Privacy Preserving Reputation Management in Social Networks

Omar Hasan and Lionel Brunie

Abstract Reputation management is a powerful security tool that helps establish the trustworthiness of users in online applications. One of the most successful use of reputation systems is on e-commerce web sites such as eBay.com and Amazon.com, which use reputation systems to root out fraudulent sellers. Reputation systems can also play an important role in social networks to enforce various security requirements. For example, a reputation system can help filter fake user profiles. However, a major challenge in developing reputation systems for social networks is that users often hesitate to publicly rate fellow users or friends due to the fear of retaliation. This trend prevents a reputation system from accurately computing reputation scores. Privacy preserving reputation systems hide the individual ratings of users about others and only reveal the aggregated community reputation score thus allowing users to rate without the fear of retaliation. In this chapter, we describe privacy preserving reputation management in social networks and the associated challenges. In particular we will look at privacy preserving reputation management in decentralized social networks, where there is no central authority or trusted third parties, thus making the task of preserving privacy particularly challenging.

1 Social Networks and Relationships

We take a look at the key social concepts of social networks and social relationships. In particular, we discuss the nature of social relationships by identifying the various attributes that characterize them.

Omar Hasan and Lionel Brunie
University of Lyon, CNRS, INSA-Lyon, LIRIS, UMR5205, F-69621, France
e-mail: {omar.hasan, lionel.brunie}@insa-lyon.fr

1.1 Social Networks

A social network is a composition of nodes and the relationships between them. The nodes in a social network may be individuals or collectives of individuals. The relationships between nodes are founded on human ties such as friendship, membership in the same family or organization, mutual interests, common beliefs, trade, exchange of knowledge, geographical proximity, etc.

1.2 Characteristics of Social Relationships

The most commonly discussed characteristics of social relationships include *roles*, *valence*, *provenance*, *history*, and *strength* [43].

Roles. A social relationship is defined by the roles that are associated with it. For example, the roles of employer and employee define the relationship of employer–employee in a professional setting. The same pair of nodes may take on different roles in a parallel relationship. For example, a employer–employee relationship may be complemented by a neighbor–neighbor relationship.

Valence. A social relationship can have positive, negative, or neutral sentiments associated with it. For example, an individual may like, dislike, or be apathetic towards another individual.

Provenance. Some attributes of a social relationship may be asymmetric, that is, perceived differently by the individual participants of the relationship. For example, a sentiment of *like* from one node may not be reciprocated by the other node in the relationship.

Relationship history. Social relationships have a temporal dimension. A social relationship may evolve with time through interactions or the absence thereof. The history of a social relationship can be considered as an indicator of the current and future status of the relationship. For example, a long positive relationship in the past is likely to be followed by a positive relationship in the present and in the near future.

Strength. Strength of a tie (or social relationship) is a quantifiable property that characterizes the link between two nodes [50]. The notion of tie strength was first introduced by sociologist Mark Granovetter in his influential paper "The strength of weak ties" [27] published in 1973. He defined the strength of a tie as a "combination of the amount of time, the emotional intensity, the intimacy (mutual confiding), and the reciprocal services which characterize the tie" [27]. The strength of a social relationship is a complex construct, which is itself composed of several properties of social relationships. We discuss the strength of social relationships in detail in the following section.

1.3 Strength of Social Relationships

Granovetter proposed four dimensions of tie strength: *amount of time*, *intimacy*, *intensity*, and *reciprocal services* [27] [25]. A number of researchers (including Burt [9], Wellman and Wortley [58], Lin et al. [40], Marsden [41]) have since studied the dimensions of tie strength and have refined and expanded the original list of four. The existing literature suggests seven dimensions of tie strength: *intensity*, *intimacy*, *duration*, *reciprocal services*, *structural factors*, *emotional support*, and *social distance* [25].

In a study on predicting tie strength between individuals based on their exchanges on social networking sites [25], Gilbert and Karahalios have identified a number of indicators that predict tie strength belonging to each of the seven dimensions. In a study with similar goals, Petroczi et al. [50] have developed a set of questions that they pose the members of a virtual social network in order to establish the strength of ties between them. In the following list, we discuss each of the seven dimensions of tie strength as well as some associated indicators and questions that yield tie strength.

Intensity. The indicators of the intensity of a tie strength include the *frequency of contact* and the *amount of information exchanged* between two nodes.

Homans presented the argument in his 1950 book “The Human Group” that “the more frequently the persons interact with one another, the stronger their sentiments of friendship for one another are apt to be” [34] [27].

Gilbert and Karahalios [25] use the amount of information exchanged (for example, the number of words and messages exchanged) on a social networking site as an indicator of the intensity of the tie strength between individuals.

Intimacy. Mutual confiding (or trust) is an indicator of the intimacy and the strength of a social tie [27] [41] [50]. Sociologist Diego Gambetta [24] characterizes trust as contextual and quantifiable as subjective probability. Petroczi et al. [50] ask the members of an online discussion forum the following question in order to determine the trust and consequently the tie strength between them: “Which participants do you trust (for example they know your real name, email address, password to your introduction sheet)?”.

Gilbert and Karahalios [25] use the variable “Relationship status”, with the possible values of *single*, *in relationship*, *engaged*, and *married*, as an indicator of the intimacy of two individuals. Other variables that they use as indicators of intimacy include “Distance between hometowns”, “Appearances together in photos”, and “Days since last communication”.

Duration. The duration or the span of the relationship is considered as an indicator of the strength of the relationship.

Gilbert and Karahalios [25] use the variable "Days since first communication" on social networking sites as a proxy for the length of the relationship between two individuals.

Reciprocal services. A social relationship is stronger if it is reciprocated by both participants. For example, a sentiment of *like* shared by both nodes would result in a strong social relationship.

Gilbert and Karahalios [25] use *the number of links and applications mutually shared* between friends as variables quantifying reciprocal services on social networking sites.

Structural factors. Ronald Burt proposed that structural factors shape tie strength, factors like network topology and informal social circles [9] [25].

A structural factor that Gilbert and Karahalios [25] use to predict tie strength is the "Number of mutual friends". They also use structural factors such as membership in common interests groups, and association with the same institutions, organizations, or geographical locations (for example, graduation from the same university, employment in the same company, or residence in a common city, etc.).

Emotional support. Wellman and Wortley argue that providing emotional support, such as offering advice on family problems, indicates a stronger tie [58] [25].

To determine the emotional support between the members of a virtual social network, Petroczi et al. [50] ask the members which other members they have requested or they feel they could request for a favor or help. Gilbert and Karahalios [25] monitor emotion words (as identified by the Linguistic Inquiry and Word Count (LIWC) dictionary [49], for example, *birthday*, *congrats*, *sweetheart*) exchanged between the members of a social networking site as indicators of emotional support.

Social distance. Lin et al. show that social distance, embodied by factors such as socioeconomic status, education level, political affiliation, race and gender, influences tie strength [40] [25].

Gilbert and Karahalios [25] measure social distance by considering parity in age, occupation, education, political, and religious views of the individuals.

2 Trust

Trust is an important indicator of the strength of a social relationship. It inherently takes into account a number of other aspects of a social relationship.

2.1 Modeling Trust

Sociologist Diego Gambetta [24] proposes the following definition of trust:

> Trust (or, symmetrically, distrust) is a particular level of the subjective probability with which an agent assesses that another agent or group of agents will perform a particular action, both before he can monitor such action (or independently of his capacity ever to be able to monitor it) and in a context in which it affects his own action.

This is one of the seminal definitions that describe trust as a quantifiable construct. Gambetta observes that trust is an agent's degree of belief (the level of subjective probability) that another entity will perform an expected action. An additional important aspect of this definition is the recognition that trust is contextual.

The advantage of Gambetta's model of trust is its quantification of trust as subjective probability. It allows trust to be modeled as a mathematical construct and to be manipulated using the wide range of tools available in probability theory. Moreover, trust modeled with subjective probability is more intuitive than trust modeled with other theories such as subjective logic and fuzzy logic.

2.2 Characteristics of Trust

From Gambetta's definition, we can infer that trust has the following characteristics:

Binary-Relational and Directional. According to the definition, "Trust ... is a particular level of the subjective probability with which *an agent* assesses that *another agent or group of agents* will perform a particular action ...". From this excerpt, it is evident that trust is a relationship between two entities. Moreover, it is also clear that trust is directional. The first entity is an agent who has trust in a second entity which may be another agent or a group of agents.

Contextual. As given in the definition, "Trust ... is a particular level of the subjective probability with which an agent assesses that another agent or group of agents will perform *a particular action* ...". We infer that trust is in the context of "a particular action" that the second entity may perform.

Quantifiable as Subjective Probability. "Trust ... is *a particular level of the subjective probability* with which an agent assesses that another agent or group of agents will perform a particular action, *both before he can monitor such action (or independently of his capacity ever to be able to monitor it)* ...". From this excerpt of the definition, we deduce that trust is quantifiable as subjective probability.

We discuss below some other characteristics of trust which are not evident from Gambetta's definition. We provide examples to support their validity as characteristics of trust. These characteristics have been previously identified by several authors (such as Capra [12]).

Non-Reflexive. An agent may or may not trust herself. For example, a patient Alice may trust her doctor to prescribe her the correct medicine, whereas she might not trust herself to do so.

Asymmetric. If an agent Alice trusts an agent Bob, then Bob may or may not trust Alice. For example, in the context of car repair, a car owner Alice may trust her mechanic Bob, however Bob may not necessarily trust Alice.

Non-Transitive. If an agent Alice trusts an agent Bob who in turn trusts an agent Carol, then Alice may or may not trust Carol. For example, an email server A might trust an email server B to not send spam. If B trusts an email server C in the same context, then A may or may not trust C depending on various factors such as its strength of trust in B, the availability of additional evidence, etc.

Dynamic. Trust may change with time. For example, let's say that an online shopper Alice has so far had good experiences with an online vendor Bob and therefore she has high trust in him. However, if her latest transaction with Bob is less than satisfactory, then her trust in Bob is likely to decrease instead of staying constant.

2.3 Inferring Trust

There are a number of techniques that enable inferring trust between entities. The first technique that we describe is *direct interaction* that requires explicit input from nodes. The other three methods that we discuss aim to infer trust from existing information.

2.3.1 Direct Interaction

The primitive method of establishing trust in an unknown entity is to directly interact with it and observe its behavior in the desired context. However, this method requires that the entity be trusted at least once without any prior background on that entity. This approach is perhaps suitable for low-risk transactions and in situations when no other recourse is available. However, when reliance on an unknown entity may lead to substantial damage, the other approaches for trust establishment are clearly preferable, since they allow the truster to base his trust on some prior knowledge provided by others.

McKnight et al. [42] introduce the notion of *initial trust*, which is described as the trust in an unfamiliar trustee – a relationship in which the actors do not yet have credible information about, or affective bonds with each other [7].

2.3.2 Trust Recommendation and Propagation

Establishing trust in an unknown entity through trust recommendation and propagation takes advantage of the possible transitivity of trust. Let's say that Alice wishes to establish trust in an unknown individual, Carol. If another individual Bob trusts Carol, then he could give a recommendation to Alice about Carol's trustworthiness. Taking Bob's trust recommendation and her own trust in Bob into account, Alice may establish a trust relationship with Carol. Thus a transitive path of trust that leads from Alice to Bob to Carol, enables Alice to develop trust in Carol. If Alice wishes to establish trust in Carol through Bob's recommendation, we say that Bob's trust in Carol has propagated to Alice.

Guha et al. [29] term the above described one-step propagation as *atomic propagation*. The term stems from the observation that the conclusion is reached based on a single argument, rather than a possibly lengthy chain of arguments. Guha et al. identify four types of atomic propagations: *direct propagation*, *co-citation*, *transpose trust*, and *trust coupling*. We briefly elaborate each of these types of atomic trust propagation:

Direct Propagation. The example given in the first paragraph represents direct propagation. If i trusts j, and j trusts k, then a direct propagation allows us to infer that i trusts k. Guha et al. refer to this particular atomic propagation as direct propagation since the trust propagates directly along an edge.

Co-Citation. Let's consider that i_1 trusts j_1 and j_2, and i_2 trusts j_2. Under co-citation, it is concluded that i_2 also trusts j_1.

Transpose Trust. In transpose trust, i's trust in j causes j to develop some level of trust towards i. Let's say that i trusts j, then transpose trust implies that j should also trust i.

Trust Coupling. Let's suppose that i and j both trust k, then trust coupling leads us to infer that i and j should trust each other since they both trust k.

Iterative propagation builds upon multiple atomic propagations to help establish trust in an unknown entity. Let's extend the example presented in the first paragraph: Alice trusts Bob and Bob trusts Carol. We further assume that Carol trusts Dave. Alice may establish trust in Dave as a result of the following two atomic propagations: 1) the first atomic propagation builds Bob's direct trust in Dave, and 2) now since Bob trusts Dave, Alice can establish trust in Dave through a second atomic propagation. This sequence of atomic propagations is referred to as iterative propagation.

2.3.3 Trust Negotiation

Trust negotiation is an approach that can enable strangers to electronically share sensitive data and services. Trust negotiation establishes trust between entities based not on their identities but their properties. For example, in the case of an individual, the properties that may be considered include their place of employment, age, membership in certain organizations etc. With trust negotiation, the trust between two entities is acquired through iterative requests for credentials and their disclosure.

An example from Bertino et al. [6]: CARS is an online car rental agency, which has an agreement with a company called CORRIER to provide rental vehicles free of charge to their employees, provided that they prove their employment status (which also implies that they are authorized to drive). Other customers (who are not employees of CORRIER) can rent a vehicle by showing a valid driving license and by providing a credit card for payment. Thus, CARS establishes trust in customers to be legitimate drivers through the exchange of multiple possible credentials.

```
Customer: Request a vehicle
CARS: Show digital employment ID from CORRIER
Customer: Not available
CARS: Show digital driving license
Customer: Digital driving license
CARS: Provide digital credit card
Customer: Digital credit card
CARS: Vehicle granted (vehicle info, pickup info, etc.)
```

2.3.4 Reputation

Reputation is the general opinion of the community about the trustworthiness of an individual or an entity. A person who needs to interact with a stranger, may analyze her reputation to determine the amount of trust that he can place in her. In the physical world, reputation often comes from word of mouth, media coverage, physical infrastructure, etc. However, the reputation of a stranger is often difficult to observe in online communities, primarily due to their global scale, the cheap availability of anonymous identities, and the relative ease of acquiring high quality digital presence.

A reputation system computes the reputation of an entity based on the feedback (quantified trust) provided by fellow entities. Reputation systems make certain that users are able to gage the trustworthiness of an entity based on the history of its behavior. The expectation that people will consider one another's pasts in future interactions constrains their behavior in the present [54].

Hoffman et al. [33] provide the following description of reputation:

> In general, reputation is the opinion of the public toward a person, a group of people, or an organization. In the context of collaborative applications such as peer-to-peer systems, reputation represents the opinions nodes in the system have about their peers. Reputation allows parties to build trust, or the degree to which one party has confidence in another within the context of a given purpose or decision. By harnessing the community knowledge in the form of feedback, reputation-based trust systems help participants decide who to trust, encourage trustworthy behavior, and deter dishonest participation by providing a means through which reputation and ultimately trust can be quantified and disseminated.

3 Privacy Preserving Reputation Systems

An accurate reputation score is possible only if the feedback is accurate. However, it has been observed that the users of a reputation system may avoid providing honest feedback [53]. The reasons for such behavior include fear of retaliation from the target entity or mutual understanding that a feedback value would be reciprocated.

A solution to the problem of fear of retaliation is computing reputation scores in a privacy preserving manner. A privacy preserving protocol for computing reputation scores does not reveal the individual feedback of any entity. Private feedback ensures that there are no consequences for the feedback provider and thus he is uninhibited to provide honest feedback.

Slandering is the act of sabotaging an honest user's reputation by assigning them unwarranted low feedback. A trade off of private feedback is that it creates the opportunity for slandering without consequences. However, we draw attention to the processes of voting and election, where the privacy of the voters is often guaranteed to allow them complete freedom of opinion. Since feedback providers in reputation systems are similarly entitled to personal opinion, it can be argued that their privacy should also be preserved. Slandering is most effective when it is carried out by a collusion of users. An important challenge to be addressed by future work is the detection of collusions in privacy preserving reputation systems.

3.1 Architecture

The architecture of a reputation system is one of the key factors in determining how the following activities are conducted:

- Feedback collection

- Feedback aggregation (reputation computation)
- Reputation dissemination

The two common architectures are: centralized and decentralized.

Centralized Reputation Systems. Centralized reputation systems are characterized by the existence of a trusted central authority. The central authority receives feedback from users, aggregates it to compute the reputation, and disseminates the reputation scores.

One of the benefits of a centralized solution is that it is straightforward to implement. Moreover, a centralized reputation system is often less vulnerable to certain attacks, such as the sybil attack, since the central authority can monitor and correlate all activities in the reputation system. Additionally, the central authority is universally trusted, therefore users can be assured that the feedback collection, aggregation, and dissemination are being done correctly.

Unfortunately, the requirement of universal trustworthiness of the central authority is also a liability. If the central authority fails or becomes compromised, then the whole reputation system crashes. Thus the central authority is a single point of failure and a high-value target for attackers. As with any other centralized system, another major disadvantage of centralized reputation systems is that they are very expensive to deploy and maintain, particularly for large numbers of users. Centralized reputation systems are also unable to cater for decentralized environments, particularly decentralized social networks.

Decentralized Reputation Systems. Decentralized environments are characterized by the absence of a central authority. Advantages of such networks include: lack of a single point of failure, no need to deploy and maintain an expensive central authority, a more democratic environment, scalability, etc.

Decentralized reputation systems are suitable for decentralized environments such as decentralized social networks as they do not assume the presence of a central entity. In decentralized reputation systems, a central location for submitting and aggregating feedback, and disseminating reputation does not exist. Feedback is commonly stored locally by the node who generates it, for example in response to his experiences with another party. Computing reputation of an entity in the system requires finding all or a portion of the nodes who carry feedback about that entity. Once the feedback providers have been located, the aggregation may be done at a single location after receiving all feedback, or a more sophisticated protocol may be employed to aggregate the feedback in a distributed manner.

3.2 Adversarial Models

Semi-Honest. In the semi-honest model, the agents do not deviate from the specified protocol. In other words, they always execute the protocol according to the specifications. The adversary abstains from wiretapping and tampering of the communication channels. However, within these constraints, the adversary passively attempts to learn the inputs of honest agents by using intermediate information received during the protocol and any other information that it can gain through other legitimate means.

Disruptive Malicious. Disruptive malicious agents are not bound to conform to the protocol. Agents under the malicious model may deviate from the protocol as and when they deem necessary. They actively attempt to achieve their objectives. They may participate in extra-protocol activities, devise sophisticated strategies, and exhibit arbitrary behavior. Specifically, malicious agents may 1) refuse to participate in the protocol, 2) provide out of range values as their inputs, 3) selectively drop messages that they are supposed to send, 4) prematurely abort the protocol, 5) distort information, and 6) wiretap and tamper with all communication channels. A malicious adversary may have one or both of the following objectives: 1) learn the inputs of honest agents, and 2) disrupt the protocol for honest agents. The reasons for disrupting the protocol may range from gaining illegitimate advantage over honest agents to completely denying the service of the protocol to honest agents.

4 Centralized Privacy Preserving Reputation Systems

We review some reputation systems that can be utilized for privacy preserving reputation management in centralized social networks.

4.1 Androulaki et al. [2] – A Reputation System for Anonymous Networks

Androulaki et al. [2] propose a reputation scheme for pseudonymous peer-to-peer systems in anonymous networks. Users in such systems interact only through disposable pseudonyms such that their true identity is not revealed. Reputation systems are particularly important for such environments since otherwise there is little incentive for good conduct. However, reputation systems are hard to implement for these environments. One of the reasons is that a user must keep his reputation even if he cycles through many pseudonyms. Moreover, the pseudonyms must be unlinkable to the user as well as to each other even though they share the same reputation score. Another issue that arises in reputation systems for anonymous networks is that a user may lend his good reputation to less reputable users through anonymous pseudonyms.

The proposed system employs the following cryptographic building blocks: anonymous credential systems, e-cash, and blind signatures. Reputation is exchanged in the form of e-coins called *repcoins*. The higher the amount of repcoins received from other users, the higher is the reputation of the user.

The system requires the presence of a *bank*, which is a centralized entity. Additionally, the system also requires that all communication take place over an anonymous network, such as Mixnet [13] or a network using Onion routing [20]. This requirement makes the solution inaccessible to applications in non-anonymous networks.

The security goals of reputation systems for anonymous networks are different than those of privacy preserving reputation systems. The reputation systems for anonymous networks aim to hide the identity of a user who interacts and assigns feedback to others. Whereas, in privacy preserving reputation systems, the goal is to hide the feedback value assigned but not the identity of the user who assigned it. The choice between the two kinds of reputation systems depends on the security objectives of the application.

4.1.1 Security Model

Some of the security requirements of the reputation system are as follows:

Unlinkability. An adversary, controlling the bank and a number of corrupted users, is unable to link a pseudonym with the identity of its non-corrupted user any better than by making a random guess. Moreover, the adversary has no advantage in telling whether two pseudonyms belong to the same non-corrupted user or not.

No Over-Awarding. A user who tries to double-award (forge) a repcoin, using one or even two different pseudonyms, gets detected and his identity is revealed.

Exculpability. Any coalition of corrupted users (including the bank) is unable to falsely accuse a user of forgery in order to expose his identity.

Reputation Unforgeability, Non-Transferability. A user cannot forge better reputation. In particular, a user U_1 cannot borrow reputation from another user U_2, unless U_2 reveals his master secret key to U_1.

4.1.2 Cryptographic Building Blocks

The following cryptographic building blocks are used for the construction of the scheme:

Anonymous Credential Systems. In anonymous credential systems (for example, [5, 10]), organizations grant credentials to pseudonymous identities of users. Verifiers are able to verify the authenticity of credentials in possession of users. However, neither an organization or a verifier is able to link a credential to the true identity of a user.

E-Cash. E-cash [14, 15] is a digital currency that offers the following properties: anonymity, unforgeability (or identification of double-spenders), and fungibility. Please see Section 5.5.1 for further detail. A centralized *bank* is a key player in an e-cash system.

Blind Signatures. In a blind signature scheme (for example, [14]), an entity signs a message for a user, however the entity does not learn the content of the message.

4.1.3 A Reputation System for Anonymous Networks

The system assumes the presence of a central entity called the bank, which is needed for implementing the above listed cryptographic schemes. The system also requires that all communication takes place over an anonymous network, for example, a Mixnet, or a network using Onion routing. The users interact with each other in a peer-to-peer manner. However, the users must also communicate with the central bank to withdraw and deposit repcoins.

From the above listed building blocks, Androulaki et al. build a reputation system in which each user has a reputation that he cannot lie about or shed. However, a user may generate as many one time pseudonyms as he needs for his transactions. All pseudonyms of a user share the same reputation. The system is robust against self-promotion attacks. Reputation is updated and demonstrated in a way such that anonymity is not compromised. The system maintains unlinkability between the identity of a user and his pseudonyms, and unlinkability among pseudonyms of the same user.

The system by Androulaki et al. follows upon the work by Dingledine et al. [19, 18, 17] on reputations systems and anonymous networks.

Table 1 Androulaki et al. [2] – A Reputation System for Anonymous Networks.

Architecture	Centralized
Target Environment	Peer-to-peer systems
Adversarial Model	Malicious (Disruptive)
Key Security Mechanisms	Anonymous credential systems, E-cash (bank), Blind signatures, Mixnets / Onion Routing
Privacy Guarantee	Satisfies unlinkability, no over-awarding, exculpability, and reputation unforgeability if the underlying primitives (anonymous credential system, e-cash system, and blind signatures) are secure
Complexity (Messages)	$O(1)$

4.2 Steinbrecher [56] – Privacy-Respecting Reputation Systems within Centralized Internet Communities

Steinbrecher [56] argues that traditional cryptographic techniques such as encryption and digital signatures can provide only "technical" security guarantees. For example, encryption and digital signatures can guarantee the confidentiality and integrity of the *text* of a reply sent by an expert to a user on a self help forum. However, these techniques cannot guarantee the misbehavior of the users themselves. For example, the user might violate confidentiality by relaying the *content* of the text to a third party, or the expert may violate integrity by giving *false advice*. It is argued that trust can mitigate these risks and that reputation systems are a suitable technology for acquiring trust.

However, the author contests that the design of current reputation systems (such as the eBay reputation system) allow open access to the interests and behavior profiles of users. A third-party may acquire information such as the time and frequency of participation, interests in specific items, feedback provided etc. Moreover, it is easy to associate the pseudonym of a user with their real identity, for example, through a mailing address.

To counter this issue, Steinbrecher presents a privacy-respecting reputation system for centralized Internet communities. The system relies on simultaneous use of multiple pseudonyms and changing them frequently to achieve anonymity and unlinkability.

4.2.1 A Generalized Model for Centralized Reputation Systems

The paper presents a generalized model for centralized reputation systems. Users use global pseudonyms tied to global reputations. The set of global pseudonyms at time t is considered as $P_t = \{p_{t,1}, \ldots, p_{t,m}\}$. The set of possible reputations that might be associated with a pseudonym is given as R. $(R, +)$ is a commutative group and $+$ an operator to combine elements from R independently of t. At time t_1, each pseudonym $p_{t_1,l}$ has the reputation $rep(t_1, p_{t_1,l}) \in R$, where $l \in 1 \ldots m$. After $p_{t_1,i}$ receives a rating r_{j,i,t_1} from $p_{t_1,j}$, the reputation of $p_{t_1,i}$ at time t_2 is computed as:

$$rep(t_2, p_{t_1,i}) = rep(t_1, p_{t_1,i}) + r_{j,i,t_1} \qquad (1)$$

where $t_2 \geq t_1$, and $p_{t_1,i}$ does not receive any rating other than r_{j,i,t_1} between t_1 and t_2.

4.2.2 Using Pseudonyms for Unlinkability and Anonymity

The system proposes simultaneous use of multiple pseudonyms by a user. The idea is to have a separate pseudonym for each context (for example, the context of a seller on an auction site, the context of an expert on a self help forum, etc.). It is suggested that this design leads to unlinkability between the different roles of a user on the Internet.

The system permits users to regularly change their pseudonyms to achieve anonymity. A new and an old pseudonym are unlinkable from the perspective of third-parties, however, the provider (central server) is able to link the two pseudonyms. The unlinkability also assumes that a large number of pseudonyms have the same reputation.

To prevent the provider from linking new and old pseudonyms, the system suggests using a set of non-colluding trustworthy third parties who make incremental changes to the pseudonym of the user.

Steinbrecher's work on reputation and privacy also includes [55, 51]. These proposals are oriented for centralized environments as well.

An adversary may compromise unlinkability by monitoring all pseudonyms with the same reputation. The adversary can deduce that a new pseudonym with the same reputation as a recently deleted pseudonym belong to the same user.

Table 2 Steinbrecher [56] – A Centralized Privacy Preserving Reputation System.

Architecture	Centralized
Target Environment	E-commerce, Self-help forums, etc.
Adversarial Model	Malicious (Disruptive)
Key Security Mechanisms	Pseudonym / Identity management
Privacy Guarantee	Unlinkability and anonymity are satisfied if the provider (central server) is honest and secure
Complexity (Messages)	$O(1)$

5 Decentralized Privacy Preserving Reputation Systems

In the following sections, we discuss reputation systems that can be deployed in decentralized social networks for privacy preserving reputation management.

5.1 Clifton et al. [16] – Secure Sum

Secure multi-party computation is the study of protocols that take inputs from distributed entities and aggregate them to produce outputs, while preserving the privacy of the inputs.

One of the well-known secure multi-party computation protocols is secure sum [16], which takes inputs from entities and computes their sum. This protocol is clearly a natural fit for the problem at hand. The protocol may be used directly to compute reputation in the form of sum or mean.

5.1.1 Secure Sum

The secure sum protocol assumes that there are three or more sites and there is no collusion between them. It is also assumed that the value to be computed, $v = \sum_{l=1}^{s} v_l$ lies in the range $[0..m]$. The sites are numbered as $1 \dots s$. Site 1 generates a random number R uniformly chosen from $[0..m]$. It then sends $R + v_1 \ mod \ m$ to site 2, where v_1 is site 1's local input. Site 2 does not learn any information about v_1 since $R + v_1 \ mod \ m$ is distributed uniformly across the range $[0..m]$ due to R. For sites $l = 2 \dots s - 1$, the protocol proceeds as follows: Site l receives:

$$V = R + \sum_{j=1}^{l-1} v_j \ mod \ m \tag{2}$$

Site l learns nothing since the value is distributed uniformly across $[0..m]$. Site l computes:

$$R + \sum_{j=1}^{l} v_j \ mod \ m = (v_l + V) \ mod \ m \tag{3}$$

Site l then sends this value to site $l + 1$. Eventually, site s also performs the above step. Site s sends the result back to site 1, who subtracts R from it to obtain the sum. Site 1 does not learn any of the private values due to the uniform distribution of the received result over the range $[0..m]$.

The protocol may be used to compute reputation as the sum of the feedback values provided as private inputs by the participants of the protocol.

The security of the secure sum protocol breaks down if the sites collude. Any two sites $l - 1$ and $l + 1$ can use the values that they send and receive respectively to compute the private input v_l of site l.

5.1.2 Other Secure Multi-Party Computation Protocols

Other secure multi-party computation protocols include: secure product [16, 1, 3, 35], secure set union [16, 39], secure set intersection [16, 39], and secure multiset operations [39]. The doctoral thesis of Wenliang Du [21] describes several secure two-party computation protocols for problems in linear programming, geometry, and statistical analysis. A seminal work in secure multi-party computation is the study of the Millionaire's problem [59], in

Table 3 Clifton et al. [16] – Secure Sum.

Architecture	Decentralized
Target Environment	Distributed environments
Adversarial Model	Semi-Honest + Agents do not collude
Key Security Mechanisms	Secure multi-party computation
Privacy Guarantee	The chances that the adversary will learn private information are no better than making a random guess over the range $[0..m]$. Probability: $\frac{1}{m+1}$
Complexity (Messages)	$O(n)$, where n = number of sites

which two parties must determine whose number is larger without disclosing their numbers. We refer the reader to [26] for a comprehensive study of secure multi-party computation.

5.2 Pavlov et al. [46] – Decentralized Additive Reputation Systems

Pavlov et al. [46] propose several protocols for decentralized additive reputation systems. Two of their protocols are secure under the semi-honest and the malicious adversarial models respectively. The protocols draw their strength from witness (feedback provider) selection schemes, which guarantee the inclusion of a certain number of honest witnesses as participants. The security mechanisms used in the protocols include secure multi-party computation, secret sharing, and discrete log commitment.

5.2.1 Problem Setting

A querying agent consults a group of n witnesses to compute the reputation of a target agent, where $0 < n < N$, and $N > 1$ is the number of potential witnesses. $b < N$ is the number of dishonest agents in N.

5.2.2 Decentralized Additive Reputation Systems

A decentralized additive reputation system is described in the article as a reputation system that satisfies the following two requirements: 1) feedback collection, combination, and propagation are implemented in a decentralized way; 2) combination of feedbacks provided by agents is calculated in an additive manner. The Beta reputation system [36] is cited as an example. The eBay reputation system is additive, however, not decentralized.

5.2.3 Impossibility of Perfect Privacy

The paper argues that it is impossible to guarantee perfect privacy for an honest feedback provider in a decentralized additive reputation protocol. The argument is that a dishonest agent may deterministically create a set of n feedback providers, with $n-1$ dishonest agents and the one honest agent under attack. Given the inputs of the $n-1$ dishonest agents and the output (the reputation score), the secret feedback of the honest agent is easily obtained.

The impossibility argument does not apply to protocols in which an honest agent may choose not to contribute his feedback. The argument also does not apply to protocols in which the set of feedback providers cannot be created deterministically.

5.2.4 Witness Selection Scheme 1 (WSS-1)

A witness selection scheme for a reputation protocol is a process that results in the creation of a set of witnesses. The witnesses in the set contribute their feedback towards computing the reputation of the target agent.

The first scheme [46, Lemma 2] guarantees that if honest agents are uniformly distributed over N, then at least two honest witnesses will be selected with probability greater than $(1 - \frac{1}{n})(\frac{N-b-1}{N-1})$. The scheme is secure under the semi-honest adversarial model, in which all agents follow the protocol correctly.

According to our analysis, the complexity of the number of messages exchanged is linear in terms of the number of potential witnesses: $O(N)$. After each witness is selected, it is probabilistically decided whether to add more witnesses, therefore the count may run up to N. If each agent sends its successor the current set of witnesses, the total bandwidth utilized is $O(N^2)$.

The complexity of the scheme is a function of the population size of the potential witnesses (N) instead of the witnesses who contribute their feedback (n). The scheme also has the potential of leaving out many honest witnesses from the reputation protocol. Moreover, the scheme works only if $b < n-1$, because otherwise $n-1$ dishonest witnesses can select themselves into the set if the first witness selected is dishonest. Even then the scheme might fail since the number of witnesses selected is probabilistic and it may be the case that the actual number of selected witnesses is less than n.

5.2.5 Witness Selection Scheme 2 (WSS-2)

The second scheme [46, Lemma 3] guarantees under the malicious adversarial model that if honest agents are uniformly distributed over N, then at least $n(\frac{N-b-n}{N})$ honest witnesses would be selected. A coin flipping scheme is utilized to grow the set of witnesses by selecting the next witness randomly from the available pool of witnesses. According to the paper, the scheme requires $O(n^3)$ messages among the n selected witnesses.

5.2.6 A Reputation Protocol based on WSS-1

In this reputation protocol, the set of source agents is created using the first witness selection scheme, which guarantees that at least two source agents are honest. Agent q chooses a random number as its secret. Each agent splits its secret into $n+1$ shares such that they all add up to the secret. Each agent keeps the $n+1^{th}$ share and sends its other n shares to the other n agents in the protocol such that each agent receives a unique share. Each agent then adds all shares received along with his $n+1^{th}$ share and sends it to the querying agent. The querying agent adds all sums received and subtracts the random number to obtain the reputation score.

The protocol guarantees the privacy of an honest source agent under the semi-honest model as long as all the other $n-1$ source agents do not collude. The probability that all other source agents will not collude is greater than $(1-\frac{1}{n})(\frac{N-b-1}{N-1})$. The number of messages exchanged is analyzed as $O(n^2)$. We estimate that the size of the messages exchanged is as follows: $O(n^2)$ IDs and $O(n^2)$ numbers.

The complexity is claimed to be $O(n^2)$, however, we believe it to be $O(N)+O(n^2)$ due to the utilization of the witness selection scheme.

Table 4 Pavlov et al. [46] – A Reputation Protocol based on WSS-1.

Architecture	Decentralized
Target Environment	Distributed environments
Adversarial Model	Semi-honest
Key Security Mechanisms	Secure multi-party computation, secret sharing
Privacy Guarantee	$(1-\frac{1}{n})(\frac{N-b-1}{N-1})$
Complexity (Messages)	$O(N)+O(n^2)$, where N = number of potential witnesses, and n = number of selected witnesses

5.2.7 A Reputation Protocol based on WSS-2

This protocol uses the Pedersen verifiable secret sharing scheme [48] and a discrete log commitment method. The Pedersen scheme is resilient up to $n/2$ malicious agents. The set of source agents is created using the second witness selection scheme. It guarantees the presence of less than $n/2$ malicious agents, if $b<\frac{N}{2}-n$.

The protocol is secure under the malicious model as long as $b < \frac{N}{2} - n$. The number of messages exchanged is $O(n^3)$, due to the second witness selection scheme.

Table 5 Pavlov et al. [46] – A Reputation Protocol based on WSS-2.

Architecture	Decentralized
Target Environment	Distributed environments
Adversarial Model	Malicious (Disruptive)
Key Security Mechanisms	Verifiable secret sharing, discrete log commitment
Privacy Guarantee	If $b < \frac{N}{2} - n$, then the adversary does not learn any more information about the private feedback of an honest witness
Complexity (Messages)	$O(n^3)$, where n = number of witnesses

5.3 Gudes et al. [28] – The Knots Reputation System

Gudes et al. [28] present several schemes that augment their Knots reputation system [23] with privacy preserving features. A defining characteristic of the Knots reputation model is the notion of subjective reputation. The reputation of a target member is computed by each querying member using a different set of feedback, thus the reputation is subjective for each querying member. The feedback that a querying member uses for computing reputation comes exclusively from the members in which he has a certain amount of pre-existing trust. An advantage of this approach is that the querying member has confidence in each of the feedback values that are used for computing reputation.

The disadvantage is that the opinion of the members whom the querying agent does not know in not taken into account. The notion of subjective reputation tends to be non-conformant with the idea of reputation, which is generally considered to be the aggregate of feedback of the community at large. The concept of subjective reputation seems closer to trust propagation than reputation.

5.3.1 The Knots Model

The Knots model differentiates between two types of users in the system. The *experts* in the system are the users who provide services and the *members* are users who consume those services. The reputation system is concerned with computing the reputation of the experts through the feedback provided by the members. Members have trust relationships among themselves in the context of providing reliable feedback about the experts.

$TrustSet_x(A)$ is defined as the set of members whom member A trusts to provide feedback about expert x. $TM(A,B)$ represents the amount of direct trust that a member A has in another member B. $DTE(A,x)$ is defined as the amount of direct trust that a member A has in an expert x. The subjective reputation of an expert x by a member A is computed as follows:

$$TE(A,x) = \frac{\Sigma_{\forall B \in TrustSet_x(A)} DTE(B,x) \cdot TM(A,B)}{\Sigma_{\forall B \in TrustSet_x(A)} TM(A,B)} \tag{4}$$

In the privacy preserving version of the Knots model, the challenge is to compute $TE(A,x)$, such that the privacy of each $DTE(B,x)$ is maintained, where $B \in TrustSet_x(A)$. The three decentralized privacy preserving schemes presented in the paper compute $\rho(A,x)$ (the numerator of the fraction in equation 4), such that A cannot learn any of the $DTE(B,x)$ values.

The privacy goal does not include preserving the privacy of the trust between the members (the TM values). It is limited to preserving the privacy of the feedback about the experts (the DTE values).

5.3.2 Reputation Scheme 1

Each member $B \in TrustSet_x(A)$ receives $TM(A,B)$ from A and then computes $E_A(DTE(B,x) \cdot TM(A,B))$ and sends it to a Trusted Third Party (TTP), Z (where $E_A(.)$ is an encryption with the public key of member A). The TTP Z relays each message to A without revealing the source member. A decrypts the messages and obtains $\rho(A,x)$.

Since A does not know the source of a message, it cannot reverse a received value to reveal the private feedback. The messages are encrypted, therefore the TTP does not learn any information either. The scheme requires $O(n)$ messages to be exchanged, where n is the cardinality of $TrustSet_x(A)$.

The scheme requires disclosure of the trust that A has in each member B. Moreover, there is heavy reliance on the TTP. If the TTP and A collude, then they can easily determine each $TM(B,x)$.

5.3.3 Reputation Scheme 2

Each member $B \in TrustSet_x(A)$ generates $E_A(DTE(B,x))$ and sends it to a TTP, Z. The TTP sends a randomly permuted vector of the messages to A, who decrypts the messages and obtains a vector (vector 1) of the DTE values. A then sends a vector of all values $TM(A,B)$ to Z, where $B \in TrustSet_x(A)$. Z permutes the vector (vector 2) according to the DTE vector (with respect to the order of the members). A and Z compute the scalar product of vectors 1 and 2 using a secure product protocol (such as [1]) to obtain $\rho(A,x)$.

Table 6 Gudes et al. [28] – Reputation Scheme 1.

Architecture	Decentralized
Target Environment	Distributed environments
Adversarial Model	Semi-honest
Key Security Mechanisms	TTP, Public-key cryptography
Privacy Guarantee	If the TTP is honest, the chances that A will learn $DTE(B, x)$ are no higher than making a random guess across $\|TrustSet_x(A)\|$ given values. $B \in TrustSet_x(A)$.
Complexity (Messages)	$O(n)$, where $n = \|TrustSet_x(A)\|$

Due to the random permutation generated by the TTP, A is unable to correlate the DTE values with individual members. The TTP does not learn any of the DTE values due to encryption. A key advantage of the scheme is that any member B does not learn $TM(A, B)$.

We analyze that the number of messages exchanged is $O(n)$, whereas the bandwidth utilized is $O(n^2)$ in terms of k-bit numbers transfered, where k is the security parameter (key length).

The privacy of the $TM(A, B)$ values is still not fully preserved since they must be disclosed to the TTP.

Table 7 Gudes et al. [28] – Reputation Scheme 2.

Architecture	Decentralized
Target Environment	Distributed environments
Adversarial Model	Semi-honest
Key Security Mechanisms	TTP, Public-key cryptography, Secure product
Privacy Guarantee	If the TTP is honest, the chances that A will learn $DTE(B, x)$ are no higher than making a random guess across $\|TrustSet_x(A)\|$ given values. Moreover, B does not learn $TM(A, B)$. $B \in TrustSet_x(A)$.
Complexity (Messages)	$O(n)$, where $n = \|TrustSet_x(A)\|$

5.3.4 Reputation Scheme 3

A executes the reputation protocol for the semi-honest model from Pavlov et al. [46] to obtain $\Sigma_{\forall B \in TrustSet_x(A)} DTE(B, x)$. A sends $TM'(A, B) = TM(A, B) + Q$ to each $B \in TrustSet_x(A)$, where Q is a random number. A executes the secure sum protocol [16] to obtain $\Sigma_{\forall B \in TrustSet_x(A)}(TM'(A, B) \cdot DTE(B, x))$. A calculates:

$$\rho(A, x) = \Sigma_{\forall B \in TrustSet_x(A)}(TM'(A, B) \cdot DTE(B, x))$$

$$-(Q \cdot \Sigma_{\forall B \in TrustSet_x(A)} DTE(B, x)) \qquad (5)$$

This scheme has the advantage that the privacy of both the $DTE(B, x)$ values and the $TM(A, B)$ values is preserved without the presence of any TTPs. The protocol requires $O(n^2)$ messages due to the inclusion of the protocol from [46].

Table 8 Gudes et al. [28] – Reputation Scheme 3.

Architecture	Decentralized
Target Environment	Distributed environments
Adversarial Model	Semi-honest + Agents do not collude
Key Security Mechanisms	Secure multi-party computation
Privacy Guarantee	A does not learn more information about $DTE(B, x)$, where $B \in TrustSet_x(A)$. The chances of B learning $TM(A, B)$ are no better than its chances of guessing the random number Q from $TM'(A, B)$.
Complexity (Messages)	$O(n^2)$, where $n = \lvert TrustSet_x(A)\rvert$

5.3.5 Proposals for the Malicious Adversarial Model

The work also includes some proposals for augmenting the schemes for the malicious adversarial model. The proposals are largely based on the assumption that a member who provides feedback (member B) would lack the motivation to act maliciously if it does not know the identity of the querying member (member A). However, this assumption does not take into account the case when an attacker may want to attack the system simply to disrupt it, for example, in a denial-of-service attack.

5.4 Hasan et al. [32] – The k-Shares Reputation Protocol

The k-shares protocol by Hasan et al. [32] [31] [30] offers the following advantages over comparable protocols such as those by Pavlov et al. [46, Section 5.2] and Gudes et al. [28]: 1) Lower message complexity of $O(n)$ as opposed to $O(n^2)$ and higher of the protocols in [46] and [28]; 2) The k-Shares protocol allows agents to quantify and maximize the probability that their privacy will be preserved before they submit their feedback.

5.4.1 Framework

An action called “preserve privacy” is defined. Agents are assumed to have trust relationships with some other agents in the context of this action. This assumption is called *trust awareness* and derives from the fact that agents have social relationships and a key component of such relationships is the trust that each other’s privacy will be preserved. For example, a user may trust his family members and close friends to help him preserve his privacy.

The adversary is considered as semi-honest and is allowed to collude. Privacy is formalized using the Ideal-Real approach. An ideal protocol for computing reputation is one in which a Trusted Third Party (TTP) receives all inputs and then locally computes the reputation. On the other hand, a real protocol computes reputation without the participation of any TTP. The real protocol is said to preserve privacy if the adversary, with high probability, cannot obtain any more information about the private input of an agent than it can learn in the ideal protocol.

5.4.2 The Protocol

A simplified version of the protocol is outlined below.

1. **Initiate.** The querying agent q retrieves the set of source agents S_t of the target agent t and sends the set to each of the source agents.
2. **Select Trustworthy Agents.** Each source agent selects up to k other agents in S_t. Each agent selects these agents such that the probability that all of them will collude to break his privacy is low. k is a constant, such that $k \ll n$, where n is the number of all source agents. The risk to privacy is thus quantified before submitting the feedback.
3. **Prepare and Send Shares.** Each agent generates k shares such that their sum is equal to the secret feedback value. The secret cannot be revealed until all shares are known. The shares are sent to the selected fellow agents.
4. **Compute Sums and Reputation.** Each agent that receives shares from fellow agents computes the sum of all shares received and sends the sum to the querying agent q. Agent q receives all the sums and computes the grand total and divides it by n to learn the reputation score.

The full version of the protocol takes measures to ensure that a share is not compromised even if it is the only share received by an agent. Moreover, the protocol also takes steps so that the protocol does not reach certain failure states.

The highlights of the protocol are as follows: 1) It requires each source agent to send only $k \ll n$ messages, which implies that the protocol requires only $O(n)$ messages. 2) The risk to privacy can be quantified before submitting feedback. Thus, an agent knows the risk and if that risk is unacceptable it can opt to not participate in the protocol. As a consequence, even up to $n-1$

dishonest agents in the protocol cannot breach the privacy of one dishonest agent.

Table 9 Hasan et al. [32] – The k-Shares Reputation Protocol.

Architecture	Decentralized
Target Environment	Distributed environments
Adversarial Model	Semi-honest
Key Security Mechanisms	Secure multi-party computation, Trust awareness, Secret sharing
Privacy Guarantee	The privacy of an agent a is preserved with high probability if it finds k trustworthy agents in the set of feedback provider agents S_t, such that $k \ll n$ and the probability that all k agents will collude to break agent a's privacy is low.
Complexity (Messages)	$O(n)$, where n is the number of feedback providers

5.4.3 Experimental Results

The work comprises of experiments on the web of trust of the Advogato.org social network. The members of Advogato rate each other in the context of being active and responsible members of the open source software developer community. The choice of feedback values are *master*, *journeyer*, *apprentice*, and *observer*, with *master* being the highest level in that order. The result of these ratings is a rich web of trust. The members of Advogato are expected to not post spam, not attack the Advogato trust metric, etc. It is therefore argued that the context "be a responsible member of the open source software developer community" comprises of the context "be honest". The four feedback values of Advogato are substituted as follows: $master = 0.99$, $journeyer = 0.70$, $apprentice = 0.40$, and $observer = 0.10$. For the experiments, the lowest acceptable probability that privacy will be preserved is defined as 0.90. This means that a set of two trustworthy agents must include either one *master* rated agent or two *journeyer* rated agents for this security threshold to be satisfied. The two experiments and their results are as follows:

Experiment 1: In the k-Shares protocol, the following assumption must hold for an agent a's privacy to be preserved: the probability that the agents to whom agent a sends shares, are all dishonest must be low. The experiment determines the percentage of instances of source agents in the Advogato data set for whom this assumption holds true.

Results: Consider the case where there are at least 50 source agents present in the protocol and $k = 2$, that is only two trustworthy agent can be selected to preserve privacy. It is observed that the assumption holds for 85.8% of instances of source agents. At $n \geq 5$, the percentage is 72.5%.

Experiment 2: The experiment observes the effect of increasing k on the percentage of instances of source agents whose privacy is preserved by the k-Shares protocol in the Advogato.org data set.

Results: Consider the case where there are at least 50 source agents present in the protocol and $k = 1$, that is only one trustworthy agent can be selected to preserve privacy. In the percentage of instances of source agents whose privacy is preserved is 75.4%. At $k = 2$, the percentage is 85.8%. The rise is due to the possibility with $k = 2$ to rely on two trustworthy agents. Increasing k over 2, even up to 500, does not result in a significant advantage (86.3% at $k = 500$). These results validate the assumption that the privacy of a large number of agents can be preserved with $k \ll n$.

5.5 Belenkiy et al. [4] – A P2P System with Accountability and Privacy

Selfish participants are a major threat to the functionality and the scalability of peer-to-peer systems. Belenkiy et al. [4] propose a content distribution peer-to-peer system that provides accountability, which makes it resilient against selfish participants. The solution is based on e-cash technology. Despite making peers accountable, the system does not compromise the privacy of the peers. The system ensures that transactions between peers remain private. The only exception is the case when there is a dispute between transacting peers.

Although the system is not directly related to reputation, we study it here because it provides insight into designing a privacy preserving system using the e-cash technology. In Section 4.1, we discuss a privacy preserving reputation system based on e-cash by Androulaki et al. [2].

5.5.1 E-Cash and Endorsed E-Cash

E-cash [14, 15] is a digital currency that offers the following properties:

Anonymity. It is impossible to trace an e-coin (the monetary unit of e-cash) to the user who spent it. This property holds even when the bank (a central entity who issues the e-coins) is the attacker.

Unforgeability. The only exception to the anonymity property is that e-cash does not guarantee the anonymity of a user who tries to double-spend an e-coin. In this case, the bank can learn the identity of the dishonest user. A forged e-coin allows the bank to trace down the user who forged it.

Fungibility. A user can use the e-coins received for services provided as payment for services received from any other user in the system.

Endorsed e-cash [11] adds the following property to e-cash:

Fair Exchange. Fair exchange means that a buyer gets the item only if the seller gets paid and vice versa.

5.5.2 A Currency based Model

The authors describe a peer-to-peer content distribution system inspired by BitTorrent [52]. However, the proposed system provides stronger accounting in its protocols that allow nodes to *buy* and *barter* data blocks from their neighbors in a fair manner.

The system requires the participation of two trusted entities: 1) A *bank*, which maintains an endorsed e-cash account for each user. Users are able to make deposits and withdrawals of e-coins. 2) An *arbiter*, which protects the fair exchange of e-cash for data.

A user has two options for acquiring the data blocks that it needs: 1) it can pay e-coins to users who own those data blocks; 2) or it can barter its own data blocks for the ones that it needs. To earn e-cash, a user has to offer data blocks that other users want and exchange them for e-coins. A user is prevented from being selfish since it cannot consume the service provided by the peer-to-peer system unless he contributes as well.

An unendorsed e-coin cannot be deposited into the seller's bank account until the buyer endorses it. Each unendorsed e-coin has a contract associated with it. The fair exchange takes place according to the contract. If the seller fulfills its commitments, then the unendorsed e-coin must be endorsed by the buyer or otherwise by the arbiter.

5.5.3 The Buy and Barter Protocols

The *buy* protocol operates as follows: Alice requests a data block from Bob. Bob encrypts the block with a random key and sends the ciphertext to Alice. Alice sends an unendorsed e-coin and a contract for the data block. If the unendorsed e-coin and the contract are formed correctly, Bob sends the decryption key for the data block to Alice. If the key decrypts the data block correctly, Alice endorses the sent e-coin, which Bob can then deposit into his account.

The protocol ensures that fair exchange of e-coins and data takes place. If Bob is dishonest and the key is incorrect, Alice does not endorse the e-coin. In case Alice is dishonest and she does not endorse the coin after receiving the key, Bob can present the arbiter with proof of his correct service (in the form of the contract and other credentials received from Alice) and have the arbiter endorse the e-coin for him.

Moreover, the privacy of the transaction is preserved since no third party involvement is required, unless there is a need for arbitration. The e-coin spent by Alice is unlinkable to her due to the anonymity provided by e-cash.

The barter protocol also provides fair exchange and privacy. Alice and Bob initially send each other an unendorsed e-coin as collateral and a contract which lets them have the arbiter endorse the coin in case the key for a bartered data block is incorrect. Alice and Bob then continue to exchange data blocks until the occurrence of fair termination or arbitration.

Endorsed e-cash requires that each received e-coin must be deposited back to the bank before it can be spent. The buy protocol therefore incurs significant overhead due to this requirement. However, the barter protocol is scalable since it does not require any involvement from the bank under normal circumstances.

The bank and the arbiter are centralized entities. This implies that the system is not fully decentralized. The two centralized entities present scalability issues (at least for the buy protocol) as well as single points of failure.

5.6 Nin et al. [45] – A Reputation System for Private Collaborative Networks

Nin et al. [45] present a reputation system that computes the reputation of a user based on the access control decisions that he makes. If a user makes good access control decisions, such as granting access to legitimate users and denying access to unauthorized users, then he receives good reputation. In contrast, making dishonest access control decisions leads to bad reputation. The privacy objective of the reputation system is to keep the trust relationships between the users private.

The system operates as follows: A node keeps record of its access control decisions. Other nodes can view anonymized details of those decisions and verify if the decisions were made according to the access control rules or not. The anonymization is derived through the multiplicative homomorphic property of the ElGamal encryption scheme. Private details are not revealed to a third-party due to the anonymization.

5.6.1 Private Collaborative Networks

A private collaborative network is described as a network of users that has the following properties: 1) the users are connected with each other through trust relationships; 2) users own resources that can be accessed by other users if sufficient trust exists; and 3) trust relationships among users remain private.

A private collaborative network is modeled as a directed labeled graph. Edges represent trust relationships between nodes (users). Each edge is labeled with the type of trust relationship as well as the weight of the trust.

Access to each resource in the network is governed by a set of access conditions. An access condition is of the form $ac = (v, rt, d_{max}, t_{min})$, where v is the owner with whom the requester of the resource must have a direct or transitive trust relationship of type rt to gain access. d_{max} and t_{min} are the required maximum depth and minimum trust respectively to obtain access.

Each trust relationship also exists in the form of a certificate signed by the truster and the trustee. Since relationships must be kept private, a certificate itself is considered a private resource. To gain access to a resource, a requester must demonstrate to the owner, the existence of a "certificate path" linking the requester to the owner.

5.6.2 The Reputation Model

The reputation system assigns good reputation to a user who performs decisions in accordance with the specified access conditions. In contrast, a user who does not correctly enforce access control rules, receives lower reputation. Reputation lies in the interval $[0, 1]$.

A user can act dishonestly in two ways: 1) deny access to a resource to a legitimate requester, or 2) allow access to a resource to an unauthorized requester. The access control decision is considered wrong if it violates either of the rt, d_{max}, t_{min} parameters in the access condition. For a wrong decision that violates the trust requirement (t_{min}), the absolute difference between the minimum amount of trust required (t_{min}) and the trust computed over the certificate path is given as wd. The values arising from all such wrong decisions are given as the set $\{wd_1, \ldots, wd_{|WD_{t_A}|}\}$, where $|WD_{t_A}|$ is the number of wrong decisions.

The values in the set $\{wd_1, \ldots, wd_{|WD_{t_A}|}\}$, which represent the wrong decisions made by user A in terms of trust, are aggregated as:

$$AGt_{AC_{SET_A}} = OWA_Q(wd_1, \ldots, wd_{|WD_{t_A}|}) \qquad (6)$$

where $AGt_{AC_{SET_A}}$ is the aggregated value of the wrong decisions with respect to trust. OWA is an Ordered Weighted Averaging function and Q is a non-decreasing fuzzy quantifier. According to the authors: "The interest of the OWA operators is that they permit the user to aggregate the values giving importance to large (or small) values".

The wrong decisions of the user that violate the depth and path requirements are aggregated as $AGd_{AC_{SET_A}}$ and $AGp_{AC_{SET_A}}$ respectively. The reputation of user A is then computed as:

$$R_A = 1 - \frac{1}{3}(AGt_{AC_{SET_A}} + AGd_{AC_{SET_A}} + AGp_{AC_{SET_A}}) \quad (7)$$

which implies that the mean of the aggregates of the three types of wrong decisions is subtracted from the perfect reputation of 1 to arrive at the reduced reputation of the user. The more dishonest decisions a user makes, the lower his reputation.

5.6.3 Anonymized Audit Files

After a user makes an access control decision, an entry about that decision is added into the user's anonymized audit file. The entry includes information such as the identity of the requester of the resource, the certificate path demonstrated by the requester, etc. However, all private information in the entry is encrypted using the ElGamal encryption scheme [22]. Therefore, a third-party who analyzes the entry is unable to acquire any information about these private elements. Due to the multiplicative homomorphic nature of the ElGamal encryption scheme, the encrypted information can be manipulated to compute reputation. A network participant who wishes to learn the reputation of a certain user, can analyze the anonymized audit file of that user and derive the reputation score without compromising privacy.

We analyze the number of messages exchanged to compute reputation as constant ($O(1)$), since all required information is provided directly by the target node.

The reputation system has the following advantages: 1) the reputation of a node is not derived from the feedback of other nodes but from objective information about its behavior (its access control decisions), and 2) a node itself manages and furnishes the evidence required for another node to judge its reputation.

The adversarial model is not specified in the paper, however, we estimate that the scheme would be secure only upto the semi-honest model since nodes are assumed to manage their audit files honestly.

Table 10 Nin et al. [45] – A Reputation System for Private Collaborative Networks.

Architecture	Decentralized
Target Environment	Private collaborative networks
Adversarial Model	Semi-honest
Key Security Mechanisms	ElGamal encryption scheme
Privacy Guarantee	Trust relationships among users remain private if the underlying encryption scheme is secure
Complexity (Messages)	$O(1)$

5.7 Kinateder and Pearson [37] – A Privacy-Enhanced P2P Reputation System

The decentralized reputation system proposed by Kinateder and Pearson [37] requires a Trusted Platform Module (TPM) chip at each agent. The TPM enables an agent to demonstrate that it is a valid agent and a legitimate member of the reputation system without disclosing its true identity. This permits the agent to provide feedback anonymously.

5.7.1 Security Goals

The reputation system sets the security requirements listed below. An attacker must not be able to:

- Provide false feedback on an honest user's behalf.
- Access an honest user's private database and modify data such as feedback, reputation, etc.
- Learn the identity of a feedback provider (which implies that a user should be able to provide feedback anonymously).

Moreover, it is required that:

- The identity of a dishonest user can be revealed if there is sufficient legal justification.

5.7.2 Trusted Platform

The reputation system presented in the paper relies on the Trusted Platform (TP) [44, 47] technology for security. A trusted platform is described as a secure computing platform that preserves the privacy of the user by providing the following three functionalities:

Protected Storage. Data on the TP is protected from unauthorized access.

Integrity. The TP can prove that it is running only the authorized software and no malicious code.

Anonymity. The TP can demonstrate that it is a genuine TP without revealing the identity of the user. The TP uses a pseudonym attested by a PKI Certification Authority (CA).

A Trusted Platform comprises of a Trusted Platform Module (TPM), which is a hardware device with cryptographic functions that enable the various security functionalities of the TP. The TPM is unforgeable and tamper-resistant.

5.7.3 System Model and Functionality

An agent in the system can take up one of following three roles at any given time: *recommender*, *requester*, and *accumulator*.

Recommender. A recommender agent has interacted directly with other agents and has feedback about them. He regularly announces the availability of feedback to other agents in the system. A recommendation comprises of the target agent's pseudonym, the recommender agent's pseudonym, and the feedback value. The recommendation is digitally signed by the recommender.

Accumulator. An accumulator agent stores feedback about other agents. However, his feedback is not based on direct experience with the target agent but formed through the feedback that he has received from other agents in the system.

Requester. A requester agent queries other agents for feedback and then locally aggregates the feedback to determine the reputation of the target agent. A requester agent propagates the query to its peer agents who in turn propagate to their peer agents. Each peer decides when to discontinue further propagation based on whether recommendations are available among its peers. The requester agent receives the feedback from the recommender and accumulator agents queried and then aggregates the feedback to learn the reputation of the target agent.

It is not elaborated how the feedback announcement and feedback query protocols work, for example, if an algorithm such as broadcast or gossip is used. As a consequence, the communication complexity of the protocols is not clear. Moreover, the mechanism for aggregating the feedback is not discussed.

5.7.4 How Security is Achieved

The security requirements are fulfilled as follows:

- An attacker is unable to provide false feedback on an honest user's behalf since each feedback is digitally signed by the recommender. A requester agent can also verify through the recommender's TP that it has not been compromised by the adversary.
- An attacker is unable to access an honest user's private database and modify data such as feedback, reputation, etc. This is achieved due to the protected data storage functionality of the TP. Therefore, a requester can be certain that the given feedback is not false.
- An attacker does not learn the true identity of a feedback provider since only pseudonyms are used. Thus, a user is able to provide feedback anonymously and without inhibition. The pseudonym is protected by the TP and the CA of the user. Moreover, the use of MIX cascades is suggested to

prevent the attacker from correlating the pseudonym with the IP address of the user.
- In case of legal justification, the CA of a user can reveal his true identity.

Voss et al. [57] and Bo et al. [8] also present decentralized systems that are based on similar lines. They both suggest using smart cards as the trusted hardware modules. A later system by Kinateder et al. [38] avoids the hardware modules, however, it requires an anonymous routing infrastructure at the network level.

The reputation systems has some disadvantages. A sale on an e-commerce system may result in the disclosure of the true identities of the seller and the buyer to each other (through mailing addresses etc.), even if they use anonymous pseudonyms. We must also consider that the privacy of the pseudonym itself may need to be protected. For example, if pseudonym A assigns pseudonym B negative feedback in retaliation, then B's reputation is adversely affected due to the lack of privacy of B's feedback. Better solutions include: preserving the privacy of the feedback, or using disposable pseudonyms, which a user may change after every transaction (such as in the solution by Androulaki et al. [2]).

Table 11 Kinateder and Pearson [37] – A Privacy-Enhanced P2P Reputation System.

Architecture	Decentralized
Target Environment	Peer-to-peer systems
Adversarial Model	Malicious (Disruptive)
Key Security Mechanisms	Trusted platform, MIX cascades, Digital signatures
Privacy Guarantee	Security goals are satisfied if the underlying primitives (trusted platform, MIX cascades, digital signatures) are secure
Complexity (Messages)	Not Provided

6 Discussion

Tables 12 and 13 provide a comparison of the reputation systems that aim to preserve privacy under the semi-honest adversarial model and the disruptive malicious adversarial model respectively.

6.1 The Semi-Honest Adversarial Model

The Secure Sum protocol is simple and efficient. However, secure sum is secure only under a restricted semi-honest adversarial model where the entities are

Table 12 Literature – Privacy under the Semi-Honest Adversarial Model.

System / Protocol	**Architecture**	**Target Environment**	**Key Security Mechanisms**	**Privacy Guarantee**	**Complexity (Messages)**
Clifton et al. [16] – Secure Sum	D	Distributed environments	Secure multi-party computation	Probability: $\frac{1}{m+1}$, only if nodes don't collude	$O(n)$, where $n =$ number of sites
Pavlov et al. [46] – WSS-1	D	Distributed environments	Secure multi-party computation, secret sharing	$(1 - \frac{1}{n})(\frac{N-b-1}{N-1})$	$O(N) + O(n^2)$, where $N =$ no. of potential witnesses, and $n =$ no. of selected witnesses
Gudes et al. [28] – Scheme 1	D	Distributed environments	TTP, Public-key cryptography	Random guess across $\lvert TrustSet_x(A)\rvert$	$O(n)$, where $n = \lvert TrustSet_x(A)\rvert$
Gudes et al. [28] – Scheme 2	D	Distributed environments	TTP, Public-key cryptography, Secure product	Random guess across $\lvert TrustSet_x(A)\rvert$	$O(n)$, where $n = \lvert TrustSet_x(A)\rvert$
Gudes et al. [28] – Scheme 3	D	Distributed environments	Secure multi-party computation	A does not learn more information about $DTE(B, x)$, where $B \in TrustSet_x(A)$	$O(n^2)$, where $n = \lvert TrustSet_x(A)\rvert$
Hasan et al. [32]	D	Distributed environments	Secure multi-party computation, Trust awareness, Secret sharing	If k trustworthy agents in the set S_t, $k \ll n$	$O(n)$, where n is the number of feedback providers
Nin et al. [45]	D	Private collaborative networks	ElGamal encryption scheme	If the underlying encryption scheme is secure	$O(1)$

not allowed to collude. The protocol is therefore not suitable for preserving privacy under the more realistic model where collusion is possible.

The schemes 1 and 2 by Gudes et al. provide security under the full semi-honest model. However, both schemes rely on Trusted Third Parties (TTPs). The issue with TTPs is that if they are not fully honest, they can learn private data with little or no effort.

The reputation system by Nin et al. is very efficient. It requires exchange of a constant number of messages. However, the system is limited to Private Collaborative Networks, where reputation is computed based on the access control decisions of an entity. The reputation system is not applicable to more general social networks.

The protocol by Pavlov et al. (based on their first witness selection scheme) is secure under the full semi-honest model. Moreover, the protocol is general purpose, that is, it may be used for many different applications. The protocol also does not rely on any TTPs or centralized constructs. The scheme 3 by Gudes et al. has similar properties. However, both these protocols have communication complexity upwards of $O(n^2)$, which is quite expensive.

The protocol by Hasan et al. builds on secure multi-party computation, trust awareness, and secret sharing to achieve a low complexity of $O(n)$ messages, where n is the number of feedback providers. The privacy of an agent a is preserved with high probability if it finds k trustworthy agents in the set of feedback provider agents S_t, such that $k \ll n$ and the probability that all k agents will collude to break agent a's privacy is low.

6.2 The Disruptive Malicious Adversarial Model

Table 13 Literature – Privacy under the Disruptive Malicious Adversarial Model.

System / Protocol	**Architecture**	**Target Environment**	**Key Security Mechanisms**	**Privacy Guarantee**	**Complexity (Messages)**
Pavlov et al. [46] – WSS-2	D	Distributed environments	Verifiable secret sharing, discrete log commitment	If $b < \frac{N}{2} - n$	$O(n^3)$, where n = number of witnesses
Androulaki et al. [2]	C	Centralized systems, Peer-to-peer systems	Anonymous credential systems, E-cash (bank), Blind signatures, Mixnets / Onion Routing	If the underlying primitives (anonymous credential system, e-cash system, and blind signatures) are secure	$O(1)$
Kinateder and Pearson [37]	D	Peer-to-peer systems	Trusted platform, MIX cascades, Digital signatures	If the underlying primitives (trusted platform, MIX cascades, digital signatures) are secure	Not Provided
Steinbrecher [56]	C	E-commerce, Self-help forums, etc.	Pseudonym / Identity management	If the provider (central server) is honest and secure	$O(1)$

The reputation systems by Androulaki et al. and Steinbrecher are very efficient. They require a constant number of messages to be exchanged despite the number of feedback providers and the size of the system. However, each of these systems relies on a centralized construct. The reputation system by Androulaki et al. is based on the E-Cash system, which uses a centralized construct called the bank. Steinbrecher's reputation system has a central server as an integral part of its architecture. These centralized entities make these two systems unsuitable for fully decentralized environments.

Kinateder et al.'s reputation system provides anonymity in peer-to-peer systems under the disruptive malicious model. However, the system requires the presence of special hardware called Trusted Platform (TP) at each peer. Additionally, the system requires that messages be exchanged using MIX cascades. These requirements limit the reputation system to specialized networks where TPs are available at each peer and where MIX cascades are in use.

The protocol by Pavlov et al. (based on their second witness selection scheme) is secure under the disruptive malicious model. The protocol does not require centralized constructs or specialized networks. However, the issue with the protocol is that it needs $O(n^3)$ messages to be exchanged, which is very expensive.

References

1. A. Amirbekyan and V. Estivill-Castro. A new efficient privacy-preserving scalar product protocol. In *Proceedings of the Sixth Australasian Conference on Data Mining and Analytics*, 2007.
2. E. Androulaki, S. G. Choi, S. M. Bellovin, and T. Malkin. Reputation systems for anonymous networks. In *Proc. of PETS'08*, 2008.
3. M. J. Atallah and W. Du. Secure multi-party computational geometry. In *Proceedings of the Seventh International Workshop on Algorithms and Data Structures (WADS 2001)*, 2001.
4. M. Belenkiy, M. Chase, C. C. Erway, J. Jannotti, A. Kupcu, A. Lysyanskaya, and E. Rachlin. Making p2p accountable without losing privacy. In *Proceedings of the 2007 ACM Workshop on Privacy in Electronic Society*, 2007.
5. M. Belenkiy, M. Chase, M. Kohlweiss, and A. Lysyanskaya. P-signatures and noninteractive anonymous credentials. In *Theory of Cryptography*, 2008.
6. E. Bertino, E. Ferrari, and A. C. Squicciarini. Trust-x: A peer-to-peer framework for trust establishment. *IEEE Transactions on Knowledge and Data Engineering*, 16(7):827 – 842, July 2004.
7. G. A. Bigley and J. L. Pearce. Straining for shared meaning in organization science: Problems of trust and distrust. *Acad. Management Rev.*, 23(3):405421, 1998.
8. Y. Bo, Z. Min, and L. Guohuan. A reputation system with privacy and incentive. In *Proceedings of the Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD'07)*, 2007.
9. R. Burt. *Structural Holes: The Social Structure of Competition.* Harvard University Press, 1995.
10. J. Camenisch and A. Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *EUROCRYPT 2001*, 2001.
11. J. Camenisch, A. Lysyanskaya, and M. Meyerovich. Endorsed e-cash. In *Proceedings of the IEEE Symposium on Security and Privacy*, 2007.
12. L. Capra. Engineering human trust in mobile system collaborations. In *Proceedings of the 12th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, Newport Beach, CA, USA, 2004.
13. D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):8488, 1981.
14. D. Chaum. Blind signatures for untraceable payments. In *Proc. Advances in Cryptology (CRYPTO '82)*, 1982.
15. D. Chaum. Blind signature systems. In *Advances in Cryptology (CRYPTO'83)*, 1983.
16. C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, and M. Y. Zhu. Tools for privacy preserving distributed data mining. *SIGKDD Explorations*, 4(2):28–34, January 2003.
17. R. Dingledine, M. J. Freedman, D. Hopwood, and D. Molnar. A reputation system to increase mix-net reliability. In *Proceedings of the 4th International Workshop on Information Hiding*, 2001.
18. R. Dingledine, N. Mathewson, and P. Syverson. Reputation in privacy enhancing technologies. In *Proceedings of the 12th Annual Conference on Computers, Freedom and Privacy*, 2002.

19. R. Dingledine, N. Mathewson, and P. Syverson. Reputation in p2p anonymity systems. In *Proceedings of the Workshop on Economics of Peer-to-Peer Systems*, 2003.
20. R. Dingledine, N. Mathewson, and P. F. Syverson. Tor: The second-generation onion router. In *Proceedings of the USENIX Security Symposium*, 2004.
21. W. Du. *A Study of Several Specific Secure Two-Party Computation Problems*. PhD thesis, Purdue University, West Lafayette, IN, USA, 2001.
22. T. ElGamal. A public-key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, IT-31(4):469472, 1985.
23. N. Gal-Oz, E. Gudes, and D. Hendler. A robust and knot-aware trust-based reputation model. In *Proceedings of the Joint iTrust and PST Conferences on Privacy, Trust Management and Security (IFIPTM 2008)*, 2008.
24. D. Gambetta. *Trust: Making and Breaking Cooperative Relatioins*, chapter Can We Trust Trust?, pages 213 – 237. Department of Sociology, University of Oxford, 2000.
25. E. Gilbert and K. Karahalios. Predicting tie strength with social media. In *In Proceedings of the Conferece on Human Factors in Computing Systems (CHI09)*, 2009.
26. O. Goldreich. *The Foundations of Crypto. - Vol. 2.* Cambridge Univ. Press, 2004.
27. M. Granovetter. The strength of weak ties. *American Journal of Sociology*, 78:1360–1380, May 1973.
28. E. Gudes, N. Gal-Oz, and A. Grubshtein. Methods for computing trust and reputation while preserving privacy. In *Proc. of DBSec'09*, 2009.
29. R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *Proceedings of the International World Wide Web Conference (WWW 2004)*, 2004.
30. O. Hasan, E. Bertino, and L. Brunie. Efficient privacy preserving reputation protocols inspired by secure sum. In *Proceedings of the 8th International Conference on Privacy, Security and Trust (PST 2010)*, Ottawa, Canada, August 17-19 2010.
31. O. Hasan, L. Brunie, and E. Bertino. k-shares: A privacy preserving reputation protocol for decentralized environments. In *Proceedings of the 25th IFIP International Information Security Conference (SEC 2010)*, pages 253–264, Brisbane, Australia, September 2023 2010.
32. O. Hasan, L. Brunie, and E. Bertino. Preserving privacy of feedback providers in decentralized reputation systems. *Computers & Security*, 31(7):816 – 826, October 2012.
33. K. Hoffman, D. Zage, and C. Nita-Rotaru. A survey of attack and defense techniques for reputation systems. *ACM Computing Surveys*, 41(4), December 2009.
34. G. Homans. *The Human Group*. Harcourt, Brace, & World, New York, 1950.
35. I. Ioannidis, A. Grama, and M. Atallah. A secure protocol for computing dot-products in clustered and distributed environments. In *Proceedings of the 2002 International Conference on Parallel Processing*, 2002.
36. A. Josang and R. Ismail. The beta reputation system. In *Proceedings of the 15th Bled Electronic Commerce Conference*, Bled, Slovenia, 2002.
37. M. Kinateder and S. Pearson. A privacy-enhanced peer-to-peer reputation system. In *Proc. of the 4th Intl. Conf. on E-Commerce and Web Technologies*, 2003.
38. M. Kinateder, R. Terdic, and K. Rothermel. Strong pseudonymous communication for peer-to-peer reputation systems. In *Proceedings of the 2005 ACM symposium on Applied computing*, 2005.
39. L. Kissner. *Privacy-Preserving Distributed Information Sharing*. PhD thesis, Computer Science Department, Carnegie Mellon University, PA, USA, July 2006. CMU-CS-06-149.
40. N. Lin, W. M. Ensel, and J. C. Vaughn. Social resources and strength of ties: Structural factors in occupational status attainment. *American Sociological Review*, 46(4):393 – 405, 1981.
41. P. V. Marsden and K. E. Campbell. Measuring tie-strength. *Social Forces*, 63:482 – 501, 1984.
42. D. H. McKnight, L. L. Cummings, and N. L. Chervany. Initial trust formation in new organizational relationships. *Acad. Management Rev.*, 23(3):473490, 1998.

43. P. Mika and A. Gangemi. Descriptions of social relations. Technical report, Department of Business Informatics, Free University Amsterdam, The Netherlands, Retrieved February 17, 2011 2011.
44. C. Mitchell, editor. *Trusted computing.* Institution of Electrical Engineers, 2005.
45. J. Nin, B. Carminati, E. Ferrari, and V. Torra. Computing reputation for collaborative private networks. In *Proceedings of the 33rd Annual IEEE International Computer Software and Applications Conference*, 2009.
46. E. Pavlov, J. S. Rosenschein, and Z. Topol. Supporting privacy in decentralized additive reputation systems. In *Proceedings of the Second International Conference on Trust Management (iTrust 2004)*, Oxford, UK, 2004.
47. S. Pearson and B. Balacheff, editors. *Trusted Computing Platforms: TCPA Technology in Context.* Prentice Hall, 2003.
48. T. P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology*, 1991.
49. J. W. Pennebaker, M. E. Francis, and R. Booth. *Linguistic Inquiry and Word Count: LIWC2001.* Erlbaum Publishers, Mahwah, NJ, 2001.
50. A. Petroczi, T. Nepusz, and F. Bazso. Measuring tie-strength in virtual social networks. *Connections*, 27(2):39 – 52, 2007.
51. F. Pingel and S. Steinbrecher. Multilateral secure cross-community reputation systems for internet communities. In *Proceedings of the Fifth International Conference on Trust and Privacy in Digital Business (TrustBus 2008)*, 2008.
52. J. A. Pouwelse, P. Garbacki, D. H. J. Epema, and H. J. Sips. The bittorrent p2p file-sharing system: Measurements and analysis. In *Peer-to-Peer Systems IV*, 2005.
53. P. Resnick and R. Zeckhauser. Trust among strangers in internet transactions: Empirical analysis of ebay's reputation system. *Volume 11 of Advances in Applied Microeconomics*, pages 127–157, 2002.
54. P. Resnick, R. Zeckhauser, E. Friedman, and K. Kuwabara. Reputation systems. *Communications of the ACM*, 43(12):4548, December 2000.
55. S. Schiffner, S. Clau, and S. Steinbrecher. Privacy and liveliness for reputation systems. In *Proc. of EuroPKI'09*, pages 209 – 224, 2009.
56. S. Steinbrecher. Design options for privacy-respecting reputation systems. In *Security and Privacy in Dynamic Environments*, 2006.
57. M. Voss, A. Heinemann, and M. Muhlhauser. A privacy preserving reputation system for mobile information dissemination networks. In *Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM)*, 2005.
58. B. Wellman and S. Wortley. Different strokes from different folks: Community ties and social support. *The American Journal of Sociology*, 96(3):558 – 588, 1990.
59. A. C. Yao. Protocols for secure computations. In *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, 1982.