗ *Role-based* strategy: The Google search engine retrieves web pages from every web server.
    - ∗ *Receiver-active* data flow: A web server responds with a web page upon a request from the Google search engine.
  - Between relying entities and the processing unit:

    * *Role-based* strategy: Internet users query the Google search engine to obtain a search result.
    * *Receiver-active* data flow: The Google search engine responds with a search result upon a request from a user.

2.7.3.4 Data processing choices

The data processing choices are summarised as follows:

- *Representation of the evaluation data*: Representation of the evaluation data in feedback and advice is as follows:
    - Feedback: *instantiation*. Feedback (a hyperlink) refers to a target (a web page).
    - Advice: *ranking*. Advice (a search result) shows a number of web pages in descending order according to their relative importance.
- *Aggregation algorithm*:
    - *Personalisation*: *non-personalised*. In this simplified example where we assume that Google uses PageRank as the only aggregation algorithm, the search result is the same for different users. However, in reality Google additionally adopts other techniques to provide more personalised advice. In this case, the aggregation algorithm provides personalised advice.
    - *Collaboration awareness*: *collaboration-unaware*. The aggregation algorithm is run by the single entity of the search engine.
    - *Manipulation resistance*: *manipulation-vulnerable*. The aggregation algorithm is not immune to adversarial manipulation of feedback [23], as a sybil identity attack, a fabrication attack and a collusion attack mentioned in Section 2.7.3.2 may result in a change of web page ranks.

2.7.3.5 Robustness solutions

The robustness solutions are summarised as follows:

- *Architecture-based solutions*:
    - *Centrality.* A trusted and centralised processing unit is adopted. This prevents an adversary acting as a processing unit from conducting a fabrication attack. In addition, this allows the centralised processing unit to adopt some data mining techniques in order to detect adversarial behaviour.
    - *Data flow.* The search engine proactively retrieves, rather than passively receives, web pages from web servers.
- *Identity-based solutions*:
    - *Proof of identity.* Every web server must have a valid web address so that it can be accessed by the the search engine.
    - *Entry barrier.* Obtaining a web address usually incurs some cost.
- *Data-processing-based solutions*:
    - *Requiring identity association.* A hyperlink contained in a webpage is associated with the web address of this web page.
    - *Data mining techniques.* Some data mining techniques may be applied by the search engine in order to identify abnormality.

2.7.3.6 Discussion

A highlight of Google is its clever choice of feedback. Google takes advantage of the hyperlink structures already existing in web pages as the source of feedback. Since web pages are freely available from web servers, Google is able to obtain rich feedback at negligible cost.

In this scheme, adversaries can act as feedback providers and data storage units. This is because the processing unit is assumed to be a trusted entity, and the targets are passive entities. Adversaries can adopt an attack strategy that is any form or combination of the basic strategies shown in Section 2.7.3.2. A typical strategy is that an adversary, or multiple colluding adversaries, create numerous attacking web pages, containing hyperlinks pointing to each other, and make them available from numerous distinct web addresses. This results in the relative importance of these web pages being boosted. The extent of boost in the relative importance of these web pages depends on the number of distinct web addresses that the adversary creates in order to host the attacking web pages. It also depends on the rationality of the adversary, as follows:

- If adversaries are *irrational*, then they may create as many distinct web addresses as they wish, which can influence the rank of a web page to any extent.
- If they are *rational*, then the number of distinct web addresses that a rational adversary is willing to create is limited to the extent that the cost is not greater than the benefit from the attack. Hence the influence of the rank of a web page will be within a limited range.

The popularity of the Google web search engine may reasonably indicate that the large majority of adversaries are rational.

2.7.4 TrustGuard

TrustGuard [115] is a decentralised reputation system. It aims to facilitate a node, for example a machine, to evaluate the reputation of another node in a decentralised network. We outline TrustGuard by decomposing it according to our model and framework as follows.

2.7.4.1 Components

We use our model to decompose the components of TrustGuard as follows:

- *Entities*: *Feedback providers*, *relying entities*, *processing units*, *data storage units* and *targets* are all represented by nodes in a decentralised network.
- *Attributes of interest*: the reputation of nodes.
- *Interactions*:

  - *Previous interactions*: previous interactions conducted among nodes.
  - *Future interactions*: potential future interactions among nodes.
- *Data*:
  - *Feedback*: a node's personal evaluation about an interaction it participated in.
  - *Advice*: the reputation of a node.

2.7.4.2 Environmental assumptions

The environmental assumptions of TrustGuard are summarised as follows:

- *Target stability*: *unstable.* The reputation of a node may change over time.
- *Capability of contributing entities*:
  - *Communication capability*:
    - *Data transmission capability*: *sufficient.* The scheme assumes that each node has sufficient capability to transmit data to every other node.
    - *Connectivity*: *fully-connected.* The scheme assumes that each node has a direct connection with every other node.
  - *Computational capability*: *sufficient.* The scheme assumes that each node has sufficient capability to perform all related computations, such as feedback aggregation and dishonest feedback filtering.
  - *Data storage capability*: *sufficient.* The scheme assumes that each node has sufficient capability to store the feedback it is designated to.
  - *Feedback provision capability*: *sufficient.* The scheme assumes that each node is able to evaluate interactions accurately.
- *Motivation of contributing entities*: *sufficient.* The scheme assumes that each node has sufficient motivation to participate in the scheme.
- *Availability of contributing entities*: *constant.* The scheme assumes that each node is constantly available in the network.

- *Trust relationships*:
  - Feedback providers: nodes as a whole are trusted with respect to the overall "quality" of feedback that they provide, i.e. meaningful reputation information can be derived by aggregating feedback. Trust in nodes with respect to feedback provision fits in the following categories:
    - *Global*: It is perceived system-wide.
    - *Static*: It does not change over the life span of the system.
    - *Group*: It is not borne by individuals but, rather, all nodes as a whole.
  - Data storage units: nodes are trusted with respect to storing feedback. Trust in nodes with respect to storing feedback fits into the following categories:
    - *Global*: It is perceived system-wide.
    - *Static*: It does not change over the life span of the system.
    - *Individual*: It is borne by an individual node.
- *Adversarial models*:
  - *Rationality*: *rational.* The scheme assumes that adversaries aim to gain advantage from their attacks.
  - *Location*: *insiders.* The scheme assumes that there are only insider adversaries.
  - *Strategy space*:
    - *Sybil identity*: An adversary may act as multiple nodes with different identities.
    - *Fabrication*: An adversary may report false feedback, acting as a feedback provider, or manipulate the feedback that it is designated to store, acting as a data storage unit.
    - *Non-participation*: An adversary may not respond when it is requested for the feedback that it is designated to store.
    - *Collusion*: Multiple adversaries may collude together to carry out an attack.

2.7.4.3 Architectural choices

The architectural choices of TrustGuard are summarised as follows:

- *Role setting*: $\{FRPST\}$. Each node acts as a feedback provider, relying entity, processing unit, data storage unit and target.
- *Centrality*: *decentralised*. There are multiple data storage units and processing units.
- *Data flow*:
  - *Content-based*. The node designated to store the feedback concerning a target node is determined by the hash value of the identifier of the target node. Feedback regarding the target node should be sent to, and retrieved from, the designated node.
  - *Receiver-passive*. Feedback reporting is receiver-passive. A node sends feedback concerning a target node to the designated node without a request from it.
  - *Receiver-active*. Feedback retrieval is receiver-active. A node sends a request to the designated node. The designated node then sends back the feedback concerning the target node.

2.7.4.4 Data processing choices

The data processing choices of TrustGuard are summarised as follows:

- *Representation of the evaluation data*: *continuous numerical score*. A continuous numerical score is used in feedback to show the quality of an interaction, and in advice to show the reputation of a node.
- *Aggregation algorithm*:
  - *Personalisation*: *non-personalised*. The reputation of a target node is computed in the same way for different nodes.

  - *Collaboration awareness*: *collaboration-unaware.* The aggregation algorithm is run by an individual node.
  - *Manipulation resistance*: *manipulation-vulnerable.* The aggregation algorithm is not immune to adversarial manipulation of feedback.

2.7.4.5 Robustness solutions

The robustness solutions of TrustGuard are summarised as follows:

- *Architecture-based solution*: *role-combination.* Each node uses the advice computed by themselves to make a decision. This prevents an adversary from acting as a processing unit to conduct a fabrication attack.
- *Data-processing-based solutions*:
  - *Requiring proof of occurrence of interaction.* Valid feedback about an interaction has to be associated with evidence showing the occurrence of the interaction.
  - *Cryptography.* The proof of occurrence of an interaction is realised by adopting some cryptographic technique of fair electronic exchange, such as [81].
  - *Data mining technique.* A node computes a value measuring its similarity with another node with respect to their commonly reported feedback.
  - *Weight assigning.* A node assigns a high weight to feedback reported by another node with whom it shares a high similarity value.

2.7.4.6 Discussion

A highlight of TrustGuard is that it is a rather complete design of a decentralised feedback mechanism. The design does not overlook any important aspect of a feedback mechanism. In addition, the scheme attempts to adopt some data-processing-based robustness solutions as summarised in Section 2.7.4.5 in addition to its architectural and data processing designs.

However, we observe that the data mining technique, one of the robustness solutions adopted as discussed in Section 2.7.4.5, is not supported by the architectural design of the scheme. Given that node A wants to compute the reputation of node B, and has retrieved feedback reported by node C concerning node B from the node designated to store all feedback reported concerning node B, this data mining technique requires node A to retrieve all feedback reported by node C, concerning nodes other than B, in order to compute the similarity between node A and C. However, we notice that the data flow specified by the scheme, as shown in Section 2.7.4.3, does not support the adoption of this data mining technique. This is because node A does not know from which designated nodes to retrieve the feedback reported by node C concerning nodes other than node B.

2.7.5 eBay Feedback Forum

eBay [41] is a typical electronic marketplace. The feedback forum operated by eBay is a reputation system that facilitates buyers and sellers to establish their reputation in an online environment. We outline the eBay Feedback Forum by decomposing it according to our model and framework as follows.

2.7.5.1 Components

We use our model to decompose the components of the eBay Feedback Forum as follows:

- *Entities*: *Feedback providers*, *relying entities* and *targets* are the traders of eBay. *Processing unit* and *data storage unit* are the central server of the eBay Feedback Forum.
- *Attribute of interest*: the reputation of traders.
- *Interactions*:
  - *Previous interactions*: previous transactions between buyers and sellers.
  - *Future interactions*: potential future transactions between buyers and sellers.

- *Data*:
  - *Feedback*: a buyer or seller's personal evaluation of a transaction in which they participated.
  - *Advice*: a buyer or seller's reputation.

2.7.5.2 Environmental assumptions

The environmental assumptions of the eBay Feedback Forum are summarised as follows:

- *Target stability*: *unstable.* The reputation of a buyer or seller may change over time.
- *Capability of contributing entities*:
  - *Communication capability*:
    - *Data transmission capability*: *sufficient.* Each trader has sufficient capability to transmit data to and from the central server.
    - *Connectivity*: *fully-connected.* Each trader has a direct communication channel with the central server.
  - *Computational capability*: *sufficient.* We assume that the central server has sufficient computational capability for processing feedback.
  - *Data storage capability*: *sufficient.* We assume that the central server has sufficient capability for storing feedback.
  - *Feedback provision capability*: We assume that traders in general are competent in judging the quality of the transactions in which they are involved.
- *Motivation of contributing entities*:
  - Traders as feedback providers: *sufficient.* A large proportion of traders are willing to provide feedback (52% of buyers and 60% of sellers leave feedback after a transaction [109]).
  - The central server: *sufficient.* It is in the central server's business interest to provide data storage and feedback processing services.

- *Availability of contributing entities*:
  - Traders as feedback providers: *intermittent*. Traders are not available constantly.
  - The central server: *constant*. The central server is assumed to be constantly available.
- *Trust relationships*:
  - Traders: They, as a whole, are trusted implicitly to provide sufficiently informative feedback. This trust fits in following categories:
    - *Global*: It is perceived marketplace-wide.
    - *Static*: It does not change over the life span of the marketplace.
    - *Group*: It is not borne by individuals but, rather, all traders as a whole.
  - The central server: It is trusted implicitly to honestly behave as a data storage unit and processing unit. This trust fits in the following categories:
    - *Global*: It is perceived marketplace-wide.
    - *Static*: It does not change over the life span of the marketplace.
    - *Individual*: It is borne by the individual entity of the central server.
- *Adversarial models*. Possible adversarial models are as follows:
  - *Rationality*: both *irrational* and *rational*. Adversaries may or may not consider the cost of their attacks as the primary factor when they try to influence the reputation of a trader.
  - *Location*: *insiders*. Adversaries may attack the system by joining the marketplace.
  - *Strategy space*. The possible adversarial strategy space includes:
    - *Sybil identity*: An adversary may act as multiple traders with different identities.
    - *Fabrication*: An adversary acting as a trader may provide false feedback.
    - *Collusion*: Multiple adversaries acting as traders may collude together to carry out an attack.

2.7.5.3 Architectural choices

The architectural choices of eBay Feedback Forum are summarised as follows:

- *Role setting*: $\{FRT,PS\}$. The traders of eBay act in the compound role of FRT and the central server of the eBay Feedback Forum acts in the compound role of PS.
- *Centrality*: *centralised.* The role of data storage unit and processing unit is performed by a single entity, the central server of eBay.
- *Data flow.* The data flow between traders and the eBay Feedback Forum is shown as follows:
  - From a trader as a feedback provider to the central server: *receiver-passive.* A trader sends feedback directly to the central server without a request from the central server.
  - From the central server to a trader as a relying entity: *receiver-active.* The central server sends a trader the reputation of another trader upon a request from the former.

2.7.5.4 Data processing choices

The data processing choices of the eBay Feedback Forum are summarised as follows:

- *Representation of the evaluation data*: Representation of the evaluation data in feedback and advice is as follows:
  - In feedback: a *discrete numerical score* from {-1, 0, 1} and some *text.*
  - Advice: a *discrete numerical score* and some *text.*
- *Aggregation algorithm*:
  - *Personalisation*: *non-personalised.* The reputation of a trader provided by the central server is the same for every trader.

  - *Collaboration awareness*: *collaboration-unaware.* The aggregation algorithm is run by the single entity of the central server.
  - *Manipulation resistance*: *manipulation-vulnerable.* The aggregation algorithm is not immune to adversarial manipulation of feedback.

2.7.5.5 Robustness solutions

The robustness solutions of the eBay Feedback Forum are summarised as follows:

- *Architecture-based solutions*: *Centrality.* A trusted and centralised processing unit and data storage is adopted. This avoids an adversary acting as a data storage unit and a processing unit. In addition, this allows the centralised processing unit to adopt some data mining techniques in order to detect adversarial behaviour.
- *Data-processing-based solutions*: *Requiring proof of occurrence of interaction.* A trader is allowed to report feedback only if the central server ascertains that the business transaction has occurred. This proof of occurrence of the business transaction is achieved through the central server's knowledge of the occurrence of the monetary transaction associated with the business transaction.

2.7.5.6 Discussion

A highlight of this scheme is its simplicity and robustness solutions. The feedback mechanism takes advantage of the existence of a centralised entity of the application environment: the marketplace. The marketplace is assigned additional roles of the centralised data storage unit and processing unit, which allows a centralised architecture for the feedback mechanism. In addition, the feedback mechanism takes advantage of the fact that marketplace has the information about the occurrence of the monetary transaction associated with a business transaction. This information is utilised as a proof of occurrence of the business transaction to provide some robustness to the scheme.

With respect to the robustness of the scheme, adversaries can act as traders, as the central server is assumed to be a trusted entity, as shown in Section 2.7.5.2. Adversaries can adopt an attack strategy that is any combination of the basic strategies shown in Section 2.7.5.2. A typical strategy is that an adversary, or multiple colluding adversaries, report numerous feedbacks in order to boost or damage the reputation of a trader. The extent of boost or damage of the reputation depends on the number of feedbacks reported by the adversary, which is influenced by the rationality of the adversary. The robustness of the system against irrational and rational adversaries is as follows:

- If adversaries are *irrational*, then they may conduct as many business transactions with the target trader as they wish and then report false feedback, which can influence the reputation of the target trader to any extent.
- If adversaries are *rational*, then the number of business transactions that a rational adversary is willing to conduct with a target trader is limited to the extent that the cost is not greater than the benefit from the attack. Hence the influence of the attack on the reputation of the target trader is limited within a range.

The popularity of eBay, however, may reasonably indicate that the large majority of adversaries are rational.

2.8 Closing Remarks

In this chapter, we provide an abstract model and a comprehensive framework for feedback mechanisms. We also use our model and framework to analyse several existing feedback mechanisms.

This work contributes towards a systematic analysis of feedback mechanisms. By identifying the components according to our abstract model, identifying the environmental constraints and decomposing the design choices according to our framework, the robustness and performance of a feedback mechanism can be better understood.

This work also contributes to a systematic design of a feedback mechanism. After identifying the environmental constraints of an application scenario, our framework helps to identify some design choices that are potentially suitable for the application scenario. Our framework also helps to understand the relationship between different design choices.

Chapter 3

A Secure Marketplace for Online Services that Induces Good Conduct

Contents

Electronic marketplaces are not always easily regulated using traditional legal systems. As a result, suitable dispute prevention and resolution mechanisms for online services are challenging to design. One approach is to adopt informal approaches based on online reputation mechanisms. However, commonly-used reputation systems are subject to a variety of abuses and attacks, many of which artificially boost reputation scores. In this chapter, we propose an electronic marketplace which can be used for trading online services such as computational resources and digital storage. This marketplace incorporates a dispute prevention and resolution mechanism that is explicitly designed to encourage the good conduct of marketplace users, as well as provide important security features and be cost-effective.

3.1 Introduction

An *electronic marketplace* on the Internet gathers users (both service providers and consumers) together to trade goods and services. It can provide a potentially sound solution to benefit the users and society in general compared to traditional marketplaces. It can dramatically increase the number of potential trading partners for a user. It can also enable automated trading, where software agents act on behalf of human users and organisations to perform the tasks of service trading [51]. These include automating the processes of choosing appropriate trading partners, conducting trading transactions and post-purchase evaluation. Consequently, the processes required by human users and organisations are offloaded and these processes can be considerably speeded up.

3.1 Introduction

Dispute prevention and resolution mechanisms are essential for viable marketplaces. However, some traditional approaches, such as state-enforced contractual guarantees, tend to be less effective in electronic marketplaces trading online services. There are several reasons for this [7, 9, 34, 67]. Firstly, the uptake of electronic commerce has tended to outpace the establishment of related legal regulation. Secondly, the one-time deal and multiple jurisdictional nature of online trading presents challenges to conventional legal systems. These problems may lead to electronic marketplace dispute resolution becoming uncomfortably expensive and time consuming.

One of the most common approaches to facilitate trust in electronic marketplaces is to adopt informal dispute prevention and resolution mechanisms based on reputation systems, for example eBay's Feedback Forum and Amazon's Marketplace Trader Ratings. Such informal mechanisms are referred to as *extralegal* [7].

However, commonly-used reputation systems are fairly lightweight mechanisms that are subject to a number of attacks [58]. Dellarocas [34] has shown that the strength of a reputation system can affect the subsequent behaviour of sellers in an online marketplace. Reputation systems must thus be adopted with care if a secure and effective marketplace is to be established.

In this chapter we show that in some cases it is possible to provide a much stronger extralegal framework than the commonly adopted reputation systems, such as eBay's Feedback Forum and Amazon's Marketplace Trader Ratings, at little additional cost. Our proposal works for electronic marketplaces where both of the following properties hold:

- The products or services exchanged in the electronic marketplace can be transformed into an "arbitrable" form at negligible cost. In other words, the violation of a service provision agreement can easily be identified by a third party. We call such products and services *arbitrable.* This condition is rarely satisfied in a conventional marketplace trading tangible goods. However, in an electronic marketplace this condition can often be met for digital goods on which a non-repudiation of origin mechanism [131] can be applied. For example, if the service is computational service, then the correctness of the result of a computational task can often be verified by a third party.

- Provision of traded products or services can easily be replicated at negligible additional cost to service providers or benefit to consumers. We call such services *replicable*. Again, this condition is almost never satisfied in a conventional marketplace, since redelivering a product or service usually involves cost to the service provider and benefit to the consumer. However, this condition often applies to digital goods. For example, if a software vendor allows a buyer to download software that they have already successfully purchased, this comes at little cost to the vendor and negligible benefit to the buyer (indeed, for digital goods such a capability is often highly desirable in the event that the goods become corrupted).

On open networks such as the Internet, arbitrable and replicable online services and digital products become increasingly available. Some typical examples are computational and data storage services. The demand and supply of such services have been growing [55]. It is reasonable to predict that the interest in electronic marketplaces for these types of service will increase.

In the proposed marketplace, we enhance the extralegal system by adopting:

- an automated electronic arbitration service; and
- a robust reputation system which can counter known major attacks.

Our proposed electronic marketplace has the following features:

- *Induction of good conduct.* Participants have a strong incentive to engage in good conduct, where we define *good conduct* to be compliance with the policy of the marketplace.
- *Cost-effectiveness.* This enables participants to find transaction partners who make the most cost-effective offer. Further, the cost of the operation of the marketplace to consumers and providers is within reasonable and acceptable limits.
- *Transaction security.* This ensures that participants receive what they expect from the transaction. In other words:

  - consumers receive satisfactory service from providers upon payment; and
  - providers receive payment in full upon the provision of satisfactory service to consumers.

- *Reputation robustness.* Providers' reputation is protected against malicious damage.
- *Penalty balance.* The financial gain of the *operator* of the electronic marketplace resulting from performing dispute resolution remains the same regardless of its arbitration decision. This ensures that it does not have an incentive to make an unfair dispute resolution that favours any disputing party during arbitration.

In order to demonstrate the scheme, we will use a concrete example of a marketplace in which computational resources are traded (since results of computational tasks are often verifiable and easily redelivered, this satisfies our usage conditions). However, the scheme can easily be abstracted to support a wider range of applications. Some of this work has been published in [72], [73] and [74].

The remainder of the chapter is organised as follows. Section 3.2 discusses some related work. Section 3.3 provides an overview of the structure of the proposed marketplace. Section 3.4 details the operation of the marketplace. In Section 3.5, we provide a preliminary introduction to the terminology and tools used in the subsequent analyses. In Section 3.6, we provide an analysis of the strategies of the marketplace participants. Section 3.7 conducts some evaluation of the scheme. In Section 3.8, we apply the framework proposed in Chapter 2 to decompose and discuss the reputation system adopted by this electronic marketplace. In Section 3.9, we illustrate our proposed marketplace by incorporating it into Grid computing. Lastly, Section 3.10 concludes and identifies further issues.

3.2 Related Work

There are many studies into the design of electronic marketplaces. Similar to the notion of a broker, the use of a "mediator" has been studied, for example [5, 87]. There are a few studies into arbitration services applied to electronic marketplaces. Milosevic et al. [85] demonstrate some benefits of an arbitration service in an electronic marketplace.

There is a great deal of research dedicated to reputation systems used in electronic marketplaces. A number of studies, for example [6, 20, 109], demonstrate the effects of reputation mechanisms in electronic marketplaces. Bakos and Dellarocas [9] provide a comparison between reputation systems and legal enforcement as institutional mechanisms, in terms of their ability to induce cooperative behaviour in electronic marketplaces. Resnick et al. [110] conducted a field experiment showing the value of reputation on a seller in eBay. Dellarocas [34] conducts a detailed case study on eBay's feedback forum, and provides a comprehensive overview of reputation from the perspective of game theory.

However, many reputation systems are vulnerable to attack [58]. In this chapter, we identify an application scenario where the products or services exchanged are *arbitrable* and *replicable*, and for which we design an electronic marketplace that incorporates robustness in the design of the reputation mechanism.

3.3 Structure of the Marketplace

In this section we provide an overview of the structure of our electronic marketplace scheme.

3.3.1 Marketplace Components

There are four main components:

- *Participants*, who are the users of the marketplace and are either consumers, providers, or both.
- A *broker*, which provides an intermediary service between consumers and providers for computational resource exchange.

- An *arbitrator*, which provides a service of arbitration and enforcement of its results.
- A *reputation server*, which collects and provides reputation information for all marketplace providers.

These components are illustrated in Figure 3.1. The functional relationship between the components is as follows:

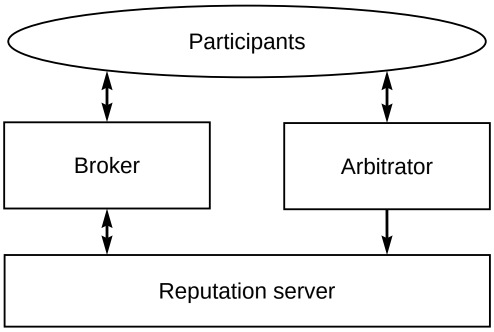

Figure 3.1: Components of the marketplace and their functional relations.

- Participants coordinate with the broker to perform transactions.
- Participants coordinate with the arbitrator to resolve possible disputes which occur during transactions.
- The reputation server coordinates with the broker to exchange reputation information concerning providers. This coordination has two facets:
  - the broker inputs data concerning provider's performance to the reputation server;
  - the reputation server provides the broker with the latest reputation information for providers.
- The arbitrator inputs additional data concerning the performance of providers to the reputation server. This information is derived from transactions involved in disputes.

We require that the broker, arbitrator and reputation server interact honestly with, and trust, each other. Although these components can be separately distributed, one convenient setting is to make them form a single entity, a *central server*, to perform all three functionalities. We will hereafter adopt this setting for simplicity.

3.3.2 Participant Status

All participants have two potential states:

- *Inactive*, which concerns participants who have either never been admitted to the system or have temporarily left.
- *Active*, which concerns participants who are admitted to the system and are engaged in buying or selling computational resources.

Shifts between state are triggered by the events illustrated in Figure 3.2. A broken line indicates that the shift of the state actuated by the corresponding event is decided by the central server, while a solid line means that the shift is decided by a participant. The events are:

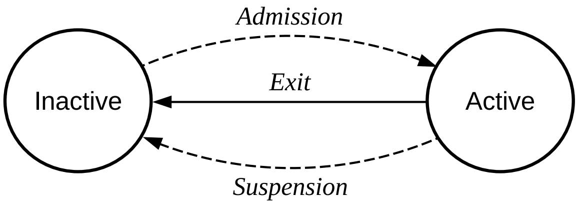

Figure 3.2: Shifts of status.

- *Admission*. An inactive participant is admitted to the system by the central server, shifting its state to active. During admission, the server has to ascertain that:
  - The participant agrees with the rules of the system.
  - It is able to enforce payment for received services, fees and penalties resulting from misbehaviour. This guarantee can take various forms, such as a financial deposit.

Upon admission, the server opens a financial account for a new participant with the balance equal to the amount of financial deposit. This account is reused for this participant for subsequent admissions.

- *Suspension.* The server stops providing service to an active participant. This is initiated by the server and the state of the participant will shift to inactive. Suspension can result from:
  - excessive consumption by a consumer, i.e. its account is in debt;
  - a provider's unacceptable performance, i.e. poor reputation score.
- *Exit.* An active participant voluntarily leaves the system, temporarily or permanently. The state will be shifted to inactive.

3.4 Operation of the Marketplace

This section details the operation of the marketplace. Section 3.4.1 lists all necessary assumptions. Section 3.4.2 introduces some notation. Section 3.4.3 discusses the operation of the reputation server. The operation of the broker is described in Section 3.4.4, while two types of possible misbehaviour which can possibly occur during the operation are illustrated in Section 3.4.5. Section 3.4.6 and 3.4.7 describe the operations of the arbitrator.

3.4.1 Assumptions

In this electronic marketplace, we make some assumptions as follows:

- All participants are rational decision makers, i.e. they do not blindly conform to, or divert from, stipulated rules, but choose their best action to maximise their payoff according to their personal preferences [94]. In a marketplace this is a reasonable assumption since the main goal of participants is to maximise their payoff.
- Entities of the marketplace have sufficient capability to apply the following cryptographic techniques:

  - a public key infrastructure, so that the identity of the central server is verifiable universally;
  - a cryptographic entity authentication mechanism, so that every consumer and provider can be authenticated to the server;
  - a cryptographic non-repudiation mechanism, so that the messages exchanged between transaction partners can be transformed into a non-repudiable form; and
  - a key management system, so that all the required cryptographic keys are securely managed (including their distribution).

  Since all entities participate in the exchange of computational resources, it is reasonable to assume that they have sufficient capability to apply these cryptographic techniques.

- Services to be bought by consumers and sold by providers can be unambiguously specified.
- There is a method for consumers to quantitatively predict the amount of services to buy, or sell, such as memory, bandwidth, and time, etc.
- There is a method to evaluate the quality of service.

3.4.2 Notation

In this section, we introduce some notation that will be used in this scheme, as shown in Table 3.1.

Table 3.1: Notation

Notation	**Meaning**
p	The pre-agreed payment of a transaction, where $p > 0$.
f_0	The service fee if no arbitration is invoked, where $p > f_0 > 0$.
f_1	The service fee if arbitration is invoked, where $p + f_0 > f_1 > f_0$.
f_2	The financial penalty when a participant is found misbehaving during arbitration for repudiable misbehaviour, where $f_2 = 2f_1$.
r^+	Positive feedback.
$r^\times$	Negative feedback.

3.4.3 Operation of the Reputation Server

We define *misbehaviour* as behaviour which results in the expected outcome of a transaction being unachievable. We say that a participant *misbehaves* when it conducts misbehaviour, and a participant *behaves* when it does not conduct misbehaviour. For example, if a provider agrees to provide a consumer with a computational service but sends an incorrect computational result, we say that the provider misbehaves. If the provider does not send any result to the consumer, this is also considered as misbehaving.

Our marketplace uses two types of feedback to describe the behaviour of a provider with respect to every transaction that the participant was previously involved in. If the provider is not found misbehaving, an r^+ feedback is reported for the provider. If it is found misbehaving, an $r^\times$ feedback is reported.

Given a provider i, the reputation server defines X_i as the number of previous transactions in which provider i was not discovered misbehaving, which is the number of r^+ feedbacks reported about i. The reputation server defines Y_i as the number of previous transactions in which i was found misbehaving, which is the number of $r^\times$ feedbacks reported about i. The reputation server uses a reputation computation method to calculate the *reputation* of i. One example of the reputation computation method is the beta reputation computation approach [58] to calculate the reputation of i:

$$R_i = \frac{X_i + 1}{X_i + Y_i + 2}.$$

Although our proposed marketplace still works without this reputation system, incorporating it enhances the performance of the marketplace. Without the reputation system, providers are still induced to engage in good conduct by the design of the broker and arbitrator. However, the reputation system helps consumers to identify providers with good performance and to isolate and avoid those with poor performance. This also offers providers with an additional incentive to good conduct.

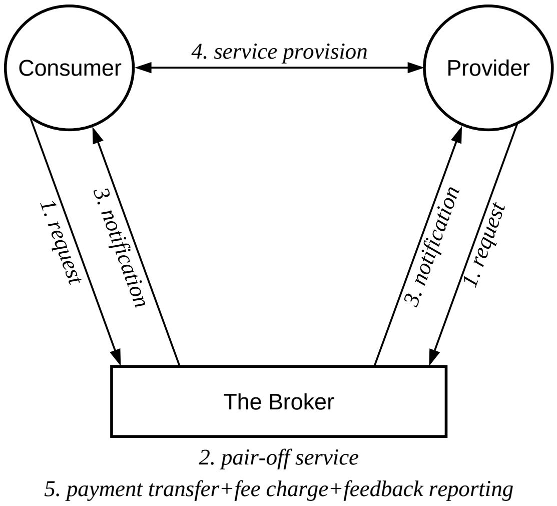

Figure 3.3: The operation of the broker.

3.4.4 Operation of the Broker

This section shows the operation of the broker, as illustrated in Figure 3.3. The procedure is as follows:

1. When a participant wants to buy or sell computational resources, it sends a request for purchase or sale along with the following information:
    - requirements (from consumers) or capabilities (from providers) of computational resources;
    - the length of time that the computational resources are required (by consumers) or offered (by providers);
    - requirements on reputation values of a future partner (from comsumers);
    - an offer of a price (from a consumer) or an expected price (from a provider);
    - optionally, priority for the previously described aspects.
2. Upon receipt of the request, the broker performs a pair-off service so that the requested participant will be partnered with the most suitable participant(s) from the other side. The pair-off procedure can be achieved in different ways, such as:

   - the broker makes a random decision amongst those who are equally qualified;
   - the participant who raises the request makes its own decision upon the broker providing information of all qualified candidates from the other side;
   - the broker negotiates with the two sides to reach a consensus between the two parties on their requirements and offers.

   If the pair-off is successful, the information concerning the new transaction will be added to an *unfinished transaction database*, which stores information of all unfinished transactions. This database has at least four fields, namely *transaction ID*, which uniquely identifies every transaction, *expiry time*, which specifies the expiry time of this transaction (this should allow enough time for participants to request arbitration), *consumer ID* and *provider ID*, which specify the identities of the participants who perform the transaction.

   If the pair-off is unsuccessful, i.e. there is no suitable participant from the other side available, the participant will be notified. It may change its service requirements specified in Step 1 and then make another service request, or request the broker to put it on a waiting list so that when a suitable participant appears, they can be paired-off and start a transaction.

3. The broker notifies the paired-off participants of the identities of their partners and the transaction ID.

4. The paired-off parties directly communicate with each other to conduct the transaction in the pre-agreed manner. Note that all messages exchanged between them should be non-repudiable, i.e. they should be accompanied by a cryptographic commitment such as a digital signature. The receiver should validate the received commitment. During the transaction, if one side is aware of its partner's misbehaviour, then it should immediately invoke an arbitration service provided by the arbitrator. The operation of the arbitrator is detailed in Sections 3.4.6 and 3.4.7.

5. Upon the expiry of the transaction, i.e. when the time reaches the expiry time recorded in the unfinished transaction database, if arbitration has been invoked, then regardless of the completion of the arbitration, the broker terminates its procedure. If no arbitration has been invoked then the broker

debits the consumer and credits the provider by their pre-agreed payment p. Also the broker charges a service fee f_0 from both the consumer and provider. Meanwhile, it reports to the reputation server an r^+ feedback for the provider. Finally, the record of the transaction in the unfinished transactions database is deleted.

The procedure just described involved a set of actions that the participants are advised to comply with in order to successfully complete a transaction. However, it does not prevent misbehaviour, which directly results in an unsatisfactory outcome. We now show how the scheme addresses misbehaviour.

3.4.5 Categorisation of Misbehaviour

Our approach to dealing with misbehaviour is based on the following idea. When misbehaviour occurs, we end it as quickly as possible and apply an appropriate penalty to the participant that misbehaves, as well as offer appropriate compensation to those who are affected. Our mechanism stipulates that whenever a participant is aware of its partner's misbehaviour, it should immediately invoke one of the two corresponding arbitration services that will be described in Section 3.4.6 and Section 3.4.7. The choice between the two arbitration services should be made according to the characteristics of the misbehaviour. We divide misbehaviour into two classes, as follows:

- *Non-repudiable misbehaviour.* The participant who misbehaves is not able to later deny its misbehaviour. This is achieved by applying cryptographic mechanisms of non-repudiation of origin to all communication between providers and consumers. For example, a provider sending an incorrect result with its digital signature to the consumer is non-repudiable misbehaviour. A consumer sending a signed task which requires more computational resources than it requested is also considered as non-repudiable misbehaviour.
- *Repudiable misbehaviour.* The participant who misbehaves is later able to deny the misbehaviour. A third party is not able to ascertain whether the accuser falsely censures the accused or the accused indeed misbehaved. For example,

when a participant accuses its partner of not responding within a pre-agreed period of time, the arbitrator is not able to judge whether the participant is falsely accusing its partner or the partner indeed did not respond within the pre-agreed period of time. Hence, not responding within a pre-agreed period of time is a type of repudiable misbehaviour.

It is stipulated that when a participant faces non-repudiable misbehaviour, it should invoke the arbitration service described in Section 3.4.6. If it faces repudiable misbehaviour then it should have recourse to the arbitration service described in Section 3.4.7.

3.4.6 Operation of the Arbitrator for Non-repudiable Misbehaviour

The operation of the arbitrator for non-repudiable misbehaviour is as follows:

1. A participant, either a consumer or a provider, sends a request to the arbitrator for arbitration, along with all related evidence.
2. The arbitrator first checks if the received transaction ID, consumer ID and provider ID matches a record in the unfinished transactions database. If so, the arbitration proceeds. If not, the participant requesting arbitration is financially punished and the arbitration terminates.
3. The arbitrator verifies the validity of the evidence.
4. If the provider is found misbehaving or making false accusation then the arbitrator punishes the provider by debiting the amount f_1 and sends the reputation server an $r^{\times}$ feedback for the provider. Meanwhile it does not charge the consumer any money.

   If the consumer is found misbehaving or making a false accusation then the arbitrator punishes the consumer by debiting the amount $p + f_1$. Meanwhile it compensates the provider by the amount p and sends the reputation server an r^{+} feedback for the provider.
5. The partnership between the two participants terminates. The record of the transaction in the unfinished transactions database is deleted. Further, the

arbitrator checks if any participant meets the criteria of being suspended. If so, the participant is suspended.

The financial punishment and compensation for the arbitration of non-repudiable misbehaviour is summarised in Table 3.2.

Table 3.2: The financial penalty and compensation for non-repudiable misbehaviour arbitration

Arbitration Result	**Provider**	**Consumer**
Provider misbehaving or false accusation	$-f_1$	0
Consumer misbehaving or false accusation	p	$-p - f_1$

3.4.7 Operation of the Arbitrator for Repudiable Misbehaviour

It is extremely challenging to arbitrate based on a report of repudiable misbehaviour without any further investigation, since it is unknown whether the accuser falsely censures the accused, or the accused indeed misbehaved. Our scheme stipulates a simple process for such a situation. This process can immediately identify accidental misbehaviour. We also show that it is not the best strategy for any participant to intentionally misbehave with respect to repudiable misbehaviour. The process is described as follows:

1. A participant, either consumer or provider, requests arbitration.
2. The arbitrator first checks if the received transaction ID, consumer ID and provider ID matches a record in the unfinished transactions database. If so, the arbitration proceeds. If not, the participant requesting arbitration is financially punished and the arbitration terminates.
3. The arbitrator requests that both consumer and provider repeat the communication between them for this transaction, i.e. all messages sent to each other during Step 4 as described in Section 3.4.4, via the arbitrator. Note that because the service provision is *replicable*, as we assumed in Section 3.1, repetition of the service provision results in negligible additional cost to the provider or benefit to the consumer.

4. Both participants communicate via the arbitrator.
5. If the provider is found misbehaving then the arbitrator debits the provider by the amount f_2, and sends an $r^{\times}$ feedback for the provider. Meanwhile, it does not charge the consumer any money. If the consumer is found misbehaving then the arbitrator debits the consumer by the amount $p + f_2$. Meanwhile, it credits the provider the amount p and sends an r^{+} feedback for the provider.
6. If no misbehaviour is detected at the end of the transaction, the arbitrator debits the consumer and credits the provider the pre-agreed payment p, since the transaction has now been completed. However, it charges a service fee f_1, instead of f_0, from both the consumer and provider. The arbitrator sends an r^{+} feedback for the provider.
7. The partnership between the two participants is terminated. The record of the transaction in the unfinished transactions database is deleted. Besides, the arbitrator checks if any participant meets the criteria of being suspended. If so, the participant is suspended.

The financial punishment and compensation for repudiable misbehaviour arbitration is summarised in Table 3.3.

Table 3.3: The financial penalty and compensation for repudiable misbehaviour arbitration

Arbitration Result	**Provider**	**Consumer**
Provider misbehaving	$-f_2$	0
Consumer misbehaving	p	$-p - f_2$
No misbehaving	$p - f_1$	$-p - f_1$

3.5 Preliminary to Rational Robustness

In this section, we introduce the terminology and tools used in the subsequent analysis of our marketplace.

We first define the *utility* of an entity. We say the *utility* of an entity is an abstract payoff, or benefit, that the entity can gain from a particular behaviour. The utility is measured by the *utility function*, which quantitatively measures the extent of payoff, or benefit, that the entity can gain from its behaviour. The concept of utility provides means of comparing the payoff, or benefit, of different behaviour.

For example, let i denote an entity. Suppose there are two behaviours, 0 and 1, that i can choose from, and the outcome of behaviour 0 and 1 are x and y, respectively. Let $\mathcal{U}_i$ denote the utility of entity i and it is measured by the utility function $\mathcal{U}_i(\cdot)$, i.e. $\mathcal{U}_i = \mathcal{U}_i(\cdot)$. The utilities of i with respect to behaviour 0 and 1 can be written as $\mathcal{U}_i^0 = \mathcal{U}_i(x)$ and $\mathcal{U}_i^1 = \mathcal{U}_i(y)$, respectively. Suppose $\mathcal{U}_i(x) < \mathcal{U}_i(y)$, then we get that $\mathcal{U}_i^0 < \mathcal{U}_i^1$, i.e. the utility from behaviour 0 is less then that from behaviour 1 for entity i.

We now define the *strategy* of an entity. We say that a *strategy* of an entity is a plan of behaviour designed to achieve a particular goal. From the previous example, suppose outcome x is the goal of entity i, then the strategy of i should be behaviour 0, since only behaviour 0 will lead to outcome x.

We now introduce the concept of *rationality.* We say that an entity is *rational* if and only if the entity always chooses a strategy that maximises its utility. From the previous example, suppose there are only two behaviours 0 and 1 that entity i can choose from. If entity i is rational, then its strategy is to conduct behaviour 1 instead of 0, since $\mathcal{U}_i^0 < \mathcal{U}_i^1$.

In this chapter, we assume that all participants are rational (see Section 3.4.1). We say that the robustness of our marketplace achieved under this assumption is *rational robustness.* In the subsequent sections, we will analyse our marketplace with respect to its rational robustness.

3.6 Participant Behaviour Analysis

In this section, we analyse the strategies of participants. We show in each case that misbehaving is not the best strategy for any participant, independent of the strategy of the other participant.

3.6.1 Utility Functions and Notation

Let $\mathcal{U}_i$ denote the overall utility, or benefit, that the participant i gains from a transaction. The utility of a consumer in one transaction is determined by three factors: service received, financial payment, and time spent. Formally we define the utility function for consumer i as follows:

$$\mathcal{U}_i = \mathcal{U}_i(\cdot)_s + \mathcal{U}_i(\cdot)_b + \mathcal{U}_i(\cdot)_t,$$

where $\mathcal{U}_i(\cdot)_s$, $\mathcal{U}_i(\cdot)_b$ and $\mathcal{U}_i(\cdot)_t$ respectively denote the utility functions of the service received, financial payment and time spent. The utility of a provider in one transaction is determined by four factors: service provision, financial gain or loss (financial loss is possible if the provider is found misbehaving), time spent and reputation gain or loss. We formally define the utility function of provider i as follows:

$$\mathcal{U}_i = \mathcal{U}_i(\cdot)_d + \mathcal{U}_i(\cdot)_b + \mathcal{U}_i(\cdot)_t + \mathcal{U}_i(\cdot)_r,$$

where $\mathcal{U}_i(\cdot)_d$, $\mathcal{U}_i(\cdot)_b$, $\mathcal{U}_i(\cdot)_t$ and $\mathcal{U}_i(\cdot)_r$ respectively denote the utility functions of service provision, financial gain or loss, time spent, and reputation gain or loss.

In our scheme a participant can be both a consumer and a provider. But in a transaction, a participant acts only in one role. Its utility then corresponds with the role that the participant acts.

We assume that individual components of the utility functions have their own algebraic properties.

With respect to $\mathcal{U}_i(\cdot)_s$, a consumer's utility function based on a received service, we assume that the service is not time critical, i.e. $\mathcal{U}_i(\cdot)_s$ remains the same when the service is received in one, or more than one, transaction. We classify the service into two categories. We denote by s^+ and $s^\times$, respectively, a received service that does, or does not, meet the pre-agreed quality requirements. We assume that:

- $\mathcal{U}_i(s^\times)_s = 0$, i.e. the utility of a received service that does not meet the pre-agreed quality requirements is zero to consumer i.
- $\mathcal{U}_i(s^+)_s > 0$, i.e. the utility of a received service that meets the pre-agreed quality requirements is greater than zero to consumer i.

With respect to $\mathcal{U}_i(\cdot)_d$, a provider's utility function based on its delivered service, we classify the service into two categories. We denote by d^+ and $d^\times$, respectively, a service that does, or does not, meet the pre-agreed quality requirements. It is reasonable to assume that:

- $\mathcal{U}_i(\cdot)_d \leq 0$, i.e. providing a service may incur some cost to provider i.
- $\mathcal{U}_i(d^+)_d < \mathcal{U}_i(d^\times)_d$, i.e. providing a satisfactory service d^+ incurs more cost than a unsatisfactory service $d^\times$ to provider i.
- $\mathcal{U}_i(\cdot)_d$ is a non-increasing function against time, i.e. the cost of providing a service d to provider i does not decrease against time.

We denote by $\mathcal{U}_i(\cdot)_b$ a participant's utility function based on the financial balance that it achieves as a result of the transaction. The financial balance consists of the pre-agreed payment p and the two types of service fee f_0 and f_1. We assume that:

- $\mathcal{U}_i(\cdot)_b$ is a linear function, i.e. $\mathcal{U}_i(b_1)_b + \mathcal{U}_i(b_2)_b = \mathcal{U}_i(b_1 + b_2)_b$.
- $\mathcal{U}_i(\cdot)_b$ is an increasing function, i.e. if $b_1 < b_2$ then $\mathcal{U}_i(b_1)_b < \mathcal{U}_i(b_2)_b$.

With respect to $\mathcal{U}_i(\cdot)_t$, a participant's utility function based on its time spent on a transaction, we assume that:

- $\mathcal{U}_i(\cdot)_t \leq 0$, i.e. a participant spending time on a transaction may incur some cost.
- $\mathcal{U}_i(\cdot)_t$ is a linear function, i.e. $\mathcal{U}_i(t_1)_t + \mathcal{U}_i(t_2)_t = \mathcal{U}_i(t_1 + t_2)_t$.
- $\mathcal{U}_i(\cdot)_t$ is a decreasing function, i.e. if $t_1 > t_2$, then $\mathcal{U}_i(t_1)_t < \mathcal{U}_i(t_2)_t$.

With respect to $\mathcal{U}_i(\cdot)_r$, a provider's utility function based on the feedback reported about a transaction, we assume that:

- $\mathcal{U}_i(r^\times)_r < 0$, i.e. negative feedback results in some negative utility.

- $\mathcal{U}_i(r^+)_r > 0$, i.e. positive feedback brings some positive utility.
- $\mathcal{U}_i(r^+)_r < \mathcal{U}_i(f_0)_b$, i.e. the utility resulting from a positive feedback is less than the utility of the service fee f_0. In another word, the possible increase of reputation resulting from a positive feedback is insignificant.
- The change (increase or decrease) of reputation of a provider resulting from one feedback gives negligible utility to other providers. In other words, the marketplace has a sufficient number of providers so that the change of reputation of a provider resulting from one feedback gives insignificant competition advantage to another provider.

3.6.2 Participant Utility Inequalities

Let T denote the pre-agreed time for a transaction. It is easily seen that if a consumer i sends a purchase request to the broker, then it is expected to benefit from the transaction if the transaction is completed smoothly. In other words, if consumer i sends a purchase request to the broker, then the overall utility of obtaining a satisfactory service s^+ at financial cost of $p + f_0$ and time cost of T is greater than zero, i.e. the following inequality holds:

$$\mathcal{U}_i(s^+)_s + \mathcal{U}_i(-p - f_0)_b + \mathcal{U}_i(T)_t > 0. \tag{3.1}$$

Likewise, if a provider i sends a sale request, then it is expected to benefit from the transaction if the transaction is completed smoothly. In other words, if consumer i sends a sale request to the broker, then the overall utility of receiving $p - f_0$ amount of financial payment and a positive feedback is greater than the cost of providing a satisfactory service d^+ using T period of time, i.e. the following inequality holds:

$$\mathcal{U}_i(d^+)_d + \mathcal{U}_i(p - f_0)_b + \mathcal{U}_i(T)_t + \mathcal{U}_i(r^+)_r > 0. \tag{3.2}$$

3.6.3 Non-repudiable Misbehaviour

In this section, we analyse participants' strategies with respect to non-repudiable misbehaviour. In particular, we will show that it is not the best strategy for a participant to conduct non-repudiable misbehaviour or to falsely accuse its transaction partner of conducting non-repudiable misbehaviour.

First, we will show that if a participant is behaving and has discovered its partner conducting some non-repudiable misbehaviour and has acquired valid evidence, then its best strategy is to request arbitration.

Lemma 1. *If a participant i is behaving and has discovered its partner conducting some non-repudiable misbehaviour and has acquired valid evidence then $\mathcal{U}_i^N < \mathcal{U}_i^Y$, where $\mathcal{U}_i^Y$ and $\mathcal{U}_i^N$ denote the utilities that i gains if it does or does not request arbitration, respectively.*

Proof. If i is a consumer then i receives an unsatisfactory service $s^\times$ (as its partner has conducted some misbehaviour). If i requests arbitration and provides the evidence, then the arbitration will result in no financial loss or gain for i (as the evidence is valid). The time that i spends on this transaction will not be longer than the pre-agreed time for the transaction T (since once i has requested arbitration, it stops spending time on this transaction.) Thus, $\mathcal{U}_i^Y$ can be formalised as:

$$\mathcal{U}_i^Y = \mathcal{U}_i(s^\times)_s + \mathcal{U}_i(0)_b + \mathcal{U}_i(T'), \text{ where } T' \leq T. \tag{3.3}$$

If i does not request arbitration, then it will be debited amount $p + f_0$ by the broker. The time that i spends on this transaction is T (as it has to remain involved in the transaction until it completes). Thus $\mathcal{U}_i^N$ can be formalised as:

$$\mathcal{U}_i^N = \mathcal{U}_i(s^\times)_s + \mathcal{U}_i(-p - f_0)_b + \mathcal{U}_i(T)_t. \tag{3.4}$$

where $T' \leq T$.

From (3.3) and (3.4), we have that:

$$\mathcal{U}_i^Y - \mathcal{U}_i^N = \mathcal{U}_i(s^\times)_s - \mathcal{U}_i(s^\times)_s + \mathcal{U}_i(0)_b - \mathcal{U}_i(-p - f_0)_b + \mathcal{U}_i(T')_t - \mathcal{U}_i(T)_t.$$

It is obvious that $\mathcal{U}_i(s^\times)_s - \mathcal{U}_i(s^\times)_s = 0$. By the assumptions stated in Section 3.6.1, we have that $\mathcal{U}_i(0)_b - \mathcal{U}_i(-p - f_0)_b > 0$, since $-p - f_0 < 0$, and $\mathcal{U}_i(T')_t - \mathcal{U}_i(T)_t \geq 0$, since $T' \leq T$. Thus, we derive that $\mathcal{U}_i^N < \mathcal{U}_i^Y$.

If i is a provider then i can only provide an unsatisfactory service $d^\times$, regardless of whether or not i requests arbitration, since its partner has conducted misbehaviour (see Section 3.4.3 about the definition of misbehaviour). If i requests arbitration and provides the evidence, then i will be credited amount p by the arbitrator. The time that i spends on this transaction will be no longer than the pre-agreed time for the transaction T (since once i has requested arbitration, it stops spending time on this transaction). In addition, i will be reported a positive feedback r^+ by the arbitrator. Thus we derive that:

$$\mathcal{U}_i^Y = \mathcal{U}_i(d^\times)_d + \mathcal{U}_i(p)_b + \mathcal{U}_i(T')_t + \mathcal{U}_i(r^+)_r, \text{ where } T' \leq T. \tag{3.5}$$

If i does not request arbitration then it will be credited amount $p - f_0$ by the broker. The time that i spends on this transaction is T (as it has to remain involved in the transaction until it completes). In addition, i will be reported a positive feedback r^+ by the broker. Thus we derive that:

$$\mathcal{U}_i^N = \mathcal{U}_i(d^\times)_d + \mathcal{U}_i(p - f_0)_b + \mathcal{U}_i(T)_t + \mathcal{U}_i(r^+)_r. \tag{3.6}$$

From (3.5) and (3.6), we have that:

$$\begin{aligned}\mathcal{U}_i^Y - \mathcal{U}_i^N = & \mathcal{U}_i(d^\times)_d - \mathcal{U}_i(d^\times)_d + \mathcal{U}_i(p)_b - \mathcal{U}_i(p - f_0)_b \\ & + \mathcal{U}_i(T')_t - \mathcal{U}_i(T)_t + \mathcal{U}_i(r^+)_r - \mathcal{U}_i(r^+)_r.\end{aligned}$$

It is obvious that $\mathcal{U}_i(d^\times)_d - \mathcal{U}_i(d^\times)_d = 0$ and $\mathcal{U}_i(r^+)_r - \mathcal{U}_i(r^+)_r = 0$. By the assumptions stated in Section 3.6.1, we have that $\mathcal{U}_i(p)_b - \mathcal{U}_i(p - f_0)_b > 0$, since $p > p - f_0$, and $\mathcal{U}_i(T')_t - \mathcal{U}_i(T)_t \geq 0$, since $T' \leq T$. Thus, we derive that $\mathcal{U}_i^N < \mathcal{U}_i^Y$.

Hence the lemma holds for both consumer and provider. □

We will now show that if a participant is behaving, conducting non-repudiable misbehaviour is not the best strategy for a participant.

Theorem 2. *If a participant i is behaving then $\mathcal{U}_i^Y < \mathcal{U}_i^N$, where $\mathcal{U}_i^Y$ and $\mathcal{U}_i^N$ denote the utilities that i gains if it conducts non-repudiable misbehaviour or continues to behave, respectively.*

Proof. Note that Lemma 1 proves that the transaction partner of i will request arbitration if i conducts non-repudiable misbehaviour.

We first consider the situation where i is a consumer. If i behaves, then it expects to receive a satisfactory service s^+, to be debited the amount $p + f_0$ by the broker and to spend T period of time on the transaction. Thus, the utility of i is as follows:

$$\mathcal{U}_i^N = \mathcal{U}_i(s^+)_s + \mathcal{U}_i(-p - f_0)_b + \mathcal{U}_i(T)_t. \tag{3.7}$$

If i conducts non-repudiable misbehaviour then i will receive an unsatisfactory service $s^\times$, because its misbehaviour causes its transaction partner to not be able to provide a satisfactory service s^+ (see Section 3.4.3 about the definition of misbehaviour). In addition, its transaction partner will request arbitration, according to Lemma 1. Consequently, i will be debited the amount $p + f_1$ by the arbitrator. The time spent on the transaction will be no longer than the pre-agreed time for the transaction T (since once i has requested arbitration, it stops spending time on this transaction). Thus the utility of i is as follows:

$$\mathcal{U}_i^Y = \mathcal{U}_i(s^\times)_s + \mathcal{U}_i(-p - f_1)_b + \mathcal{U}_i(T')_t, \text{ where } T' \leq T. \tag{3.8}$$

From (3.7) and (3.8), we have that:

$$\begin{aligned} \mathcal{U}_i^Y - \mathcal{U}_i^N = {} & \mathcal{U}_i(s^\times)_s - \mathcal{U}_i(s^\times)_s + \mathcal{U}_i(-p - f_1)_b - \mathcal{U}_i(-p - f_0)_b \\ & + \mathcal{U}_i(T')_t - \mathcal{U}_i(T)_t. \end{aligned}$$

By a similar argument to the proof of Lemma 1, we derive that $\mathcal{U}_i^Y < \mathcal{U}_i^N$.

We now consider the situation where i is a provider. If i behaves, then it expects to deliver a satisfactory service d^+, to receive $p - f_0$ amount of financial payment, to spend T period of time on this transaction and to receive a positive feedback r^+. Thus the utility of i is as follows:

$$\mathcal{U}_i^N = \mathcal{U}_i(d^+)_d + \mathcal{U}_i(p - f_0)_b + \mathcal{U}_i(T)_t + \mathcal{U}_i(r^+)_r. \tag{3.9}$$

If i conducts non-repudiable misbehaviour then i will provide an unsatisfactory service $d^\times$. In addition, its transaction partner will request arbitration, according to Lemma 1. Consequently, i will be debited the amount f_1 by the arbitrator. The time spent on the transaction will be no longer than the pre-agreed time for the transaction T. The arbitrator will provide a negative feedback $r^{\times}$ for i. Thus the utility of i is as follows:

$$\mathcal{U}_i^Y = \mathcal{U}_i(d^{\times})_d + \mathcal{U}_i(-f_1)_b + \mathcal{U}_i(T')_t + \mathcal{U}_i(r^{\times})_r, \text{ where } T' < T. \tag{3.10}$$

From (3.10), since $\mathcal{U}_i(d^{\times})_d \leq 0$, $\mathcal{U}_i(-f_1)_b < 0$, $\mathcal{U}_i(T')_t \leq 0$ and $\mathcal{U}_i(r^{\times})_r < 0$, we have that $\mathcal{U}_i^Y < 0$. Besides, by (3.2), we have that $\mathcal{U}_i^N > 0$. Hence $\mathcal{U}_i^Y < \mathcal{U}_i^N$.

Therefore this theorem holds for both consumer and provider. □

We will now show that a participant falsely accusing its transaction partner of conducting non-repudiable misbehaviour is not a good strategy.

Theorem 3. *If a participant i is behaving and has not discovered its partner conducting any non-repudiable misbehaviour then $\mathcal{U}_i^Y < \mathcal{U}_i^N$, where $\mathcal{U}_i^Y$ and $\mathcal{U}_i^N$ denote the utilities that a participant i gains if it does or does not request arbitration, respectively.*

Proof. If i is a consumer, then let $\mathcal{U}_i^{Y_0}$ and $\mathcal{U}_i^{Y_1}$ denote the utilities of i in the cases where i has received a satisfactory or unsatisfactory service at the moment of requesting arbitration, respectively. If i is a provider, then let $\mathcal{U}_i^{Y_0}$ and $\mathcal{U}_i^{Y_1}$ denote the utilities of i in the cases where i has provided a satisfactory or unsatisfactory service at the moment of requesting arbitration, respectively. We show that $\mathcal{U}_i^{Y_0} < \mathcal{U}_i^N$ and $\mathcal{U}_i^{Y_1} < \mathcal{U}_i^N$.

We first consider the situation where i is a consumer. If i does not request arbitration, by the same argument to derive (3.7), we have that:

$$\mathcal{U}_i^N = \mathcal{U}_i(s^+)_s + \mathcal{U}_i(-p - f_0)_b + \mathcal{U}_i(T)_t. \tag{3.11}$$

If i requests arbitration and has received a satisfactory service s^+, then i will be debited the amount $p + f_1$ by the arbitrator. The time spent on the transaction will be T, as it takes T to provide a satisfactory service. Thus we derive that:

$$\mathcal{U}_i^{Y_0} = \mathcal{U}_i(s^+)_s + \mathcal{U}_i(-p - f_1)_b + \mathcal{U}_i(T)_t. \tag{3.12}$$

From (3.11) and (3.12), since $-p - f_0 > -p - f_1$, we have that $\mathcal{U}_i(-p - f_0)_b > \mathcal{U}_i(-p - f_1)_b$. Hence $\mathcal{U}_i^{Y_0} < \mathcal{U}_i^N$.

If i requests arbitration and has received a unsatisfactory service $s^{\times}$, the time spent on the transaction will not be longer than T. Thus we derive that:

$$\mathcal{U}_i^{Y_1} = \mathcal{U}_i(s^{\times})_s + \mathcal{U}_i(-p - f_1)_b + \mathcal{U}_i(T')_t, \text{ where } T' < T. \tag{3.13}$$

By (3.1), we have that $\mathcal{U}_i^N > 0$. From (3.13), since $\mathcal{U}_i(s^{\times})_s = 0$, $\mathcal{U}_i(-p - f_1)_b < 0$ and $\mathcal{U}_i(T')_t \leq 0$, we have that $\mathcal{U}_i^{Y_1} < 0$. Hence $\mathcal{U}_i^{Y_1} < \mathcal{U}_i^N$.

We now consider the situation where i is a provider. If i does not request arbitration then by the same argument to derive (3.9), we have that:

$$\mathcal{U}_i^N = \mathcal{U}_i(d^+)_d + \mathcal{U}_i(p - f_0)_b + \mathcal{U}_i(T)_t + \mathcal{U}_i(r^+)_r. \tag{3.14}$$

If i requests arbitration and has provided a satisfactory service d^+, then i will be debited the amount f_1 by the arbitrator. The time spent on the transaction will be T. Additionally, i will receive a negative feedback reported by the arbitrator. Thus we derive that:

$$\mathcal{U}_i^{Y_0} = \mathcal{U}_i(d^+)_d + \mathcal{U}_i(-f_1)_b + \mathcal{U}_i(T)_t + \mathcal{U}_i(r^{\times})_r. \tag{3.15}$$

From (3.14) and (3.15), since $\mathcal{U}_i(-f_1)_b < \mathcal{U}_i(p - f_0)_b$ and $\mathcal{U}_i(r^{\times})_r < \mathcal{U}_i(r^+)_r$, we have that $\mathcal{U}_i^{Y_0} < \mathcal{U}_i^N$.

If i requests arbitration and has provided a unsatisfactory service $d^{\times}$, the time spent on the transaction will be no longer than T. Thus we derive that:

$$\mathcal{U}_i^{Y_1} = \mathcal{U}_i(d^{\times})_d + \mathcal{U}_i(-f_1)_b + \mathcal{U}_i(T')_t + \mathcal{U}_i(r^{\times})_r, \text{ where } T' < T. \tag{3.16}$$

By (3.2), we have that $\mathcal{U}_i^N > 0$. From (3.16), since $\mathcal{U}_i(d^{\times})_d \leq 0$, $\mathcal{U}_i(-f_1)_b < 0$, $\mathcal{U}_i(T')_t \leq 0$ and $\mathcal{U}_i(r^{\times})_r < 0$, we have that $\mathcal{U}_i^{Y_1} < 0$. Hence $\mathcal{U}_i^{Y_1} < \mathcal{U}_i^N$.

Therefore the lemma holds for both consumer and provider. □

To summarise, it is not the best strategy for a rational participant to conduct non-repudiable misbehaviour or falsely accuse its partner of conducting non-repudiable misbehaviour.

3.6.4 Repudiable Misbehaviour

In this section, we analyse participants' strategies with respect to repudiable misbehaviour. In particular, we will show that it is not the best strategy for a participant to conduct repudiable misbehaviour or to falsely accuse its transaction partner of conducting repudiable misbehaviour.

Firstly, we show that if a consumer has conducted a repudiable misbehaviour but its partner does not request arbitration, then it is the best strategy for the consumer itself to request arbitration.

Lemma 4. *If a consumer i has conducted a repudiable misbehaviour but its transaction partner does not request arbitration then $\mathcal{U}_i^N < \mathcal{U}_i^Y$, where $\mathcal{U}_i^Y$ and $\mathcal{U}_i^N$ denote the utilities that i gains if it itself does or does not request arbitration, respectively.*

Proof. If i does not request arbitration then i will receive an unsatisfactory service $s^\times$ (as it has conducted some misbehaviour). Besides, i will be debited the amount $-p - f_0$. The time that i spends on this transaction will be T. Thus, we derive that:

$$\mathcal{U}_i^N = \mathcal{U}_i(s^\times)_s + \mathcal{U}_i(-p - f_0)_b + \mathcal{U}_i(T)_t. \tag{3.17}$$

If i request arbitration then i will receive an satisfactory service s^+ during the arbitration. In addition, i will be debited the amount $p + f_1$. The time that i spends on this transaction will be $2T$, twice as long as the pre-agreed time for the transaction, because the transaction is repeated once. Thus we derive that:

$$\mathcal{U}_i^Y = \mathcal{U}_i(s^+)_s + \mathcal{U}_i(-p - f_1)_b + \mathcal{U}_i(2T)_t. \tag{3.18}$$

From (3.17) and (3.18) we have that:

$$\mathcal{U}_i^Y - \mathcal{U}_i^N = \mathcal{U}_i(s^+)_s - \mathcal{U}_i(s^\times)_s + \mathcal{U}_i(-p - f_1)_b - \mathcal{U}_i(-p - f_0)_b + \mathcal{U}_i(2T)_t - \mathcal{U}_i(T)_t.$$

By the assumptions stated in Section 3.6.1, we have that:

$$\begin{aligned}&\mathcal{U}_i(s^{\times})_s = 0,\\ &\mathcal{U}_i(-p - f_1)_b - \mathcal{U}_i(-p - f_0)_b = \mathcal{U}_i(f_0 - f_1)_b, \text{ and}\\ &\mathcal{U}_i(2T)_t - \mathcal{U}_i(T)_t = \mathcal{U}_i(T)_t.\end{aligned}$$

Hence, we have that:

$$\mathcal{U}_i^Y - \mathcal{U}_i^N = \mathcal{U}_i(s^+)_s + \mathcal{U}_i(f_0 - f_1)_b + \mathcal{U}_i(T)_t.$$

Since $f_1 < p + f_0$ (see Table 3.1), we have that $f_0 - f_1 > -p - f_0$. Hence, $\mathcal{U}_i(f_0 - f_1)_b > \mathcal{U}_i(-p - f_0)_b$. Thus we have that:

$$\mathcal{U}_i^Y - \mathcal{U}_i^N > \mathcal{U}_i(s^+)_s + \mathcal{U}_i(-p - f_0)_b + \mathcal{U}_i(T)_t > 0,$$

by (3.1). □

We now show that if a provider is behaving and has discovered its partner conducting some repudiable misbehaviour, its best strategy is to request arbitration.

Lemma 5. *If a provider i is behaving and has discovered its partner conducting some repudiable misbehaviour then $\mathcal{U}_i^N \leq \mathcal{U}_i^Y$, where $\mathcal{U}_i^Y$ and $\mathcal{U}_i^N$ denote the utilities that i gains if it does or does not request arbitration, respectively.*

Proof. We will prove this lemma in two steps. Firstly we will consider the situation where the consumer will not behave again before the transaction expires. Secondly we will consider the situation where the consumer will do so.

With respect to the former case. If i does not request arbitration then i will provide an unsatisfactory service $d^{\times}$ (since its transaction partner has conducted misbehaviour, i is not able to provide a satisfactory service s^+). It will be debited the amount $p - f_0$ by the broker. The time spent on the transaction is T and i will be reported a positive feedback r^+ by the broker. Thus we derive that:

$$\mathcal{U}_i^N = \mathcal{U}_i(d^{\times})_d + \mathcal{U}_i(p - f_0)_b + \mathcal{U}_i(T)_t + \mathcal{U}_i(r^+)_r. \tag{3.19}$$

If i requests arbitration then i will provide an unsatisfactory service $d^{\times}$ (since its transaction partner will not behave again, i is not able to provide a satisfactory service s^+ during arbitration). It will be debited the amount p by the arbitrator. The time spent on the transaction is less than T. A positive feedback r^+ will be reported for i by the arbitrator. Thus we derive that:

$$\mathcal{U}_i^Y = \mathcal{U}_i(d^\times)_d + \mathcal{U}_i(p)_b + \mathcal{U}_i(T')_t + \mathcal{U}_i(r^+)_r, \text{ where } T' < T. \tag{3.20}$$

From (3.19) and (3.20), since $p > p - f_0$ and $T' < T$, we have that $\mathcal{U}_i(p)_b > \mathcal{U}_i(p - f_0)_b$ and $\mathcal{U}_i(T')_t > \mathcal{U}_i(T)_t$. Then we derive that $\mathcal{U}_i^N < \mathcal{U}_i^Y$.

Now we consider the latter case. We prove in Lemma 4 that the consumer will trigger arbitration if the provider does not do so. Therefore arbitration will be requested by either the provider or the consumer. The utility of i will remain the same no matter whether or not i requests arbitration. Hence $\mathcal{U}_i^N = \mathcal{U}_i^Y$.

Therefore, in both cases, $\mathcal{U}_i^N \leq \mathcal{U}_i^Y$. □

We now show that if a consumer is behaving and has discovered its partner conducting some repudiable misbehaviour, its best strategy is to request arbitration.

Lemma 6. *If a consumer i is behaving and has discovered its partner conducting some repudiable misbehaviour then $\mathcal{U}_i^N < \mathcal{U}_i^Y$, where $\mathcal{U}_i^Y$ and $\mathcal{U}_i^N$ denote the utilities that i gains if it does or does not request arbitration, respectively.*

Proof. Let $\mathcal{U}_i^{Y_0}$ and $\mathcal{U}_i^{Y_1}$ denote i's utilities if it requests arbitration and during the arbitration the provider behaves or continues misbehaving, respectively. We show that $\mathcal{U}_i^N < \mathcal{U}_i^{Y_0}$ and $\mathcal{U}_i^N < \mathcal{U}_i^{Y_1}$.

We first consider the situation where i does not request arbitration. Consumer i will receive an unsatisfactory service $s^\times$ (since its transaction partner has conducted some misbehaviour). It will be debited the amount $p + f_0$. The time spent on the transaction will be T. Thus we have that:

$$\mathcal{U}_i^N = \mathcal{U}_i(s^\times)_s + \mathcal{U}_i(-p - f_0)_b + \mathcal{U}_i(T)_t. \tag{3.21}$$

We now consider the situation where i requests arbitration and its transaction partner behaves during the arbitration. Consumer i will receive a satisfactory service

s^+. It will be debited the amount $-p - f_1$. The time spent on the transaction will be $2T$. Thus we have that:

$$\mathcal{U}_i^{Y_0} = \mathcal{U}_i(s^+)_s + \mathcal{U}_i(-p - f_1)_b + \mathcal{U}_i(2T)_t. \tag{3.22}$$

If i requests arbitration and its transaction partner continues misbehaving then i will receive an unsatisfactory service $s^\times$. It will not be credited or debited any money. The time spent on the transaction will be less than T. Thus we have that:

$$\mathcal{U}_i^{Y_1} = \mathcal{U}_i(s^\times)_s + \mathcal{U}_i(0)_b + \mathcal{U}_i(T')_t, \text{ where } T' < T. \tag{3.23}$$

From (3.21) and (3.22) we have that:

$$\begin{aligned}\mathcal{U}_i^{Y_0} - \mathcal{U}_i^N &= \mathcal{U}_i(s^+)_s + \mathcal{U}_i(f_0 - f_1)_b + \mathcal{U}_i(T)_t \\ &> \mathcal{U}_i(s^+)_s + \mathcal{U}_i(-p - f_0)_b + \mathcal{U}_i(T)_t \\ &> 0,\end{aligned}$$

by a similar argument to the proof of Lemma 4. Hence $\mathcal{U}_i^N < \mathcal{U}_i^{Y_0}$.

From (3.21) and (3.23) we have that:

$$\mathcal{U}_i^{Y_1} - \mathcal{U}_i^N = \mathcal{U}_i(p + f_0)_b + \mathcal{U}_i(T')_t - \mathcal{U}_i(T)_t.$$

Since $p + f_0 > 0$, we have that $\mathcal{U}_i(p + f_0)_b > 0$. Since $T' < T$, we have that $\mathcal{U}_i(T')_t - \mathcal{U}_i(T)_t > 0$. Hence, $\mathcal{U}_i^N < \mathcal{U}_i^{Y_1}$. □

Now, we will show that if a participant is behaving then conducting repudiable misbehaviour is not the best strategy.

Theorem 7. *If a participant i is behaving then $\mathcal{U}_i^N < \mathcal{U}_i^Y$, where $\mathcal{U}_i^Y$ and $\mathcal{U}_i^N$ denote the utilities that i gains if it conducts repudiable misbehaviour or continues to behave, respectively.*

Proof. Lemma 5 and Lemma 6 prove that if i conducts repudiable misbehaviour then its partner will request arbitration. Let $\mathcal{U}_i^{Y_0}$ and $\mathcal{U}_i^{Y_1}$ denote i's utilities if i conducts repudiable misbehaviour and then behaves or continues to misbehave during the arbitration, respectively. We show that $\mathcal{U}_i^{Y_0} < \mathcal{U}_i^N$ and $\mathcal{U}_i^{Y_1} < \mathcal{U}_i^N$

We first consider the situation where i is a consumer. If i behaves, by the argument to derive (3.7), we have that:

$$\mathcal{U}_i^N = \mathcal{U}_i(s^+)_s + \mathcal{U}_i(-p - f_0)_b + \mathcal{U}_i(T)_t. \tag{3.24}$$

If i conducts repudiable misbehaviour and then behaves during the arbitration, then i will receive a satisfactory service s^+. It will be debited the amount $p + f_1$. The time spent on the transaction will be $2T$. Thus we have that:

$$\mathcal{U}_i^{Y_0} = \mathcal{U}_i(s^+)_s + \mathcal{U}_i(-p - f_1)_b + \mathcal{U}_i(2T)_t. \tag{3.25}$$

From (3.24) and (3.25), since $-p - f_0 > -p - f_1$, we have that:

$$\mathcal{U}_i(-p - f_0)_b > \mathcal{U}_i(-p - f_1)_b.$$

Since $T < 2T$, we have that $\mathcal{U}_i(T)_t > \mathcal{U}_i(2T)_t$. Hence, we have that $\mathcal{U}_i^{Y_0} < \mathcal{U}_i^N$.

If i conducts repudiable misbehaviour and then continues to misbehave during the arbitration, then i will receive a unsatisfactory service $s^\times$. It will be debited the amount $-p - f_2$. The time spent on the transaction will be less than T. Thus we have that:

$$\mathcal{U}_i^{Y_1} = \mathcal{U}_i(s^\times)_s + \mathcal{U}_i(-p - f_2)_b + \mathcal{U}_i(T')_t, \text{ where } T' < T. \tag{3.26}$$

From (3.26), since $\mathcal{U}_i(s^\times)_s = 0$, $\mathcal{U}_i(-p - f_2)_b < 0$ and $\mathcal{U}_i(T')_t < 0$, we have that $\mathcal{U}_i^{Y_1} < 0$. Besides, $\mathcal{U}_i^N > 0$ by (3.1). Hence we have that $\mathcal{U}_i^{Y_1} < \mathcal{U}_i^N$.

We now consider the situation where i is a provider. If i behaves, by the argument to derive (3.9), we have that:

$$\mathcal{U}_i^N = \mathcal{U}_i(d^+)_d + \mathcal{U}_i(p - f_0)_b + \mathcal{U}_i(T)_t + \mathcal{U}_i(r^+)_r. \tag{3.27}$$

If i conducts repudiable misbehaviour and then behaves during the arbitration, then i will provide a satisfactory service s^+. It will be credited the amount $p - f_1$. The time spent on the transaction will be $2T$. Thus we have that:

$$\mathcal{U}_i^{Y_0} = \mathcal{U}_i(d^+)_d + \mathcal{U}_i(p - f_1)_b + \mathcal{U}_i(2T)_t + \mathcal{U}_i(r^+)_r. \tag{3.28}$$

From (3.27) and (3.28), since $\mathcal{U}_i(p - f_0)_b > \mathcal{U}_i(p - f_1)_b$ and $\mathcal{U}_i(T)_t > \mathcal{U}_i(2T)_t$, we have that $\mathcal{U}_i^{Y_0} < \mathcal{U}_i^N$.

If i conducts repudiable misbehaviour and then continues to misbehave during the arbitration, then i will provide an unsatisfactory service $s^{\times}$. It will be debited the amount $-f_2$. The time spent on the transaction will be less than T. Thus we have that:

$$\mathcal{U}_i^{Y_1} = \mathcal{U}_i(d^{\times})_d + \mathcal{U}_i(-f_2)_b + \mathcal{U}_i(T')_t + \mathcal{U}_i(r^{\times})_r, \text{ where } T' < T. \tag{3.29}$$

From (3.29), since $\mathcal{U}_i(d^{\times})_d \leq 0$, $\mathcal{U}_i(-f_2)_b < 0$, $\mathcal{U}_i(T')_t \leq 0$ and $\mathcal{U}_i(r^{\times})_r \leq 0$, we have that $\mathcal{U}_i^{Y_1} < 0$. Besides, by (3.2), we have that $\mathcal{U}_i^N > 0$. Hence $\mathcal{U}_i^{Y_1} < \mathcal{U}_i^N$. □

Finally, we show that if a participant is behaving and has not discovered its partner conducting any repudiable misbehaviour, then it is not the best strategy for the participant to request arbitration.

Theorem 8. *If a participant i is behaving and has not discovered its partner conducting any repudiable misbehaviour then $\mathcal{U}_i^Y < \mathcal{U}_i^N$, where $\mathcal{U}_i^Y$ and $\mathcal{U}_i^N$ denote i's utilities if it does or does not request arbitration, respectively.*

Proof. Let $\mathcal{U}_i^{Y_0}$ and $\mathcal{U}_i^{Y_1}$ denote i's utilities if it requests arbitration and behaves, or misbehaves, during the arbitration, respectively.

We first consider the situation where i is a consumer. If i behaves, by the argument to derive (3.7), we have that:

$$\mathcal{U}_i^N = \mathcal{U}_i(s^+)_s + \mathcal{U}_i(-p - f_0)_b + \mathcal{U}_i(T)_t. \tag{3.30}$$

If i requests arbitration and then behaves during the arbitration, then i will receive a satisfactory service s^+. It will be debited the amount $p + f_1$. The time spent on the transaction will be $2T$. Thus we have that:

$$\mathcal{U}_i^{Y_0} = \mathcal{U}_i(s^+)_s + \mathcal{U}_i(-p - f_1)_b + \mathcal{U}_i(2T)_t. \tag{3.31}$$

From (3.30) and (3.31), we have that $\mathcal{U}_i^{Y_0} < \mathcal{U}_i^N$.

If i request arbitration and then continues to misbehave during the arbitration, then i will receive an unsatisfactory service $s^{\times}$. It will be debited the amount $-p - f_2$. The time spent on the transaction will be less than T. Thus we have that:

$$\mathcal{U}_i^{Y_1} = \mathcal{U}_i(s^{\times})_s + \mathcal{U}_i(-p - f_2)_b + \mathcal{U}_i(T')_t, \text{ where } T' < T. \tag{3.32}$$

From (3.32), since $\mathcal{U}_i(s^{\times})_s = 0$, $\mathcal{U}_i(-p - f_2)_b < 0$ and $\mathcal{U}_i(T')_t < 0$, we have that $\mathcal{U}_i^{Y_1} < 0$. Besides, $\mathcal{U}_i^N > 0$ by (3.1). Hence we have that $\mathcal{U}_i^{Y_1} < \mathcal{U}_i^N$.

We now consider the situation where i is a provider. If i behaves, by the argument to derive (3.9), we have that:

$$\mathcal{U}_i^N = \mathcal{U}_i(d^+)_d + \mathcal{U}_i(p - f_0)_b + \mathcal{U}_i(T)_t + \mathcal{U}_i(r^+)_r. \tag{3.33}$$

If i requests arbitration and then behaves during the arbitration, then i will provide a satisfactory service s^+. It will be credited the amount $p - f_1$. The time spent on the transaction will be $2T$. Thus we have that:

$$\mathcal{U}_i^{Y_0} = \mathcal{U}_i(d^+)_d + \mathcal{U}_i(p - f_1)_b + \mathcal{U}_i(2T)_t + \mathcal{U}_i(r^+)_r. \tag{3.34}$$

From (3.33) and (3.34), we have that $\mathcal{U}_i^{Y_0} < \mathcal{U}_i^N$.

If i requests arbitration and then continues to misbehave during the arbitration, then i will provide a unsatisfactory service $s^{\times}$. It will be debited the amount $-f_2$. The time spent on the transaction will be less than T. Thus we have that:

$$\mathcal{U}_i^{Y_1} = \mathcal{U}_i(d^{\times})_d + \mathcal{U}_i(-f_2)_b + \mathcal{U}_i(T')_t + \mathcal{U}_i(r^{\times})_r, \text{ where } T' < T. \tag{3.35}$$

From (3.35), since $\mathcal{U}_i(d^{\times})_d \leq 0$, $\mathcal{U}_i(-f_2)_b < 0$, $\mathcal{U}_i(T')_t \leq 0$ and $\mathcal{U}_i(r^{\times})_r \leq 0$, we have that $\mathcal{U}_i^{Y_1} < 0$. Besides, by (3.2), we have that $\mathcal{U}_i^N > 0$. Hence $\mathcal{U}_i^{Y_1} < \mathcal{U}_i^N$. □

To summarise, it is not the best strategy for a participant to conduct repudiable misbehaviour or falsely accuse its partner of conducting repudiable misbehaviour.

3.7 Evaluation of the Scheme

In this section, we assess the marketplace scheme to justify the features claimed in Section 3.1.

3.7.1 Induction of Good Conduct

In Section 3.1, we claim that our marketplace features induction of good conduct, i.e. participants have an incentive to engage in good conduct. We show this feature as follows.

Theorems 2, 3, 7 and 8 show that conducting misbehaviour or false accusation are not the best strategies for any participant in our scheme. Further, the establishment of a good reputation is an additional incentive to induce good conduct from providers. Therefore, the best strategy for a participant is to engage in good conduct, i.e. complying with the operational procedures specified in Sections 3.4.4, 3.4.6 and 3.4.7.

3.7.2 Cost-effectiveness

In Section 3.1, we claim that our marketplace features cost-effectiveness, i.e. participants will find transaction partners who make the most cost-effective offer and the cost of the operation of the marketplace is within reasonable and acceptable limits.

With respect to the cost-effectiveness of transactions in our marketplace, the broker performs a pair-off service so that the most suitable consumer and provider are paired-off to conduct a transaction. With assistance of the broker, a participant is able to be paired-off with a transaction partner who makes the most cost-effective offer.

With respect to the cost of the operation of the marketplace, the following aspects need to be considered:

- Computational cost. Our marketplace requires that all messages exchanged between transaction partners should be non-repudiable, i.e. they should be accompanied by a cryptographic commitment such as a digital signature. The receiver should validate the received commitment (see Step 4 of the operation of the broker in Section 3.4.4). These are the main additional computational costs introduced by the marketplace. Since this scheme is designed for participants to exchange digital services, it is likely that the participants are devices that have reasonably abundant computational power, such as personal computers. Hence it is sensible to assume that the computational cost associated with generating and validating the cryptographic commitment can be easily borne by the participants.

- Communication cost. Our marketplace requires participants to interact with the server during the execution of a transaction and the resolution of a dispute (see Step 1 of the operation of the broker in Section 3.4.4, Step 1 of the operation of the arbitrator for non-repudiable misbehaviour in Section 3.4.6, and Steps 1 and 4 of the operation of the arbitrator for repudiable misbehaviour in Section 3.4.7). These are the main additional communication costs introduced by the marketplace. Since this scheme is designed for participants to exchange digital services, it is sensible to assume that each participant has reasonably abundant communication bandwidth for data transmission. Hence, the communication cost associated with interacting with the server can be easily borne.

- Time cost. In our marketplace, there is a time delay between a participant sending a request to the server and then getting paired-off with a suitable transaction partner (see Step 2 of the operation of the broker in Section 3.4.4). The length of this time delay mainly depends on the availability of a suitable transaction partner in the marketplace. If there is already a suitable transaction partner waiting, then the participant will be immediately paired-off with the suitable transaction partner. Otherwise, the participant may need to change its service requirements or wait until its suitable transaction partner appears in the marketplace. In our marketplace, the pair-off service is performed by the broker, a centralised entity who has information about all of the participants in the marketplace and their service requirements. This setting reduces the time delay to a minimum.

- Service fee cost. In our marketplace, participants have to pay a service fee, f_0 or f_1, to the server on a per transaction basis if a the transaction is successfully completed (see Step 5 of the operation of the broker in Section 3.4.4, Step 4 of the operation of the arbitrator for non-repudiable misbehaviour in Section 3.4.6, and Steps 5 and 6 of the operation of the arbitrator for repudiable misbehaviour in Section 3.4.7). The service fees f_0 and f_1 can be configured by the server at a reasonable and acceptable level, which can be driven by, for example, competition among multiple marketplaces.

Thus the various costs of the operation of the marketplace can be justifiably claimed to be reasonable and acceptable.

3.7.3 Transaction Security

In Section 3.1, we claim that our marketplace features transaction security, i.e. consumers will receive satisfactory service from providers upon payment, and providers will receive payment in full upon the provision of satisfactory service to consumers. The transaction security of our marketplace can be shown as follows:

- From the perspective of consumers, if they do not misbehave then there are only two possible outcomes from a transaction:
  - They receive satisfactory service and pay the amount $p + f_0$ (see Step 5 of the operation of the broker in Section 3.4.4) or $p + f_1$ (see Step 6 of the operation of the arbitrator for repudiable misbehaviour in Section 3.4.7). In this case, consumers receive satisfactory service upon payment.
  - They receive unsatisfactory service but with no financial loss (see Step 4 of the operation of the arbitrator for non-repudiable misbehaviour in Section 3.4.6 and Step 5 of the operation of the arbitrator for repudiable misbehaviour in Section 3.4.7). In this case, the consumer can complete the task by repeatedly requesting new transactions until it receives satisfactory service.
- From the perspective of providers, if they do not misbehave then they will receive the amount $p - f_0$ (see Step 5 of the operation of the broker in Section 3.4.4), $p - f_1$ (see Step 6 of the operation of the arbitrator for repudiable misbehaviour in Section 3.4.7) or p (see Step 5 of the operation of the arbitrator for repudiable misbehaviour in Section 3.4.7). In this case, providers will receive payment in full upon provision of satisfactory service.

Hence, transaction security is achieved for consumers and providers.

3.7.4 Reputation Robustness

In this section, we discuss the robustness of the reputation system incorporated into the marketplace. In Section 3.1, we claim that the reputation assets of providers maintained by the marketplace is protected against malicious damage. We will discuss the robustness of the reputation system according to our framework of feedback mechanisms of Chapter 2. In Section 2.5.6.3, we abstract some basic strategies that insider adversaries can adopt in order to attack a feedback mechanism, which includes our reputation system. We show how our proposed marketplace makes the reputation system immune to these attacks, as follows:

- *Sybil identity.* In our marketplace, this can be interpreted as that an adversary may act as multiple consumers with different identities. These different consumer identities can be used by the adversary to conduct a fabrication attack or a collusion attack. However, as we will show subsequently, adversaries are induced not to conduct fabrication attacks and collusion attacks. Hence a sybil identity attack will not have any impact on the robustness of the reputation system.

- *Fabrication.* In our marketplace, this can be interpreted as that an adversary acting as a consumer may falsely request arbitration when they do not find misbehaviour from their partners, or not request arbitration when they do discover misbehaviour from their partners. However, Lemma 1, Theorem 3, Lemma 6 and Theorem 8 have shown that conducting misbehaviour or false accusation are not the best strategies for any consumer. Hence, under the assumption that all participants are rational decision makers, as stated in Section 3.4.1, consumers are induced not to falsely request arbitration when they do not find misbehaviour from their transaction partners, and to request arbitration when they do discover misbehaviour from their transaction partners. Therefore adversaries are induced not to conduct fabrication attack.

- *Non-participation.* In our marketplace, this can be interpreted as that an adversary acting as a consumer does not report feedback for its transaction partner. However, in our marketplace, in each transaction feedback is reported by the broker or the arbitrator on behalf of the consumer (see Step 5 of the operation of the broker in Section 3.4.4, Step 4 of the operation of the arbitrator for non-repudiable misbehaviour in Section 3.4.6, and Steps 5 and 6 of the operation of the arbitrator for repudiable misbehaviour in Section 3.4.7). Hence, the reputation server will receive feedback regarding every transaction. A non-participation attack is thus inapplicable to our marketplace.

- *Collusion.* In our marketplace, this can be interpreted as that multiple adversaries acting as consumers may collude together to carry out an attack. We will discuss the robustness of the reputation system with respect to the following two types of collusion attack:
  - Multiple adversaries acting as consumers collude together to unfairly influence the reputation of a provider by each adversary targeting the provider and conducting a fabrication attack. Since adversaries are induced not to conduct fabrication attacks, they are also induced not to conduct this type of collusion attack.
  - A consumer and a provider collude together and conduct a non-existing transaction so that a positive feedback r^+ will be reported for the provider by the broker. This will unfairly increase the reputation of the provider. In our marketplace, each participant in a transaction has to pay a service fee f_0 when the transaction is successfully completed without arbitration (see Step 5 of the operation of the broker in Section 3.4.4). Hence, under the assumption that all participants are rational decision makers, as stated in Section 3.4.1, and the assumption that $\mathcal{U}_i(r^+)_r < \mathcal{U}_i(f_0)_b$, i.e. the utility resulting from a positive feedback is less than the utility of the service fee f_0, as stated in Section 3.6.1, adversaries are induced not to conduct this type of collusion attack.

  To summarise, in our marketplace, adversaries are induced not to conduct a collusion attack.

3.7.5 Penalty Balance

In Section 3.1, we claim that our marketplace features penalty balance, i.e. the financial gain of the *operator* of the electronic marketplace from performing dispute resolution remains the same regardless of its arbitration decision. This ensures that the operator does not have an incentive to make an unfair dispute resolution that favours any disputed party during arbitration. We will show this property as follows:

- With respect to arbitration for non-repudiable misbehaviour, the arbitrator gains the same amount of service fee f_1 regardless of its arbitration result (see Table 3.2). Therefore, the arbitrator has no incentive to make an unfair arbitration to favour any disputed participant with respect to non-repudiable misbehaviour.

- With respect to arbitration for repudiable misbehaviour, the arbitrator gains the same service fee f_2 regardless of its arbitration decision (see Table 3.3). Therefore, the arbitrator has no incentive to make an unfair arbitration to favour any disputed participant with respect to repudiable misbehaviour.

3.7.6 Relaxation of Participants' Rationality

In Section 3.4.1, we assume that participants are rational decision makers. However, this assumption can be relaxed to an extent without affecting the robustness of the mechanism. In more specific terms, our marketplace is robust as long as participants are rational about requesting arbitration, regardless of their rationality about conducting misbehaviour.

This is because, even if a participant is intentionally misbehaving (this is irrational behaviour as it reduces the utilities of the misbehaving participant), its transaction partner will request arbitration if its transaction partner is rational (see Lemmas 1, 5 and 6). In this situation, our marketplace still works. For example, our marketplace still works even if a provider intentionally provides a unsatisfactory service (irrational behaviour) as long as the consumer is rational about requesting arbitration (requesting arbitration when they are aware of their transaction partner's misbehaviour).

3.8 Fitting the Reputation System into our Framework

In our marketplace, we adopt a reputation system to evaluate the reputation of providers (see Section 3.4.3, Step 5 of the operation of the broker in Section 3.4.4, Step 4 of the operation of the arbitrator for non-repudiable misbehaviour in Section 3.4.6, and Steps 5 and 6 of the operation of the arbitrator for repudiable misbehaviour in Section 3.4.7).

In this section, we use our proposed model and framework of Chapter 2, to discuss the reputation system used in our marketplace. We will first identify the components of the reputation system according to the model described in Section 2.3. We then decompose the environmental assumptions according to Section 2.5 and the design choices of the reputation system according to Section 2.6. Lastly, we perform a brief analysis according to the decomposition of the reputation system.

3.8.1 Components

The components of the reputation system are as follows:

- *Entities*: *Feedback providers* are the consumers. The *relying entity* is the broker. *Targets* are the providers. *Processing unit* and *data storage unit* are the reputation server.
- *Attribute of interest*: the reputation of providers.
- *Interactions*:
  - *Previous interactions*: previous transactions between consumers and providers.
  - *Future interactions*: potential future transactions between consumers and providers.
- *Data*:
  - *Feedback*: successful completion of a transaction and a consumer' request for arbitration.
  - *Advice*: a provider's reputation.

3.8.2 Environmental Assumptions

The environmental assumptions of the reputation system are summarised as follows:

- *Target stability*: *unstable.* The reputation of a provider may change over time.

- *Capability of contributing entities*:
  - *Communication capability*:
    - *Data transmission capability*: *sufficient.* Each trader has sufficient capability to transmit data to and from the central server.
    - *Connectivity*: *fully-connected.* Each trader has a direct communication channel with the central server.
  - *Computational capability*: *sufficient.* We assume that the reputation server has sufficient computational capability for processing feedback.
  - *Data storage capability*: *sufficient.* We assume that the reputation server has sufficient capability for storing feedback.
  - *Feedback provision capability*: We assume that consumers are able to comply with the operation protocol of the marketplace.
- *Motivation of contributing entities*:
  - Consumers as feedback providers: *sufficient.* For every transaction, a feedback will be reported for the provider by the broker or the arbitrator on the behalf of the consumer.
  - The reputation server: *sufficient.* It is the business interest of the reputation server to provide data storage and feedback processing services.
- *Availability of contributing entities*:
  - Consumers as feedback providers: *intermittent.* Consumers are not available constantly.
  - The reputation server: *constant.* The reputation server is assumed to be constantly available.
- *Trust relationships*:
  - Traders: They are assumed to make rational decision. This trust fits in following categories:
    - *Global*: It is perceived marketplace-wide.
    - *Static*: It does not change over the life span of the marketplace.
    - *Individual*: It is borne by the individual trader.

  - The reputation server: We assume that the reputation server is a trusted entity behaving as a data storage unit and processing unit. This trust fits in the following categories:
    - *Global*: It is perceived marketplace-wide.
    - *Static*: It does not change over the life span of the marketplace.
    - *Individual*: It is borne by the individual entity of the reputation server.

- *Adversarial models.* Possible adversarial models are as follows:
  - *Rationality*: *rational.* We assume that all the participants are rational decision makers.
  - *Location*: *insiders.* Our robustness analysis mainly focuses on insider adversaries.
  - *Strategy space.* The possible adversarial strategy space includes:
    - *Sybil identity*: An adversary may act as multiple consumers with different identities.
    - *Fabrication*: An adversary acting as a consumer may falsely request arbitration when they do not find misbehaviour from their partners, or not request arbitration when they do discover misbehaviour from their partners.
    - *Non-participation.* In our marketplace, this can be interpreted as that an adversary acting as a consumer does not report feedback for its transaction partner.
    - *Collusion*: Multiple adversaries acting as consumers may collude together to carry out an attack.

3.8.3 Architectural Choices

The architectural choices of the reputation system are summarised as follows:

- *Role setting*: $\{PS,F,R,T\}$. The reputation server acts in the compound role of PS. The consumers act in the role of F. The broker act in the role of R and the providers act in the role of T.

- *Centrality*: *centralised.* The role of data storage unit and processing unit is performed by a single entity.
- *Data flow.* The data flow between consumers and the reputation server is shown as follows:
  - From a consumer as a feedback provider to the reputation server: *receiver-passive.* The reputation server passively receives feedback reported by the broker or the arbitrator on behalf of the consumer.
  - From the reputation server to the broker as a relying entity: *receiver-active* or *receiver-passive.* The reputation server may send the broker the reputation of a provider upon, or without, a request from the broker.

3.8.4 Data Processing Choices

The data processing choices of the reputation system are summarised as follows:

- *Representation of the evaluation data*: Representation of the evaluation data in feedback and advice is as follows:
  - In feedback: a *binary score.*
  - Advice: a *continuous numerical score.*
- *Aggregation algorithm*:
  - *Personalisation*: *non-personalised.* The reputation of a provider is the same for every consumer.
  - *Collaboration awareness*: *collaboration-unaware.* The aggregation algorithm is run by the single entity of the reputation server.
  - *Manipulation resistance*: *manipulation-vulnerable.* The aggregation algorithm is not immune to adversarial manipulation of feedback.

3.8.5 Robustness Solutions

The robustness solutions of the reputation system are summarised as follows:

- *Centrality.* A trusted and centralised processing unit and data storage is adopted. This avoids an adversary acting as a data storage unit and a processing unit.
- *Requiring proof of truthfulness of feedback.* Arbitration request from a consumer has to be verified by the arbitrator before a negative feedback is reported.
- *Incentive mechanism.* For every transaction, each trader has to bear a service fee. This discourages adversaries from boosting the reputation of a colluded provider by faking transactions.

3.8.6 Discussion

In our framework of feedback mechanisms of Chapter 2, we show that the robustness solution of requiring proof of truthfulness of feedback is not applicable in many application scenarios (see Section 2.6.3.3). The reputation system used in our marketplace, however, successfully adopts this robustness solution (see Section 3.8.5). The main reasons why the reputation system is able to adopt this robustness solution are as follows:

- We require that products or services exchanged in the marketplace are *arbitrable.* This condition can often be met for digital goods on which a non-repudiation of origin mechanism can be applied (see Section 3.1). This enables our marketplace to require that all messages exchanged between transaction partners should be non-repudiable, i.e. they should be accompanied by a cryptographic commitment such as a digital signature (see Step 4 of the operation of the broker in Section 3.4.4). As a result, non-repudiable misbehaviour can be identified by the arbitrator (see Section 3.4.6).
- We require that products or services exchanged in the marketplace are *replicable.* This condition can often applies to digital goods (see Section 3.1). This enables that the marketplace to require the transaction partners to repeat their transaction via the arbitrator in order to help the arbitrator to identify the misbehaving participant, once an arbitration for repudiable misbehaviour is requested (see Section 3.4.7).

3.9 Incorporating the Marketplace into Grid Computing

Grid computing enables heterogeneous machines to share computational resources across different organisations and various geographic locations. It has great potential to benefit many different applications [55]. However, a lack of incentive for people to contribute spare computational resources, low quality of service and “free-riding” problems have considerably limited the development of Grid computing [1, 43].

It is believed that introducing an economic-based and market-like mechanism into Grid computing significantly benefits participants [13, 25]. For example, Beck *et al.* [10] argue that a promising way to incentivise participants to share computational resources is the introduction of a pricing mechanism for the use of Grid-based resources. Mills and Dabrowski [84] show that an economic-based strategy benefits system welfare in different supply-demand relationships. Nimis *et al.* [92] claim that Grid technological solutions can be enhanced by setting the right incentives to reveal information about demand and supply accurately.

In previous sections, we discussed some preliminary ideas about how a viable marketplace trading online services could be created. In this section, we discuss the applicability of our marketplace to a Grid computing scenario. In addition, we show how to incorporate the marketplace into Grid computing for exchanging computational resources.

3.9.1 Applicability of our Marketplace in Grid Computing

In this section, we discuss the applicability our marketplace in Grid computing scenario from the following aspects:

- In Section 3.1, we require that products or services exchanged in the marketplace should be *arbitrable* and *replicable*. In Grid computing scenario, the main purpose is to share computational resources. As we discussed in Section 3.1, since results of computational tasks are often verifiable, the service of computational resources exchange is *arbitrable*. Besides, since computational tasks and results are easily redelivered, the service of computational resources exchange is *replicable*.

- Our marketplace requires that each participant has sufficient communication and computational capabilities (see Section 3.8.2). It is reasonable to assume that each machine in a Grid computing network has a communication channel with the central server and sufficient data transmission capability to and from the central server. Besides, it is reasonable to assume that each machine has sufficient computational capability to apply the cryptographic techniques stated in Section 3.4.1.
- Our marketplace requires that participants are rational decision makers (see Section 3.4.1). It is reasonable to assume that entities who demand and supply computational resources are rational decision makers with respect to computational resource exchange. The main goal of providers is to maximise their utilities, i.e. their financial revenue, by providing computational services. The main goal of consumers is to maximise their utilities, i.e. receiving computational services at minimal financial cost.

Hence, Grid computing seems a suitable application scenario for our marketplace.

3.9.2 Implementation Architecture

In this section, we indicate how to implement our marketplace in a Grid computing scenario.

We provide a layered architecture to integrate the proposed marketplace into Grid computing. This architecture is motivated by [91]. This architecture has three layers, as depicted in Figure 3.4.

The Application Layer comprises Grid applications and resources, which represent the applications that demand or supply computational resources, respectively. For consumers, Grid applications in this layer send tasks that require computational resources to, and receives the corresponding results from, the Middleware Layer. For providers, computational resources in this layer are consumed to perform tasks

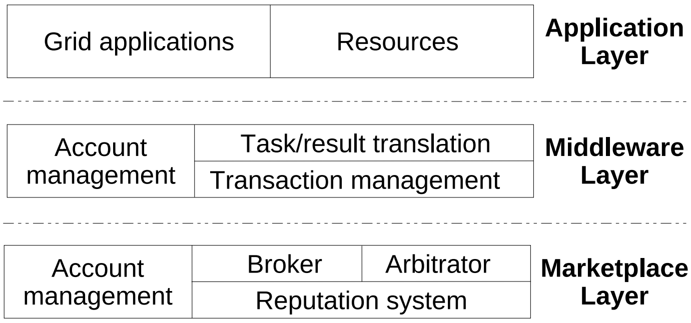

Figure 3.4: Layered architecture for integrating the marketplace into Grid computing

received from the Middleware Layer, to which the corresponding results are returned.

The Middleware Layer comprises functionalities of task/result translation, transaction management and account management. The component of task/result translation "translates" tasks and results from various formats that are used by the Application Layer into a standard and mutually understandable format, and vice versa. The transaction management component handles the generation of unambiguous and quantifiable requests for purchase or sale of computational resources on the behalf of consumers or providers, based on the amount of computational resources to be traded, their economic preferences and the current state of the marketplaces. This component also acts on the behalf of its user to interact with the marketplace and the same component of its user's transaction partner to carry out a transaction, as specified in Section 3.4. The component of account management facilitates a participant to manage its financial account maintained by the marketplace.

The Marketplace Layer is made up with the proposed marketplace. It provides the Middleware Layer with a platform for exchanging computational resources. It comprises the broker, arbitrator, reputation system and account management. The operation of the broker, arbitrator and reputation system is elaborated in Section 3.4. The account management component maintains all participants' financial accounts and provides them with access to their own financial accounts.

3.10 Conclusion and Future Work

In this chapter, we have proposed a cost-effective and secure marketplace which induces good conduct for exchanging digital products or services that are *arbitrable* and *replicable*. We have discussed the implementation the proposed marketplace into Grid computing.

Some issues arise from our proposed marketplace which are worth addressing in future work:

- Successful operation of an electronic marketplace based on our scheme relies on the components of the central server. Therefore, the security of the central server components becomes crucial. It would be interesting to explore how to reduce risks of attacks on these components, perhaps by distributing their functionalities.
- We have only provided an example of the reputation aggregation algorithm that can be applied in the proposed marketplace. Developing a suitable reputation system that can provide richer information about providers' reputation for adoption in this type of marketplace remains an interesting open problem.
- We only provide a conceptual architecture for integrating the proposed marketplace into Grid computing. Real world implementation of the proposed marketplace is an interesting task.

Chapter 4

An Alternative to Reputation Systems for Electronic Marketplaces

Contents

This chapter provides a novel feedback mechanism for electronic marketplaces. In this setting, the role of feedback is no longer a "shadow of the future", but a "shadow of the present". In other words, feedback directly impacts on the seller's payoff for the current transaction. By changing the fundamental functionality of feedback, many inherent problems of reputation systems are overcome.

4.1 Introduction

A mechanism that induces cooperation is essential for an electronic marketplace where buyers often do not have prior information about the trustworthiness of their transaction partners (sellers). A feedback-based mechanism like a reputation system is a common approach to induce sellers' good conduct in electronic marketplaces. A reputation system invites buyers to rate the "quality" of the behaviour of the sellers involved in transactions. This information is collected and aggregated as the reputation of sellers, and made available in the marketplace. Sellers with high reputation are more likely to be selected by potential future buyers. Reputation in this situation acts as a "shadow of the future" [106] to induce good conduct from sellers when they carry out transactions, even with unfamiliar buyers. This metaphor for reputation means that the expectation of reciprocity or retaliation in future interactions constrains sellers' behaviour in the present [106].

Although this setting induces good conduct of sellers, it has some inherent problems. For example:

- A reputation system is unable to induce good conduct for every single transaction. The *exit problem* [64], where sellers leaving the marketplace can misbehave without any fear, is an inherent problem of reputation systems. This is because the reputation of a departing seller has no future impact. In other words, a "shadow of the future" no longer exists for the departing seller.
- The *bootstrapping problem* [93], where new sellers have difficulty starting their businesses because they have no initial reputation in the marketplace, is a long-standing problem of reputation systems.
- Another problem is *reputation delay* [57], where there is a time delay in updating a seller's reputation. This is because there is a gap between the time when a transaction is conducted and when a feedback regarding this transaction is reported.
- Reputation systems are also vulnerable to some robustness threats [57, 52], such as *collusion attacks* (see Section 2.5.6.3) where colluding buyers and sellers conduct fake transactions and then report false positive feedbacks to inflate

the sellers' reputation.

These inherent problems are very challenging to address within the framework of reputation systems. This is mainly because feedback in reputation systems has impact only on future transactions. This raises the interesting question of whether the impact of feedback can be exerted on transactions other than those occurring in the future.

In this chapter, we propose a novel feedback mechanism for electronic marketplaces in which the role of feedback is no longer a "shadow of the future", but a "shadow of the present". In other words, feedback directly impacts on the seller's payoff for the current transaction. By changing the fundamental functionality of feedback, some of the previously mentioned problems of reputation systems are overcome.

Our approach is analogous to law enforcement to an extent: sellers have to be responsible for every single transaction that they participate. However, our mechanism relies solely on feedback provided by buyers, instead of factual evidence, to impose punishment, because it is unrealistic to obtain factual evidence about the quality of service for every transaction in an electronic marketplace.

The rest of this chapter is arranged as follows. We first discuss related work in Section 4.2. In Section 4.3, we outline the marketplace setting where our proposed feedback mechanism can be applied. We then present our feedback mechanism in Section 4.4 and discuss the features of the mechanism in Section 4.5. Subsequently, we refine our scheme in Section 4.6. In Section 4.7, we apply the framework proposed in Chapter 2 to decompose and discuss our proposed feedback mechanism. Lastly, we outline future work in Section 4.8.

4.2 Related Work

Dellarocas [33] proposed a similar "non-reputation-based" feedback mechanism for auction marketplaces. In this scheme, the operator of an auction marketplace first charges a listing fee from the seller before every transaction. At the end of a transaction, it will provide a reward to the seller if the buyer reports positive feedback, or impose a punishment if the buyer reports a negative feedback.

A drawback of this scheme is that it only facilitates a binary rating. It seems non-trivial to support a continuous rating. Additionally, this mechanism is limited only to auction marketplaces. This is because the operation of the scheme requires multiple buyers bidding for a product or service offered by one seller.

Further, this scheme is not robust against buyer-seller collusion attacks. This is because the reward is greater than the listing fee. A colluding seller can first pay a listing fee for a transaction. Its colluding buyer then reports positive feedback. This will enable the seller to receive the reward, which is greater than the listing fee.

Jurca and Faltings [61] proposed a sanctioning feedback mechanism. After a transaction, both seller and buyer are asked to rate the transaction. The seller and buyer will be sanctioned if their ratings contradict. A disadvantage of this mechanism is that it only allows binary rating, and it also seems challenging to facilitate continuous rating. Further, this feedback mechanism is designed only for an application environment where there are repeated transactions between a seller and a buyer. In addition, a buyer will be punished for reporting negative feedback.

Kerr and Cohen [64] proposed the idea of Trunits, reputation units that can be traded within a marketplace. A new seller can purchase some initial reputation value. Sellers are also allowed to sell their reputation. This approach solves the bootstrapping problem, exit problem and feedback delay problem. However, it suffers from the problem of devaluation of reputation units. More specifically, the number of reputation units available in the marketplace rises with the increase of the number of transactions conducted in the marketplace. This results in the devaluation of reputation units. Very frequent sellers earn reputation units at a faster rate than the devaluation rate of reputation units, while infrequent sellers earn reputation units at a slower rate than the devaluation rate. This puts more frequent sellers in a more advantageous position than less frequent sellers. Besides, this scheme suffers from collusion attacks, because a seller and a buyer may collude together and pretend to conduct transactions in order to earn reputation units.

Some side payment mechanisms, such as that of Jurca and Faltings [59] and Miller et al. [83], are proposed to induce honest feedback reporting. A buyer will receive some side payment if its rating for a seller is the same as the subsequent rating provided by another buyer for the same seller. This scheme aims to achieve that by employing a proper scoring rule to determine the amount of side payment, providing truthful feedback is the optimal strategy. But this scheme overlooks the collusion attacks where colluding buyers consecutively report feedback with the same rating for a seller in order to receive side payments as well as to influence the reputation of the seller. However, it may be possible to integrate some ideas from side payment mechanisms into our proposed scheme to induce honest feedback reporting from buyers, as we point out in Section 4.8 for future work.

4.3 Marketplace Setting

In this section, we describe the type of marketplace where our proposed feedback mechanism could be employed. We assume that:

- The operator of the marketplace is trusted. This means that the operator is trusted by all participants of the marketplace and acts in a trustworthy manner.
- The marketplace applies *controlled anonymity* [31] for every participant. This means that the operator conceals and periodically re-randomises the identities of sellers and buyers. In this way each participant is represented by a temporary anonymised identity, while only the operator knows the real identity. The operator also keeps track of all feedback reported by a buyer regarding a seller.

The straightforward steps involved in a transaction are as follows:

1. When a new seller S is admitted to the marketplace, the operator has to ascertain whether it is able to enforce a punishment Ψ on S for any transaction involving S. This assurance can take various forms, such as a financial deposit.

2. S places an *advertisement* (p, q_a, m) in the marketplace, where p denotes the advertised product or service, q_a denotes the necessary quality of service (QoS) to deliver p, and m denotes the monetary price. We assume that $\mathcal{U}_S(m) > \mathcal{U}_S(q_a)$ while the advertisement remains displayed, where $\mathcal{U}_S(m)$ denotes the utility to S of possessing m monetary units. $\mathcal{U}_S(q_a)$ denotes the cost to S of delivering p with QoS q_a.

3. A *potential transaction* is found if a buyer B finds an advertisement (p, q_a, m) such that $\mathcal{U}_B(q_a) > \mathcal{U}_B(m)$, where $\mathcal{U}_B(q_a)$ is the utility to B of receiving product p with QoS q_a, and $\mathcal{U}_B(m)$ is the cost to B of losing m monetary units.

4. If the buyer B wishes to proceed with the transaction, it first makes a payment of m monetary units to the seller S.

5. Seller S delivers product p to buyer B.

6. Buyer B assesses the QoS of the received product p and reports it, as feedback denoted by F, to the operator.

7. The operator decides whether a punishment Ψ needs to be imposed on S, according to the feedback F. If so, it decides upon the level of the punishment to be imposed and then enforces the punishment.

Our proposed feedback mechanism is applied to Steps 6 and 7. We will specify the range of feedback that can be accepted from buyers, and also formulate the level of punishment Ψ that should be imposed on sellers.

In this feedback mechanism, no reward or punishment is provided or imposed on the buyer. This is to avoid additional incentive to influence the feedback reported by buyers.

The main difference between this setting and a marketplace using a reputation system is as follows:

Feedback in this scheme is not used to establish the seller's reputation, as in reputation systems. Instead, it directly impacts on the seller's payoff for the current transaction. Hence the impact of feedback is only on the current transaction.

4.4 Our Proposed Feedback Mechanism

In this section, we describe our feedback mechanism. For the purpose of easy understanding, we first describe a binary mechanism. In this mechanism, we assume that the quality of service is represented by a binary level. We then show a more complex continuous mechanism. In this mechanism, we assume that the quality of service can be represented by a continuous scale.

4.4.1 Binary Mechanism

In this binary mechanism, we assume that quality of service Q is measured by only two levels: q_0 and q_1, respectively representing *unsatisfactory* service and *satisfactory* service. In this case, q_1 is equivalent to the advertised quality of service q_a which is mentioned in the steps of a transaction described in Section 4.3. We assume that $\mathcal{U}_S(q_0) = 0$, i.e. it costs nothing to S to deliver an unsatisfactory service, since if S decides to deliver an unsatisfactory service, it is better for S to deliver the service that costs as little as possible.

In this scenario, a transaction and the utilities of the buyer B and the seller S can be described in Figure 4.1. The transaction is conducted as follows:

1. Buyer B moves first and has two options: *in* and *out*, which denote, respectively, that B does or does not proceed with the transaction by paying m monetary units to S.

2. If the buyer chooses *out* then this transaction is terminated. In this case the utility of both equals zero. Otherwise, S has two options: $Q = q_1$ and $Q = q_0$, which, respectively, denote that S does or does not deliver a satisfactory service to B.

3. Buyer B then reports feedback $F = 1$ or $F = 0$, where $F = 1$ denotes $Q = q_1$ and $F = 0$ denotes $Q = q_0$. There are four cases:

    - S chooses $Q = q_0$ and B reports $F = 1$, i.e. S provides an unsatisfactory service, but B falsely reports that S provides a satisfactory service. In this case the utility of B equals $\mathcal{U}_B(q_0) - U_B(m)$, and the utility of S equals $\mathcal{U}_S(m) - \mathcal{U}_S(q_0)$.

- S chooses $Q = q_0$ and B reports $F = 0$, i.e. S provides an unsatisfactory service, and B honestly reports that S provides an unsatisfactory service. In this case the utility of B equals $\mathcal{U}_B(q_0) - U_B(m)$, but the utility of S becomes $\mathcal{U}_S(m) - \mathcal{U}_S(q_0) - \mathcal{U}_S(\psi)$.
- S chooses $Q = q_1$ and B reports $F = 1$, i.e. S provides a satisfactory service, and B honestly reports that S provides a satisfactory service. In this case the utility of B equals $\mathcal{U}_B(q_1) - U_B(m)$, and the utility of S equals $\mathcal{U}_S(m) - \mathcal{U}_S(q_1)$.
- S chooses $Q = q_1$ and B reports $F = 0$, i.e. S provides a satisfactory service, but B falsely reports that S provides an unsatisfactory service. In this case the utility of B equals $\mathcal{U}_B(q_1) - U_B(m)$, but the utility of S becomes $\mathcal{U}_S(m) - \mathcal{U}_S(q_1) - \mathcal{U}_S(\psi)$.

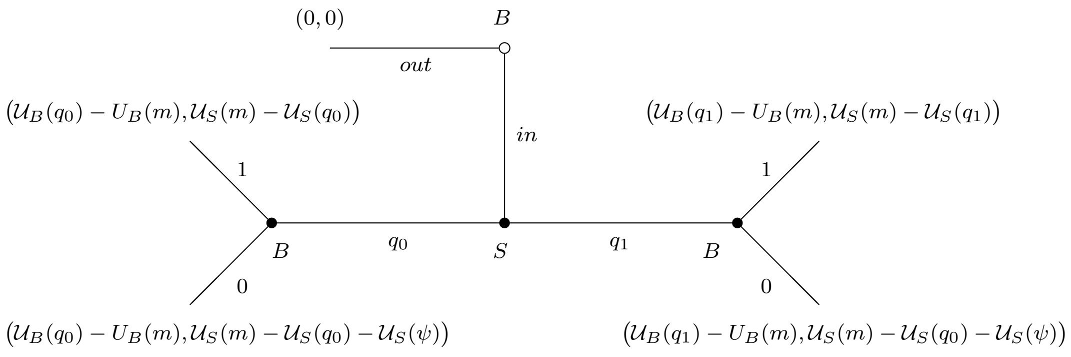
(0, 0)
B
out
in
(𝒰_B(q_0) − U_B(m), 𝒰_S(m) − 𝒰_S(q_0))
(𝒰_B(q_1) − U_B(m), 𝒰_S(m) − 𝒰_S(q_1))
1
1
B
q_0
S
q_1
B
0
0
(𝒰_B(q_0) − U_B(m), 𝒰_S(m) − 𝒰_S(q_0) − 𝒰_S(ψ))
(𝒰_B(q_1) − U_B(m), 𝒰_S(m) − 𝒰_S(q_0) − 𝒰_S(ψ))

Figure 4.1: A binary transaction

We suggest that buyer B and seller S follow Procedure 1.

```
Procedure 1:
Buyer $B$ chooses $in$;
Seller $S$ chooses $Q = q_1$;
$B$ reports $F = 1$ if $Q = q_1$; or $F = 0$ if $Q = q_0$;
Impose a punishment $\Psi = \psi(1 - F)$ on $S$, where $\psi$ is a pre-determined
constant.
```

We will show that if the punishment ψ is appropriately set, then Procedure 1 is the optimal strategy for both B and S.

We denote by $g(f|q)$ the probability that a feedback $F = f$ will be reported given the quality of the provided service $Q = q$ in the marketplace, i.e.:

$$g(f|q) = \Pr(F = f|Q = q).$$

The probability distribution $g(f|q)$ may vary among different marketplaces, since the feedback reporting behaviour may differ among different marketplaces. One extreme example is that if the vast majority of buyers in a marketplace are honest with respect to reporting feedback then $g(f|q)$ reflects, to a large extent, the overall quality of service exchanged in the marketplace. Another extreme example is that if the vast majority of buyers in a marketplace are dishonest with respect to reporting feedback, then $g(f|q)$ reveals little information about the overall quality of service exchanged in the marketplace.

The estimate of $g(f|q)$ can be obtained through, for example, a marketplace-level statistical analysis on buyer feedback reporting behaviour. In this chapter we do not provide a concrete method of estimating $g(f|q)$ as part of our current study.

We now show that if $g(f|q)$ satisfies some requirement, then the punishment ψ can be appropriately set such that Procedure 1 is the optimal strategy for both B and S.

Theorem 9. *If $g(1|q_1)+g(0|q_0)-1 > 0$ and ψ is set such that $\mathcal{U}_S(\psi) > \frac{\mathcal{U}_S(m)}{g(1|q_1)+g(0|q_0)-1}$, then B choosing in and S choosing $Q = q_1$ is the optimal strategy for B and S.*

Proof. We first consider the situation where the buyer B chooses *in*. In this case, we show that $Q = q_1$ is a better strategy than $Q = q_0$ for S.

We derive the expected utilities of strategies $Q = q_1$ and $Q = q_0$ for the seller S, denoted by $\mathcal{P}_S^{q_1}$ and $\mathcal{P}_S^{q_0}$ respectively, as follows:

$$\mathcal{P}_S^{q_1} = \mathcal{U}_S(m) - \mathcal{U}_S(q_1) - g(0|q_1)\mathcal{U}_S(\psi); \tag{4.1}$$

$$\mathcal{P}_S^{q_0} = \mathcal{U}_S(m) - \mathcal{U}_S(q_0) - g(0|q_0)\mathcal{U}_S(\psi). \tag{4.2}$$

From (4.1) and (4.2) we derive that:

$$\begin{aligned}\mathcal{P}_S^{q_1} - \mathcal{P}_S^{q_0} &= \mathcal{U}_S(q_0) - \mathcal{U}_S(q_1) + (g(0|q_0) - g(0|q_1))\mathcal{U}_S(\psi)\\ &= \mathcal{U}_S(q_0) - \mathcal{U}_S(q_1) + (g(0|q_0) + g(1|q_1) - 1)\mathcal{U}_S(\psi). \qquad (4.3)\end{aligned}$$

Recall that $\mathcal{U}_S(q_0) = 0$ and $\mathcal{U}_S(\psi) > \frac{\mathcal{U}_S(m)}{g(1|q_1)+g(0|q_0)-1}$. Then from (4.3) we derive that:

$$\begin{aligned}\mathcal{P}_S^{q_1} - \mathcal{P}_S^{q_0} > & \frac{(g(0|q_0) + g(1|q_1) - 1)\mathcal{U}_S(m)}{g(1|q_1) + g(0|q_0) - 1} - \mathcal{U}_S(q_1) \\ = & \mathcal{U}_S(m) - \mathcal{U}_S(q_1) \\ > & 0.\end{aligned}$$

Hence given that the buyer B chooses *in*, $Q = q_1$ is the best strategy for S.

We now show that B choosing *in* is the best strategy. Since, if B chooses *in* then $Q = q_1$ is the best strategy for S. Hence the utility of B in this case is $\mathcal{U}_B(q_1) - U_B(m)$. Recall that $\mathcal{U}_B(q_1) > U_B(m)$. On the other hand, if B chooses *out* then its utility equals 0. Hence, B choosing *in* is the best strategy. □

We now show the condition for seller S to be profitable in the marketplace.

Theorem 10. *If ψ is further set such that $g(0|q_1)\mathcal{U}_S(\psi) < \mathcal{U}_S(m) - \mathcal{U}_S(q_1)$, then it is profitable for S to join the marketplace.*

Proof. From Theorem 9 we conclude that B choosing *in* and S choosing $Q = q_1$ is the optimal strategy if the punishment ψ is set as required in the theorem. Hence the expected profit for S to conduct a transaction by following its optimal strategy in the marketplace is as follows:

$$\mathcal{P}_S^{q_1} = \mathcal{U}_S(m) - \mathcal{U}_S(q_1) - g(0|q_1)\mathcal{U}_S(\psi).$$

It is easy to check that if ψ is further set such that $g(0|q_1)\mathcal{U}_S(\psi) < \mathcal{U}_S(m) - \mathcal{U}_S(q_1)$, then $\mathcal{P}_S^{q_1} > 0$, i.e. it is profitable for S to join the marketplace. □

Finally, we show that the requirements on the punishment ψ from Theorems 9 and 10 are not contradictory.

Theorem 11. *If the marketplace satisfies that $\frac{g(0|q_0)}{g(0|q_1)} > \frac{\mathcal{U}_S(m)}{\mathcal{U}_S(m)-\mathcal{U}_S(q_1)} + 1$, then ψ can be set such that the conditions in Theorems 9 and 10 can both be satisfied.*

Proof. From the condition stated in this theorem, we derive that:

$$\begin{aligned} \frac{g(0|q_0)}{g(0|q_1)} &> \frac{\mathcal{U}_S(m)}{\mathcal{U}_S(m)-\mathcal{U}_S(q_1)}+1 \\ \Longrightarrow \frac{\mathcal{U}_S(m)}{\mathcal{U}_S(m)-\mathcal{U}_S(q_1)} &< \frac{g(0|q_0)-g(0|q_1)}{g(0|q_1)} \\ \Longrightarrow \frac{\mathcal{U}_S(m)}{g(1|q_1)+g(0|q_0)-1} &< \frac{\mathcal{U}_S(m)-\mathcal{U}_S(q_1)}{g(0|q_1)}. \end{aligned} \tag{4.4}$$

From Theorem 10 we learn that the requirement on ψ is that:

$$\begin{aligned} g(0|q_1)\mathcal{U}_S(\psi) &< \mathcal{U}_S(m)-\mathcal{U}_S(q_1) \\ \Longrightarrow \mathcal{U}_S(\psi) &< \frac{\mathcal{U}_S(m)-\mathcal{U}_S(q_1)}{g(0|q_1)}. \end{aligned} \tag{4.5}$$

From (4.4) and (4.5) we derive that ψ can be set such that $\mathcal{U}_S(\psi)$ is between $\frac{\mathcal{U}_S(m)}{g(1|q_1)+g(0|q_0)-1}$ and $\frac{\mathcal{U}_S(m)-\mathcal{U}_S(q_1)}{g(0|q_1)}$, which satisfies both the conditions in Theorems 9 and 10. □

From Theorems 9, 10 and 11 we learn that if a marketplace meets the conditions that $g(1|q_1)+g(0|q_0)-1>0$ and $\frac{g(0|q_0)}{g(0|q_1)} > \frac{\mathcal{U}_S(m)}{\mathcal{U}_S(m)-\mathcal{U}_S(q_1)}+1$, then ψ can be set as required by Theorems 9 and 10. Following Procedure 1 is thus optimal for both buyer and seller and profitable for the seller.

In Figure 4.2, we visually show the relationship between the buyer honesty level $g(1|q_1)$ and the punishment $\mathcal{U}_S(\psi)$. For simplicity, we assume that $g(0|q_0)=1$, i.e. given that a buyer has received a dissatisfactory transaction, it will report negative feedback. The figure shows the relationship as follows:

- The buyer honesty level $g(1|q_1)$ must be greater than $\frac{\mathcal{U}_S(m)}{2\mathcal{U}_S(m)-\mathcal{U}_S(q_1)}$. This requirement can be derived from the condition stated in Theorem 11.
- The punishment ψ must have a larger utility than the monetary income m, i.e. $\mathcal{U}_S(\psi) > \mathcal{U}_S(m)$. This requirement can be derived from the condition stated in Theorem 9.
- The condition that this mechanism works is that the coordinate $(g(1|q_1),\mathcal{U}_S(\psi))$ is within the shaded area marked in Figure 4.2. This requirement can be derived from the conditions stated in Theorems 9 and 10.

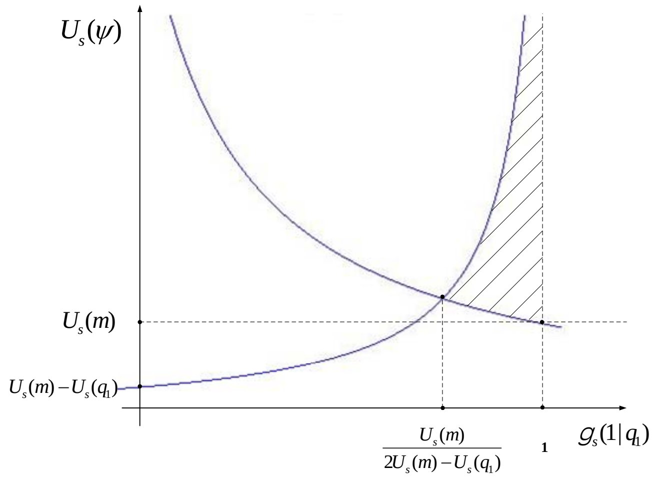

Figure 4.2: The relationship between the buyer honesty and the punishment.

We will discuss the applicability of this feedback mechanism in Section 4.7.6.

4.4.2 Continuous Mechanism

We now describe a continuous feedback mechanism, which takes partial satisfaction into consideration. In this mechanism, the seller S can choose quality of service $Q = \delta q_a$, where $0 \leq \delta \leq 1$. The notation δ denotes the extent to which the delivered service is in line with the advertised quality of service q_a. We assume that $\mathcal{U}_S(\delta q_a) = \delta \mathcal{U}_S(q_a)$, i.e. the cost to deliver a partially satisfactory service δq_a is proportional to the extent of satisfaction δ. After receiving the service, the buyer B reports a feedback $F = f$, where $0 \leq f \leq 1$, indicating the extent to which B believes that the received service or product is in line with the advertised quality q_a.

We suggest that the seller S and buyer B follow Procedure 2. We will also show that if ψ is appropriately set, then our suggested procedure is the optimal strategy for both B and S.

Procedure 2:

1 Buyer B chooses in;
2 Seller S chooses $Q = \delta q_a$ where $\delta = 1$;
3 B reports $F = f$ where f is δ for the actual received service;
4 Impose a punishment $\Psi = \psi(1-f)$ on S, where ψ is a pre-determined constant.

Let $G(f|q) = \Pr(F \leq f|Q = q)$ denote the cumulative distribution function of feedback F given $Q = q$. The cumulative distribution function $G(f|q)$ may vary among different marketplaces. The estimate of $G(f|q)$ can be obtained through, for example, a marketplace-level statistical analysis on buyer feedback reporting behaviour. In this chapter we do not provide a concrete method of estimating $G(f|q)$ as part of our current study.

We now show that if $G(f|q)$ satisfies some requirement then the punishment ψ can be appropriately set such that Procedure 2 is the optimal strategy for both B and S. We then have the following theorems.

Theorem 12. *Let $\mathbb{G}(q) = \int_0^1 G(f|q)\mathrm{d}f$. If $\mathbb{G}(\delta q_a) - \mathbb{G}(q_a) > 0\ \forall \delta \neq 1$ and ψ is set such that $\mathcal{U}_S(\psi) > \mathcal{U}_S(m)\frac{(1-\delta)}{\mathbb{G}(\delta q_a)-\mathbb{G}(q_a)}$, then B choosing in and S choosing $Q = q_a$ is the optimal strategy for B and S.*

Proof. We prove this theorem in a similar way to the proof of Theorem 9. Given that B chooses *in* and S chooses $Q = q_a$, the expected punishment Ψ is as follows:

$$\Psi = \int_0^1 g(f|q_a)(1-f)\psi \mathrm{d}f.$$

We then derive the utility of S, denoted by $\mathcal{P}_S^{q_a}$, as follows:

$$\begin{aligned}
\mathcal{P}_S^{q_a} &= \mathcal{U}_S(m) - \mathcal{U}_S(q_a) - \mathcal{U}_S\bigg(\int_0^1 g(f|q_a)(1-f)\psi \mathrm{d}f\bigg) \\
&= \mathcal{U}_S(m) - \mathcal{U}_S(q_a) - \mathcal{U}_S\bigg(\int_0^1 \psi \cdot g(f|q_a)\mathrm{d}f - \int_0^1 \psi \cdot f \cdot g(f|q_a)\mathrm{d}f\bigg) \\
&= \mathcal{U}_S(m) - \mathcal{U}_S(q_a) - \mathcal{U}_S\bigg(\psi - \psi\int_0^1 f \cdot g(f|q_a)\mathrm{d}f\bigg) \qquad (4.6) \\
&= \mathcal{U}_S(m) - \mathcal{U}_S(q_a) - \mathcal{U}_S\bigg(\psi\int_0^1 G(f|q_a)\mathrm{d}f\bigg) \qquad (4.7) \\
&= \mathcal{U}_S(m) - \mathcal{U}_S(q_a) - \mathcal{U}_S\bigg(\psi\mathbb{G}(q_a)\bigg) \\
&= \mathcal{U}_S(m) - \mathcal{U}_S(q_a) - \mathbb{G}(q_a)\mathcal{U}_S(\psi). \qquad (4.8)
\end{aligned}$$

Note that from (4.6) to (4.7), we use integration by parts.

Given that B chooses *in* and S chooses $Q = \delta q_a$, where $\delta \neq 1$, in a similar way we derive the utility of S, denoted by $\mathcal{P}_S^{\delta q_a}$, as follows:

$$\begin{aligned}
\mathcal{P}_S^{\delta q_a} &= \mathcal{U}_S(m) - \mathcal{U}_S(\delta q_a) - \mathcal{U}_S\bigg(\int_0^1 g(f|\delta q_a)(1-f)\psi \mathrm{d}f\bigg) \\
&= \mathcal{U}_S(m) - \mathcal{U}_S(\delta q_a) - \mathcal{U}_S\bigg(\int_0^1 \psi \cdot g(f|\delta q_a)\mathrm{d}f - \int_0^1 \psi \cdot f \cdot g(f|\delta q_a)\mathrm{d}f\bigg) \\
&= \mathcal{U}_S(m) - \mathcal{U}_S(\delta q_a) - \mathcal{U}_S\bigg(\psi - \psi\int_0^1 f \cdot g(f|\delta q_a)\mathrm{d}f\bigg) \\
&= \mathcal{U}_S(m) - \mathcal{U}_S(\delta q_a) - \mathcal{U}_S\bigg(\psi\int_0^1 G(f|\delta q_a)\mathrm{d}f\bigg) \\
&= \mathcal{U}_S(m) - \mathcal{U}_S(\delta q_a) - \mathcal{U}_S\bigg(\psi\mathbb{G}(\delta q_a)\bigg) \\
&= \mathcal{U}_S(m) - \mathcal{U}_S(\delta q_a) - \mathbb{G}(\delta q_a)\mathcal{U}_S(\psi). \qquad (4.9)
\end{aligned}$$

From (4.8) and (4.9) we derive that:

$$\begin{aligned}
\mathcal{P}_S^{q_a} - \mathcal{P}_S^{\delta q_a} &= \mathcal{U}_S(\delta q_a) - \mathcal{U}_S(q_a) + \mathbb{G}(\delta q_a)\mathcal{U}_S(\psi) - \mathbb{G}(q_a)\mathcal{U}_S(\psi) \\
&= \mathcal{U}_S(\psi)\bigg(\mathbb{G}(\delta q_a) - \mathbb{G}(q_a)\bigg) - (1-\delta)\mathcal{U}_S(q_a) \\
&> \mathcal{U}_S(m)(1-\delta) - (1-\delta)\mathcal{U}_S(q_a) \qquad (4.10) \\
&= (1-\delta)\bigg(\mathcal{U}_S(m) - \mathcal{U}_S(q_a)\bigg) \\
&> 0. \qquad (4.11)
\end{aligned}$$

Note that (4.10) follows from the stated assumption in the theorem that:

$$\mathcal{U}_S(\psi) > \mathcal{U}_S(m)\frac{(1-\delta)}{\mathbb{G}(\delta q_a) - \mathbb{G}(q_a)}.$$

Note also that (4.11) follows since $1 - \delta > 0$, and $\mathcal{U}_S(m) - \mathcal{U}_S(q_a) > 0$, as stated in Section 4.3.

Hence, given that B chooses *in*, $Q = q_a$ is the best strategy for S.

We now show that B choosing *in* is the best strategy. If B chooses *in*, then $Q = q_a$ is the best strategy for S. Hence the utility of B in this case is $\mathcal{U}_B(q_a) - U_B(m)$. Recall that $\mathcal{U}_B(q_a) > U_B(m)$. If B chooses *out* then its utility equals 0. Hence, B choosing *in* is the best strategy. □

We now show the condition for seller S to be profitable in the marketplace.

Theorem 13. *If ψ is further set such that $\mathbb{G}(q_a)\mathcal{U}_S(\psi) < \mathcal{U}_S(m) - \mathcal{U}_S(q_a)$, then it is profitable for S to join the marketplace.*

Proof. We prove this theorem in a similar way to the proof of Theorem 10. From Theorem 12 we conclude that B choosing *in* and S choosing $Q = q_a$ is the optimal strategy if the punishment ψ is set as required in the theorem. From (4.8), the expected profit for S to conduct a transaction by following the optimal strategy is as follows:

$$\mathcal{P}_S^{q_a} = \mathcal{U}_S(m) - \mathcal{U}_S(q_a) - \mathbb{G}(q_a)\mathcal{U}_S(\psi).$$

It follows immediately that if $\mathbb{G}(q_a)\mathcal{U}_S(\psi) < \mathcal{U}_S(m) - \mathcal{U}_S(q_a)$, then $\mathcal{P}_S^{q_a} > 0$, i.e. it is profitable for S to join the marketplace. □

We now show that the requirements on the punishment ψ from Theorems 12 and 13 are not contradictory.

Theorem 14. *If the marketplace satisfies that $\frac{\mathbb{G}(\delta q_a)}{\mathbb{G}(q_a)} > \frac{\mathcal{U}_S(m)(1-\delta)}{\mathcal{U}_S(m)-\mathcal{U}_S(q_a)} + 1$, $\forall \delta \neq 1$, then ψ can be set such that the conditions in Theorem 12 and 13 can be both satisfied.*

Proof. Similarly to the proof of Theorem 11, we derive that:

$$\begin{aligned}
\frac{\mathbb{G}(\delta q_a)}{\mathbb{G}(q_a)} &> \frac{\mathcal{U}_S(m)(1-\delta)}{\mathcal{U}_S(m) - \mathcal{U}_S(q_a)} + 1 \\
\Longrightarrow \frac{\mathbb{G}(\delta q_a)}{\mathbb{G}(q_a)} - 1 &> \frac{\mathcal{U}_S(m)(1-\delta)}{\mathcal{U}_S(m) - \mathcal{U}_S(q_a)} \\
\Longrightarrow \frac{\mathbb{G}(\delta q_a) - \mathbb{G}(q_a)}{\mathbb{G}(q_a)} &> \frac{\mathcal{U}_S(m)(1-\delta)}{\mathcal{U}_S(m) - \mathcal{U}_S(q_a)} \\
\Longrightarrow \frac{\mathcal{U}_S(m)(1-\delta)}{\mathbb{G}(\delta q_a) - \mathbb{G}(q_a)} &< \frac{\mathcal{U}_S(m) - \mathcal{U}_S(q_a)}{\mathbb{G}(q_a)}.
\end{aligned} \tag{4.12}$$

From Theorem 13 we learn the requirement on δ is that:

$$\begin{aligned}
\mathbb{G}(q_a)\mathcal{U}_S(\psi) &< \mathcal{U}_S(m) - \mathcal{U}_S(q_a) \\
\Longrightarrow \mathcal{U}_S(\psi) &< \frac{\mathcal{U}_S(m) - \mathcal{U}_S(q_a)}{\mathbb{G}(q_a)}.
\end{aligned} \tag{4.13}$$

From (4.12) and (4.13) we derive that ψ can be set such that $\mathcal{U}_S(\psi)$ is between $\frac{\mathcal{U}_S(m)(1-\delta)}{\mathbb{G}(\delta q_a)-\mathbb{G}(q_a)}$ and $\frac{\mathcal{U}_S(m)-\mathcal{U}_S(q_a)}{\mathbb{G}(q_a)}$, which satisfies both the conditions in Theorems 12 and 13. □

From Theorems 12, 13 and 14 we learn that if a marketplace meets the conditions that $\mathbb{G}(\delta q_a) - \mathbb{G}(q_a) > 0$ and $\frac{\mathbb{G}(\delta q_a)}{\mathbb{G}(q_a)} > \frac{\mathcal{U}_S(m)(1-\delta)}{\mathcal{U}_S(m)-\mathcal{U}_S(q_a)} + 1$, $\forall \delta \neq 1$, then ψ can be set as required by Theorems 12 and 13. Following Procedure 2 is thus the optimal strategy for both buyer and seller and profitable for the seller.

In summary, we have proven that:

- It is profitable for sellers to join the marketplace that applies the proposed binary or continuous feedback mechanism.
- Buyers are induced to proceed with potential transactions.
- Sellers are induced to deliver satisfactory services.

4.5 Discussion

In this section, we discuss the advantages and disadvantages of our proposed mechanism. In our proposed mechanism, because feedback directly impacts on the seller's payoff for the current transaction, it has several advantages over traditional reputation systems for marketplaces:

1. It induces good conduct for every potential transaction. This is reflected as follows:
    - In our scheme, whenever there is a potential transaction, the buyer is induced to proceed with the transaction, as we prove in Theorems 9 and 12. On the other hand, in a marketplace supported by a reputation system, a buyer may proceed with the transaction only if the reputation of the seller is sufficiently good (this is actually the main principle of using a reputation system in a marketplace).
    - In our scheme, whenever there is a potential transaction, the seller is induced to deliver the product or service as advertised, as proven in Theorems 9 and 12. Sellers remain induced to provide good service regardless of any possible plans to leave the marketplace. However, in a marketplace

supported by a reputation system, a departing seller can deliver products or services with a poor quality without any fear, due to the exit problem discussed in Section 4.1.

2. It does not require new sellers to establish reputation over time. Instead, they have the same chance as existing sellers of being selected as business partners by buyers. But in a marketplace supported by a reputation system, new sellers are disadvantaged compared with existing sellers, due to the bootstrapping problem discussed in Section 4.1.

3. In our scheme, the delay between the time when a transaction is carried out and when the involved buyer reports a feedback has no impact on its induction of sellers' good conduct. But in a marketplace supported by a reputation system, a longer delay implies that the reputation is less fresh, because it represents the seller's behaviour of a longer time ago, as we discussed in Section 4.1.

4. Our scheme is immune to a buyer-seller collusion attack, where some buyers and sellers collude together to report false feedback in order to gain unfair benefit from other buyers or sellers. However, in our scheme, feedback from one transaction only impacts on this transaction, but not any other transaction. Colluding buyers and sellers thus cannot gain benefit from other transactions. However, in a marketplace supported by a reputation system, buyer-seller collusion is a robustness threat, as we discussed in Section 4.1.

On the other hand, our scheme has some disadvantages, as follows:

1. It is not practical to implement in certain application scenarios. First, there is a minimum requirement on the overall buyer honesty level with respect to feedback reporting behaviour. For example, the binary mechanism requires that $g(1|q_1) + g(0|q_0) - 1 > 0$ and $\frac{g(0|q_0)}{g(0|q_1)} > \frac{\mathcal{U}_S(m)}{\mathcal{U}_S(m) - \mathcal{U}_S(q_1)} + 1$, as shown in Section 4.4.1. The continuous mechanism requires that $\mathbb{G}(\delta q_a) - \mathbb{G}(q_a) > 0$ and $\frac{\mathbb{G}(\delta q_a)}{\mathbb{G}(q_a)} > \frac{\mathcal{U}_S(m)(1-\delta)}{\mathcal{U}_S(m) - \mathcal{U}_S(q_a)} + 1$, $\forall \delta \neq 1$, as shown in Section 4.4.2. In addition, our proposed marketplace requires controlled anonymity, as shown in Section 4.3. If a marketplace does not satisfy these requirements, then our scheme is not applicable to it.

2. Buyers' false negative feedback results in sellers being unfairly punished. As a result, sellers are in a more vulnerable position than buyers, contrary to reputation systems, in which buyers are in a more vulnerable position than sellers. In the next section, we will address this issue by proposing a refinement of our scheme.

4.6 Further Refinement

In this section, we present a refinement of both the binary and continuous mechanisms to reduce some unfair punishment imposed on sellers that results from false negative feedback. Meanwhile, we show that after the refinement, the optimal strategy for the seller S remains $Q = q_a$.

4.6.1 Refining the Binary Mechanism

Let $\mathcal{T}$ be a statistical tool analysing all reported feedback in the marketplace to detect false negative feedback. In other words, $\mathcal{T}$ predicts the actual quality q of a service given that a feedback $F = 0$ is reported. More formally, $\mathcal{T}$ outputs the probability $h(q|0)$, where $q \in \{0, 1\}$. Let α denote the overall ratio of the number of satisfactory services to the total number of services delivered in the marketplace. The estimate of $h(q|0)$ can be obtained through, for example, a marketplace-level statistical analysis on buyer feedback reporting behaviour. In this chapter we do not discuss how to realise the statistical tool $\mathcal{T}$ or how to obtain the overall ratio α, but assume their existence.

In this refinement, we propose to replace Step 4 of Procedure 1 with Procedure 3.

In this refinement, if the probability that a negative feedback is false is greater than α, then the seller will not be punished. It hence reduces the seller's cost caused by some false negative feedback provided by dishonest buyers.

We now show that the seller's best strategy after the refinement remains $Q = q_1$. This will ensure that our feedback mechanism still works after adoption of the re-

```
Procedure 3:
if $F = 0$ then
    Operator computes $h(q|0)$ using $\mathcal{T}$ on all feedback available in the marketplace;
    if $h(q_1|0) \leq \alpha$ then
        $F' = 0$;
    else
        $F' = 1$;
    Impose a punishment $\Psi = \psi \cdot (1 - F')$ on $S$.
```

finement.

Theorem 15. *After replacing Step 4 of Procedure 1 with Procedure 3, $Q = q_1$ still remains the optimal strategy of S.*

Proof. Let N denote the total number of transactions conducted in the marketplace. Let N^{q_1} and N^{q_0} denote, respectively, the number of transactions with $Q = q_1$, and $Q = q_0$. Hence, $\alpha = \frac{N^{q_1}}{N}$.

Suppose that the punishment ψ is set such that $\mathcal{U}_S(\psi) > \frac{\mathcal{U}_S(m)}{g(1|q_1)+g(0|q_0)-1}$, where $g(1|q_1) + g(0|q_0) - 1 > 0$, as required by Theorem 9. We prove in Theorem 9 that choosing $Q = q_1$ is the seller S's optimal strategy.

The refinement changes both $g(1|q_1)$ and $g(0|q_0)$. Let $g'(1|q_1)$ and $g'(0|q_0)$ respectively denote their corresponding values after the refinement blocks the feedback $F = 0$ in one transaction. We derive that $g'(1|q_1) = g(1|q_1) + h(q_1|0)\frac{1}{N^{q_1}}$, while $g'(0|q_0) = g(0|q_0) - h(q_0|0)\frac{1}{N^{q_0}}$. Hence, we obtain that:

$$\begin{aligned} g'(1|q_1) + g'(0|q_0) - g(1|q_1) - g(0|q_0) &= h(q_1|0)\frac{1}{N^{q_1}} - h(q_0|0)\frac{1}{N^{q_0}} \\ &= \left(N^{q_0}h(q_1|0) - N^{q_1}h(q_0|0)\right)\frac{1}{N^{q_1}N^{q_0}} \\ &= \left((N^{q_1} + N^{q_0})h(q_1|0) - N^{q_1}\right)\frac{1}{N^{q_1}N^{q_0}} \\ &= \left(Nh(q_1|0) - N^{q_1}\right)\frac{1}{N^{q_1}N^{q_0}} \\ &> 0, \end{aligned}$$

since $h(q_1|0) > \alpha$, i.e. $h(q_1|0) > \frac{N^{q_1}}{N}$, and thus $Nh(q_1|0) > N^{q_1}$.

Therefore, after the refinement, $\mathcal{U}_S(\psi) > \frac{\mathcal{U}_S(m)}{g'(1|q_1)+g'(0|q_0)-1}$, where $g'(1|q_1)+g'(0|q_0)-1 > 0$ still holds. Hence $Q = q_1$ is still the optimal strategy for S. □

4.6.2 Refining the Continuous Mechanism

Let $\mathcal{T}$ be a statistical tool used by the operator to analyse the probability of the actual quality δq_a of a service given that a feedback $F = f$ is reported. In other words, $\mathcal{T}$ outputs the probability density function $h(\delta q_a|f)$. Let $H(\delta q_a|f)$ denote the cumulative distribution function of $h(\delta q|f)$, i.e. $H(\delta q_a|f) = \int_0^\delta h(xq_a|f)\mathrm{d}x$.

In this refinement, we propose to replace step 4 of Procedure 2 with Procedure 4.

```
Procedure 4:
if  $f < 1$ then
    Operator computes $h(\delta q_a|f)$, $H(\delta q_a|f)$ using $\mathcal{T}$ on all feedback
    available in the marketplace;
    Imposes a punishment $\Psi = \psi \cdot (1 - f')$ on $S$, where
    $f' = \max\left(f, \frac{\mathcal{U}_S(q_a)}{\mathcal{U}_S(\psi)}\left(1 - \int_0^1 H(\delta q_a|f)\mathrm{d}\delta\right)\right)$.
```

Upon adoption of this refinement, if $f < \frac{\mathcal{U}_S(q_a)}{\mathcal{U}_S(\psi)}\left(1 - \int_0^1 H(\delta q_a|f)\mathrm{d}\delta\right)$, then $f' > f$. This means that the actual punishment $\Psi = \psi(1 - f')$ imposed on S is less than $\psi(1 - f)$, the punishment when the refinement is not adopted.

We now show that the seller's best strategy after the refinement remains $Q = q_q$. This will ensure that that our feedback mechanism still works after adoption of the refinement.

Theorem 16. *After replacing Step 4 of Procedure 2 with Procedure 4, $Q = q_a$ remains the optimal strategy of S.*

Proof. Theorem 13 shows that the utility of S for choosing the optimal strategy $Q = q_a$ is greater than zero, i.e. $\mathcal{P}_S^{q_a} > 0$.

From Procedure 4 we know that the punishment Ψ will be modified by the refinement only if $f < \frac{\mathcal{U}_S(q_a)}{\mathcal{U}_S(\psi)}\big(1 - \int_0^1 H(\delta q_a|f)\mathrm{d}\delta\big)$. When $f \geq \frac{\mathcal{U}_S(q_a)}{\mathcal{U}_S(\psi)}\big(1 - \int_0^1 H(\delta q_a|f)\mathrm{d}\delta\big)$, the punishment Ψ is the same as that from using Procedure 4 alone without the refinement. Hence, in this case, the refinement does not affect the utility of the seller. Thus the optimal strategy of $Q = q_a$ remains unchanged.

We now show that if $f < \frac{\mathcal{U}_S(q_a)}{\mathcal{U}_S(\psi)}\big(1 - \int_0^1 H(\delta q_a|f)\mathrm{d}\delta\big)$, the expected utility of S from this transaction after the feedback adjustment in Procedure 4 is still less than zero. We learn that given the feedback $F = f$ is reported, the expected cost to deliver the service for S is $\mathcal{U}_S(q_a)\bigg(\int_0^1 \delta h(\delta q_a|f)\mathrm{d}\delta\bigg)$. Then we derive the overall utility of S from conducting this transaction as follows:

$$
\begin{aligned}
&\mathcal{U}_S(m) - \mathcal{U}_S(q_a)\bigg(\int_0^1 \delta h(\delta q_a|f)\mathrm{d}\delta\bigg) - \mathcal{U}_S\big((1-f')\psi\big) && (4.14)\\
=&\mathcal{U}_S(m) - \mathcal{U}_S(q_a)\bigg(1 - \int_0^1 H(\delta q_a|f)\mathrm{d}\delta\bigg) - \mathcal{U}_S\big((1-f')\psi\big) && (4.15)\\
=&\mathcal{U}_S(m) - \mathcal{U}_S(q_a)\bigg(1 - \int_0^1 H(\delta q_a|f)\mathrm{d}\delta\bigg) - \\
&\mathcal{U}_S(\psi)\bigg(1 - \frac{\mathcal{U}_S(q_a)}{\mathcal{U}_S(\psi)}\big(1 - \int_0^1 H(\delta q_a|f)\mathrm{d}\delta\big)\bigg)\\
=&\mathcal{U}_S(m) - \mathcal{U}_S(q_a)\bigg(1 - \int_0^1 H(\delta q_a|f)\mathrm{d}\delta\bigg) - \\
&\mathcal{U}_S(\psi) + \mathcal{U}_S(q_a)\bigg(1 - \int_0^1 H(\delta q_a|f)\mathrm{d}\delta\bigg)\\
=&\mathcal{U}_S(m) - \mathcal{U}_S(\psi)\\
<&\ 0.
\end{aligned}
$$

Note that we get from (4.14) to (4.15) using integration by parts.

Therefore, in both cases, the refinement does not provide S with an incentive to deviate from the optimal strategy $Q = q_a$. □

4.6.3 Robustness of the Statistical Tools

In Sections 4.6.1 and 4.6.2, we propose an approach to utilise the statistical tool $\mathcal{T}$ to reduce some unfair punishment imposed on sellers that results from false negative feedback. The robustness of the statistical tool $\mathcal{T}$ against malicious manipulation becomes an important factor influencing its accuracy.

We suggest that obfuscation is an approach to improve the robustness of the statistical tool $\mathcal{T}$. This can be achieved by different approaches, as follows:

- The operator may find a group of statistical tools $\mathfrak{T} = \{\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_m\}$ and choose a subset to use, perhaps even dynamically changing the choice over time.
- The operator may keep confidential its choice of the statistical tools in use.
- The operator may keep confidential from the buyer whether or not a punishment is eventually imposed on the seller.
- The operator may keep confidential from the seller what feedback is reported by the buyer, if no punishment is applied.
- The operator may keep confidential all feedback reported and punishment imposed in the marketplace.
- It is worth noting that the controlled anonymity, as described in Section 4.3, also contributes to protection measures against malicious attacks.

4.7 Fitting the Feedback Mechanism into our Framework

In this section, we use our model and framework of Chapter 2 to discuss our proposed feedback mechanism. We will first identify the components of the feedback mechanism according to the model described in Section 2.3. We then decompose the environmental assumptions according to Section 2.5 and the design choices of the feedback mechanism according to Section 2.6. Lastly, we perform a brief analysis according to the decomposition of the feedback mechanism.

4.7.1 Components

The components of the feedback mechanism are as follows:

- *Entities*: *Feedback providers* are the buyers. The *relying entities* are the buyers. *Targets* are the sellers. *Processing unit* and *data storage unit* are the operator of the marketplace.
- *Attribute of interest*: quality of service provided by sellers.
- *Interactions*:
  - *Previous interactions*: previous transactions between buyers and sellers.
  - *Future interactions*: potential future transactions between buyers and sellers.
- *Data*:
  - *Feedback*: a buyer's rating on the quality of a received service.
  - *Advice*: once a potential transaction is found, the buyer should proceed with the transaction.

4.7.2 Environmental Assumptions

The environmental assumptions of the reputation system are summarised as follows:

- *Target stability*: *unstable.* The quality of service provided by a seller may change over time.
- *Capability of contributing entities*:
  - *Communication capability*:
    - *Data transmission capability*: *sufficient.* Each trader has sufficient capability to transmit data to and from the operator of the marketplace.
    - *Connectivity*: *fully-connected.* Each trader has a direct communication channel with the operator of the marketplace.
  - *Computational capability*: *sufficient.* We assume that the operator of the marketplace has sufficient computational capability for processing feedback.

  - *Data storage capability*: *sufficient.* We assume that the operator of the marketplace has sufficient capability for storing feedback.
  - *Feedback provision capability*: We assume that consumers are able to evaluate the quality of received services.

- *Motivation of contributing entities*:
  - Buyers as feedback providers: *sufficient.* We assume that buyers have sufficient motivation to report feedback.
  - The operator of the marketplace: *sufficient.* It is in the business interest of the operator to provide data storage and feedback processing services.

- *Availability of contributing entities*:
  - Buyers as feedback providers: *intermittent.* Buyers may not be available constantly.
  - The operator of the marketplace: *constant.* The operator is assumed to be constantly available.

- *Trust relationships*:
  - Traders: They are assumed to make rational decision. This trust fits into following categories:
    - *Global*: It is perceived marketplace-wide.
    - *Static*: It does not change over the life span of the marketplace.
    - *Individual*: It is borne by the individual trader.
  - The operator of the marketplace: We assume that the operator is a trusted entity behaving as a data storage unit and processing unit. This trust fits in the following categories:
    - *Global*: It is perceived marketplace-wide.
    - *Static*: It does not change over the life span of the marketplace.
    - *Individual*: It is borne by the individual entity of the reputation server.

- *Adversarial models.* Possible adversarial models are as follows:
  - *Rationality*: *rational.* We assume that all the traders are rational decision maker.

  - *Location*: *insiders.* Our robustness analysis mainly focuses on insider adversaries.
  - *Strategy space.* The possible adversarial strategy space includes:
    - *Fabrication*: An adversary acting as a buyer may report false feedback.
    - *Collusion*: Adversaries acting as the buyer and seller of a transaction may collude together to pretend to conduct a transaction and then the buyer reports a positive feedback for the seller.

4.7.3 Architectural Choices

The architectural choices of the feedback mechanism are summarised as follows:

- *Role setting*: $\{PS,F,R,T\}$. The operator of the marketplace acts in the compound role of PS. The buyers act in the role of F. The buyers act in the role of R and the sellers act in the role of T.
- *Centrality*: *centralised.* The role of data storage unit and processing unit is performed by a single entity.
- *Data flow*: *receiver-passive.* The operator of the marketplace passively receives feedback reported by the buyers.

4.7.4 Data Processing Choices

The data processing choices of the feedback mechanism are summarised as follows:

- *Representation of the evaluation data.* Representation of the evaluation data in feedback and advice is as follows:
  - In feedback: a *binary score* in the binary mechanism, and a *continuous numerical score* in the continuous mechanism.
  - In advice: a constant advice that buyers should proceed, once a potential transaction is found.

- *Aggregation algorithm.* No aggregation algorithm is required.

4.7.5 Robustness Solutions

The robustness solutions of the reputation system are summarised as follows:

- *Centrality.* A trusted and centralised processing unit and data storage is adopted. This avoids an adversary acting as a data storage unit and a processing unit.
- *Identity disguise.* Our marketplace requires controlled anonymity, i.e. the identities of buyers and sellers are randomised for every single transaction. This prevent an adversary acting as a buyer from targeting a particular seller by reporting false negative feedback for the seller.
- *Data mining techniques.* The refinement of the feedback mechanism utilise the statistical tool $\mathcal{T}$.
- *Incentive mechanism.* For every transaction, the seller is induced to provide a satisfactory service.

4.7.6 Discussion

Some interesting features of our proposed feedback mechanism are as follows:

- There is only a constant advice in this feedback mechanism, i.e. buyers should proceed with the transaction once they find a potential transaction. This is because in every potential transaction, the buyer is induced to proceed with the transaction.
- No aggregation algorithm is required in our feedback mechanism. This is because feedback only has impact on the transaction from which the feedback is reported, and thus no aggregation algorithm is needed.

On the other hand, our feedback mechanism has some drawbacks. For example, our feedback mechanism requires that the overall feedback reporting behaviour meets certain conditions. In other words, the binary mechanism requires that $g(1|q_1) + g(0|q_0) - 1 > 0$ and $\frac{g(0|q_0)}{g(0|q_1)} > \frac{\mathcal{U}_S(m)}{\mathcal{U}_S(m) - \mathcal{U}_S(q_1)} + 1$ (see Section 4.4.1). The continuous mechanism requires that $\mathbb{G}(\delta q_a) - \mathbb{G}(q_a) > 0$ and $\frac{\mathbb{G}(\delta q_a)}{\mathbb{G}(q_a)} > \frac{\mathcal{U}_S(m)(1-\delta)}{\mathcal{U}_S(m) - \mathcal{U}_S(q_a)} + 1$, $\forall \delta \neq 1$ (see Section 4.4.2). These mean that the overall feedback reporting behaviour in the marketplace needs to achieve a certain level of honesty. This limits the applicability of our feedback mechanism.

However, in some application scenarios, this may be achieved by some external factors that ensure sufficient honest feedback report. For example:

- If the buyers of a marketplace are strongly influenced by a culture that encourages people to behave in a honest manner then it is reasonable to expect a high level of honest feedback report.
- We may integrate a suitable incentive mechanism to induce honest feedback reporting, as we discussed in Section 4.2 and will point out in Section 4.8 as a possible future work.

4.8 Conclusion and Future Work

In this chapter, we propose a novel feedback mechanism for electronic marketplaces. Buyer feedback impacts on the seller's payoff only for the current transaction but not other transactions. Our mechanism is able to induce good conduct for every transaction. The exit, bootstrapping, and feedback delay problems of reputation systems do not apply to our scheme. It is also robust against buyer-seller collusion attacks.

It is of interest to explore more approaches to further protect sellers. For example:

- Sellers may be allowed to defend themselves by presenting evidence of the quality of the delivered service if a negative feedback is reported from a buyer. If the evidence is valid, then the punishment can be dropped.

- The marketplace operator may conduct random transaction monitoring. In other words, the operator randomly chooses some transactions and witnesses the quality of the service delivery. If a buyer reports an untruthful feedback for a monitored transaction, it may be punished. As a result, buyers may be induced to provide honest feedback.

- Some ideas from existing side payment mechanisms may be applied to induce buyer honesty in feedback reporting.

Chapter 5

A Reputation-based Announcement Scheme for VANETs

Contents

This chapter provides a novel announcement scheme for VANETs based on a reputation system that allows evaluation of message reliability. We present a secure and efficient scheme which is robust and fault tolerant against temporary unavailability of the central server.

5.1 Introduction

A vehicular ad hoc network (VANET) is formed by roadside infrastructure and mobile nodes embedded within vehicles which are connected in a self-organised way. Active research in VANETs is demonstrated by numerous papers in the academic literature, for example [99, 21, 29, 37, 39, 48, 53, 54, 65, 68, 75, 86, 98, 101, 102, 103, 104, 112, 121] and ongoing projects [16, 69] in industry. VANETs allow vehicles to generate and broadcast messages about road conditions, such as traffic congestion, accidents and road conditions. We call these kinds of messages *road-related messages* and a scheme that facilitates vehicles to generate and broadcast road-related messages an *announcement scheme*. Broadcast of road-related messages may help vehicles to be aware of the situation ahead of them and, as a result, may provide a safer driving environment. In addition, it also has the capability to improve efficiency of traffic on road networks. However, these benefits can only be realised if the road-related messages generated by vehicles are reliable.

We say that a message is *reliable* if it reflects reality. Unreliable messages may result in various consequences, for example journey delays or accidents. Unreliable messages may be the result of vehicle hardware malfunction. For example, if a sensor in a vehicle is faulty then the messages generated based on the information provided by the faulty sensor may be false. Unreliable messages can also be generated intentionally. For example, some vehicles may generate and broadcast false road congestion messages with the intention to deceive other vehicles into avoiding certain routes. In the extreme case, unreliable message may lead to injuries and even deaths. Hence, evaluation of the reliability of vehicle-generated messages is of importance in VANETs.

In a large VANET environment, vehicles are assumed to have a weak (or no) trust relationship with each other [21]. This raises the question: how do vehicles decide whether to rely on a message? In this chapter, we address this problem by proposing a novel reputation-based announcement scheme for VANETs. The reliability of a message is evaluated according to the *reputation* of the vehicle that generates this message. A message is considered reliable provided that the vehicle that generates the message has a sufficiently high reputation. The reputation of a vehicle is represented by a numerical score. This reflects the extent to which the vehicle has announced reliable messages in the past. It is computed based on *feedback* reported by other vehicles. Feedback contains a numerical score representing the feedback-reporting vehicle's personal evaluation of the reliability of the message. The score is collected, updated and certified by a trusted party (e.g. a reputation server). The reputation score evolves, as time elapses, based on the reliability of messages that the vehicle announces. Vehicles tend to give positive feedback for reliable messages. This increases the reputation score. Meanwhile, a reputation score decreases when negative feedback is reported.

The rest of the chapter is organised as follows. We discuss related work in Section 5.2. In Section 5.3, we introduce the entities involved in our scheme and their relationships. We also introduce the notation and algorithms needed in our scheme, and show how to initialise a system that applies our scheme. We then elaborate our scheme in Section 5.4. We analyse the robustness of the scheme in Section 5.5. In Section 5.6, we discuss other properties of our scheme and some related issues. In Section 5.7, we show some simulation results about the performance of our announcement scheme. In Section 5.8, we apply our proposed model and framework of Chapter 2 to decompose and discuss the reputation system used by our announcement scheme. Section 5.9 shows some possible approaches to extending the scheme. We conclude in Section 5.10 and discuss future work.

5.2 Related Work

There have been a number of announcement schemes proposed in order to evaluate the reliability of announcement messages in VANETs. Generally, a message is considered reliable if:

- the integrity of the message is valid;
- the message was generated and announced by a legitimate vehicle; and
- there is a means of "measuring" message reliability.

Digital signatures are commonly used to satisfy the first two requirements [99, 21, 29, 39, 48, 65, 68, 86, 101, 112, 121]. To achieve the third requirement, different techniques have been proposed. These include the threshold method [21, 29, 68, 101, 121], network modelling [48], and trust-based and reputation-based models [99, 39, 86, 112]. We will discuss those most closely related to our work.

A majority of the schemes in the literature uses the threshold method, for example [21, 29, 68, 101, 121]. In this mechanism, a vehicle accepts a message if it receives messages with the same content that have been announced by a number of distinct vehicles that exceeds a threshold within a time interval. The threshold may be a fixed system-wide parameter [29, 101] or a flexible parameter [21, 68, 121]. The threshold has to be chosen carefully. It should not be so high that insufficient endorsement occurs and vehicles are not able to utilise the information received. It should not be so low that the decision may be affected by the presence of adversaries. In our scheme we do not require multiple messages from other vehicles in order to evaluate the reliability of a message. Indeed we may only need to verify one message provided that the reputation of the announcing vehicle is sufficiently high. This allows vehicles to make decisions and act upon messages quickly.

Golle et al. [48] proposed the evaluation of message reliability by modelling the network. They present a scheme that allows vehicles to detect and correct malicious messages in VANETs. Vehicles are assumed to maintain a “model” of the VANET, which contains all the knowledge that the vehicles possess about the VANET. A vehicle can then compare the messages received against the model of the VANET. A message that is consistent and agrees with the vehicle’s model is likely to be accepted as valid. Inconsistent messages are addressed using a heuristic approach. A vehicle will search for explanations for the inconsistent messages and rank all possible explanations according to the heuristic approach. The message with the highest scoring explanation will be validated. However, requiring vehicles to possess a wide knowledge of the network may be infeasible and impractical. In our work, we propose a simpler and more practical model. We evaluate messages based on the simple principle of reputation, where the reliability of a message generated by a vehicle is reflected by its reputation score.

Several trust-based and reputation-based models, for example [99, 39, 86, 112], have been presented in the literature. In these schemes, a decentralised infrastructure is adopted. However, the issue associated with decentralised infrastructures is that robustness is often not guaranteed.

In [39], Dötzer et al. proposed a reputation system based on a mechanism called *Opinion Piggybacking*. In this approach, a vehicle generates a message and broadcasts it to neighbouring vehicles. A receiving vehicle will append its own opinion about the reliability of the message, which may be based on the content of the message or the aggregated opinions already appended to the message. Upon receiving a message, a vehicle is required to compute and aggregate previous opinions appended to the message before it decides and generates its own opinion. This may create a computational burden on receiving vehicles. In addition, details of implementation such as the initialisation of the reputation system and the updating of reputation scores of vehicles were not discussed. Issues of revocation and robustness against possible collusion of adversaries were also not addressed.

In the scheme by Minhas et al. in [86], message reliability is evaluated by modelling the trustworthiness of the message generator. In this scheme, vehicle trustworthiness is modelled based on the combination of three trust models: role-based trust, experience-based trust and majority-based trust. *Role-based trust* exploits certain predefined roles that are enabled through the identification of vehicles. For example, vehicles may have more trust towards traffic patrol or law enforcing authorities compared to other vehicles. To avoid impersonation attacks, each vehicle is required to possess a certificate that includes its name, role and public key, issued by a trusted authority for authentication purposes. *Majority-based trust* is similar to the threshold method that we discussed earlier. *Experience-based trust* is established based on direct interactions: a vehicle determines who to trust based on how truthful they have been in their past interactions. However, such a model requires vehicles to establish a long-term relationship with each other, which may not be practical in a large VANET environment. Furthermore, it also requires vehicles to store information regarding vehicles that they have encountered in the past. This may lead to storage problems. A similar approach of experienced-based trust was proposed by Patwardhan et al. in [99].

Schmidt et al. proposed a framework for vehicle behaviour analysis in [112]. A vehicle's behaviour refers to all observable information, including its movement and position in the past and present. A receiving vehicle accumulates a sequence of messages from a broadcasting vehicle and these may provide sufficient information for behaviour analysis. The result of this analysis will help to determine a vehicle as trustworthy, neutral or untrustworthy. In this approach, vehicles are required to make observations before a decision can be made. This may not be desirable in VANETs, since vehicles are not able to act quickly upon the messages received.

Compared with these trust-based and reputation-based approaches, our work features as follows:

- We take advantage of the already existing centralised infrastructure in a highly dynamic and distributed environment of VANETs. This allows us to design a secure and efficient announcement scheme.
- We design a comprehensive announcement scheme using a reputation system that allows evaluation of message reliability that is practical, efficient and robust against adversaries. Vehicles may provide feedback for messages received. These feedbacks accumulate to a vehicle's reputation score. Hence short-term encounters between vehicles may lead to long-term trust, which is represented

by reputation scores.

- Vehicles can quickly decide whether to rely on a message or not based on the reputation score. The reputation score reflects the extent to which a vehicle has announced reliable messages in the past, which reflects the likelihood that it will announce reliable messages in the future.

Here, we focus only on the research related to the issue of evaluating message reliability for an announcement scheme in VANETs. For other issues, we refer the reader to [21, 68, 17, 119] for wider overviews of the topic.

5.3 Preliminaries

In this section, we introduce the entities involved in our scheme and their relationships. We also introduce some algorithms and notation. Lastly, we will describe how to initialise the system.

5.3.1 Entities

Our system consists of three types of entity: *a reputation server*, *access points* and *vehicles*.

5.3.1.1 Reputation Server

We rely on a centralised reputation server which we assume is a trusted authority. One role of the reputation server is to maintain the reputation of vehicles. This includes collecting feedback, aggregating feedback to produce reputation and propagating reputation. The reputation server is also in charge of admitting vehicles into, and revoking them from, the system.

There are several justifications for adopting a centralised architecture. First, it is a common practice that vehicles are regulated and governed by some centralised authority, such as the Driver and Vehicle Licensing Agency (DVLA) in the United Kingdom. Hence it is natural to adopt a centralised architecture. In addition, a centralised architecture has some advantages over a decentralised system. For example, it is often easier to manage, control and secure a centralised system.

We assume that the reputation server is equipped with a clock.

5.3.1.2 Access Points

Our scheme relies on *access points*, which are physical wireless communication devices. These are connected with the reputation server, acting as a communication interface between vehicles and the reputation server. The purpose of access points is to allow vehicles to communicate with the centralised reputation server in a convenient and frequent manner. It is worth noting that our scheme does not require a vehicle to be able to communicate with the reputation server all the time. Further, our scheme does not require a secure communication channel between an access point and the reputation server. Rather, it suffices that a public communication channel connects an access point and the reputation server.

We envisage that access points are installed at locations frequently visited by vehicles such as fuel stations, service stations and traffic lights. The number of access points required depends on the size of the system, the road topology and traffic patterns, etc.

5.3.1.3 Vehicles

Vehicles are the end users of the system. They broadcast and receive messages to and from their neighbouring vehicles. In our scheme, a vehicle comprises the actual vehicle and its human user. We assume that there is no prior trust between vehicles. Upon receipt of a message, the receiving vehicle needs to evaluate the reliability of the message before considering how to act upon it.

We assume that a vehicle is equipped with a computing device called an on-board unit (OBU), which has wireless communication capability to broadcast and receive messages to and from other OBUs on neighbouring vehicles. In addition, we assume that trusted hardware is embedded as part of an OBU, so that any secret data cannot be learnt by anyone, including the vehicle itself. The trusted hardware can securely store keys and perform embedded cryptographic operations, such as digital signatures. We also assume that a secure clock is embedded within the trusted hardware.

5.3.2 Algorithm Components and Notation

The algorithms needed in our scheme are described as follows. We provide a summary of all notation used in this chapter in Section 5.11.

- Our scheme requires a reputation aggregation algorithm Aggr. It computes a reputation score for each vehicle based on feedback reported by other vehicles. We will discuss it in more detail in Section 5.4.6.
- We need a time discount function, denoted by $\mathsf{TimeDiscount}$. This is a non-increasing function whose range is $[0, 1]$. It takes as input a non-negative value representing a time difference, and outputs a number between 0 and 1. One simple example is:

  $$\mathsf{TimeDiscount}(t) = \begin{cases} 1 - t/\Psi_{td} & \text{if } t < \Psi_{td}; \\ 0 & \text{if } t \geq \Psi_{td}, \end{cases}$$

  where $\Psi_{td} > 0$ is a public parameter, determining how quickly the time discount function decreases as t increases.
- We require two secure digital signature schemes, denoted by $\mathsf{DS}_1 = (\mathsf{KeyGen}_1,$ Sign_1, $\mathsf{Verify}_1)$ and $\mathsf{DS}_2 = (\mathsf{KeyGen}_2, \mathsf{Sign}_2, \mathsf{Verify}_2)$, where KeyGen, Sign and Verify denote key generation, signing and verification algorithms, respectively. We use two digital signature schemes because they will be used for different purposes, and hence there may be different requirements for each scheme.
- We require a secure cryptographic hash function, denoted by H.
- We require a secure message authentication code (MAC) algorithm, denoted by MAC.

- We also require a vehicle clock regulation protocol, denoted by VCRP. It consists of a server-side protocol, denoted by VCRP_S, and a vehicle-side protocol, denoted by VCRP_V. The purpose of VCRP is to ensure that only the reputation server is able to regulate the secure clock embedded in the trusted hardware of a vehicle. An entity authentication protocol can be applied to achieve the protocol VCRP.
- We require three configurable public parameters Ψ_{rs}, Ψ_t and $\mathbb{T}$. The parameter Ψ_{rs} acts as a threshold and is used by a vehicle to determine whether or not another vehicle is reputable. It is a constant between 0 and 1. The parameter Ψ_t also acts as a threshold and is used to determine whether or not a message tuple is sufficiently fresh for feedback reporting. The parameter $\mathbb{T}$ is a large time interval, over which a *sufficiently large* number of vehicles report feedback relating to a vehicle.

5.3.3 Initialisation of the System

The initialisation of the system includes initialisation of the reputation server, new vehicles and new access points.

5.3.3.1 Initialisation of the Reputation Server

When a new announcement scheme is set up, the reputation server is initialised as follows. The reputation server:

- Installs the reputation aggregation algorithm Aggr.
- Installs the algorithms KeyGen_1, Sign_1, KeyGen_2 and Verify_2.
- Generates its own public and private key pair (pk_S, sk_S) using KeyGen_1. The private key sk_S is then kept confidential.
- Installs the server-side protocol VCRP_S.
- Regulates its own clock.

- Creates a database which will store the following data for every vehicle in the system: the identity, public key, MAC key, current reputation score and all feedback reported for the vehicle.

5.3.3.2 Admission of New Vehicles

When a new vehicle V is admitted into the system, it is initialised as follows. The reputation server:

- Assigns it a unique identifier, denoted by id_V.
- Generates a public and private key pair, denoted by (pk_V, sk_V), for the vehicle using the algorithm KeyGen_2.
- Generates a MAC key mk_V for the vehicle.
- Embeds the private key sk_V, the MAC key mk_V and the algorithm Sign_2 into the trusted hardware of the vehicle. It also embeds the vehicle clock regulation algorithm $\mathsf{VCRP_V}$ into the trusted hardware. We assume that this procedure is conducted in a secure environment.
- Applies the server-side protocol $\mathsf{VCRP_S}$ to send a clock regulation instruction in oder to regulate the clock embedded within the trusted hardware of V.
- Installs the hash function H, the algorithms Verify_1 and Verify_2, its own public key pk_S and the thresholds Ψ_{rs} and Ψ_t into the OBU of the vehicle. Note that these are not necessarily installed into the trusted hardware of the vehicle.
- Creates a record in its database for vehicle V containing id_V, pk_V and mk_V. The initial reputation score field is set to 0 and the feedback field is left empty.

5.3.3.3 Installation of New Access Points

When a new access point is installed in the system, a communication channel needs to be established between the access point and the reputation server. Subsequently, the access point serves as a communication interface between vehicles and the reputation server.

5.4 Operation of the Announcement Scheme

We describe our scheme by showing how reputation of a vehicle is formed, propagated, updated and utilised in order to determine the reliability of a message sent by the vehicle. The operation of the scheme consists of the following phases: *reputation certificate retrieval*, *message announcement*, *message reliability evaluation*, *feedback reporting*, *reputation update* and *vehicle revocation*.

5.4.1 Reputation Certificate Retrieval

In this phase, a vehicle retrieves its latest reputation certificate from the reputation server. When a vehicle V_b drives into wireless communication range of an access point, it retrieves its own reputation certificate from the central server via the access point as follows:

1. V_b sends its identity id_{V_b} to the server via the access point.
2. The reputation server generates a *reputation certificate* C for the vehicle, where:
$$C = (id_{V_b}, pk_{V_b}, t_c, rs_{V_b}, \sigma),$$
in which t_c denotes the time when C is generated and it is obtained from the reputation server's clock, rs_{V_b} denotes the reputation score of V_b at time t_c, and $\sigma = \mathsf{Sign}_1(id_{V_b}, pk_{V_b}, t_c, rs_{V_b})_{sk_S}$ denotes a digital signature using the signature algorithm Sign_1 and private key sk_S on $(id_{V_b}, pk_{V_b}, t_c, rs_{V_b})$.
3. The reputation server sends C to V_b via the access point.
4. Once V_b obtains C, it stores the reputation certificate locally. Previously obtained reputation certificates can then be deleted.

Note that in this procedure, V_b is not required to authenticate itself to the reputation server. This is because the reputation certificate is not confidential and can be retrieved by any vehicle. We will show later that there is no point in one vehicle retrieving the reputation certificate of another vehicle.

5.4.2 Message Broadcast

In this phase, V_b generates a road-related message and broadcasts it to its neighbouring vehicles. This is described as follows:

1. V_b converts the information obtained, for example from its sensors or driver, into a message m. The technical detail of how this is done is beyond the scope of this chapter. It computes the hash value $\mathsf{H}(m)$, which it then submits to its trusted hardware.
2. The trusted hardware retrieves the current time t_b from its embedded clock and generates a time-stamped signature $\theta = \mathsf{Sign}_2(t_b, \mathsf{H}(m))_{sk_{V_b}}$, and outputs t_b and θ.
3. V_b forms a *message tuple* $M = (m, t_b, \theta, C)$ and broadcasts M to its neighbouring vehicles.

5.4.3 Message Reliability Evaluation

Upon receiving the message tuple $M = (m, t_b, \theta, C)$, a receiving vehicle V_r performs the following procedure:

1. V_r submits θ to its trusted hardware.
2. The trusted hardware retrieves the current time t_r from its embedded clock, stores the tuple (t_r, θ), and then outputs t_r to V_r.
3. V_r checks:
   (a) whether the time-discounted reputation score is acceptable, i.e.:
   $$rs_{V_b} \cdot \mathsf{TimeDiscount}(t_r - t_c) \geq \Psi_{rs},$$
   where t_c is extracted from C;
   (b) whether the message tuple M is sufficiently fresh, i.e. $t_r - t_b \leq \Psi_t$;
   (c) whether $\sigma \in C$ is valid, by using the verification algorithm Verify_1 and the public key of the reputation server pk_S; and

(d) whether θ is valid, by using the verification algorithm Verify_2 and the public key pk_{V_b}, which can be extracted from C.

If all checks are positive, then vehicle V_b is considered to be reputable. Message m is thus considered as reliable and can be taken into consideration. The message tuple M is stored for future feedback reporting. Otherwise, V_b is not considered to be reputable and m is not considered to be reliable. However, if at least Steps 3b, 3c and 3d are positive, then the message tuple M is still stored for future feedback reporting; otherwise it is discarded.

5.4.4 Feedback Reporting

In this phase, when vehicle V_r has its own experience about the event that the message m describes, it is able to judge the reliability of the message. Then if V_r wants to report feedback to the reputation server, it performs the following procedure.

1. V_r generates a feedback rating $fr \in \{0, 1\}$, where $fr = 1$ represents that m is reliable, while $fr = 0$ represents that m is false. In this chapter, we only use binary feedback rating for simplicity.
2. V_r submits $(id_{V_b}, id_{V_r}, fr, t_b, \mathsf{H}(m), \theta)$ to its trusted hardware.
3. The trusted hardware retrieves t_r from the tuple (t_r, θ) that was previously stored during the message reliability evaluation phase, computes a message authentication code (MAC) value $\delta = \mathsf{MAC}(id_{V_b}, id_{V_r}, fr, t_b, t_r, \mathsf{H}(m), \theta)_{mk_{V_r}}$, and then outputs t_r and δ.
4. V_r forms a *feedback tuple* $F = (id_{V_b}, id_{V_r}, fr, t_b, t_r, \mathsf{H}(m), \theta, \delta)$. We say that F is *positive* feedback if $fr = 1$ and *negative* feedback if $fr = 0$.
5. When V_r drives into wireless communication range of an access point, it sends the feedback tuple F to the reputation server via the access point.

Note that V_r is not required to authenticate itself to the reputation server during feedback upload. This is because F contains the MAC value θ, which can only be generated by V_r and the reputation server.

5.4.5 Reputation Update

In this phase, the reputation server updates the reputation score rs_{V_b} of vehicle V_b on receipt of a feedback tuple $F = (id_{V_b}, id_{V_r}, fr, t_b, t_r, \mathsf{H}(m), \theta, \delta)$, as follows:

1. The reputation server first checks:
   (a) whether $t_r - t_b \leq \Psi_t$;
   (b) whether the MAC value δ is valid, by computing a MAC on the tuple $(id_{V_b}, id_{V_r}, fr, t_b, t_r, \mathsf{H}(m), \theta)$ using mk_{V_r} and checking whether it matches δ; and
   (c) whether the signature θ is valid, by using the algorithm Verify_2 and pk_{V_b}.

   If any check fails then this procedure is terminated and F is discarded.

2. If the checks pass then the reputation server considers the feedback tuple F as valid and stores it in the database.

3. The reputation server applies the reputation aggregation algorithm Aggr on all stored feedback relating to V_b in order to compute the latest reputation score rs_{V_b} for vehicle V_b. It then replaces the previous reputation score in the database with rs_{V_b}.

5.4.6 The Reputation Aggregation Algorithm

In this section we discuss how the reputation aggregation algorithm Aggr works. We will show how Aggr produces the latest reputation score rs_V for vehicle V based on all stored feedback, as follows:

1. The aggregation algorithm Aggr first selects all feedback reported for V whose corresponding message tuple was broadcast from time $\mathbb{T}$ in the past until now. More formally, let t_a denote the time when this aggregation is running. The algorithm Aggr selects a subset of feedback $\mathcal{F}$ where:

$$\mathcal{F} = \{F : (id_{V_b} = id_V) \wedge (t_b \geq t_a - \mathbb{T})\}.$$

Feedback whose corresponding message was broadcast earlier than time $\mathbb{T}$ in the past is ignored, and deleted if necessary for the sake of data storage efficiency.

2. Multiple feedback reported by one vehicle V_i for V is aggregated into one intermediate value $\hat{r}_{V_i}$. Let $\mathcal{F}_{V_i}$ denote the set of feedback reported by the vehicle V_i for V and whose corresponding message was broadcast from time $\mathbb{T}$ in the past until now, i.e.:

   $$\mathcal{F}_{V_i} = \{F : (id_{V_b} = id_V) \wedge (id_{V_r} = id_{V_i}) \wedge (t_b \geq t_a - \mathbb{T})\}.$$

   The value $\hat{r}_{V_i}$ can be aggregated using a weighted average as follows:

   $$\hat{r}_{V_i} = \frac{\sum\limits_{F \in \mathcal{F}_{V_i}} fr \cdot \big(\mathbb{T} - (t_a - t_b)\big)}{\sum\limits_{F \in \mathcal{F}_{V_i}} \big(\mathbb{T} - (t_a - t_b)\big)}. \tag{5.1}$$

   This gives more recent feedback a greater weight than less recent feedback.

   Let $\mathcal{V}$ denote the set of vehicles that have each reported at least one feedback for V in the past $\mathbb{T}$ time, i.e.:

   $$\mathcal{V} = \{V_i : (id_{V_b} = id_V) \wedge (id_{V_r} = id_{V_i}) \text{ for some } F \in \mathcal{F}\}.$$

   The value $\hat{r}_{V_i}$ is computed for each vehicle $V_i \in \mathcal{V}$.

3. Let $\mathcal{V}^-$ denote the set of vehicles reporting at least one negative feedback for V in the past $\mathbb{T}$ time, i.e.:

   $$\mathcal{V}^- = \{V_i : (id_{V_b} = id_V) \wedge (id_{V_r} = id_{V_i}) \wedge (fr = 0) \text{ for some } F \in \mathcal{F}\}.$$

   The latest reputation score rs_V is computed as follows:

   $$rs_V = \begin{cases} \frac{\sum_{V_i \in \mathcal{V}} \hat{r}_{V_i}}{|\mathcal{V}|} & \text{if } |\mathcal{V}^-| < \Psi_{nf}, \\ 0 & \text{otherwise}, \end{cases} \tag{5.2}$$

   where Ψ_{nf} is a configurable public parameter. It impacts on the robustness of the scheme and its configuration will be discussed in Sections 5.5.2 and 5.6.6. The intuition of this equation is that rs_V is computed as the average of $\hat{r}_{V_i}$ if not too many vehicles reporting negative feedback for V in the past $\mathbb{T}$ time; otherwise rs_V decreases to 0, indicating that V has conducted a message fraud attack, which will be discussed in Section 5.5.2.

5.4.7 Vehicle Revocation

The reputation server revokes a vehicle from the system if $|\mathcal{V}^-| < \Psi_{nf}$. If a vehicle is revoked, then the reputation server stops providing new reputation certificates for it. Feedback reported by the revoked vehicle will not be considered as valid. Note that previously issued reputation certificates will gradually expire as time elapses.

5.5 Robustness Analysis

In this section, we analyse the robustness of our scheme in the presence of adversaries with respect to the following attacks:

- *Message fraud.* In this attack an adversary deceives a vehicle complying with the scheme into believing that a false message m' is reliable.
- *Reputation manipulation.* In this attack an adversary unfairly inflates or deflates the reputation score of a target vehicle. This target vehicle can be the adversary itself.

  Note that reputation manipulation may lead to message fraud, since an adversarial vehicle can get its reputation unfairly inflated by a reputation manipulation attack and then launch a message fraud attack.

We categorise adversaries into two groups:

- *External adversaries* attack the system without joining as legitimate vehicles.
- *Internal adversaries* are legitimate vehicles that attack the system.

We define a notion of robustness: an announcement scheme provides (Φ_{MF}, Φ_{RM})-*robustness* if:

- Φ_{MF} is the maximum number of vehicles that an internal adversary can deceive during a time period of length $\mathbb{T}$ without itself getting revoked. This evaluates the extent to which the scheme is robust against a message fraud attack.

- Φ_{RM} is the maximum value by which the reputation score of a vehicle can be unfairly manipulated (increased or decreased) by adversaries. This evaluates the extent to which the scheme is robust against a reputation manipulation attack.

We say that an announcement scheme provides *strong robustness* if it provides $(0, 0)$ robustness, i.e. $\Phi_{MF} = 0$ and $\Phi_{RM} = 0$.

5.5.1 Robustness against External Adversaries

5.5.1.1 Robustness against Message Fraud

Claim 1. *Our scheme provides strong robustness against external adversaries conducting message fraud.*

Proof. In order to perpetrate a message fraud attack, an external adversary can engage in any of the following strategies:

1. obtain a valid reputation certificate C for a vehicle V and then forge a message tuple $M' = (m', t_b, \theta, C)$ containing a false message m' in the name of V;
2. forge a reputation certificate C' and then create a valid message tuple $M' = (m', t, \theta, C')$ containing a false message m';
3. corrupt a vehicle V that is about to generate and broadcast a message tuple $M = (m, t_b, \theta, C)$ and then replace m with a false message m', so that V will generate and broadcast $M' = (m', t_b, \theta, C)$.

An external adversary is not able to forge a valid reputation certificate C or a valid message tuple M unless the adversary has access to either the private key sk_S or sk_V. Hence, assuming that the digital signature schemes used are secure and the reputation server and vehicles manage keys appropriately, then the adversary is not able to succeed using the first two strategies. It is also reasonable to assume that an external adversary is not able to corrupt a vehicle in order to replace the message m generated by the vehicle with a false message m' before the message tuple M is generated. Hence we can regard our scheme as providing strong robustness against message fraud attacks. □

5.5.1.2 Robustness against Reputation Manipulation

Claim 2. *Our scheme provides strong robustness against external adversaries conducting reputation manipulation.*

Proof. In order to conduct a reputation manipulation attack, an external adversary can engage in any of the following strategies:

1. forge and report valid feedback in the name of vehicle V_r for a target vehicle V with its own choice of feedback rating;
2. corrupt a vehicle V_r that is about to generate and report a feedback $F = (id_V, id_{V_r}, fr, t_b, t_r, \mathsf{H}(m), \theta, \delta)$ for the target vehicle V and replace fr with a value of its own choice;
3. corrupt the reputation server and directly modify the stored reputation score rs_V of the target vehicle V.

Forging valid feedback involves forging a valid MAC value δ generated using the MAC key of a legitimate vehicle. Assuming the use of a secure MAC algorithm and that vehicles manage their MAC keys appropriately, an external adversary is not able to forge δ using the first strategy. It is also reasonable to assume that that an external adversary is not able to corrupt a vehicle in order to replace fr with a false feedback rating or corrupt the reputation server in order to modify the stored reputation score. Hence our scheme provides strong robustness against external adversaries conducting reputation manipulation attacks. □

5.5.2 Robustness against Internal Adversaries

It is straightforward to see that our scheme does *not* provide strong robustness against internal adversaries. This is because an internal adversary with a time-discounted reputation score greater than Ψ_{rs} can deceive its neighbouring vehicles into believing that a false message m' is reliable. Further, when an internal adversary receives a message tuple $M = (m, t_{V_b}, \theta, C)$ from a target vehicle it can always intentionally report false feedback. In this section, we will analyse to what extent our scheme is robust against internal adversaries.

5.5.2.1 Robustness against Reputation Manipulation

We consider the worst situation where all adversaries collude together to attack the same target vehicle V with the same goal (to inflate or deflate the reputation score of V). Recall that in order to form valid feedback, an internal adversary has to obtain a valid message tuple $M = (m, t_b, \sigma, C)$ generated by V, and obtain it before time $t_b + \Psi_t$. We assume that Ψ_t is set such that only those vehicles physically within wireless communication range of a broadcasting vehicle are able to obtain a valid message tuple before time $t_b + \Psi_t$.

Claim 3. *Let $\mathcal{V}$ denote all vehicles that have each reported at least one valid feedback relating to V, and let $\mathcal{V}_a \subseteq \mathcal{V}$ denote all internal adversaries among $\mathcal{V}$. The robustness against internal adversaries conducting reputation manipulation is $\Phi_{RM} = \frac{|\mathcal{V}_a|}{|\mathcal{V}|}$.*

Proof. Let $V_a \in \mathcal{V}_a$ be an internal adversary and $\hat{r}_{V_a}$ be the intermediate value aggregated from all feedback reported by V_a for V, according to Equation (5.1). It is easily seen that all false feedback reported by V_a for V only changes the intermediate value $\hat{r}_{V_a}$. The maximum influence of the intermediate value $\hat{r}_{V_a}$ on the reputation score rs of the target vehicle V equals $\frac{1}{|\mathcal{V}|}$. Hence the maximum extent of reputation manipulation due to one internal adversary is $\frac{1}{|\mathcal{V}|}$. Therefore the maximum extent of reputation manipulation due to all members of $\mathcal{V}_a$ equals to $\frac{|\mathcal{V}_a|}{|\mathcal{V}|}$, i.e. $\Phi_{RM} = \frac{|\mathcal{V}_a|}{|\mathcal{V}|}$. □

If $|\mathcal{V}_a|$ is relatively small compared with the size of $\mathcal{V}$, then the maximum unfair impact of internal adversaries conducting reputation manipulating attack is still small. In this case, $\mathcal{V}_a$ only adds a small noise into the reputation score of the target vehicle. It is reasonable to assume that in a VANET there is only a small proportion of internal adversaries compared with the entire population of vehicles. Hence, the unfair impact of internal adversaries conducting reputation manipulating attack remains small.

5.5.2.2 Robustness against Message Fraud

With respect to a message fraud attack, apart from those strategies mentioned in Section 5.5.1, which can be used by external adversaries, internal adversaries have an additional attack strategy. This strategy is for an internal adversary to exploit its own reputation, as described in the beginning of this section. However, an internal adversary cannot use this strategy to conduct message fraud persistently.

Claim 4. *Let p denote the overall probability that vehicles will report negative feedback upon being deceived by a false message. Let the public parameter Ψ_{nf} be set such that $\Psi_{nf} = |\mathcal{V}_a| + \Delta$, where Δ is a safe margin. The robustness against internal adversaries conducting message fraud is $\Phi_{MF} = \frac{|\mathcal{V}_a|+\Delta}{p}$.*

Proof. If an internal adversary deceives more than $\frac{\Psi_{nf}}{p}$ vehicles during a time period of length $\mathbb{T}$, then the number of negative feedbacks reported for it is likely to be greater than Ψ_{nf}. This results in its reputation score decreasing to 0, as shown in Equation (5.2). The internal adversary will thus be revoked. Hence, the maximum number of vehicles that an internal adversary can deceive during a time period of length $\mathbb{T}$ without getting revoked equals $\frac{\Psi_{nf}}{p}$.

Note that a vehicle is revoked if $|\mathcal{V}^-| \geq \Psi_{nf}$. Given $\Psi_{nf} = |\mathcal{V}_a| + \Delta$, then $|\mathcal{V}^-| < \Psi_{nf}$, meaning that the internal adversaries $\mathcal{V}_a$ are not able to get the target vehicle V revoked by reputation manipulation. The robustness of the scheme against an internal adversary conducting message fraud is $\Phi_{MF} = \frac{|\mathcal{V}_a|+\Delta}{p}$. □

By combining Claims 3 and 4, we can conclude that our scheme is ($\Phi_{MF} = \frac{|\mathcal{V}_a|+\Delta}{p}$, $\Phi_{RM} = \frac{|\mathcal{V}_a|}{|\mathcal{V}|}$)-robust against internal adversaries.

5.6 Discussion

In this section, we discuss other properties and issues related to our scheme.

5.6.1 Fault Tolerance

One important advantage of our scheme is its fault tolerance. This is shown from two perspectives: temporary unavailability of the reputation server and temporary unavailability of access points.

Recall that during the message broadcast and message reliability evaluation phases, the reputation server is not involved. In other words, the reputation server is off-line with respect to message broadcast and message reliability evaluation. From the perspective of a vehicle, the reputation server is only needed for reputation certificate retrieval and feedback reporting. Temporary unavailability of the reputation server only affects those vehicles which happen to retrieve their reputation certificates when the reputation server is unavailable. These vehicles have to continue using their existing reputation certificates. This negative effect only lasts until they successfully retrieve their new reputation certificates the next time that the reputation server is available again. The operation of the system is largely unaffected during the time when the reputation server is temporarily unavailable.

Access points which become temporarily unavailable also do not greatly affect the operation of the system. An unavailable access point only affects those vehicles which happen to drive into wireless communication range of the access point for retrieving reputation certificates. In most cases vehicles can be expected to drive into wireless communication range of another working access point within a reasonable time period.

5.6.2 Privacy

Privacy is often an important criteria of an announcement scheme for VANETs. There has been active research into this topic, e.g. [17, 21, 38, 68, 96]. While privacy has not been the main focus of this chapter, it is worth noting that this scheme provides a certain level of privacy for vehicles, as follows:

- The identity of a vehicle can easily be anonymised by using a pseudonym instead of the real identity. Our scheme then provides a vehicle with anonymity with respect to all entities except for the reputation server.
- It is possible for the reputation server to issue multiple pseudonyms and public keys for a vehicle. This requires the the reputation server to pre-embed multiple private keys into the trusted hardware of the vehicle. This provides the vehicle with an extent of unlikability with respect to messages broadcast: other entities (except for the reputation server) cannot link messages broadcast under different pseudonyms.
- The reputation server does not learn messages from feedback, as only the hash value of a message is contained in the feedback (see Section 5.4.4).

5.6.3 Incentive to Participation

One issue is a vehicle's incentive for participating in the announcement scheme. This has two facets, as follows:

- Vehicles may lack incentive to broadcast a message to other vehicles. This directly reduces the utilisation of the announcement scheme.
- Vehicles may lack incentive to provide feedback. This results in degradation of the accuracy and robustness of the scheme, the latter arising since the probability that vehicles will report negative feedback upon being deceived by a false message is reduced.

One possible approach to increase vehicles' participation is to introduce some incentives. For example, the reputation server can introduce some policy that rewards a vehicle, with some points for example, if it constantly has a high reputation score or reports a large amount of feedback. Because the reputation server acts as the central authority and maintains all reputation and feedback information, it is easy for the reputation server to introduce such rewarding policy.

5.6.4 Bootstrapping

Another issue is bootstrapping a new vehicle. In our scheme, we specify that the initial reputation score of a new vehicle is zero. This configuration often causes a *bootstrapping* problem in a reputation system, where a newcomer has difficulty establishing its reputation. However, in our scheme, a new vehicle with zero initial reputation score is still able to establish its reputation. This is because, although messages broadcast by the new vehicle will not be considered as reliable, the receiving vehicles are still able to report feedback for these messages. Gradually, the new vehicle will be able to establish its own reputation.

It is also worth noting that assigning zero initial reputation score to a new vehicle, as described in our scheme, is conservative. The purpose of this is to discourage a vehicle with bad reputation from whitewashing its reputation by re-joining the system with a new identity. This is useful when the cost of re-joining the system with a new identity is negligible. However, in a VANET it is often difficult or costly for a vehicle to re-enter the system with a different identity. In this case a new vehicle could be initialised with a positive base reputation score. For example, the based reputation score can be set to the current average reputation score of all vehicles. This setting enables a new vehicle to establish its reputation more quickly.

5.6.5 Use of Data Mining Techniques

Data mining techniques could be used to further improve the accuracy and robustness of our scheme. In our scheme, all feedback is kept by the reputation server. This makes it possible for the reputation server to using data mining techniques to distinguish false feedback from honest feedback, and vehicles reporting false feedback from those reporting honest feedback. In addition, the richness of feedback may aid data mining techniques to improve the detection accuracy. For example:

- Feedback is linked to its reporting vehicle.
- Time information is contained in feedback.
- Feedback reported by different vehicles regarding the same message can be linked together (as they share the same $\mathsf{H}(m)$ entry in feedback tuple).

Such rich information may help data mining techniques to improve the detection accuracy.

5.6.6 The Impact of Configuration of Ψ_{nf}

Configuration of the public parameter Ψ_{nf} has a direct impact on the robustness of our scheme. A lower Ψ_{nf} means a better robustness against message fraud attack (see Claim 4). However, if it is too low, then adversaries may be able to deceive the reputation server into revoking a honest vehicle which complies with the scheme, by collusively conducting reputation manipulation attacks (see Equation 5.2). The best tradeoff is $\Psi_{nf} = |\mathcal{V}_a| + \Delta$, where $|\mathcal{V}_a|$ is the number of all internal adversaries and Δ is a safe margin (see Claim 4). However, if $|\mathcal{V}_a|$ cannot be accurately estimated, then a larger safe margin Δ is needed, in order to preventing honest vehicles from been revoked as a result of reputation manipulation attacks. But the price of a large safe margin Δ is that an adversary can conduct more message fraud attacks without getting revoked.

5.7 Simulation

In this section, we show some simulation results about the performance of our announcement scheme. This is evaluated from the following aspects:

- *Message drop rate*: the average rate that reliable messages are rejected by a receiving vehicle due to low reputation scores of broadcasting vehicles after time discount, as described in Section 5.4.3.

- *Temporary unavailable reputation server*: the average increase of message drop rate due to temporary unavailability of reputation server.
- *Temporary unavailable access points*: the average increase of message drop rate due to temporary unavailability of some access points.

5.7.1 Simulation Setup

We use an event-based real street map vehicular network simulator GrooveNet [77] and extend it to incorporate our scheme into the simulator. The road network used in simulations is an urban area of ten square kilometres chosen from the city of Pittsburgh, Pennsylvania, USA. This map data is extracted from the US Census Bureau's TIGER/Line database [15]. The communication range is 300 metres. The duration of each experiment is 30 minutes. The configurations of these simulations are in line with many studies in the literature, such as [121].

An experiment is configured and then conducted as follows:

- Access points are generated and populated randomly over the selected road network.
- Vehicles are generated, populated randomly and move in the selected road network. Their mobility models are as follows: a vehicle follows the vehicle in front, and a vehicle moves at the speed limit of a street when it is leading on the street. Their trip models are as follows: a vehicle randomly moves until it is ten kilometres from its starting point; the vehicle then takes the shortest path back to the starting point and starts again along a different path.
- Road events randomly occur in the road network throughout the experiment. The time that an event will last is set randomly from 1 to 120 seconds.
- Vehicles which are sufficiently close to an event can "experience" the event. The distance for a vehicle to experience an event is set randomly from 1 to 100 meters.
- A vehicle broadcasts a message regarding an event that it experiences, along with its latest reputation certificate.

- A message receiving vehicle determines whether it accepts the received message by evaluating the reputation of the broadcasting vehicle, as specified by Section 5.4.3. The reputation threshold parameter Ψ_{rs} is set conservatively to 0.8. The time discount parameter Ψ_{td} is set conservatively to one hour. Note that Ψ_{td} in a real-world implementation should be much longer than one hour, perhaps a few days or even longer. The purpose of setting it to one hour is to make the effect of the time discount function more visible during the experiments, and also to make it in line with 30 minutes of experiment time.

- A message receiving vehicle may report feedback if it later experiences the event described by the message within the time when the event still exists. The probability that the vehicle will report a feedback is set conservatively to 0.1.

- When a vehicle moves into communication range of an access point, it retrieves and then updates its latest reputation certificate, and reports all feedback that it has generated and not yet reported.

- The reputation server updates the reputation of each vehicle based on feedback received from all vehicles and generates a new reputation certificate accordingly, as specified by Sections 5.4.5 and 5.4.6. The time interval $\mathbb{T}$ is set to ten minutes. Note that $\mathbb{T}$ in a real-world implementation should be much longer than ten minutes, perhaps weeks or even longer. The purpose of setting such a short time interval $\mathbb{T}$ in the experiments is, again, to make it in line with 30 minutes of experiment time.

5.7.2 Simulation Results

Figure 5.1 shows the simulation results of message drop rate, with respect to different density of access points and vehicles. From Figure 5.1, the results of experiments show that the message drop rate decreases when the density of access points increases. A sharp decrease of message drop rate is seen when the number of access points is increased from one to two per square kilometre. Then the decrease of message drop rate becomes relatively slow when the number of access points is increased from two to five per square kilometre. This is natural since if there are more access points then vehicles tend to encounter them more often, and thus tend to retrieve the latest reputation certificate more frequently from the reputation server. As a result, vehicles tend to broadcast messages with more “fresh” reputation certificates, and the reputation scores will tend to be discounted less by the receiving vehicles using the time discount function TimeDiscount. This results in less rejection of reliable messages, and thus a decrease in the message drop rate.

The density of vehicles also impacts on the message drop rate. We observe a decrease of message drop rate when the density of vehicles increases. A modest but noticeable decrease is seen when the density of vehicles increases from 100 to 500 vehicles in the selected road network of ten square kilometres. This is reasonable because more feedback tends to be reported for a vehicle in a vehicle-dense road network. Consequently, it is more likely that feedback whose corresponding message tuple was broadcast within the past $\mathbb{T}$ time is reported for a vehicle, and thus a reputation certificate becomes available for the vehicle. This results in the reliable messages broadcast subsequently by the vehicle being accepted by the receiving vehicles, given that the broadcasting vehicle has a sufficiently high reputation score. Hence we observe a decrease in the message drop rate.

However, this observed difference in the message drop rate due to density of vehicles may not be as significant as shown in our experiments. This is because in our experiments, the time interval $\mathbb{T}$ is set to ten minutes, which is much shorter compared to a real-world implementation. This causes a reputation certificate to be less likely to be available to a vehicle, compared with an implementation with a much longer $\mathbb{T}$.

Figure 5.2 shows the simulation results of the increase of the message drop rate due to temporary unavailability of the reputation server with respect to various densities of access points. In these experiments, we deployed and populated 500 simulated vehicles. From Figure 5.2, the results of the experiments show that the increase of the message drop rate is approximately proportional to the unavailable time of the reputation server when the unavailable time is less then 12 minutes. When the unavailable time reaches 12 minutes, the message drop rate increases to 1. This is reasonable because in our experiments we set the time discount parameter Ψ_{td} to one hour and the reputation threshold parameter Ψ_{rs} to 0.8 (see Section 5.7.1). With these configurations, the time discounted reputation score of a vehicle cannot

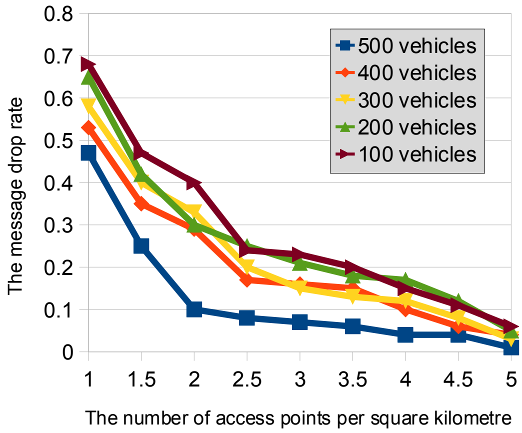

Figure 5.1: Message drop rate.

exceed the reputation threshold if the reputation certificate was obtained from the reputation server more than 12 minutes ago.

However, in a real-world implementation in which a much longer time discount parameter Ψ_{td} is expected, the rate of increase in the message drop rate due to temporary unavailabilities of the reputation server is expected to be significantly slower compared to the experiments. The minimum unavailable time of the reputation server that will result in a complete message drop is expected to extend long beyond 12 minutes.

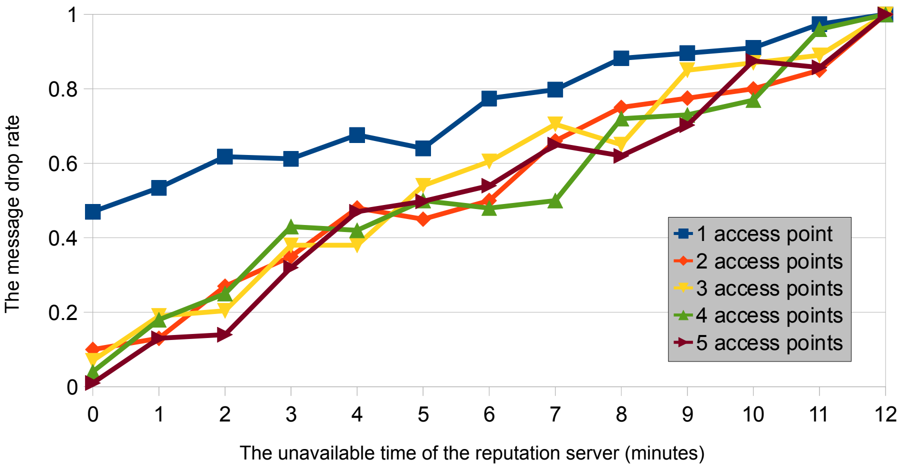

Figure 5.2: The increase of message drop rate due to temporary unavailabilities of the reputation server.

Figure 5.3 shows the simulation results of the increase of the message drop rate due to temporary unavailability of some access points. In these experiments, we deployed and populated 500 simulated vehicles and 50 access points. We examined the increase in the message drop rate caused by various proportions of access points being unavailable for different periods of time, from 5 to 25 minutes. From Figure 5.3, the results of the experiments show that the temporary unavailable access points slightly contribute to the increase in the message drop rate. This is reasonable, since when a vehicle comes across an unavailable access point, it can later retrieve its reputation certificate and report feedback via another working access point.

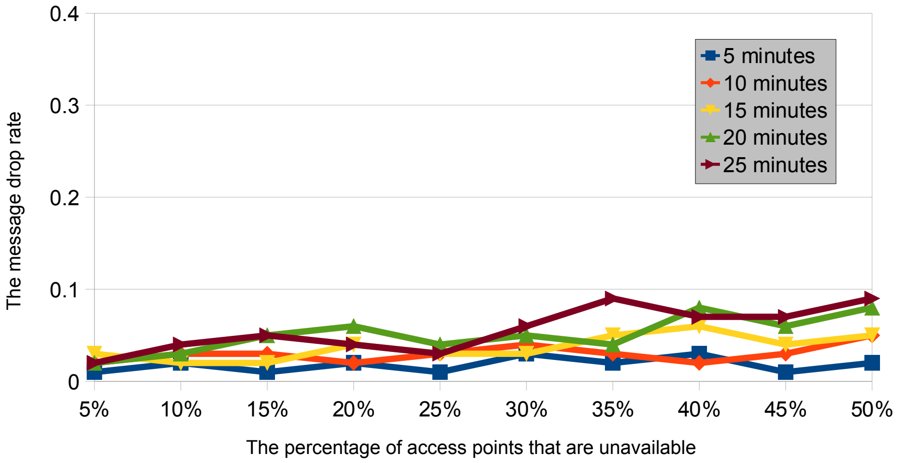

Figure 5.3: The increase of message drop rate due to temporary unavailabilities of some access points.

5.8 Fitting the Reputation System into our Framework

In this section, we use our proposed model and framework of Chapter 2, to decompose and discuss the reputation system used by our announcement scheme. We will first identify the components of the reputation system according to the model described in Section 2.3. We then decompose the environmental assumptions according to Section 2.5 and the design choices of the reputation system according to Section 2.6. Lastly, we perform a brief analysis according to the decomposition of the reputation system.

5.8.1 Components

The components of the reputation system are as follows:

- *Entities*: *Feedback providers*, *relying entities* and *targets* are the vehicles. *Processing unit* and *data storage unit* are the reputation server.
- *Attribute of interest*: reputation of vehicles.
- *Interactions*:
  - *Previous interactions*: previous messages received by vehicles.
  - *Future interactions*: future messages to be received by vehicles.
- *Data*:
  - *Feedback*: a vehicle's evaluation of the reliability of a received message.
  - *Advice*: the reputation of a vehicle.

5.8.2 Environmental Assumptions

The environmental assumptions of the reputation system are summarised as follows:

- *Target stability*: *unstable.* The reputation of a vehicle may change over time.
- *Capability of contributing entities*:
  - *Communication capability*:
    - *Data transmission capability*: *sufficient.* Each vehicle has sufficient capability to transmit feedback tuples to the reputation server via an access point. The reputation server has sufficient capability to transmit reputation certificates to vehicles via access points. Each vehicle has sufficient capability to broadcast a message tuple, which contains a reputation certificate, to its neighbouring vehicles.
    - *Connectivity*: *partially-connected.* Each vehicle only has a direct communication channel with its neighbouring vehicles and access points.

  - *Computational capability*: *sufficient.* We assume that the reputation server has sufficient computational capability for processing feedback tuples. We also assume that each vehicle has sufficient computational capability for generating feedback tuples and reputation certificates.
  - *Data storage capability*: *sufficient.* We assume that the reputation server has sufficient capability for storing feedback tuples.
  - *Feedback provision capability*: We assume that when a vehicle has its own experience about the event that a received message describes, it is able to judge the reliability of the message.

- *Motivation of contributing entities*:
  - Vehicles: *sufficient.* We assume that vehicles have sufficient motivation to report feedback.
  - The reputation server: *sufficient.* It is the administrative responsibility of the reputation server to provide data storage and feedback processing services.

- *Availability of contributing entities*:
  - Vehicles: *intermittent.* Vehicles are not constantly connected with the reputation server.
  - The reputation server: *constant.* The reputation server is assumed to be constantly available.

- *Trust relationships*:
  - Vehicles: These, as a whole, are trusted implicitly to provide sufficient feedback. This trust fits into the following categories:
    - *Global*: It is perceived system-wide.
    - *Static*: It does not change over the life span of the announcement scheme.
    - *Group*: It is not borne by individual vehicles, but all vehicles as a whole.
  - The reputation server: We assume that the reputation server is a trusted entity. This trust fits into the following categories:

        - *Global*: It is perceived system-wide.
        - *Static*: It does not change over the life span of the announcement scheme.
        - *Individual*: It is borne by the individual entity of the reputation server.

- *Adversarial models*. Possible adversarial models are as follows:
    - *Rationality*: both *irrational* and *rational*. Adversaries may or may not consider the cost of their attacks as the primary factor when they try to influence the reputation of a vehicle.
    - *Location*: both *outsiders* and *insiders*. Our robustness analysis focuses on both outsider (external) and insider (internal) adversaries.
    - *Strategy space*. The possible adversarial strategy space includes:
        - *Fabrication*: An adversary acting as a vehicle may report false feedback.
        - *Collusion*: Adversaries acting as vehicles may collude together to target a vehicle by conducting fabrication attacks.

5.8.3 Architectural Choices

The architectural choices of the reputation system are summarised as follows:

- *Role setting*: {*PS,FRT*}. The reputation server acts in the compound role of PS. The vehicles act in the compound role of FRT.
- *Centrality*: *centralised*. The role of data storage unit and processing unit is performed by a single entity.
- *Data flow*: both *receiver-active* and *receiver-passive*. The reputation server passively receives feedback reported by the vehicles. The vehicles passively receive reputation certificates, as a part of message tuples, from their neighbouring vehicles. The vehicles actively retrieve reputation certificates from the reputation server.

5.8.4 Data Processing Choices

The data processing choices of the reputation system are summarised as follows:

- *Representation of the evaluation data.* Representation of the evaluation data in feedback and advice is as follows:
  - In feedback: a *binary score.* Feedback is represented by a binary score.
  - In advice: a *continuous value.* The reputation of a vehicle is represented by a continuous value.
- *Aggregation algorithm*:
  - *Personalisation*: *non-personalised.* The reputation of a vehicle is the same for every vehicle.
  - *Collaboration awareness*: *collaboration-unaware.* The aggregation algorithm is run by the single entity of the reputation server.
  - *Manipulation resistance*: *manipulation-vulnerable.* The aggregation algorithm is not immune to adversarial manipulation of feedback.

5.8.5 Robustness Solutions

The robustness solutions of the reputation system are summarised as follows:

- *Centrality.* A trusted and centralised processing unit and data storage is adopted. This avoids an adversary acting as a data storage unit and a processing unit.
- *Proof of identity.* A vehicle is required to use the identity issued by the reputation server. This prevents an adversary from conducting sybil identity attacks.
- *Proof of occurrence of interaction.* Every feedback tuple contains a proof showing that the feedback reporting vehicle has received a message broadcast by the target vehicle.

- *Requiring time information.* Every feedback tuple contains time information about when the message is broadcast and received. Every reputation certificate contains time information about when the reputation certificate is generated.
- *Data filtering.* The reputation server only selects sufficiently "fresh" feedback to compute the reputation of vehicles.
- *Cryptography.* Cryptographic algorithms, such as hash function, MAC algorithm and digital signature algorithm, are used to protect feedback and reputation certificate, provide proof of identity and proof of occurrence of interaction.
- *Data mining techniques.* The reputation server may use data mining techniques to distinguish false feedback from honest feedback.
- *Incentive mechanism.* The reputation server may introduce some policy that rewards a vehicle if it constantly has a high reputation score or reports a large amount of feedback.

5.8.6 Discussion

A highlight of this reputation system is its centralised architecture. A centralised architecture seems counter-intuitive, since vehicles are usually highly mobile and have short-range wireless communication capability, which prevents vehicles from communicating with a centralised entity freely. This may explain why many existing approaches from the literature mainly focus on decentralised architectures (see Section 5.2).

However, we have designed a centralised architecture for this reputation system. This is because of the following:

- There usually already exists a centralised entity in a VANET, i.e. the regulatory authority of vehicles (see Section 5.3.1.1). This becomes a suitable candidate for the centralised entity of our reputation system.
- By taking advantage of the high mobility of vehicles and by introducing a new type of entity that constantly connects with the centralised entity, namely

access points, vehicles are able to frequently communicate (indirectly) with the centralised entity.

- A vehicle can retrieve, from the centralised entity, and then keep and broadcast its own reputation certificate (see Section 5.4.1). This enables its reputation information to be available to its neighbouring vehicles without assistance of the centralised entity.

By having a centralised architecture, it becomes easier for us to find many other robustness solutions (see Section 2.6.3.1).

Another highlight of this reputation system is its careful use of cryptography. The use of embedded trusted hardware in the OBU of vehicles, the abundant computational resources that vehicles have and the existence of a trusted authority, all provide a suitable environment for applying different cryptographic tools. In this reputation system, various cryptographic techniques are applied to achieve the following aspects of robustness:

- Data integrity of reputation certificates and feedback tuples is provided.
- Proof of identities of a broadcasting vehicle and a feedback reporting vehicle can be obtained from a reputation certificate and a feedback tuple, respectively.
- Proof of occurrence of interaction can be obtained. A vehicle can generate a valid feedback tuple only for a neighbouring vehicle upon receiving a message from it.

5.9 Extended and Simplified Variants

In this section, we discuss some possible approaches to extend our standard scheme, in order to increase its efficiency and flexibility. We will discuss how to facilitate multiple message broadcast to improve efficiency, and how to enable a richer reputation evaluation to improve flexibility.

We also demonstrate how the proposed scheme can be simplified in order to reduce some of the hardware requirements on vehicles. The price for such simplification is weakened robustness against internal adversaries. We will discuss a simplified variant which does not require vehicles to have a secure clock, and another simplified variant where vehicles do not require either a secure clock or trusted hardware.

5.9.1 Multiple Message Broadcast

In this section, we discuss how to facilitate multiple message broadcast in order to improve efficiency of our scheme. In the standard scheme, a message tuple contains only one message m. If vehicle V_b intends to broadcast n messages $(m_1, m_2, \ldots, m_n)$, it has to generate n message tuples. A receiving vehicle then has to evaluate the reliability of each individual message tuple. In this section we extend the standard scheme to facilitate multiple message broadcast.

Suppose vehicle V_b wants to broadcast a message vector $\mathcal{M} = (m_1, m_2, \ldots, m_n)$ containing n messages. We briefly describe the modification of the standard scheme as follows. During the message broadcast phase, V_b computes the hash value $\mathsf{H}(m_i)$ for every $m_i \in \mathcal{M}$. It then computes a hash value as follows:

$$h = \mathsf{H}(\mathsf{H}(m_1), \mathsf{H}(m_2), \ldots, \mathsf{H}(m_n)).$$

It then submits h to the trusted hardware in order to obtain a time-stamped signature $\theta = \mathsf{Sign}_2(t_b, h)_{sk_{V_b}}$. Then a message tuple $M = (\mathcal{M}, t_b, \theta, C)$ is formed. During the message reliability evaluation phase, V_r checks the validity of θ on the tuple $(t_b, \mathsf{H}(\mathsf{H}(m_1), \mathsf{H}(m_2), \ldots, \mathsf{H}(m_n)))$.

During the feedback reporting phase, V_r first generates a feedback rating vector $R = (fr_1, fr_2, \ldots, fr_n)$. If it provides a feedback rating for message m_i then it assigns $fr_i \in \{0, 1\}$; otherwise, it assigns fr_i with $\perp$, which denotes that it assigns no rating for message m_i. Then V_r submits the following to its trusted hardware:

$$(id_{V_b}, id_{V_r}, R, t_b, \mathsf{H}(m_1), \mathsf{H}(m_2), \ldots, \mathsf{H}(m_n), \theta).$$

This is to obtain a MAC value δ as follows:

$$\delta = \mathsf{MAC}(id_{V_b}, id_{V_r}, R, t_b, t_r, \mathsf{H}(m_1), \mathsf{H}(m_2), \ldots, \mathsf{H}(m_n), \theta)_{mk_{V_r}}.$$

Lastly V_r forms a feedback tuple as follows:

$$F = (id_{V_b}, id_{V_r}, R, t_b, t_r, \mathsf{H}(m_1), \mathsf{H}(m_2), \ldots, \mathsf{H}(m_n), \theta, \delta).$$

During the reputation update phase, if all verifications are successful, then the reputation server uses all feedback ratings $fr_i \neq \perp$ to update the reputation score of V_b.

By adopting this extension, a vehicle is able to simultaneously broadcast multiple messages. A receiving vehicle can also simultaneously verify the reliability of all messages in a message vector. The additional computational cost of this extension is negligible. Compared with the standard scheme, the broadcasting vehicle in this extended scheme only performs n extra hash operations in order to broadcast a message vector with n messages. A receiving vehicle also only performs n extra hash operations in order to verify the reliability of all messages in the message vector. This extension incurs some additional communication overhead when a receiving vehicle reports a feedback tuple to the reputation server. A feedback tuple has to include the feedback rating for every message $fr_1, fr_2, \ldots, fr_n$ and the hash value of every message $\mathsf{H}(m_1), \mathsf{H}(m_2), \ldots, \mathsf{H}(m_n)$. The length of a feedback tuple in this extension is longer than that of the standard scheme.

5.9.2 Multi-Level Reputation Evaluation

In this section, we discuss how to enable a richer reputation evaluation to improve flexibility of the scheme. In the standard scheme, a vehicle maintains only one threshold Ψ_{rs} to compare against the time discounted reputation score when making a decision as to whether a received message is reliable. However, some messages tend to be more critical than others. We may accept a critical message only if it is provided by a highly reputable vehicle. Similarly, we may accept an unimportant message if it is provided by a reasonably reputable vehicle.

The standard scheme can be easily extended to facilitate the above-mentioned multi-level reputation evaluation. The reputation server simply installs multiple thresholds $(\Psi_{rs}^1, \Psi_{rs}^2, \ldots, \Psi_{rs}^n)$ into each vehicle. These correspond to different levels of importance for different messages. When a vehicle receives a message, it just selects the corresponding threshold to compare against the time-discounted reputation score.

5.9.3 Simplified Variant 1

In our standard scheme, we assume that each vehicle is equipped with a secure clock. In this section, we relax this assumption: we assume that each vehicle has a clock that is not protected by the trusted hardware, i.e. the vehicle is able to modify the time information output by the clock. We outline this simplified variant by modifying the standard scheme as follows.

The vehicle clock regulation protocol VCRP and the public parameter Ψ_t are no longer required. Vehicles periodically synchronise their clocks with the reputation server. During the message broadcast phase, V_b retrieves the current time t_b from its clock, which is not protected by the trusted hardware. During the message reliability evaluation phase, Steps 1, 2 and 3b are removed. During the feedback reporting phase, V_r forms a feedback tuple $F = (id_{V_b}, id_{V_r}, t_b, fr, \mathsf{H}(m), \theta, \delta)$. Note that t_r in the standard scheme is removed in this variant. During the reputation update phase, Step 1a is removed.

This variant still provides *strong* robustness against external adversaries but it is less robust against internal adversaries. In this variant, the restriction removed from the standard scheme is that a vehicle is only able to generate valid feedback if it receives a message tuple before the time $t_b + \Psi_t$. Removing this restriction means that there is no time limitation on receiving a message tuple in order to generate valid feedback. Hence internal adversaries can engage in the following strategy. Once an internal adversary obtains a message from the target vehicle, it later forwards it to another internal adversary when they drive within wireless communication range of each other. This message tuple can be further propagated to other internal adversaries in the same manner. All internal adversaries receiving the message tuple, regardless of the receiving time, report feedback relating to the target vehicle.

Let $\mathcal{V}'_a$ denote the set of internal adversaries obtaining at least one message tuple generated by the target vehicle. The robustness of this variant becomes ($\Phi_{MF} = \frac{|\mathcal{V}'_a|}{p}, \Phi_{RM} = \frac{|\mathcal{V}'_a|}{|\mathcal{V}|}$), by the same argument in Section 5.5.2. It is straightforward to see that the size of $\mathcal{V}'_a$ is greater than or equal to that of $\mathcal{V}_a$. Hence the robustness of this variant may be less than that of the standard scheme. However, if the size of $\mathcal{V}'_a$ is still sufficiently small such that $\Phi_{MF} = \frac{|\mathcal{V}'_a|}{p}$ and $\Phi_{RM} = \frac{|\mathcal{V}'_a|}{|\mathcal{V}|}$ are still acceptable, then this variant can be an option for implementation.

5.9.4 Simplified Variant 2

In this variant, we remove the restriction from the standard scheme that each vehicle is equipped with trusted hardware and a secure clock. Instead, we assume that the onboard unit (OBU) of a vehicle is equipped with a computing device without trusted hardware storage and a non-protected clock. Note that in this variant, we do not assume that the OBU has a tamper-resistant device. Hence the vehicle itself is able to access its private key and MAC key, which is prevented in the standard scheme. We outline this variant by modifying the standard scheme as follows.

The vehicle clock regulation protocol VCRP and the public parameter Ψ_t are no longer required. Vehicles themselves periodically synchronise their clocks with the reputation server. During admission of a new vehicle, the reputation server sends its private key and MAC key over a secure channel to the vehicle. These are no longer kept confidential from the vehicle. During the message broadcast phase, V_b retrieves the current time t_b from the non-protected clock. Instead of the trusted hardware, V_b itself generates the signature $\theta = \mathsf{Sign}_2(t_b, \mathsf{H}(m))_{sk_{V_b}}$. During the message reliability evaluation phase, Steps 1, 2 and 3b are removed. During the feedback reporting phase, Step 2 is removed. During the reputation update phase, Step 1a is removed.

This variant also provide *strong* robustness against external adversaries. But it is less robust against internal adversaries than the standard scheme and Variant 1. In this variant, the restriction further removed from Variant 1 is that a vehicle is not able to access its private key and MAC key. Removing this restriction means that internal adversaries can engage in another strategy. An internal adversary distributes its MAC key to another colluding internal adversary. Consequently, one internal adversary is able to generate feedback on behalf of another colluding internal adversary. This provides internal adversaries with a convenient way of conducting a reputation manipulation attack. Given every internal adversary possesses the MAC key of every other internal adversary from a colluding group, once an internal adversary receives a message tuple from a target vehicle, it can generate and report feedback on behalf of every colluding internal adversary.

Let $\mathcal{V}_a^*$ denote the set of all internal adversaries. Then the robustness of this variant becomes $(\Phi_{MF} = \frac{|\mathcal{V}_a^*|}{p}, \Phi_{RM} = \frac{|\mathcal{V}_a^*|}{|\mathcal{V}|})$. It is easily seen that $|\mathcal{V}_a| \leq |\mathcal{V}_a'| \leq |\mathcal{V}_a^*|$. Hence the robustness of this variant may be less than that of the standard scheme and Variant 1. But if the size of $\mathcal{V}_a^*$ is relatively small compared with that of $\mathcal{V}$, then Variant 2 is also another option for implementation.

5.10 Conclusion and Future Work

In this chapter, we present a novel reputation-based announcement scheme for VANETs in order to evaluate message reliability. We have shown that our scheme is robust against external adversaries and robust against internal adversaries to a reasonably good level.

In future work, it might be of interest to investigate the following aspects:

- Although the current scheme already provides a certain level of privacy, it might be of interest to further enhance the privacy protection of the scheme.
- In the current scheme a vehicle and its human driver are represented by a single entity. It might be of interest to extend our scheme to reflect the potentially different reputations of human drivers and vehicles separately.
- In the current scheme a message broadcast by a vehicle is only utilised by its neighbouring vehicles. It might be of interest to extend the current scheme in such a way that a message can be utilised by vehicles in a greater area.
- In this chapter, we present a simple feedback aggregation algorithm based on binary feedback ratings. It might be of interest to investigate alternative approaches which allow continuous feedback ratings and thus provide richer results.
- It might be interest to investigate some concrete data mining techniques that can be used to further improve the robustness of the scheme.

5.11 A Summary of Notation

For ease of reference, we now provide a summary of all notation used in this chapter.

Table 5.1: Entity Related Notation

V	A vehicle.
V_b	A broadcasting vehicle.
V_r	A receiving vehicle.
id_V	The identifier of the vehicle V.
pk_S	The public key of the reputation server.
sk_S	The private key of the reputation server.
pk_V	The public key of vehicle V.
sk_V	The private key of vehicle V.
mk_V	The MAC key of vehicle V.

Table 5.2: Algorithms

Aggr	A reputation aggregation algorithm.
$\mathsf{TimeDiscount}$	A time discount function.
$(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$	A secure digital signature scheme.
H	A secure hash function.
MAC	A secure MAC algorithm.
$(\mathsf{VCRP_S}, \mathsf{VCRP_V})$	A vehicle clock regulation protocol.

Table 5.3: Public Parameters

Ψ_{td}	The public parameter determining how quickly the time discount function decreases.
Ψ_{rs}	The threshold used to determine whether or not a vehicle is reputable.
Ψ_t	The threshold used to determine whether or not a message tuple M is sufficiently fresh for feedback reporting.

Table 5.4: Time Related Notation

t	A time difference.
t_c	The time when a reputation certificate is generated by the reputation server.
t_b	The time when a message tuple is generated by vehicle V_b.
t_r	The time when a message tuple is received by vehicle V_r.
t_a	The time when the reputation server runs the reputation aggregation algorithm Aggr.
$\mathbb{T}$	A large time interval.

Table 5.5: Reputation Related Notation

m	A message.
$\mathcal{M} = (m_1, m_2, \ldots, m_n)$	A message vector.
rs_V	A reputation score of vehicle V.
fr	A feedback rating.
$R = (fr_1, fr_2, \ldots, fr_n)$	A feedback rating vector.
$\hat{r}^*_{V_i}$	An intermediate value computed by aggregating all feedback reported by vehicle V_i for a target vehicle.

Table 5.6: Cryptographic Digests

$\sigma = \mathsf{Sign}_1(id_{V_b}, pk_{V_b}, t_c, rs_{V_b})_{sk_S}$	A digital signature signed by the reputation server in order to produce a reputation certificate for vehicle V_b.
$\theta = \mathsf{Sign}_2(t_b, \mathsf{H}(m))_{sk_{V_b}}$	A digital signature signed by vehicle V_b in order to produce a message tuple.
$\delta = \mathsf{MAC}(id_{V_b}, id_{V_r}, fr, t_b, t_r, \mathsf{H}(m), \theta)_{mk_{V_r}}$	A MAC value generated by V_r in order to produce a feedback tuple.
$h = \mathsf{H}(\mathsf{H}(m_1), \mathsf{H}(m_2), \ldots, \mathsf{H}(m_n))$	A hash value of a collection of hash values.

Table 5.7: Notation for Compound Data

$C = (id_{V_b}, pk_{V_b}, t_c, rs_{V_b}, \sigma)$	A reputation certificate about vehicle V_b.
$M = (m, t_b, \theta, C)$	A message tuple generated by vehicle V_b.
$F = (id_{V_b}, id_{V_r}, fr, t_b, t_r, \mathsf{H}(m), \theta, \delta)$	A feedback tuple generated by vehicle V_r for V_b.
$\mathcal{F} = \{F : (id_{V_b} = id_V) \wedge (t_b \geq t_a - \mathbb{T})\}$	The set of all feedback reported for V whose corresponding message tuple was broadcast from time $\mathbb{T}$ in the past until now.
$\mathcal{F}_{V_i} = \{F : (id_{V_b} = id_V) \wedge (id_{V_r} = id_{V_i}) \wedge (t_b \geq t_a - \mathbb{T})\}$	The set of all feedback reported by V_i for V whose corresponding message tuple was broadcast from time $\mathbb{T}$ in the past until now.
$\mathcal{V} = \{V_i : (id_{V_b} = id_V) \wedge (id_{V_r} = id_{V_i})$ for some $F \in \mathcal{F}\}$.	The set of all vehicles that each has reported at least one feedback for V in the past $\mathbb{T}$ time.
$\mathcal{V}^- = \{V_i : (id_{V_b} = id_V) \wedge (id_{V_r} = id_{V_i}) \wedge (fr = 0)$ for some $F \in \mathcal{F}\}$	The set of vehicles that each has reported at least one negative feedback for V in the past $\mathbb{T}$ time.

Table 5.8: Notation for Robustness Analysis

Φ_{MF}	The maximum number of vehicles that an internal adversary can deceive during $\mathbb{T}$ period of time without its reputation score decreasing to 0 and then itself getting revoked.
Φ_{RM}	The maximum value by which the reputation score of a vehicle can be unfairly manipulated (increased or decreased) by adversaries.
p	The overall probability that vehicles will report negative feedback upon being deceived by a false message.
x	The number of internal adversaries that each obtains at least one valid message tuple from a target vehicle before the message expires for feedback reporting.
x'	The number of internal adversaries that obtain at least one message tuple generated by the target vehicle, directly from a target vehicle or indirectly from other internal adversaries.
x^*	The total number of internal adversaries.
Δ	A safe margin.

Chapter 6

Concluding Remarks

The main focus of this thesis has been on design and analysis of electronic feedback mechanisms:

- We provided an abstract model and a framework for feedback mechanisms.
- We proposed an online marketplace which can be used for trading arbitrable and replicable services such as computational resources and digital storage.
- We proposed a novel feedback mechanism for online marketplaces in which feedback directly impacts on the seller's payoff for the current transaction, unlike the reputation approach where feedback impacts on the seller's payoff for the future transactions.
- We proposed a novel announcement scheme for VANETs based on a reputation system that allows evaluation of message reliability

We list a few future research directions as follows:

- It might be worthwhile to further refine the main factors that influence the design of feedback mechanisms based on the categorisations that we provided in Chapter 2. This might provide a more systematic view about feedback mechanisms. This might also provide more concrete options about the design choices of feedback mechanisms and facilitate a more efficient and better design of feedback mechanisms.

- It might be interesting to propose a procedure for the systematic design of a feedback mechanism. By following this procedure, a designer might be able to quickly find a set of suitable options on the design choices of a feedback mechanism for a chosen application scenario. This might reduce the possibility that some important factors are overlooked or some suitable design options are ignored, and hence might improve the overall quality of feedback mechanism design.

- It might be interesting to apply our model and framework of Chapter 2 to decompose and analyse more existing feedback mechanisms. This might provide a comprehensive understanding about the feedback mechanisms proposed in the literature. A collection of the summary of existing feedback mechanisms that are decomposed and analysed according to our model and framework might contribute to the design, analysis and application of feedback mechanisms by providing a single point of reference and a systematic, structural and comparable analysis.

- It might be useful to provide some customised models and frameworks based on our generic one of Chapter 2 for some particular application scenarios. This enables the customised framework to provide a more detailed examination of the environmental constraints concerning of a particular application. This might also simplify and more suitably tailor the framework to the designated application scenario.

- It remains interesting challenges to design more concrete feedback mechanisms for different application environments. There are plenty of application scenarios for which efficient and robust feedback mechanisms that utilise indirect experience might be interesting to design.

In addition, we have identified some more specific future research problems in Sections 3.10, 4.8 and 5.10.

Last, but not least, the writing of this thesis has made the author aware of some more fundamental questions that arise from the employment of modern information technologies, such as how to refine the information flows within a community so that the utilisation of indirect experience can be enhanced for both individuals and the community while also satisfying requirements concerning information security and privacy.

The completion of this thesis is thus not an end, rather a launch point for future investigations.

Bibliography

[1] E. Adar and B.A. Huberman. Free riding on gnutella. *First Monday*, 5(10), 2000.

[2] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.

[3] Amazon, 2011. http://www.amazon.com.

[4] R. Aringhieri, E. Damiani, S.D.C. Di Vimercati, S. Paraboschi, and P. Samarati. Fuzzy techniques for trust and reputation management in anonymous peer-to-peer systems. *Journal of the American Society for Information Science and Technology*, 57(4):528–537, 2006.

[5] I. Ashlagi, D. Monderer, and M. Tennenholtz. Mediators in position auctions. *Games and Economic Behavior*, 67(1):2–21, 2009.

[6] S. Ba and P.A. Pavlou. Evidence of the effect of trust building technology in electronic markets: price premiums and buyer behavior. *MIS quarterly*, 26(3):243–268, 2002.

[7] S. Ba, A.B. Whinston, and H. Zhang. Building trust in online auction markets through an economic incentive mechanism. *Decision Support Systems*, 35(3):273–286, 2003.

[8] M. Babaioff, J. Chuang, and M. Feldman. *Incentives in peer-to-peer systems*, chapter 23, pages 593–612. Cambridge University Press, 2007.

[9] Y. Bakos and C. Dellarocas. Cooperation without enforcement? a comparative analysis of litigation and online reputation as quality assurance mechanisms.

In *Proceedings of the 23rd International Conference on Information Systems*, pages 127–142, 2002.

[10] R. Beck, M. Schwind, and O. Hinz. Grid economics in departmentalized enterprises. *Journal of Grid Computing*, 6(3):277–290, 2008.

[11] S. Braynov and T. Sandholm. Trust revelation in multiagent interaction. In *Proceedings of CHI*, volume 2, pages 57–60, 2002.

[12] J.S. Breese, D. Heckerman, C. Kadie, et al. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pages 43–52, 1998.

[13] J. Broberg, S. Venugopal, and R. Buyya. Market-oriented grids and utility computing: The state-of-the-art and future directions. *Journal of Grid Computing*, 6(3):255–276, 2008.

[14] Better Business Bureau, 2011. http://www.bbb.org.

[15] US Census Bureau, 2011. http://www.census.gov/geo/www/tiger.

[16] C2CC. *The Car-to-Car Communication Consortium*, 2011. http://www.car-to-car.org.

[17] G. Calandriello, P. Papadimitratos, J. Hubaux, and A. Lioy. Efficient and robust pseudonymous authentication in VANET. In *Proceedings of the 4th ACM International Workshop on Vehicular Ad Hoc Networks*, pages 19–28. ACM, 2007.

[18] E. Carrara and G. Hogben. *Reputation-based systems: a security analysis.* ENISA, 2007. Position paper.

[19] D.W. Chadwick. Operational models for reputation servers. In *Proceedings of the iTrust*, pages 108–115, 2005.

[20] K. Chen, T. Hogg, and N. Wozny. Experimental study of market reputation mechanisms. In *Proceedings of the ACM Conference on Electronic Commerce*, pages 234–235, 2004.

[21] L. Chen, S.L. Ng, and G. Wang. Threshold anonymous announcement in vanets. *IEEE Journal on Selected Areas in Communications*, 29(3):605–615, 2011.

[22] A. Cheng and E. Friedman. Sybilproof reputation mechanisms. In *Proceedings of the 2005 ACM SIGCOMM Workshop on Economics of Peer-to-peer Systems*, page 132. ACM, 2005.

[23] A. Cheng and E. Friedman. Manipulability of PageRank under sybil strategies. In *Proceedings of the First Workshop on the Economics of Networked Systems (NetEcon06)*, 2006.

[24] W.W. Cohen, R.E. Schapire, and Y. Singer. Learning to order things. *Journal of Artificial Intelligence Research*, 10:243–270, 1999.

[25] C. Courcoubetis, M. Dramitinos, T. Rayna, S. Soursos, and G.D. Stamoulis. Market mechanisms for trading grid resources. In *GECON*, volume 5206 of *Lecture Notes in Computer Science*, pages 58–72. Springer, 2008.

[26] E. Damiani, S. Paraboschi, P. Samarati, and F. Violante. A reputation-based approach for choosing reliable resources in peer-to-peer networks. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pages 207–216. ACM New York, NY, USA, 2002.

[27] R.K. Dash, S.D. Ramchurn, and N.R. Jennings. Trust-based mechanism design. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems.* IEEE Computer Society, 2004.

[28] N. Daswani, P. Golle, S. Marti, H. Garcia-Molina, and D. Boneh. Evaluating reputation systems for document authenticity. Technical report, Computer Science Department, Stanford University, 2003.

[29] V. Daza, J. Domingo-Ferrer, F. Sebé, and A. Viejo. Trustworthy privacy-preserving car generated announcements in vehicular ad hoc networks. *IEEE Transaction on Vehicular Technology*, 58(4):1876 – 1886, 2009.

[30] J. Delgado and N. Ishii. Memory-based weighted-majority prediction for recommender systems. In *Proceedings of ACM SIGIR99 Workshop on Recommender Systems: Algorithms and Evaluation*, 1999.

[31] C. Dellarocas. Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior. In *Proceedings of the 2nd ACM Conference on Electronic Commerce*, pages 150–157. ACM New York, NY, USA, 2000.

[32] C. Dellarocas. The design of reliable trust management systems for electronic trading communities. *Arbeitspapier, MIT*, 2002.

[33] C. Dellarocas. Efficiency through feedback-contingent fees and rewards in auction marketplaces with adverse selection and moral hazard. In *Proceedings of the 4th ACM Conference on Electronic Commerce*, pages 11–18. ACM New York, NY, USA, 2003.

[34] C. Dellarocas. The digitization of word of mouth: Promise and challenges of online feedback mechanisms. *Management Science*, 49(10):1407–1424, 2003.

[35] C. Dellarocas, F. Dini, and G. Spagnolo. Designing reputation mechanisms. In N. Dimitri, G. Piga, and G. Spagnolo, editors, *Handbook of Procurement*, chapter 18, pages 446–482. Cambridge University Press, 2006.

[36] C. Desrosiers and G. Karypis. A comprehensive survey of neighborhood-based recommendation methods. *Handbook on Recommender Systems, Springer*, 2009.

[37] J. Domingo-Ferrer and Q. Wu. Safety and privacy in vehicular communications. In *Privacy in Location-based Applications*, volume 5599 of *Lecture Notes in Computer Science*, pages 173–189. Springer, 2009.

[38] F. Dötzer. Privacy issues in vehicular ad hoc networks. In *Proceedings of Privacy Enhancing Technologies*, volume 3856 of *Lecture Notes in Computer Science*, pages 197–209. Springer, 2005.

[39] F. Dötzer, L. Fischer, and P. Magiera. VARS: A vehicle ad hoc network reputation system. In *Proceedings of the Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks*, volume 1, pages 454–456, 2005.

[40] J.R. Douceur. The sybil attack. In *Proceedings for the 1st International Workshop on Peer-to-Peer Systems (IPTPS '02)*, pages 251–260, 2002.

[41] eBay, 2011. http://www.ebay.com.

[42] Epinions, 2011. http://www.epinions.com.

[43] T. Eymann, S. König, and R. Matros. A framework for trust and reputation in grid environments. *Journal of Grid Computing*, 6(3):225–237, 2008.

[44] Y. Freund, R. Iyer, R.E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *The Journal of Machine Learning Research*, 4:933–969, 2003.

[45] E. Friedman, P. Resnick, and R. Sami. Manipulation-resistant reputation systems. In N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, editors, *Algorithmic Game Theory*, chapter 27, pages 677–697. Cambridge University Press, 2007.

[46] Eric J. Friedman and Paul Resnick. The social costs of cheap pseudonyms. *Journal of Economics and Management Strategy*, 10:173–199, 2001.

[47] D. Goldberg, D. Nichols, B.M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):70, 1992.

[48] Philippe Golle, Daniel H. Greene, and Jessica Staddon. Detecting and correcting malicious data in VANETs. In *Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks*, pages 29–37. ACM, 2004.

[49] Google, 2011. http://google.com.

[50] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *Proceedings of the 13th International Conference on World Wide Web*, pages 403–412. ACM, 2004.

[51] R.H. Guttman, A.G. Moukas, and P. Maes. Agent-mediated electronic commerce: A survey. *The Knowledge Engineering Review*, 13(02):147–159, 2001.

[52] K. Hoffman, D. Zage, and C. Nita-Rotaru. A survey of attack and defense techniques for reputation systems. *ACM Computing Surveys (CSUR)*, 42(1):1–31, 2009.

[53] J. Hubaux, S. Capkun, and J. Luo. The security and privacy of smart vehicles. *IEEE Security & Privacy*, 2(3):49–55, 2004.

[54] J. Hubaux, P. Papadimitratos, and M. Raya. Securing vehicular communications. *IEEE Wireless Communications Magazine*, 13(5):8–15, 2006.

[55] The Insight Research Corporation. *Grid Computing: A Vertical Market Perspective 2006—2011*, 2006.

[56] A. Jøsang. The beta reputation system. In *Proceedings of the 15th Bled Conference on Electronic Commerce*, Bled, Slovenia, 2002.

[57] A. Jøsang and J. Golbeck. Challenges for robust of trust and reputation systems. In *Proceedings of the 5th International Workshop on Security and Trust Management, STM*, 2009.

[58] A. Jøsang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(2):618–644, 2007.

[59] R. Jurca and B. Faltings. An incentive compatible reputation mechanism. In *Proceedings of the IEEE Conference on E-Commerce (CEC)*. Published by the IEEE Computer Society, 2003.

[60] R. Jurca and B. Faltings. CONFESS — an incentive compatible reputation mechanism for the online hotel booking industry. In *Proceedings of the IEEE Conference on E-Commerce*. IEEE Computer Society, 2004.

[61] R. Jurca and B. Faltings. Obtaining reliable feedback for sanctioning reputation mechanisms. *Journal of Artificial Intelligence Research*, 29(1):391–419, 2007.

[62] S.D. Kamvar, M.T. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *WWW*, pages 640–651, 2003.

[63] R. Kerr and R. Cohen. Smart cheaters do prosper: Defeating trust and reputation systems. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 993–1000, 2009.

[64] R. Kerr and R. Cohen. Trust as a tradable commodity: A foundation for safe electronic marketplaces. *Computational Intelligence*, 26(2):160–182, 2010.

[65] T. H. Kim, A. Studer, R. Dubey, X. Zhang, A. Perrig, F. Bai, B. Bellur, and A. Iyer. VANET alert endorsement using multi-source filters. In *Proceedings of the Seventh ACM International Workshop on VehiculAr InterNETworking*, VANET '10, pages 51–60, New York, NY, USA, 2010. ACM.

[66] J.M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999.

[67] P. Kollock. The production of trust in online markets. *Advances in Group Processes*, 16(1):99–123, 1999.

[68] G. Kounga, T. Walter, and S. Lachmund. Proving reliability of anonymous information in VANETs. *Transactions on Vehicular Technology*, 58(6):2977–2989, 2009.

[69] R. Kroh, A. Kung, and F. Kargl. VANETs security requirements. Technical report, Secure Vehicle Communication (SeVeCom), 2006.

[70] R. Levien. *Attack Resistant Trust Metrics*. PhD thesis, University of California at Berkeley, 2003.

[71] F. Li and J. Wu. Mobility reduces uncertainty in MANETs. In *IEEE INFOCOM 2007. 26th IEEE International Conference on Computer Communications*, pages 1946–1954, 2007.

[72] Q. Li and K.M. Martin. A secure marketplace for online services that induces good conduct. In *Short Paper Proceedings of the IFIP WG 11.11 International Conference on Trust Management*, 2010. ISSN:2079-2263.

[73] Q. Li and K.M. Martin. A viable grid marketplace. In *Proceedings of the 2010 International Symposium on Parallel and Distributed Processing with Applications (ISPA2010)*, pages 427 –432, 2010.

[74] Q. Li, K.M. Martin, and J. Zhang. Design of a multiagent-based e-marketplace to secure service trading on the Internet. In *Proceedings of the 13th International Conference on Electronic Commerce (ICEC 2011)*, 2011.

[75] X. Lin, X. Sun, and P. Ho. GSIS: secure vehicular communications with privacy preserving. In *IEEE Transactions on Vechicular Technology*, volume 56, pages 3442–3456, 2006.

[76] Z. Malik and A. Bouguettaya. Reputation bootstrapping for trust establishment among web services. *IEEE Internet Computing*, 13:40–47, 2009.

[77] R. Mangharam, D. Weller, R. Rajkumar, P. Mudalige, and F. Bai. Groovenet: A hybrid simulator for vehicle-to-vehicle networks. In *Mobile and Ubiquitous Systems: Networking & Services, 2006 Third Annual International Conference on*, pages 1–8. IEEE, 2006.

[78] N.B. Margolin, B.N. Levine, N.B. Margolin, and B.N. Levine. Quantifying and discouraging sybil attacks. *Computer Science Technical Report*, 67, 2005.

[79] S. Marti and H. Garcia-Molina. Economic design of reputation systems. Technical report, Stanford University, 2004.

[80] S. Marti and H. Garcia-Molina. Taxonomy of trust: categorizing p2p reputation systems. *Computer Networks*, 50(4):472–484, 2006.

[81] S. Micali. Simple and fast optimistic protocols for fair electronic exchange. In *Proceedings of the Twenty-second Annual Symposium on Principles of Distributed Computing*, pages 12–19. ACM, 2003.

[82] B.N. Miller, I. Albert, S.K. Lam, J.A. Konstan, and J. Riedl. MovieLens unplugged: experiences with an occasionally connected recommender system. In *Proceedings of the 8th International Conference on Intelligent User Interfaces*, pages 263–266. ACM, 2003.

[83] N. Miller, P. Resnick, and R. Zeckhauser. Eliciting informative feedback: The peer-prediction method. *Management Science*, 51(9):1359–1373, 2005.

[84] K.L. Mills and C. Dabrowski. Can economics-based resource allocation prove effective in a computation marketplace? *Journal of Grid Computing*, 6(3):291–311, 2008.

[85] Z. Milosevic, A. Jøsang, T. Dimitrakos, and M.A. Patton. Discretionary enforcement of electronic contracts. In *Proceedings of the Sixth International Enterprise Distributed Object Computing Conference*, pages 39–50, 2002.

[86] U.F. Minhas, J. Zhang, T. Tran, and R. Cohen. Towards expanded trust management for agents in vehicular ad hoc networks. *International Journal of Computational Intelligence Theory and Practice*, pages 3–15, 2010.

[87] D. Monderer and M. Tennenholtz. Strong mediated equilibrium. *Artificial Intelligence*, 173(1):180–195, 2009.

[88] M. Montaner, B. López, and J.L.D.L. Rosa. A taxonomy of recommender agents on the internet. *Artificial Intelligence Review*, 19(4):285–330, 2003.

[89] A. Nakamura and N. Abe. Collaborative filtering using weighted majority prediction algorithms. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 395–403. Morgan Kaufmann Publishers Inc., 1998.

[90] A. Nandi, T.W. Ngan, A. Singh, P. Druschel, and D. Wallach. Scrivener: providing incentives in cooperative content distribution systems. *Middleware 2005*, pages 270–291, 2005.

[91] D. Neumann, J. Stößer, C. Weinhardt, and J. Nimis. A framework for commercial grids - economic and technical challenges. *Journal of Grid Computing*, 6(3):325–347, 2008.

[92] J. Nimis, A. Anandasivam, N. Borissov, G. Smith, D. Neumann, N. Wirström, E. Rosenberg, and M. Villa. SORMA — business cases for an open grid market: concept and implementation. In *GECON*, volume 5206 of *Lecture Notes in Computer Science*, pages 173–184. Springer, 2008.

[93] Z. Noorian and M. Ulieru. The state of the art in trust and reputation systems: a framework for comparison. *Journal of Theoretical and Applied Electronic Commerce Research*, 5:97–117, 2010.

[94] M.J. Osborne. *An Introduction to Game Theory.* Oxford University Press, 2004.

[95] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: bringing order to the web. Technical report, Stanford University, 1998.

[96] P. Papadimitratos, L. Buttyan, J. Hubaux, F. Kargl, A. Kung, and M. Raya. Architecture for secure and private vehicular communications. In *Proceedings of the 7th International Conference on ITS Telecommunications (ITST)*, 2007.

[97] T.G. Papaioannou and G.D. Stamoulis. A mechanism that provides incentives for truthful feedback in peer-to-peer systems. In *Proceedings of IEEE/ACM CCGRID 2005 (Workshop on Global P2P Computing)*, 2005.

[98] B. Parno and A. Perrig. Challenges in securing vehicular networks. In *Proceedings of Workshop on Hot Topics in Networks (HotNets-IV)*, 2005.

[99] A. Patwardhan, A. Joshi, T. Finin, and Y. Yesha. A data intensive reputation management scheme for vehicular ad hoc networks. In *Proceedings of the 3rd Annual International Conference on Mobile and Ubiquitous Systems*, pages 1–8, 2006.

[100] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content-addressable network. In *Proceedings of the 2001 Conference on*

Applications, Technologies, Architectures, and Protocols for Computer Communications, pages 161–172. ACM, 2001.

[101] M. Raya, A. Aziz, and J. Hubaux. Efficient secure aggregation in VANETs. In *Proceedings of the 3rd International Workshop on Vehicular Ad Hoc Networks*, pages 67–75. ACM, 2006.

[102] M. Raya and J. Hubaux. The security of vehicular ad hoc networks. In *Proceeding of SASN*, pages 11–21. ACM, 2005.

[103] M. Raya and J. Hubaux. Securing vehicular ad hoc networks. *Journal of Computer Security*, 15(1):39–68, 2007.

[104] M. Raya, P. Papadimitratos, V.D. Gligor, and J. Hubaux. On data-centric trust establishment in ephemeral ad hoc networks. In *INFOCOM*, pages 1238–1246. IEEE, 2008.

[105] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. GroupLens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, pages 175–186. ACM, 1994.

[106] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman. Reputation systems. *Communications of the ACM*, 43(12):45–48, 2000.

[107] P. Resnick and R. Sami. Sybilproof transitive trust protocols. In *Proceedings of the Tenth ACM Conference on Electronic Commerce*, pages 345–354. ACM, 2009.

[108] P. Resnick and H.R. Varian. Recommender systems. *Communications of the ACM*, 40(3):58, 1997.

[109] P. Resnick and R. Zeckhauser. Trust among strangers in internet transactions: empirical analysis of eBay's reputation systems. In *The Economics of the Internet and E-Commerce*, volume 11 of *Advances in Applied Microeconomics*, pages 127–157. Elsevier Science, 2002.

[110] P. Resnick, R. Zeckhauser, J. Swanson, and K. Lockwood. The value of reputation on eBay: a controlled experiment. *Experimental Economics*, 9:79–101, 2006.

[111] S. Ruohomaa, L. Kutvonen, and E. Koutrouli. Reputation management survey. In *ARES*, pages 103–111. IEEE Computer Society, 2007.

[112] R.K. Schmidt, T. Leinmüller, E. Schoch, A. Held, and G. Schäfer. Vehicle behavior analysis to enhance security in VANETs. In *Proceedings of 4th Workshop on Vehicle to Vehicle Communications (V2VCOM 2008)*. Citeseer, 2008.

[113] Slashdot, 2012. http://www.slashdot.org.

[114] J.D. Sonnek and J.B. Weissman. A quantitative comparison of reputation systems in the grid. In *GRID*, pages 242–249. IEEE, 2005.

[115] M. Srivatsa, L. Xiong, and L. Liu. TrustGuard: countering vulnerabilities in reputation management for decentralized overlay networks. In *Proceedings of the 14th International Conference on World Wide Web*, pages 422–431. ACM, 2005.

[116] T. Tran and R. Cohen. A learning algorithm for buying and selling agents in electronic marketplaces. In *Proceedings of the 15th Conference of the Canadian Society for Computational Studies of Intelligence on Advances in Artificial Intelligence*, pages 31–43. Springer, 2002.

[117] T. Tran and R. Cohen. Improving user satisfaction in agent-based electronic marketplaces by reputation modelling and adjustable product quality. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, volume 2, pages 828–835, Los Alamitos, CA, USA, 2004. IEEE Computer Society.

[118] K. Walsh and E.G. Sirer. Experience with an object reputation system for peer-to-peer filesharing. In *Proceedings of the Symposium on Networked System Design and Implementation (NSDI)*, 2006.

[119] P. Wex, J. Breuer, A. Held, T. Leinmüller, and L. Delgrossi. Trust issues for vehicular ad hoc networks. In *Vehicular Technology Conference (VTC Spring)*, pages 2800–2804. IEEE, 2008.

[120] Wikipedia, 2011. http://www.wikipedia.org.

[121] Q. Wu, J. Domingo-Ferrer, and U. González-Nicolás. Balanced trustworthiness, safety, and privacy in vehicle-to-vehicle communications. *Vehicular Technology, IEEE Transactions on*, 59(2):559–573, 2010.

[122] L. Xiong and L. Liu. PeerTrust: supporting reputation-based trust for peer-to-peer electronic communities. *IEEE Transaction on Knowledge and Data Engineering*, pages 843–857, 2004.

[123] B. Yu. *Emergence and evolution of agent-based referral networks*. PhD thesis, North Carolina State University, 2001.

[124] B. Yu and M.P. Singh. A multiagent referral system for expertise location. In *Working Notes of the AAAI Workshop on Intelligent Information Systems*, pages 66–69, 1999.

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

[126] B. Yu and M.P. Singh. Detecting deception in reputation management. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 73–80. ACM, 2003.

[127] H. Yu, P.B. Gibbons, M. Kaminsky, and F. Xiao. Sybillimit: A near-optimal social network defense against sybil attacks. In *IEEE Symposium on Security and Privacy*, pages 3–17. Citeseer, 2008.

[128] H. Yu, M. Kaminsky, P.B. Gibbons, and A. Flaxman. Sybilguard: defending against sybil attacks via social networks. In *Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, page 278. ACM, 2006.

[129] H. Yu, C. Shi, M. Kaminsky, P.B. Gibbons, and F. Xiao. Dsybil: Optimal sybil-resistance for recommendation systems. In *IEEE S&P*, 2009.

[130] J. Zhang. *Promoting Honesty in Electronic Marketplaces: Combining Trust Modeling and Incentive Mechanism Design*. PhD thesis, University of Waterloo, 2009.

[131] J. Zhou. *Non-repudiation in Electronic Commerce*. Artech House, 2001.

[132] P. Zimmermann. *The Official PGP User's Guide*. MIT press Boston, 1995.