Stan Gurtler* and Ian Goldberg

SoK: Privacy-Preserving Reputation Systems

Abstract: Trust and user-generated feedback have become increasingly vital to the normal functioning of the modern internet. However, deployed systems that currently incorporate such feedback do not guarantee users much in the way of privacy, despite a wide swath of research on how to do so spanning over 15 years. Meanwhile, research on systems that maintain user privacy while helping them to track and update each others' reputations has failed to standardize terminology, or converge on what privacy guarantees should be important. Too often, this leads to misunderstandings of the tradeoffs underpinning design decisions. Further, key insights made in some approaches to designing such systems have not circulated to other approaches, leaving open significant opportunity for new research directions. This SoK investigates 42 systems describing privacy-preserving reputation systems from 2003–2019 in order to organize previous work and suggest directions for future work. Our three key contributions are the systematization of this body of research, the detailing of the tradeoffs implied by overarching design choices, and the identification of underresearched areas that provide promising opportunities for future work.

Keywords: anonymity, privacy, reputation

DOI 10.2478/popets-2021-0007
Received 2020-05-31; revised 2020-09-15; accepted 2020-09-16.

1 Introduction

Significant attention has been given to the internet and its ability to connect people with unprecedented amounts of information, as well as to large-scale changes to society that followed this connection. *Disruption* is the term of art, with companies like Amazon, AirBnB, and Uber all causing dramatic changes to the industries they inhabit [3]. In many cases, this disruption is founded upon the ability to connect people with each other, rather than with information. Without the ability to refer to feedback shared by others on their experiences purchasing goods, staying in short-term rentals, or ridesharing, users would have a great deal of difficulty placing their trust into such online platforms.

Outside of the realm of these disruptors, the internet has had a marked effect to enable the formation and maintenance of communities. Some of these communities were limited by restrictive policies, such as LGBTQ [41] individuals from geopolitical regions which may be hostile towards such individuals [63]. Others were limited more by the fact that members did not previously realize there were others with which to form a community, like the many fandoms that crop up for even obscure media on sites like Reddit [52, 7:45]. However, communities on the internet frequently must endure bad actors. Through sites like You Got Posted [57], 8chan [2], and Kiwi Farms [51], malicious individuals gather to compare notes, stalk, and harass marginalized communities and their members, frequently infiltrating such groups to gather information on members. Communities have responded to these infiltrations with varying success, like warning local police that malevolent actors may attempt to deceive dispatchers into sending violently armed police response teams to their addresses, or withdrawing entirely from the internet [2]. Infiltrations have had effects on real-world events as well, causing the cancellations of gatherings offline [38] or the pre-emptive expulsion of individuals believed to be harmful to the group from such gatherings [21]. Online communities have to this point found it difficult to protect themselves from such infiltrations. Some websites, such as Reddit, employ *reputation* as a measure to help users identify and deal with misbehaviour in their communities.

In both the case of online platforms (*e.g.*, Amazon) and online communities, users depend on feedback generated by other users. They may do so to determine who to engage in transactions with or to properly weigh the input of a user based on what opinion the community holds of them. In either case, the feedback helps guide users in their interactions. It makes sense that in transactional relationships like those found on Amazon or eBay, which both employ reputation, users may desire some or all of their transactions to be conducted privately. This desire creates a potential for tension between maintaining the privacy of one's actions using such services while still being able to find and provide

***Corresponding Author: Stan Gurtler:** University of Waterloo, E-mail: tmgurtler@uwaterloo.ca
Ian Goldberg: University of Waterloo, E-mail: iang@uwaterloo.ca

suitable, valid feedback to help guide future users. Users sometimes desire anonymity in community settings as well. Notably, previous work has investigated how ratings may reveal sensitive attributes about users [36]. On the face of it, preserving anonymity seems incompatible with maintaining reputation of any sort. However, work has been conducted on diminishing this incompatibility. Some systems, for example, allow users to generate new pseudonyms that continue to carry their old reputation.

Despite the bevy of work in this area, there has been little standardization or convergence. Papers often reinvent their own terminology, making works difficult to compare. When authors do compare their work to others', there is not consistent agreement on what privacy properties are actually important to provide. This disagreement leads to comparisons that obfuscate and ignore the tradeoffs underpinning certain design choices. Few papers spend time considering how reputation scores are determined from the feedback provided, which can have dramatic impact on the mechanisms that provide certain privacy properties. Some papers explicitly leave this open-ended, purporting to support a variety of reputation mechanisms, but in fact only support reputation mechanisms of a specific nature.

In this work, we investigate 42 systems on privacy-preserving reputation systems dating between 2003 and 2019. We identify five broad approaches in the designs of these systems and specify the tradeoffs that these large-scale design choices imply. We specify a variety of reputation functions of multiple types, and highlight how design choices enable or limit the use of these reputation functions. Our contributions are threefold:

1. We organize the disparate works, so that future research may more easily compare itself to previous works. We propose a set of criteria that is able to systematize research and readily identify the improvements new work makes upon the state of the art.
2. We detail the tradeoffs implied by overarching design choices, so that future research may identify bounds on where it could expect to make improvements, and directly enable researchers to identify promising areas for future exploration.
3. We identify areas where properties considered in certain approaches have not circulated to other approaches. These areas suggest the opportunity for large changes even within existing approaches, and future works may be able to apply these changes to allow for systems with a more useful combination of properties. In particular, we observe that:
   - there are underresearched paths to decentralization other than blockchain approaches;
   - the ability to accrue reputation without linking one's actions together would be of significant benefit in community settings and should be explored more deeply; and
   - work to this point does not explore the utility of reputation for or by groups.

In this paper, we proceed as follows. In Section 2, we describe how we chose the papers we consider in this systematization. In Section 3, we define terminology for several high-level design choices relevant to all privacy-preserving reputation systems. In Section 4, we systematize classes of reputation functions and outline the design choices necessary to enable them in a reputation system. In Section 5, we systematize the literature into five broad approaches and provide insight into the important differences in provided properties they imply. In Section 6, we identify opportunities for future research, and in Section 7, we conclude.

2 Methodology

In this systematization, we conducted our search for papers by starting with one seed paper, AnonRep [66]. From this seed, we examined every paper that it cites and that cites it (as recognized by Google Scholar in September 2019). We found 73 such papers. Papers were then included in our systematization if and only if both of the following were true. First, they described systems that supported a "vote" operation, where one or more voters gave feedback representing their opinion of a votee. To allow for variety, this criterion was not specified further. Second, they preserved at least one of our recognized privacy properties during said vote operation. These conditions captured 14 and excluded 59 papers from this set. All 59 excluded papers failed to implement a vote operation meeting our definition. This procedure was iteratively repeated for all included papers from this set until convergence was reached, resulting in 42 systems described across 45 papers. The mapping of properties in Appendix A (see Section 3) and the classification of systems were both coded by a single author.

3 Terminology

As noted before, previous work has not converged on a standard set of terms in order to describe their systems. In cases where the terms themselves are consistent, the definitions used with these terms have at times obscured important distinctions in underlying design. In Appendix A, we examine the many different terms that have been used in previous work across all the categories we describe in this section. Throughout this paper, we will use a few specific terms to refer to participants in reputation systems. We refer to a user who contributes feedback for another party as a "voter". We refer to a user for whom feedback is contributed as a "votee". In both cases, these users may refer to an individual or an organization as necessary. Where relevant, a user who requests a reputation of a votee is a "requester". In this section, we elaborate on three key areas that have been addressed inconsistently in the past: architecture, reputation directionality, and privacy properties.

3.1 Architecture

In reputation systems, the integrity of reputations must be preserved. If votees were allowed to interfere with voters who would rate them negatively and prevent those ratings, they could artificially raise their own reputation scores. On the other hand, if malevolent actors were allowed to post ratings indiscriminately, they could artificially lower the reputation scores of votees. In Sybil attacks specifically, users may be able to perform this ballot-stuffing and badmouthing by creating arbitrary numbers of identities with which to participate in the system. This style of attack has been well described in previous work [22, 23, 25, 42, 58, 65]. The designers of reputation systems may turn to a variety of strategies in order to protect against such malicious actions; largely, they rely on one of the following three:

Third-Party Mediation: A reputation system may designate one or more trusted third parties (TTPs) to be responsible for the integrity of the reputation scores. Reputation systems may also designate one or more TTPs to be responsible for the privacy of the users in the system. The TTPs' involvement can take several forms, and differing amounts of trust may be placed in them. In some systems, the TTPs bootstrap the system but may not be required for its ongoing operation, such as when group signature schemes are used. In others, the TTPs only serve to audit interactions. In still others, the TTPs intermediate all interactions. Some systems use only one TTP, others may use multiple, and still others require multiple, often to try to break apart centralized roots of trust. These systems are frequently called "centralized".

Ephemeral Mesh Topology: In some systems, reputation is not a global, persistent value, but is instead calculated when it is requested. Requesters are responsible for interacting directly with voters to solicit their individual evaluations of a votee. So long as requesters can confirm they are interacting with the voters they intend to, the procedure used to combine reputation scores in such systems guarantees that each participant may only contribute one evaluation. Some systems additionally allow requesters to weight the importance of voters' contributions by how much they themselves trust the voters. Requesters are typically free to choose which voters they intend to query and are not required to always choose the same voters for each request. We term these systems "user-defined decentralized".

Proofs of Validity: In some systems, voters contribute their feedback for votees directly to all other users, such as via an append-only public bulletin board. Proofs of the integrity and validity of votes, then, must be derived using additional information. This often takes the form of proofs of knowledge of specific secret values that indicate a voter has undergone a transaction with the votee, without specifying which transaction. In such a system, careful attention must be placed on how the bulletin board is maintained. The system would not be useful if users could not agree on which feedback is valid, and so the system must remove the potential for abuse in reaching this agreement. While this approach is certainly decentralized, it has clear differences in its manner of decentralization than systems that use an Ephemeral Mesh Topology. As such, we term these systems "system-defined decentralized".

While a majority of systems in the literature use Third-Party Mediation, Proofs of Validity have been an increasingly attractive approach to designers of reputation systems. Methods of incorporating such proofs even where TTPs are still being used may help distribute trust in the system away from centralized nodes.

3.2 Reputation Directionality

eBay was one of the earliest-used reputation systems. In eBay's reputation system, buyers and sellers both participate in rating one another, and reputation has different roles in determining how to interact with buyers and sellers. In other systems, such as Amazon's, buyers rate sellers, but there is no clear mechanism for buyers themselves to be rated. In still other systems, such as Reddit's, all participants rate each other, with no distinctions being made between "types" of user. We suggest a classification of reputation systems into three kinds according to how their ratings are organized.

Simplex Reputation Systems ($C \rightarrow S$): In a simplex reputation system, there are two sets of participants. One set, C, represents the clients or consumers in the system. The other set, S, represents the servers or sellers in the system. Clients may assign ratings, but have no ratings associated with themselves; even when there is an overlap between C and S, a participant acting as a client does not have their server rating associated with their client activity. On the other hand, servers receive reputations, and have these reputations displayed in a manner that clients can observe and use to inform their decisions about future interactions. Amazon is an example of a simplex reputation system.

Half-Duplex Reputation Systems ($C \leftrightharpoons S$): In a half-duplex reputation system, there are two sets of participants. One set, C, represents the clients or consumers in the system. The other set, S, represents the servers or sellers in the system. Clients may assign ratings to servers, and servers may assign ratings to clients. When there is an overlap between C and S, a participant has two different ratings that do not impact one another, and are only used in the appropriate settings where they behave as a client or as a server. As both clients and servers receive reputations, both clients and servers can observe the role-specific reputations of one another and base decisions about future interactions upon them. eBay is an example of a half-duplex reputation system.

Full-Duplex Reputation Systems ($P \leftrightarrow P$): In a full-duplex reputation system, there is only one set of participants, P, representing the peers or participants in the system. Peers assign ratings to one another, and there are no structural distinctions between peers who give ratings and peers who receive ratings. Peers can observe the reputations of one another and base decisions about future interactions upon them. Reddit is an example of a full-duplex reputation system.

3.3 Privacy Properties

While privacy-preserving reputation systems naturally must do something to protect user privacy, the exact nature of these privacy protections varies between systems. We highlight four privacy properties — two for voters and two for votees — that a privacy-preserving reputation system may provide. In all cases, it may be possible to provide said property with respect to one of the following sets: all parties not involved in a transaction, all parties except TTPs, or all parties without restriction. We note that, following the example set by Kuhn *et al.* [40], we avoid the word "anonymity" in the names of these properties, as we feel that term may be unclear and overloaded. We refer the interested reader to Appendix A for further discussion of such choices.

Voter Privacy Properties

Voter-Vote Unlinkability: In order to avoid concerns that a voter may face coercion or backlash for their vote, it may be desirable for a voter to cast a vote secretly. That is, Voter-Vote Unlinkability is provided when a voter cannot be associated with a vote they cast, or with the fact that they voted for a particular votee.

Two-Vote Unlinkability: While voters may be unlinkable to their votes, this does not preclude the possibility that users may be able to identify that two votes came from the same voter. This may be undesirable, as more votes cast reduces a voter's anonymity set and allows behavioural tracking. Thus, Two-Vote Unlinkability is provided when it is not possible to distinguish whether two votes were cast by the same voter or not.

Votee Privacy Properties

Reputation-Usage Unlinkability: It may be desirable for votees to be provided privacy as well. In these cases, reputation still must have some meaning, and must still be able to be accumulated, but it may be desirable for votees to produce a proof of their reputation without linking themselves to a long-term pseudonym associated with that reputation. Thus, Reputation-Usage Unlinkability is provided when a votee can display or use their reputation and accumulate new votes without enabling others to identify that another specific reputation use was also performed by the same votee.

Exact Reputation Blinding: For the purposes that reputation serves, it can be sufficient to know that a votee's reputation is above some threshold. Displaying a votee's precise reputation score may in fact be undesirable, as it can be observed to track the votee across usages or to infer a voter's vote for a votee. Thus, Exact Reputation Blinding is provided when a system provides a mechanism for votees to display or use their reputation without giving an exact score.

4 Reputation Functions

In order to provide simple and interpretable reputation scores for requesters to observe, reputation systems typically feature a method of summarizing the frequently large set of ratings received about participants into a single value. We note that systems in the literature only consider reputation referring to opinions about specific users. However, reputation in the real world can often refer to opinions about groups instead. For example, the approval of a populace towards its government and elected officials is frequently used as a measure of the government's performance. Extending the use of reputation systems to these cases, such as by evaluating the combination of reputation scores about multiple users within systems, could prove useful to explore further.

As deployed in the real world, reputation systems frequently feature one of two different methods of summarizing ratings. First, reputation may be represented as an average of ratings; second, reputation may be represented as a sum of ratings. We term such summarizations *reputation functions*. Though reputation systems in practice largely use one of these two reputation functions, previous work [19, 54] has investigated a wider range of potential options. We present here a categorization of reputation functions that have been used and suggested in previous work or in real-world deployments.

4.1 Voter-agnostic Reputation Functions

The most common reputation functions in use do not take into account the voter who assigned a rating when calculating a reputation score. (Note that this notion is distinct from the question of *when* a particular voter is allowed to submit a vote, as we will discuss in Section 5.) An example is a system like Reddit's, where voters may vote up or down on posts made by other votees, and all of those votes have equal impact on that votee's reputation score regardless of who made the votes. Here, we highlight three such functions from previous work.

Accrue Stars: In Accrue Stars, a votee's reputation is the sum of the votes cast for them. In some systems, a voter may only either vote to indicate approval or abstain to indicate disapproval. However, in other systems, a voter may have three options: a positive vote, abstention, or a negative vote. Where relevant, we specify the version of Accrue Stars that incorporates negative votes as "Accrue Stars — Negative".

Average Stars: In Average Stars, a votee's reputation is the mean of the votes cast for them. Voters typically have a range of options to give feedback for a votee (*e.g.*, one to five stars). This is the style of ratings eBay, among others, uses. System designers may find it useful to consider alterations of this function, particularly when more dimensionality is added, such as by taking a mean weighted by the recency of votes.

Gompertz function: The Gompertz function is specifically suggested by Huang *et al.* [32]. This function takes several parameters that must be chosen deliberately such that the function allows reputations to slowly increase but quickly decrease. In this way, the Gompertz function is intended to model the trust of humans in social interactions, which takes time to build up, but can be lost quickly. Votes are real numbers between 0 and 1, inclusive, as are the summary scores. The system operates with respect to regular divisions of time or "epochs". Votes are cast once per epoch, and more recent epochs are weighted more highly in the output of the function. Votes are also normalized per epoch; if all votees receive high ratings in one epoch, the effect is the same as if all votees receive low ratings. For more detail on this function, see Appendix B.

4.2 Voter-conscious Reputation Functions

Less commonly, reputation functions do account for the voter who assigned a rating when calculating reputation scores. The most common cases of this occur when voters can only give one rating per votee but may update this rating. That is, when they vote, they do not add a new input to the reputation calculation. Additionally, by taking the voters into account, we can generalize the above sums and averages of votes into weighted sums and averages, using information about the voter, such as their own reputation, as a weight for each vote.

Further, we note that, when every voter in a reputation system contributes exactly one feedback for a votee, it is natural to interpret the combination of this feedback as the consensus of the voters about a votee. Consensus in this manner clearly can be useful for an individual to determine their level of interaction with a votee. However, reputation in the real world can also often refer to the opinions of specific groups of individuals, and the opinions of different groups may have different meanings to an individual. That is, whereas above, we discussed the utility of systems that allow users to calculate and observe reputations *of* specified groups, such as political parties, here we are considering the utility of systems that allow users to calculate and observe reputations created *by* specified groups. While some systems in the literature can support the calculation of such voter-group reputation, further support may be beneficial.

Here, we highlight two functions that do account for the voters when calculating summary scores.

Short-term Memory Consensus: In Short-term Memory Consensus (STMC), each voter has one mutable vote to assign to each votee. A votee's rating at a particular time represents the consensus of what all voters think of them at that specific moment. In some scenarios, the voters' own reputations may affect how the votes they cast are weighted in computing the votee's summary reputation score. In such cases, a highly rated voter would have more influence on ratings, due to the increased trust placed in them by the community.

Long-term Memory Consensus: In Long-term Memory Consensus (LTMC), each voter has one mutable vote to assign to each votee. Like the Gompertz function, systems that use LTMC operate with respect to epochs. At the end of each epoch, the votes are tabulated as in STMC. The resulting score for each votee is then averaged with the score they held during the previous epoch; the weighting of the two values impacts the speed with which reputation updates. As votes are tabulated the same way as in STMC, the same voter-conscious rating weighting can also be used in LTMC.

5 Privacy-Preserving Reputation Systems

Though reputation systems have been widely used in the real world, systems that have been deployed frequently do not guarantee privacy to their users. Most commonly, these systems are highly linkable. That is, a user's actions in the reputation system can be linked together; these actions may include the votes they cast, the votes they receive, and the times they validate their reputation. Privacy-preserving reputation systems have been a line of research dating back to 2003 on how to build reputation systems that can prevent these linkages while preserving the integrity of the reputation system.

In Table 1, we systematize the strategies taken and the properties provided in privacy-preserving reputation systems in the literature. First, we identify four factors relating to a reputation system's structure. As defined in Section 3, we indicate centralization and reputation directionality. We also highlight two properties relevant to reputation scores. Reputation scope is identified as global (each votee has one score that all requesters see) or local (each score is dependent on which voters a requester works with to obtain a score). Reputation ownership is identified as votee-owned (a votee displays its own score with appropriate validation), TTP-owned (a requester obtains a votee's score from some third party with appropriate validation), or voter-owned (a requester obtains votes from voters for each votee). Table 1 also includes Amazon, eBay, and Reddit as well-known reputation systems for comparison's sake.

Second, we identify five factors related to the level of trust placed on third parties in the system. We identify when correctness is guaranteed by the protocols used in a system, versus when errors can be recognized and flagged by anyone, versus when it is left to the TTP (or, in one particular case, blockchain miners, who effectively act as a sort of distributed TTP) to handle correctness. We indicate whether TTPs are relied on to protect the privacy of users (that is, whether or not users can always have their behaviour linked by a TTP). The minimum number of TTPs required to use a system is identified, both in initial setup and for ongoing usage. Systems allowing additional TTPs to be added in an *anytrust* relationship — that is, the system's guarantees are upheld if *any one* of the TTPs is honest — are specially noted.

Third, we identify the privacy properties (as defined in Section 3.3) provided by each system.

Finally, we identify the reputation functions (as defined in Section 4) supported by each system.

Notably, we do not identify, nor does this section further elaborate on, evaluations of systems and threat models. Evaluations were not universally present in the systems under study. However, even if they were, it is difficult to compare evaluations that were performed in different environments and measured different components of the various systems. Threat models, likewise, were not universally present in the systems under study.

In Table 1, we systematize the strategies taken in the literature to design privacy-preserving reputation systems into five approaches. Figure 1 visualizes the interactions typical in each approach and demonstrates how the interactions proceed. The rest of this section further elaborates on these five approaches:

- Coin-based Reputation Systems
- Signature-based Reputation Systems
- Reputation Transfer
- SMC-based Reputation Systems
- Ticket-based Reputation Systems

5.1 Coin-based Reputation Systems

Among privacy-preserving reputation systems, the earliest work was done on *coin-based reputation systems*. These systems are based upon e-cash designs; reputation is treated as a currency. Voters are granted reputation points (or "repcoins") that they may hand out to votees in the system. These repcoins are limited in some fashion to prevent reputation inflation; for example, voters may only get a set number of points to spend per epoch. Voters spend repcoins by sending them to

Table 1. Privacy-Preserving Reputation Systems

Name	*Year*	*Centralization*	*Directionality*	*Rep. Scope*	*Rep. Ownership*	*Correctness*	*Unlinkable to TTP*	*# TTP for Setup*	*# TTP Ongoing*	*More via Anytrust*	*V-V Unlinkability*	*2V Unlinkability*	*R-U Unlinkability*	*Exact Rep. Blinding*	*Acc. Stars*	*Acc. Stars — Neg.*	*Avg. Stars*	*Gompertz*	*STMC*	*LTMC*
System		**Structure**				**Trust**					**Privacy**				**Rep.**					
Coin-based Reputation Systems																				
Ismail et al. [35]	2003	★	↔	∀	☽	◐	-	2	2	-	●	●	◐	●	●	-	●	-	-	-
Voss [59]	2004	★	↔	∀	☽	◐	-	1	1	-	◐	-	◐	●	●	●	-	-	-	-
Androulaki et al. [4]	2008	★	↔	∀	☽	◐	◐	1	1	-	●	●	◐	●	●	-	-	-	-	-
Signature-based Reputation Systems																				
iClouds [60]	2005	★	↔	∀	∇	-	-	1	N	-	◐	-	◐	-	-	-	-	-	●	-
Signatures of Reputation [11]	2010	∴	↔	∀	∇	●	●	1	0	-	◐	-	●	●	●	-	-	-	-	-
Reputation Transfer																				
Anwar and Greer [5]	2006	★	↔	∀	☽	◐	-	1	1	-	-	-	●	-	*	*	*	*	*	*
RuP [45]	2006	★	↔	∀	☽	-	●	1	1	-	-	●	●	-	*	*	*	*	*	*
DARep [27]	2007	★	↔	∀	☽	-	-	1	N	-	-	●	●	-	-	●	-	-	-	-
Hao et al. [26]	2008	★	↔	∀	☽	-	-	1	1	-	-	●	●	-	-	-	-	-	-	●
Wei and He [62]	2009	★	↔	∀	☽	-	●	1	1	-	-	●	●	-	-	-	-	-	-	●
Peng et al. [49]	2010	★	↔	∀	☽	-	●	1	1	-	-	●	●	-	*	*	*	*	*	*
Huang et al. [33]	2012	★	→	∀	☽	●	-	1	1	-	-	-	●	●	-	-	-	●	-	-
IncogniSense [17]	2013	★	→	∀	☽	●	-	1	1	-	-	-	●	●	*	*	*	*	*	*
k-Anonymous Reputation [19]	2013	★	↔	∀	☽	-	-	1	1	-	●	●	-	-	*	*	*	*	*	*
SMC-based Reputation Systems																				
Kinateder and Pearson [39]	2003	☍	↔	∃	⊙	-	-	0	N	-	●	●	-	-	-	-	-	-	●	-
DARS [48]	2004	☍	↔	∃	⊙	●	●	0	0	-	●	●	-	-	-	-	-	-	●	●
PDSPP [64]	2007	☍	↔	∃	⊙	●	●	0	0	-	●	●	-	-	-	-	-	-	●	●
3PRep [47]	2009	☍	↔	∃	⊙	●	●	0	0	-	●	●	-	-	-	-	-	-	●	●
CRDSPP [1]	2009	☍	↔	∃	⊙	●	●	0	0	-	●	●	-	-	-	-	-	-	●	●
k-Shares [28–31]	2010	☍	↔	∃	⊙	●	●	0	0	-	●	●	-	-	-	-	-	-	●	●
PFWRAP [67]	2016	☍	↔	∃	⊙	●	●	0	0	-	●	●	-	-	-	-	-	-	●	●
Dyn-PDRS [18]	2017	☍	↔	∃	⊙	●	●	0	0	-	●	●	-	-	-	-	-	-	●	●
M2M-REP [7]	2018	☍	↔	∀	⊙	●	●	0	0	-	●	-	-	-	-	-	-	-	●	●
Ticket-based Reputation Systems																				
[TTP Approaches]																				
Amazon	1994	★	→	∀	☽	-	-	1	1	-	-	-	-	-	-	-	●	-	-	-
eBay	1995	★	⇌	∀	☽	-	-	1	1	-	-	-	-	-	-	●	-	-	-	-
Reddit	2005	★	↔	∀	☽	-	-	1	1	-	●	●	-	-	-	●	-	-	-	-
TrustMe [55]	2003	★	↔	∀	☽	●	●	1	0	-	●	-	-	-	●	●	●	-	-	-
Boyd et al. [13]	2004	★	↔	∀	∇	-	-	1	1	-	●	●	-	-	●	●	●	-	-	-
ARM4FS [46]	2008	★	→	∀	⊙	-	-	1	2	-	●	●	●	●	●	●	●	-	-	-
Kerschbaum [37]	2009	★	↔	∀	☽	◐	●	0	2	-	●	●	-	-	●	●	●	-	-	-
Hussain and Skillicorn [34]	2011	★	↔	∀	☽	-	●	1	1	-	●	●	-	-	●	●	●	-	-	-
ARTSense [61]	2013	★	→	∀	☽	●	-	1	1	-	-	-	●	-	●	●	●	-	-	-
Petrlic et al. [50]	2014	★	↔	∀	☽	●	●	1	1	-	●	●	-	-	●	●	●	-	-	-
AnonRep [66]	2016	★	↔	∀	☽	◐	◐	2	2	●	●	●	●	●	●	●	●	-	-	-
Bazin et al. [9]	2016	★	↔	∀	∇	◐	-	1	1	●	●	●	-	-	●	●	●	-	-	-
Busom et al. [14]	2017	★	⇌	∀	☽	●	●	1	1	-	●	●	-	-	●	●	●	-	-	-
Garms et al. [24]	2017	★	↔	∀	☽	●	-	1	1	-	●	●	-	-	●	●	●	-	-	-
Blömer et al. [12]	2018	★	↔	∀	☽	●	●	1	1	-	●	●	-	-	●	●	●	-	-	-
CLARC [10]	2018	★	↔	∀	☽	◐	●	1	1	-	●	-	-	-	●	●	●	-	-	-
PrivRep [8]	2018	★	↔	∀	⊙	-	●	1	1	-	●	●	-	-	-	-	-	-	●	-
pRate [44]	2019	★	↔	∀	∇	●	●	1	1	-	●	●	●	●	-	-	●	-	-	-
[Public Log Approaches]																				
Beaver [56]	2016	∴	→	∀	☽	◐	●	0	0	-	●	●	-	-	●	●	●	-	-	-
Schaub et al. [53]	2016	∴	→	∀	☽	◐	●	0	0	-	●	●	-	-	●	●	●	-	-	-
PrivBox [6]	2018	∴	→	∀	☽	◐	●	0	0	-	●	●	-	-	●	●	●	-	-	-
ARS-PS [43]	2019	∴	→	∀	☽	●	●	1	1	-	●	●	-	-	●	●	●	-	-	-

Centralization:	★ = Third-Party Mediation	☍ = Ephemeral Mesh Topology	∴ = Proofs of Validity
Directionality:	→ = Simplex	⇌ = Half-Duplex	↔ = Full-Duplex
Scope:	∀ = Global	∃ = Local	
Ownership:	∇ = Votee-owned	☽ = TTP-owned	⊙ = Voter-owned
Correctness: via...	● = ...protocol guarantees	◐ = ...errors are traceable	- = ...TTP/miners
Trust Unlinkability: TTP can link...	● = ...nothing	◐ = ...misbehaviour	- = ...everything

Privacy Unlinkability: ◐ = Participants to a transaction can link each other

Reputation: * = This work considers reputation functions to be outside its scope.

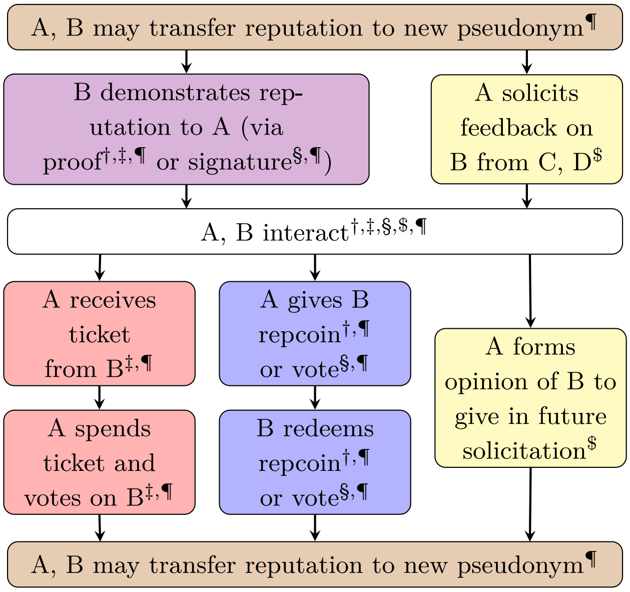
A, B may transfer reputation to new pseudonym¶
B demonstrates reputation to A (via proof†,‡,¶ or signature§,¶)
A solicits feedback on B from C, D$
A, B interact†,‡,§,$,¶
A receives ticket from B‡,¶
A gives B repcoin†,¶ or vote§,¶
A forms opinion of B to give in future solicitation$
A spends ticket and votes on B‡,¶
B redeems repcoin†,¶ or vote§,¶
A, B may transfer reputation to new pseudonym¶

Fig. 1. A system model visualization of the approaches discussed here. At each step, † (blue , violet) represents coin-based approaches, § (blue , violet) represents signature-based approaches, ¶ (brown) represents reputation transfer, $ (yellow) represents SMC-based approaches, and ‡ (red , violet) represents ticket-based approaches. A is a voter interacting with a votee B; C, D are other voters. Approaches skip forward when they cannot act.

votees. It is left open ended as to when this may occur; coin-based reputation systems do not typically require specific transactions to take place between participants in order to exchange repcoins. Upon receiving a repcoin, a votee then engages in a protocol to deposit the repcoin, raising their reputation score in the process. At any point, a votee may generate a proof that confirms their reputation level to a requester.

While all coin-based reputation systems in previous work have required a third party (which we will call “the bank”) to facilitate portions of these transactions, this is not strictly required. Work on cryptocurrencies could potentially be adapted for use with a coin-based reputation system, much as they have (as we will see in Section 5.5) for ticket-based reputation systems.

A typical interaction might look something like the approach described by Androulaki *et al.* [4]: Alice and Bob seek to interact with one another, and generate unique pseudonyms for this particular interaction. They both generate proofs that the pseudonyms correspond to a votee with their reputation levels and exchange these proofs. Alice and Bob, considering the reputation levels of the other, decide to continue their interaction. After concluding, Alice decides to spend a repcoin on Bob. Receiving this repcoin, Bob deposits it in a two-step process. First, using his pseudonym, he exchanges the repcoin for a blind signature (as introduced by Chaum [15]) from the bank. Then, under his long-term identity, Bob unblinds the signature and transmits it back to the bank, which in turn increases his reputation score.

The bank is typically trusted to faithfully follow its protocols. It is responsible for distributing repcoins according to whatever limitations the system imposes. It is also responsible for maintaining records of each votee’s deposited repcoins and corresponding reputation levels. However, although the bank is trusted to follow its protocols, the bank is not fully trusted with user privacy. In particular, the bank is not trusted to learn the linkage between a user and her pseudonym, or with what other users a given user may be interacting.

Coin-based reputation systems tend to be limited in terms of what reputation functions they support, due to the nature of how they use repcoins. They do, however, tend to provide useful, uncommon privacy properties, such as Exact Reputation Blinding. Frequently, this property is provided by votees having the ability to present zero-knowledge proofs of statements. These statements might include that their reputation is above some threshold, as mentioned by Ismail *et al.* [35] and implemented by Androulaki *et al.* [4].

Ismail *et al.* [35] designed the first privacy-preserving reputation system work we identify, in 2003. They design the bank as two entities: TI (for “token issuer”) and CA (for “certificate authority”). TI handles distributing repcoins and is trusted to see the interactions between users to verify that feedback comes from real interactions. CA handles maintaining reputation scores for votees, and restricts votees to only see their own scores directly. Votees disseminate their scores via a designated verifier scheme, so only their intended recipient may see their reputation.

Voss [59] presents a system in which repcoins are used as collateral in interactions. Voters request a number of repcoins of their choosing as the collateral for any interaction, and may invalidate any or all of them as punishment for bad behaviour. Alternately, they may award a single repcoin as positive reinforcement for good behaviour. Privacy exists for users from one another but not from the bank in this scheme.

Androulaki *et al.* [4] form their guarantees against misbehaviour by threatening that users who misbehave will implicate themselves and reveal the secret key to their long-term identity by doing so. Uniquely, anonymous credential systems, similar in concept to those introduced by Chaum [16], form the basis for the votees’ proofs of reputation levels in this work.

5.2 Signature-based Reputation Systems

Signature-based reputation systems are another approach to manage reputation in an unlinkable way. Signature-based reputation systems were designed specifically to address the problem of voters ballot-stuffing, inflating others' (or their own) reputations by spending multiple repcoins on a target. Importantly, in signature-based reputation systems, a voter may only vote for any single votee once. That is, they can vote for as many votees as they like, but for any individual votee, their reputation score is determined by the number of *unique* voters who voted for them. These votes come in the form of signatures, which include information to bind a vote to a voter's and votee's long-term identities. This binding is carefully constructed to avoid linking voters' or votees' actions.

A typical interaction might look something like the approach taken with Signatures of Reputation [11]: Alice and Bob seek to interact with one another, and as before, generate unique, short-term pseudonyms for this interaction based off of their long-term pseudonyms. They both use these short-term pseudonyms and votes they have previously received to sign messages with a specialized signature scheme. This scheme is designed such that Alice can prove the input votes were given by r_A distinct voters, where r_A is Alice's reputation score (and similarly for Bob). Alice and Bob, considering the reputation levels of the other, decide to continue their interaction. After concluding, Alice decides to vote for Bob, and generates said vote using her long-term pseudonym. Receiving this vote, Bob retains it for future proofs; if Alice has never previously voted for Bob, his maximum claimable score increases by one.

Unlike coin-based systems, signature-based reputation systems by design have the property that voters may cast at most one useful vote for any given votee. These systems are also limited in what reputation functions they can support, due to the manner in which they use these votes in their signatures for reputation. They do, however, present interesting opportunities to design a system which has a smaller or more decentralized approach to trust, either requiring only trusted platform modules (TPMs) run by users, or requiring a TTP only to set up the system, and not in its ongoing operation. Additionally, more advanced work to support a less common privacy property, Reputation-Usage Unlinkability, was done in this line of work early on.

The prototypical iClouds [60] relied on TPMs in its design. The work was expected to be used on mobile phones for information dissemination networks, and Voss *et al.* proposed that TPMs would allow votees to carry and produce their own reputation scores. Specifically, when voting for others, voters encrypt their votes in such a way that only the TPM will be able to open it, and the TPM can add together scores. The TPM, upon seeing a new vote by the same voter for a recipient, simply replaces the original with the new vote. In order to participate, users first sign up with a central authority to be given a certified long-term public key, with which they conduct all actions. In this system, voters do link their own actions together, in the view of their votees, but the value of the votes they cast is private.

Signatures of Reputation [11] have a much lower bound for trust, at the cost of functionality. A certificate authority is required so that users may only join the system with one long-term public key. However, from that key, users are able to generate as many short-term pseudonyms as they desire without needing to interact with any TTP. Importantly, votes cast for a votee cannot be rescinded; that is, Accrue Stars — Negative is not supported. The authors of this work argue that the reputation in their system is only meant to be a barrier against spam, but support for negative votes is recognized as a valuable area for future work.

5.3 Reputation Transfer

Unlike the previous two approaches, research on *reputation transfer* is agnostic to design choices around how voters may rate one another and how those ratings are tallied. Reputation transfer largely leaves the calculation of reputation up to the implementer, and instead focuses on one specific problem; namely, when users participate in any long-term system, having only one pseudonym for the entire length of participation is only a minor upgrade from using one's true identity.

As the calculation of reputation is typically left open, there is no typical full interaction for this approach. However, all works within this approach feature some procedure for votees to generate new pseudonyms that inherit the reputation of previous pseudonyms. These previous pseudonyms become invalid for future use. An archetypical example of this behaviour is given by RuP [45], where systems operate in epochs, and transfers are allowed between adjacent epochs. These transfers are executed by first requesting a TTP to strip information pertaining to previous pseudonyms from information used to prove scores, then requesting the TTP to bind that score information to a new pseudonym.

At large, works within Reputation Transfer advanced a recognition of the usefulness of Reputation-Usage Unlinkability and draw attention to the fact that Reputation-Usage Unlinkability becomes significantly stronger when combined with Exact Reputation Blinding. However, they rely heavily on TTPs in order to accomplish their transfer. How to best decentralize this procedure is an open question.

Anwar and Greer [5] initiate this line of research by suggesting that enabling users to transfer their reputation between pseudonyms would help them safeguard their privacy. They suggest using third-party guarantors that users can trust to properly transfer reputation scores between pseudonyms upon request. In their design, the guarantors are fully aware of the transfer, and users must trust the guarantor to keep the relationship between the pseudonyms secret.

RuP [45] adds restrictions in the name of preserving the correctness of the reputation system. Users may transfer their scores between pseudonyms, but they are only allowed to use one pseudonym during any given epoch. The pseudonym they use is signed by a TTP for a given epoch. RuP also uses blind signatures, so that users do not have to rely on the TTP to protect the privacy of their pseudonym linkage.

DARep [27] uses TPMs to generate secret pseudonyms for each user in a consistent manner. All users change pseudonyms simultaneously by changing a parameter sent to the TPM. How this parameter is changed is left open to implementers; a central authority may direct it, or users may reach a consensus on the timing. Votees' reputations are held by other users, and this set of users changes whenever pseudonyms change. After this change, the users who previously held a votee's reputation can identify the new pseudonym, but other users do not have enough information to perform this linkage. Interestingly, in DARep, when a voter votes for a votee, there is a cost imposed to their own reputation, in order to disincentivize ballot-stuffing attacks.

Hao *et al.* [26] in 2008 note that RuP is not robust against Sybil attacks, and identify the problem of users' exact score values potentially deanonymizing them on transfer. Hao *et al.*also make alterations to the blind signature scheme of RuP for efficiency gains, and Wei and He [62] propose similar alterations.

Peng *et al.* [49] in 2010 also modify RuP to improve efficiency. They too recognize the issue of users' scores deanonymizing them on transfer. As a cost-saving alternative to blind signatures, their design supports a protocol they call "group confusion", where several users with the same reputation all request reputation transfers at the same time. Their design does not mandate this behaviour, but it recognizes that the size of anonymity sets for users who transfer reputations is important.

Huang *et al.* [33] further recognize this problem and specifically attempt to solve it. Inspired by k-anonymity, they fuzz reputation scores when publicly announcing them such that there is always a group of users for each reported reputation value at each time interval. Their system minimizes the difference between actual and reported score while preserving their k-anonymity goal. IncogniSense [17] takes a very similar approach, and specifically analyzes a set of different methods to cloak reputation scores for accuracy and usefulness.

k-Anonymous Reputation [19] is also inspired by k-anonymity, but in a different manner. It recognizes that users may desire to keep pseudonyms for longer periods of time. Persistence of pseudonyms can be useful for users, as both name and score serve as markers of their reputation. As opposed to the technique used by Huang *et al.* [33], where reputation scores are fuzzed, users are required to wait to get a new pseudonym until a large enough group willing to change pseudonyms forms, so that anonymity may be preserved for all of them.

5.4 SMC-based Reputation Systems

Secure multiparty computation (SMC) forms the basis for another early line of research on privacy-preserving reputation systems, *SMC-based reputation systems.* Unlike previously mentioned approaches, this line has persisted and new research has continued up to contemporary work. SMC-based approaches largely arose as a unique and useful application of SMC techniques, rather than as a tool for reputation within familiar paradigms of transactions and communications, and thus take a drastically different shape from most other approaches.

In particular, SMC-based approaches tend to envision reputation as belonging to those who assign it, rather than belonging to those it describes. Put another way, other approaches typically view reputation as a score or value that a votee may display in a verified manner, and it certifies that over some period of activity they have accrued a specified reputation. However, SMC-based approaches view reputation instead as the collection of ratings of voters, applied to a votee. To determine the nature of a requester's interaction with a votee, the requester will seek out and combine the ratings voters would give to the votee. A votee does not own their own reputation and does not display it; instead, they only display sufficient information for requesters to be able to identify the votee to voters.

A typical interaction might look something like the approach of Decentralized Additive Reputation Systems (DARS) [48]: Alice is deciding whether to interact with Bob. Alice solicits feedback from Carol and Dave, which is securely composited into a reported score for Alice to consider. She decides to interact with him, and based on Carol and Dave's reported score and her own interactions with Bob, Alice forms her own score for him. Later, when Carol asks Alice (among others) for a new evaluation of Bob, Alice provides this score (as part of a securely composited score) to Carol.

SMC-based reputation systems start from a place of (user-defined) decentralization that other approaches frequently do not. These systems do not provide for a verified global scoreboard, as other systems do. Instead, requesters are expected to choose the voters from whom they solicit ratings themselves. Although this could be every eligible voter in the system, these systems are designed with the intention that requesters only request votes from a subset of voters (in particular, these systems are typically inefficient for large numbers of voters). Requesters are able to get pinpointed ratings from voters they trust, assuming they already trust some voters. Some systems allow requesters to weight ratings from voters based on how much stock they place in their recommendations. A different way of viewing this property is in terms of identifying a votee's reputation among specific communities or subgroups within a system. SMC-based approaches by their very nature turn voter-agnostic reputation functions into voter-conscious reputation functions, and consistently provide Voter-Vote Unlinkability and Two-Vote Unlinkability. However, again due to their nature, they are completely unable to provide Reputation-Usage Unlinkability, as they do not provide for short-term pseudonyms without needing to start over on reputation for each new pseudonym.

Kinateder and Pearson's [39] design is atypical compared to later approaches, in that it does not directly draw from cryptographic SMC techniques. Although the overall structure of the system is still similar (requesters soliciting ratings from voters about a votee in a manner that can be calculated without any individual voter's rating being revealed), this is accomplished through TPMs instead of cryptography. The design is more of a rough sketch than a fully fleshed out system, but it gives the overall idea of the approach.

DARS [48] is more closely related to the later cryptographic approaches. In this work, the authors design a reputation system through an application of secure sum (through multiple different techniques, including secret sharing), an SMC technique. Further work among SMC-based reputation systems largely only modifies this approach or the output of the algorithm, rather than using completely different SMC techniques.

One such work is the Private Distributed Scalar Product Protocol (PDSPP) [64]. Instead of secure sum, it calculates a scalar product. Reputation is represented as the inner product of a vector of reputation ratings given by voters and a vector of ratings of trust a requester places in each voter. Put another way, it is a weighted sum of votes. The Collusion-Resistant Distributed Scalar Product Protocol (CRDSPP) [1] expanded on this work and observed a security flaw in the PDSPP related to collusion between certain agents in the semi-honest model. The Privacy-Friendly Weighted-Reputation Aggregation Protocol (PFWRAP) [67] further expanded and made efficiency gains.

3PRep [47] expands upon a non-privacy preserving decentralized reputation system, P2PRep [20], attempting to add private computation of reputations to the system. Unlike other systems within the SMC approach, this system is not fully decentralized, at times relying on pre-trusted peers within the system to prepare certain computations or hold specific encryption keys separate from a requester. However, neither is it fully centralized; ratings still come directly from a variety of voters that work in concert to evaluate the "Ordered Weighted Average", an average that gives higher weighting to low or repeated reputation scores collected from voters.

A series of works defining k-Shares [28–31] present several incremental improvements to the SMC-based approach. The works primarily improve communication complexity and, over their span, move from semi-honest models to malicious adversary models. In brief, these works largely derive their efficiency gains from reducing the burden on requesters to receive ratings from every voter in the system in order to guarantee that they do not collude, instead only requiring that they can receive ratings from k of them that they trust.

Dyn-PDRS [18] focuses on a single specific problem within Ephemeral Mesh Topology designs. Specifically, the authors note that when users leave such decentralized systems, their ratings leave as well. Dyn-PDRS provides mechanisms for voters' ratings to continue to be used after a voter exits the system, by giving them to other voters to propagate. This does introduce the question of how long a voters' rating should still be considered accurate for a votee after the voter leaves the system. That being said, it is an interesting consideration to look at how systems handle the recommendations of voters who no longer continue to participate.

M2M-REP [7] has a different form than the other SMC-based works. In particular, it returns to a notion of global reputation. This is achieved by voters posting an encrypted version of their vote (and a proof that the vote is of a particular form) to a public bulletin board. The particular form of the vote takes advantage of the fact that votes can only be -1, 0, or 1, and allows them to be combined in a way that outputs a plaintext summary score. Due to the construction used, this plaintext output can only occur when all votes are combined.

5.5 Ticket-based Reputation Systems

Ticket-based reputation systems are the most consistently researched approach for privacy-preserving reputation systems. Ticket-based reputation systems form a natural extension of coin-based reputation systems, and the first ticket-based system was proposed in the same year as the first coin-based system. Ticket-based systems have, like SMC-based systems, continued to be researched through contemporary work. More papers have been published in this approach than any other.

In a ticket-based reputation system, instead of being able to award coins for favourable interactions as in coin-based reputation systems, a voter is given some kind of authorization (or “ticket”) to give a rating to a votee. This ticket is frequently the straightforward result of a one-to-one transaction — the archetypical example of this being a sale on a service like eBay, where the buyer and seller are given the opportunity through the service to rate each other after conducting their business. However, the ticket in some systems may be more naturally understood as one-to-many (such as, voters rating posts made by votees in a forum) or many-to-many (such as, all voters being given opportunities to revote once per some set epoch for all votees). In order to vote, a voter “spends” their ticket along with giving a rating value. A ticket may only be spent once by any individual voter in order to prevent ballot-stuffing attacks.

A typical interaction might look something like this. Alice and Bob seek to interact with one another. They both identify the others’ reputation levels (sometimes via a proof, sometimes via a public bulletin board, sometimes via a TTP). Alice and Bob, considering the reputation levels of the other, decide to continue their interaction, in the process exchanging tickets for each other. After concluding, Alice decides to rate Bob. She redeems her ticket, posting her feedback either to a TTP or to a public bulletin board (typically in a way unlinkable to her long-term identity). This feedback directly affects future calculations of Bob’s reputation. We note in particular that these two approaches for posting feedback constitute their own branches of this approach, which we term *TTP ticket-based reputation systems* and *public log ticket-based reputation systems*. Boyd *et al.* [13] provide a good archetype for a TTP ticket-based reputation system, and Beaver [56] is a good archetype for a public log ticket-based reputation system.

Significant work in this approach has been focused on minimizing the trust placed in any centralized party. There is, however, a direct tension between the efficiency and flexibility of such systems with the amount of trust placed in a centralized party to accomplish it. Centralized systems can be designed to compute reputation functions without incurring large costs associated with the cryptography usually needed to guarantee privacy in decentralized systems or systems that trust their TTPs less. Further, in order to provide a more diverse range of reputation functions (particularly complex ones), systems need more details about votes, such as the identity of the voter or their own reputation. This is challenging to provide in a privacy-preserving way. It is perhaps unsurprising that systems intended for commerce have focused more directly on decentralizing trust, while systems in other settings have focused on providing a wider array of privacy properties and improving efficiency without centralizing trust.

5.5.1 Trusted Third Party Approaches

By definition, TTP ticket-based reputation systems involve entities in which some amount of trust must necessarily be placed. However, how much trust is placed and in how many such entities varies between systems. In some cases, the TTP’s role is largely relegated to verifying new identities in order to prevent Sybil attacks; in others, the TTP sees most details associated with interactions in the system. In general, research has shifted over time towards smaller amounts of trust placed in TTPs, and distributing trust over multiple entities.

Notwithstanding this trend, in 2003 Singh and Liu present TrustMe [55], which places only a small level of trust in its TTP. In TrustMe, a bootstrap server is used to randomly assign each votee a set of different users in the system, called the trust-holding agents (THAs), who are responsible for holding, updating, and reporting the votee’s reputation. A bootstrap server is used to preserve the correctness of this arrangement. When requesters broadcast reputation queries, the THAs return the reputation scores signed with an appropriate key given by the bootstrap server. Relatively little trust needs to be placed in the bootstrap server to randomly assign THAs, and the probability of THAs colluding to falsify a votee's score decreases with the size of the system and the number of THAs utilized. However, TrustMe's privacy protections largely amount to allowing voters to cast votes that votees will not see directly.

Boyd *et al.* [13], in contrast, use a more invasive TTP. Before initiating a transaction, voters and votees jointly register the transaction with the TTP, and the TTP responds with a token for the voter to use to prove their vote is the result of a valid transaction. After the transaction, the voter sends the token along with their feedback to the TTP and receives a new signed nonce, then sends that along with their vote to the votee. The votee verifies the validity of the nonce, then acknowledges receipt of the vote to the TTP, who then provides a signed list of all vote values given for the votee. The votee uses this to display their reputation score.

ARM4FS [46] specifically examines the use case of file-sharing systems, where multiple users may have a file, but different users' versions of the files may be of varying qualities. One TTP is used for identity verification, so that users may not create arbitrary accounts in the system. Another TTP is used when a user uploads a file to the service, which tags the file with a nonce corresponding to that user. Each file receives a different nonce, even when the same user uploads them. Then, voters submit this nonce along with their vote, which the TTP directs to the correct user. This successfully allows a votee to use their reputation without linking themselves across files, which few other schemes achieve, but does not offer much to protect voter privacy.

Kerschbaum [37] proposes a system using a combination of pairing-based cryptography and traditional asymmetric encryption to achieve its privacy properties. When two users engage in a transaction, they generate tokens for each other of specific forms such that quick verification of the validity of the token is possible. The voter then submits an encrypted vote to one TTP, along with the token. This first TTP periodically forwards all received votes to a second TTP, who can decrypt the vote and verify the tokens. The second TTP collates the votes and publishes votees' scores. So long as the two TTPs do not collude, voters' votes remain private. Petrlic *et al.* [50] use a similar construction, but use homomorphic encryption such that only one TTP is needed. The TTP combines ratings, and gives the encrypted result to the votee to decrypt. This does change one important aspect, namely that the TTP is thus able to see when transactions between two users occur, which was not possible in Kerschbaum's system. Blömer *et al.* [12] also use a similar construction; their work is largely notable for approaching the problem in the Universal Composability Framework. As such, it places more emphasis on its security proofs than other works have.

Hussain and Skillicorn [34] describe a system based around what they term "personas". Their system describes using what are effectively anonymous credentials, such that service providers can be provided reputational feedback by their customers. This feedback is collated by a TTP, who observes the validity of the anonymous credential in order to accept feedback. CLARC [10] takes a similar approach, explicitly using anonymous credentials. In CLARC, however, the TTP is not needed to collate feedback; instead, feedback and proofs of validity are published openly for requesters to collate themselves. Rather, the TTP is used to trace misbehaving users and expel them from the system. This is done by requiring that all users register with the TTP separately from the other mechanisms of the system.

ARTSense [61] is a system focused on providing reputation to participatory sensing in a privacy-preserving manner, and thus has a few unusual properties. Primarily, reputation votes come from a central server based on the performance of a user during data collection. In this system, after data is collected and sent to the central server, the central server responds with a ticket that contains the server's vote on the user's reputation, encrypted so the user cannot know whether it is positive or negative. The user then redeems the ticket, and their score is updated accordingly. A nonce inside the ticket is used so that users cannot replay previous tickets, and blind signatures are incorporated so that the central server cannot link a user redeeming a ticket to the instance it was issued to them.

Modeled as a reputation system for use in forum settings, AnonRep [66] allows users to make posts and tag those posts with their own reputation. AnonRep also gives votees the ability to blind their own reputation, such that they may instead prove that their reputation is above a threshold of the votee's choosing. Voters use linkable ring signatures to cast votes on posts, so that they may only vote once or be traceable as misbehaving. Importantly, though, the TTPs are a set of servers that engage in a verifiable shuffle in order to allow users to periodically change their pseudonyms in a reliable, unlinkable manner. This has the effect that, for every epoch where this verifiable shuffle is performed, users may make posts and receive votes unlinkably from their own previous posts. Unlike almost all other systems, AnonRep notably allows the system to add additional TTPs in an anytrust relationship. That is, if any one TTP is honest, the system's guarantees are upheld, making it desirable to add additional TTPs.

In Bazin *et al.*'s work [9], the TTPs are less involved in transactions. Instead, they audit votees for honest reporting. When two users engage in a transaction, the votee gives the voter a blind signature on a ticket. The voter reports the transaction to the TTPs. The voter then unblinds the signature and sends their vote and unblinded signature directly to the votee through an anonymous channel. Periodically, all votees submit their feedbacks to the TTPs. The TTPs sign the votes, and votees go on to display their votes with this signature from the TTPs to new potential transaction partners. If a voter's feedback is missing, however, the voter can report this to the TTPs with proof that their vote was valid. As long as one TTP audits honestly, that TTP can be relied upon to enforce the system rules.

Busom *et al.* [14] recommend a multi-tiered system of reputation — the system is nominally half-duplex (voters and votees are distinct sets), but a caveat is added to this. Voters may *endorse* other voters' feedbacks as useful, and upon enough of these endorsements, a voter may gain additional status. Other than these tweaks, the reputation system is largely similar to Kerschbaum's [37] or that of Petrlic *et al.* [50]. Importantly, the addition of the endorsement mechanism does not force all of a voter's feedbacks to be linkable.

Garms *et al.* [24], like AnonRep [66], examines reputation in a forum setting. The TTP is used as the manager in a group signature scheme, so that they and no one else can link those who submit a post to keep track of their reputation. All feedback is directed through the TTP, who updates votees' reputations periodically.

In PrivRep [8], the TTP has an additional duty to decide which voters are considered trustworthy. Untrustworthy voters' votes are silently discarded and not used in reputation calculations. PrivRep also incorporates an idea from SMC-based reputation systems, where all voters get one vote for each votee (which may be updated on subsequent polls) rather than tying votes to things like transactions. The ticket in this case is just valid participation in the system, and the privacy of votes is secured through a series of non-interactive zero knowledge proofs of knowledge (NIZKPoKs) that allow the TTP to calculate an overall reputation score but that do not reveal any participant's individual ratings.

pRate [44] is particularly notable for drawing attention to the utility votees may gain by blinding their exact reputation scores. pRate instead allows for statements such as that a votee's reputation is above a specified threshold. pRate also specifically allows a user to prove statements about their reputation without linking to their long-term identity, much like coin-based systems and systems like AnonRep [66]. It accomplishes this through pairing-based cryptography and NIZKPoKs.

5.5.2 Public Log Approaches

As mentioned above, public log ticket-based reputation systems have largely followed the rise of interest in blockchain technologies. As such, they carry a unique set of concerns from other systems. While TTP ticket-based reputation systems largely assume their TTPs have sufficient incentive to provide their services correctly and in a trustworthy manner, such a claim would naturally need greater elaboration when dealing with a set of blockchain miners. How systems consider their miners is only one important way they can vary. Their variance also comes from issues in common with TTP approaches, such as how tickets are formed.

Beaver [56] is directly associated with commerce settings. Votees' reputations are tied to the items they sell, and can choose whether to link these items together through NIZKPoKs of private keys associated with the other items. Voters are granted privacy in their evaluations through linkable ring signatures (as was done in AnonRep [66]) across all public keys associated with a transaction for an item; anyone can vote once if they have participated in a transaction, but voting multiple times will implicate a voter. Voters are encouraged to generate new keypairs for each transaction, but may use NIZKPoKs of values committed in previous reviews to link them together if they so desire.

Schaub *et al.* [53] use blinded tokens transferred during a transaction for voters to give a rating for votees. Voters wait for others to transact with a votee to enlarge their anonymity set, but how voters determine that other users have transacted with a votee after them is not obvious. The authors also propose a mechanic in which the currency associated with their system will be used by the votees themselves to generate tokens, so that they cannot issue arbitrary new tokens.

PrivBox [6] only requires a public bulletin board as opposed to a full blockchain. While the bulletin board can of course be implemented via a blockchain, this work does not focus on that aspect. Users are given tokens of an unspecified form in order to give their feedback, which comes in the form of an encrypted 1 or 0. When every vote is combined, the blinding factors involved in each cancel out, and brute search can reveal the score from 0 to a maximum of the total number of votes.

ARS-PS [43] makes open use of a TTP, in order to help prevent Sybil attacks. The TTP is responsible for making sure users are only able to join the system once, and can be used later on to identify misbehaving users (although misbehaviour is detectable without identification). ARS-PS also employs an alternative underpinning for its blockchain relative to the other works in this approach. Namely, it relies on Proof of Stake instead of Proof of Work. The miners are votees, and their stake in the consensus protocol is directly tied to their own reputation scores. Voters submit votes homomorphically encrypted in such a manner that only all of a set of votees working together could decrypt it. That set adds all votes for a votee together, then decrypts the sum, and publishes it, using that to determine the stake for the next epoch in the blockchain.

5.6 Tradeoffs between Approaches

Naturally, different approaches have different strengths and weaknesses. When creating new reputation systems, discerning system designers should weigh their options with a specific mind for the goals of their particular system. We highlight some of the most notable of these tradeoffs with the aim of making more clear why certain approaches may be more or less desirable.

Coin-based systems were the first to implement Reputation-Usage Unlinkability and Exact Reputation Blinding. Though designers may desire these properties, they will find great difficulty in implementing many reputation functions with the repcoins inherent to coin-based systems, particularly due to the difficulty in implementing negative coins. A similar problem occurs for signature-based reputation systems; due to their design around specific novel signature schemes, rescinding votes seems difficult to implement. In both cases, complications arise in implementation when designers desire votes to contain more nuanced information than merely affirming a positive (or negative) interaction.

Both signature-based and SMC-based reputation systems have a considerable amount of decentralization inherent to their architecture. In both, this comes with a cost that reputation functions must be relatively simple. Due to SMC-based systems' approach involving soliciting other users for their feedback for a votee directly, it would be difficult to receive correct feedback without divulging which votee is being inquired about. Thus, SMC-based systems have considerable difficulty providing votee privacy properties like Reputation-Usage Unlinkability and Exact Reputation Blinding. SMC-based systems also generally require that voters always be online to give their feedback, which is not necessarily the case for other approaches, and though improvements have been made over time, SMC still often involves significant calculation overheads.

Ticket-based reputation systems feature the most variety, due to the large amount of work done in the approach. Similar to SMC-based reputation systems, public log approaches to ticket-based systems have difficulty providing votee privacy properties; knowing who feedback is intended for is difficult in their public setting without direct identification. Public log approaches do feature significant decentralization. However, that decentralization is subject to familiar concerns around hijacking consensus in the blockchains used. Public log approaches also may have difficulty implementing voter-conscious reputation functions without sacrificing Voter-Vote or Two-Vote Unlinkability. SMC-based systems implement these functions without this sacrifice by asking voters to jointly calculate these functions before individual votes reach the requester. Where these functions have been implemented elsewhere, avoiding that sacrifice is typically accomplished by carefully relying on TTPs to perform the calculations.

This reliance on TTPs is the main drawback with TTP ticket-based systems. Though a significant variety of privacy properties and reputation functions is possible with ticket-based approaches, this has often been accomplished with increased reliance on TTPs. Where centralization is a concern, it may be difficult to justify using TTP approaches despite the breadth of privacy properties and reputation functions available.

6 Opportunities for Future Research

From the systematization of literature we performed in Section 5, we observe combinations of properties that have not been implemented together. Where this is not the result of a tradeoff forcing properties to be left behind, these absent combinations suggest opportunities for future work. We identify three such opportunities.

Decentralization does not have to mean blockchain. Several recent systems, designed for use in transactional business settings, have focused on blockchain approaches as a means towards decentralization. This makes sense, but blockchain approaches have limitations. Ignoring the problems specific to blockchains themselves, blockchain approaches to reputation systems thus far have not been able to provide privacy properties such as Reputation-Usage Unlinkability. This is despite the fact that older schemes, such as Signatures of Reputation [11], which also do not rely on trusted third parties after initial setup, do provide Reputation-Usage Unlinkability. Blockchain approaches have also not provided reputation functions beyond the more common voter-agnostic reputation functions. However, the user-defined decentralized SMC-based reputation systems are able to provide voter-conscious reputation functions. Whether it is possible to provide such functions while providing Voter-Vote Unlinkability in a blockchain approach is an open question.

Systems that examine how to distribute trust across multiple central nodes, such as AnonRep [66] and the work of Bazin *et al.* [9], are also important to note in this conversation. Both use an anytrust model for their central nodes. While blockchain approaches are more decentralized, decentralization is not all-or-nothing. In particular, forums and other settings intended for community usage have existed as federated or similar structures for much of their history, rather than fully decentralized designs. In such cases, anytrust systems are a positive improvement on current techniques. AnonRep deserves particular note for providing every privacy property we measure. Anytrust systems have shown significant promise in providing a wider arrangement of privacy properties than blockchain approaches, while still moving away from fully centralized techniques.

Reputation-usage unlinkability and voter-conscious reputation functions would make a particularly potent combination for community usage. Reputation-usage unlinkability has been, to this point, of more interest in community settings than in business settings. Although providing sellers of goods more privacy is perhaps underexplored, it is natural to recognize that, for example, a forum has utility in offering increased anonymity to its participants, if it can be done without inviting poor community behaviour. Allowing people to explore new ideas more openly can be very useful, and Reputation-Usage Unlinkability is one way to give that opportunity. It is also in such a setting that an idea of a consensus opinion of all users regarding a votee has the most meaning. In a setting where a limited number of individuals may actually all interact with one another, voter-conscious reputation functions are most meaningful, as it reflects an actual knowledge of the community. If in this setting, a votee cannot earn positive ratings from a reasonable number of voters, it seems reasonable to conclude that votee may not be a good fit for (or is not acting in good faith in) a given community. As of time of writing, only Hao *et al.* [26] and Wei and He [62] have described systems where Reputation-Usage Unlinkability has been paired with a voter-conscious reputation function, and in both cases a large amount of trust is still placed in the TTP to function correctly. A system with more decentralized trust, like AnonRep [66], that could provide both of these properties would have significant promise in such community settings.

At this time in the literature, reputation is limited to applying by and to individual users. There are two important ways the literature could expand its uses for reputation. First, a votee's rating could be decided by specific subsets of voters. Local scope reputation systems do currently have the ability to support this usage, as they report the reputation as decided by the voters who a requester queries. However, they can only tell us the opinion of one such group at a time for each votee; knowing the opinions of multiple different groups may be valuable to a requester. Second, in no systems in the literature does reputation refer to the opinions of voters towards a group of votees. These votees might represent, for example, all the sellers of a certain kind of product, and reputation would indicate satisfaction with that kind of product overall. Alternatively, the votees could be groups of individuals, such as politicians, such that voters might express their approval or disapproval. Expanding reputation beyond the individual holds promise for new research in multiple directions.

7 Conclusion

In this work, we examined 42 systems detailing privacy-preserving reputation systems, and were able to identify large-scale trends in how the papers approached this problem. We elaborated on issues of terminology that had hindered previous works' ability to effectively communicate the importance and novelty of their work. We identified important tradeoffs underpinning the design choices that separate different approaches to this problem. We observed three key areas where previous work has not yet explored and should make for exciting avenues of research. While privacy and reputation have some natural tension between one another, we think that there are promising opportunities, particularly for application in community settings rather than transactional settings, for new research to make a large impact.

Acknowledgements

We thank the anonymous reviewers and our shepherd, Thomas Groß, for their helpful comments in improving this work. We thank the Royal Bank of Canada and NSERC grant CRDPJ-534381 for funding this work. This research was undertaken, in part, thanks to funding from the Canada Research Chairs program.

References

[1] Carlos Aguilar Melchor, Boussad Ait-Salem, and Philippe Gaborit. A collusion-resistant distributed scalar product protocol with application to privacy-preserving computation of trust. In *2009 Eighth IEEE International Symposium on Network Computing and Applications*, pages 140–147, July 2009.

[2] Jay Allen. The invasion boards that set out to ruin lives. https://boingboing.net/2015/01/19/invasion-boards-set-out-to-rui.html, January 2015.

[3] Larry Alton. How Purple, Uber and Airbnb are disrupting and redefining old industries. https://www.entrepreneur.com/article/273650, April 2016.

[4] Elli Androulaki, Seung Geol Choi, Steven M. Bellovin, and Tal Malkin. Reputation systems for anonymous networks. In Nikita Borisov and Ian Goldberg, editors, *Privacy Enhancing Technologies*, pages 202–218. Springer Berlin Heidelberg, 2008.

[5] Mohd Anwar and Jim Greer. Reputation management in privacy-enhanced e-learning. In *The Proceedings of the 3rd Annual Scientific Conference of the LORNET Research Network (I2LOR 2006)*, November 2006.

[6] Muhammad Ajmal Azad, Samiran Bag, and Feng Hao. PrivBox: Verifiable decentralized reputation system for online marketplaces. *Future Generation Computer Systems*, 89:44–57, 2018.

[7] Muhammad Ajmal Azad, Samiran Bag, Feng Hao, and Khaled Salah. M2M-REP: Reputation system for machines in the internet of things. *Computers and Security*, 79:1–16, 2018.

[8] Samiran Bag, Muhammad Ajmal Azad, and Feng Hao. A privacy-aware decentralized and personalized reputation system. *Computers and Security*, 77:514–530, 2018.

[9] Rémi Bazin, Alexander Schaub, Omar Hasan, and Lionel Brunie. A decentralized anonymity-preserving reputation system with constant-time score retrieval. Cryptology ePrint Archive, Report 2016/416, 2016. https://eprint.iacr.org/2016/416.

[10] Kai Bemmann, Johannes Blömer, Jan Bobolz, Henrik Bröcher, Denis Diemert, Fabian Eidens, Lukas Eilers, Jan Haltermann, Jakob Juhnke, Burhan Otour, Laurens Porzenheim, Simon Pukrop, Erik Schilling, Michael Schlichtig, and Marcel Stienemeier. Fully-featured anonymous credentials with reputation system. In *Proceedings of the 13th International Conference on Availability, Reliability and Security*, ARES 2018, pages 42:1–42:10, New York, NY, USA, 2018. ACM.

[11] John Bethencourt, Elaine Shi, and Dawn Song. Signatures of reputation. In Radu Sion, editor, *Financial Cryptography and Data Security*, pages 400–407, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[12] Johannes Blömer, Fabian Eidens, and Jakob Juhnke. Practical, anonymous, and publicly linkable universally-composable reputation systems. In Nigel P. Smart, editor, *Topics in Cryptology — CT-RSA 2018*, pages 470–490. Springer International Publishing, 2018.

[13] Colin Boyd, Roslan Ismail, Audun Jøsang, and Selwyn Russell. Private reputation schemes for P2P systems. In Fernandex-Medina, Castro, and Villalba, editors, *Proceedings of the 2nd International Workshop on Security In Information Systems, WOSIS 2004*, pages 196–206, Porto, Portugal, 2004. INSTICC Press.

[14] Núria Busom, Ronald Petrlic, Francesc Sebé, Christoph Sorge, and Magda Valls. A privacy-preserving reputation system with user rewards. *Journal of Network and Computer Applications*, 80:58–66, 2017.

[15] David Chaum. Blind signatures for untraceable payments. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *Advances in Cryptology*, pages 199–203, Boston, MA, 1983. Springer US.

[16] David Chaum. Security without identification: Transaction systems to make Big Brother obsolete. *Commun. ACM*, 28(10):1030–1044, October 1985.

[17] Delphine Christin, Christian Roßkopf, Matthias Hollick, Leonardo A. Martucci, and Salil S. Kanhere. IncogniSense: An anonymity-preserving reputation framework for participatory sensing applications. *Pervasive and Mobile Computing*, 9(3):353–371, 2013. Special Issue: Selected Papers from the 2012 IEEE International Conference on Pervasive Computing and Communications (PerCom 2012).

[18] Michael R. Clark, Kyle Stewart, and Kenneth M. Hopkinson. Dynamic, privacy-preserving decentralized reputation systems. *IEEE Transactions on Mobile Computing*, 16(9):2506–2517, September 2017.

[19] Sebastian Clauß, Stefan Schiffner, and Florian Kerschbaum. k-anonymous reputation. In *Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security*, ASIA CCS '13, pages 359–368, New York, NY, USA, 2013. ACM.

[20] Ernesto Damiani, Sabrina De Capitani di Vimercati, Stefano Paraboschi, and Pierangela Samarati. Managing and sharing servents' reputations in P2P systems. *IEEE Transactions on Data and Knowledge Engineering*, 15(4):840–854, 2003.

[21] EJ Dickson. Furries got an alt-right troll banned from their convention. https://www.rollingstone.com/culture/culture-news/milo-yiannopolous-furry-convention-884960/, September 2019.

[22] Minghong Fang, Neil Zhenqiang Gong, and Jia Liu. Influence function based data poisoning attacks to top-n recommender systems. In *Proceedings of The Web Conference 2020*, WWW '20, pages 3019–3025, New York, NY, USA, 2020. Association for Computing Machinery.

[23] Minghong Fang, Guolei Yang, Neil Zhenqiang Gong, and Jia Liu. Poisoning attacks to graph-based recommender systems. In *Proceedings of the 34th Annual Computer*

Security Applications Conference, ACSAC '18, pages 381–392, New York, NY, USA, 2018. Association for Computing Machinery.

[24] Lydia Garms, Keith Martin, and Siaw-Lynn Ng. Reputation schemes for pervasive social networks with anonymity (short paper). In *2017 15th Annual Conference on Privacy, Security and Trust (PST)*, pages 311–316, August 2017.

[25] Neil Zhenqiang Gong, Mario Frank, and Prateek Mittal. SybilBelief: A semi-supervised learning approach for structure-based Sybil detection. *IEEE Transactions on Information Forensics and Security*, 9(6):976–987, 2014.

[26] Liming Hao, Songnian Lu, Junhua Tang, and Aixin Zhang. A low cost and reliable anonymity scheme in P2P reputation systems with trusted third parties. In *IEEE GLOBECOM 2008 — 2008 IEEE Global Telecommunications Conference*, pages 1–5, November 2008.

[27] Liming Hao, Shutang Yang, Songnian Lu, and Gongliang Chen. A dynamic anonymous P2P reputation system based on trusted computing technology. In *IEEE GLOBECOM 2007 - IEEE Global Telecommunications Conference*, pages 332–337, November 2007.

[28] Omar Hasan, Elisa Bertino, and Lionel Brunie. Efficient privacy preserving reputation protocols inspired by secure sum. In *2010 Eighth International Conference on Privacy, Security and Trust*, pages 126–133, August 2010.

[29] Omar Hasan, Lionel Brunie, and Elisa Bertino. k-Shares: A privacy preserving reputation protocol for decentralized environments. In Kai Rannenberg, Vijay Varadharajan, and Christian Weber, editors, *Security and Privacy—Silver Linings in the Cloud*, pages 253–264, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[30] Omar Hasan, Lionel Brunie, and Elisa Bertino. Preserving privacy of feedback providers in decentralized reputation systems. *Computers and Security*, 31(7):816–826, 2012. IFIP/SEC 2010 "Security and Privacy—Silver Linings in the Cloud".

[31] Omar Hasan, Lionel Brunie, Elisa Bertino, and Ning Shang. A decentralized privacy preserving reputation protocol for the malicious adversarial model. *IEEE Transactions on Information Forensics and Security*, 8(6):949–962, June 2013.

[32] Kuan Lun Huang, Salil S. Kanhere, and Wen Hu. Are you contributing trustworthy data? The case for a reputation system in participatory sensing. In *Proceedings of the 13th ACM International Conference on Modeling, Analysis, and Simulation of Wireless and Mobile Systems*, MSWIM '10, pages 14–22, New York, NY, USA, 2010. Association for Computing Machinery.

[33] Kuan Lun Huang, Salil S. Kanhere, and Wen Hu. A privacy-preserving reputation system for participatory sensing. In *37th Annual IEEE Conference on Local Computer Networks*, pages 10–18, October 2012.

[34] Mohammed Hussain and David B. Skillicorn. Mitigating the linkability problem in anonymous reputation management. *Journal of Internet Services and Applications*, 2(1):47–65, July 2011.

[35] Roslan Ismail, Colin Boyd, Audun Jøsang, and Selywn Russel. Strong privacy in reputation systems. In *Proceedings of the 4th International Workshop on Information Security Applications (WISA)*, August 2003.

[36] Jinyuan Jia, Binghui Wang, Le Zhang, and Neil Zhenqiang Gong. AttriInfer: Inferring user attributes in online social networks using Markov random fields. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, pages 1561–1569, 2017.

[37] Florian Kerschbaum. A verifiable, centralized, coercion-free reputation system. In *Proceedings of the 8th ACM Workshop on Privacy in the Electronic Society*, WPES '09, pages 61–70. ACM, 2009.

[38] Eric Killelea. Does the furry community have a Nazi problem? https://www.rollingstone.com/culture/culture-features/does-the-furry-community-have-a-nazi-problem-194282/, April 2017.

[39] Michael Kinateder and Siani Pearson. A privacy-enhanced peer-to-peer reputation system. In Kurt Bauknecht, A. Min Tjoa, and Gerald Quirchmayr, editors, *E-Commerce and Web Technologies*, pages 206–215, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[40] Christiane Kuhn, Martin Beck, Stefan Schiffner, Eduard Jorswieck, and Thorsten Strufe. On privacy notions in anonymous communication. *Proceedings on Privacy Enhancing Technologies*, 2019(2):105–125, 2019.

[41] KW Counselling Services. What Does LGBTQ+ Mean? https://ok2bme.ca/resources/kids-teens/what-does-lgbtq-mean/, 2020.

[42] Shyong K. Lam and John Riedl. Shilling recommender systems for fun and profit. In *Proceedings of the 13th International Conference on World Wide Web*, WWW '04, pages 393–402, New York, NY, USA, 2004. Association for Computing Machinery.

[43] Dongxiao Liu, Amal Alahmadi, Jianbing Ni, Xiaodong Lin, and Xuemin Shen. Anonymous reputation system for IIoT-enabled retail marketing atop PoS blockchain. *IEEE Transactions on Industrial Informatics*, 15(6):3527–3537, June 2019.

[44] Jia Liu and Mark Manulis. pRate: Anonymous star rating with rating secrecy. In Robert H. Deng, Valérie Gauthier-Umaña, Martín Ochoa, and Moti Yung, editors, *Applied Cryptography and Network Security*, pages 550–570. Springer International Publishing, 2019.

[45] Hugo Miranda and Luis Rodrigues. A framework to provide anonymity in reputation systems. In *2006 Third Annual International Conference on Mobile and Ubiquitous Systems: Networking Services*, pages 1–4, July 2006.

[46] Wolf Müller, Henryk Plötz, Jens-Peter Redlich, and Takashi Shiraki. Sybil proof anonymous reputation management. In *Proceedings of the 4th International Conference on Security and Privacy in Communication Netowrks*, SecureComm '08, pages 7:1–7:10, New York, NY, USA, 2008. ACM.

[47] Rishab Nithyanand and Karthik Raman. Fuzzy privacy preserving peer-to-peer reputation management. Cryptology ePrint Archive, Report 2009/442, January 2009. https://eprint.iacr.org/2009/442.

[48] Elan Pavlov, Jeffrey S. Rosenschein, and Zvi Topol. Supporting privacy in decentralized additive reputation systems. In Christian Jensen, Stefan Poslad, and Theo Dimitrakos, editors, *Trust Management*, pages 108–119, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[49] Hao Peng, Song-nian Lu, Dan-dan Zhao, and Ai-xin Zhang. Low cost and reliable anonymity protocols in P2P reputation

systems. *Journal of Shanghai Jiaotong University (Science)*, 15(2):207–212, April 2010.

[50] Ronald Petrlic, Sascha Lutters, and Christoph Sorge. Privacy-preserving reputation management. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, SAC '14, pages 1712–1718. ACM, 2014.

[51] Margaret Pless. Kiwi Farms, the web's biggest community of stalkers. https://nymag.com/intelligencer/2016/07/kiwi-farms-the-webs-biggest-community-of-stalkers.html, July 2016.

[52] Mike Rugnetta. Mike Rugnetta, Idea Channel - XOXO Festival (2013). https://www.youtube.com/watch?v=-D9Xq3Xr8aE, October 2013.

[53] Alexander Schaub, Rémi Bazin, Omar Hasan, and Lionel Brunie. A trustless privacy-preserving reputation system. In Jaap-Henk Hoepman and Stefan Katzenbeisser, editors, *ICT Systems Security and Privacy Protection*, pages 398–411. Springer International Publishing, 2016.

[54] Stefan Schiffner, Andreas Pashalidis, and Elmar Tischhauser. On the limits of privacy in reputation systems. In *Proceedings of the 10th Annual ACM Workshop on Privacy in the Electronic Society*, WPES '11, pages 33–42, New York, NY, USA, 2011. ACM.

[55] Aameek Singh and Ling Liu. TrustMe: anonymous management of trust relationships in decentralized P2P systems. In *Proceedings Third International Conference on Peer-to-Peer Computing (P2P 2003)*, pages 142–149, September 2003.

[56] Kyle Soska, Albert Kwon, Nicolas Christin, and Srinivas Devadas. Beaver: A decentralized anonymous marketplace with secure reputation. Cryptology ePrint Archive, Report 2016/464, 2016. https://eprint.iacr.org/2016/464.

[57] Adam Steinbaugh. Kevin Bollaert sentenced to 18 years over revenge porn site "You Got Posted". http://adamsteinbaugh.com/2015/04/03/kevin-bollaert-sentenced-to-years-over-revenge-porn-site-you-got-posted/, April 2015.

[58] Nguyen Tran, Bonan Min, Jinyang Li, and Lakshminarayanan Subramanian. Sybil-resilient online content voting. In *Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation*, NSDI '09, pages 15–28, USA, 2009. USENIX Association.

[59] Marco Voss. Privacy preserving online reputation systems. In Yves Deswarte, Frédéric Cuppens, Sushil Jajodia, and Lingyu Wang, editors, *Information Security Management, Education and Privacy*, pages 249–264, Boston, MA, 2004. Springer US.

[60] Marco Voss, Andreas Heinemann, and Max Muhlhauser. A privacy preserving reputation system for mobile information dissemination networks. In *First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SecureComm '05)*, pages 171–181, September 2005.

[61] Xinlei (Oscar) Wang, Wei Cheng, Prasant Mohapatra, and Tarek Abdelzaher. ARTSense: Anonymous reputation and trust in participatory sensing. In *2013 Proceedings IEEE INFOCOM*, pages 2517–2525, April 2013.

[62] Yunzhao Wei and YanXiang He. A pseudonym changing-based anonymity protocol for P2P reputation systems. In *2009 First International Workshop on Education Technology and Computer Science*, volume 3, pages 975–980, March 2009.

[63] Jonathan Wells. Tyler Oakley: How the internet revolutionised LGBT life. https://www.telegraph.co.uk/men/thinking-man/tyler-oakley-how-the-internet-revolutionised-lgbt-life/, November 2015.

[64] Danfeng Yao, Roberto Tamassia, and Seth Proctor. Private distributed scalar product protocol with application to privacy-preserving computation of trust. In Sandro Etalle and Stephen Marsh, editors, *Trust Management*, pages 1–16, Boston, MA, 2007. Springer US.

[65] Haifeng Yu, Michael Kaminsky, Phillip B. Gibbons, and Abraham Flaxman. SybilGuard: Defending against Sybil attacks via social networks. In *Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '06, pages 267–278, New York, NY, USA, 2006. Association for Computing Machinery.

[66] Ennan Zhai, David Isaac Wolinsky, Ruichuan Chen, Ewa Syta, Chao Teng, and Bryan Ford. AnonRep: Towards tracking-resistant anonymous reputation. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, pages 583–596. USENIX Association, March 2016.

[67] Mingwu Zhang, Yong Xia, Ou Yuan, and Kirill Morozov. Privacy-friendly weighted-reputation aggregation protocols against malicious adversaries in cloud services. *International Journal of Communication Systems*, 29(12):1863–1872, 2016.

A Comparison of Terminology from Previous Work

In Section 3, we noted that previous work has not converged on a standard set of terms in order to describe their systems. Even when the same terms are used across works, the meanings ascribed to the terms are often different, obscuring underlying differences in the systems. In this appendix, we provide specific mappings from the terms we use to the terminology from previous work.

We first note that, though a scant few papers do recognize the different ideas encapsulated by our terms "Simplex", "Half-Duplex", and "Full-Duplex" in reference to the directionality of reputation systems, no previous works actually develop these differences into specific terminology. The same is true of our terms "Voter-agnostic" and "Voter-conscious" in reference to reputation functions.

Table 2, referring to the terms from Section 3.1, demonstrates one of the clearest case of similar terminology obscuring underlying differences. To describe what we call "Third-Party Mediation", most papers call the architectures of competing systems "centralized" (though, interestingly, a few papers we classify as

Table 2. Mapping of Architecture Terms

Third-Party Mediation	Ephemeral Mesh Topology	Proofs of Validity
Centralized [6–9, 18, 24, 30, 31, 37, 43, 48, 53, 56, 59]	Decentralized [6, 7, 9, 18, 28–31, 48, 53]	Decentralized [6, 8, 43, 53, 56]
Semi-centralized [53]	Distributed [7, 8, 24, 37, 39, 59, 64]	
Decentralized [8, 9]		
Distributed [7, 59]		

Table 3. Mapping of Voter Privacy Properties

Voter-Vote Unlinkability	Two-Vote Unlinkability
Anonymity [10–12, 14, 17, 19, 26, 27, 35, 43, 45, 46, 49, 50, 53, 55, 59, 61, 62, 66]	Unlinkability [10–12, 14, 34, 43, 44, 53, 59, 66]
Privacy [1, 5, 7, 8, 13, 18, 24, 28–31, 33, 34, 39, 47, 48, 60, 64, 67]	Anonymity [44]
Peer-Pseudonym Unlinkability [4]	Pseudonym-Pseudonym Unlinkability [4]
Review-Payment Unlinkability [56]	Review-Review Unlinkability [56]
Transaction-Rating Unlinkability [9]	Rating-Rating Unlinkability [9]
Confidentiality [37, 43]	
Rating Secrecy [19, 44]	
“secret, unlinkable, and anonymous” [6]	

Table 4. Mapping of Votee Privacy Properties

Reputation-Usage Unlinkability	Exact Reputation Blinding
Identity Anonymity [66]	Reputation Budget [66]
Signer Anonymity [11]	“cloaking of reputation scores” [17]

Third-Party Mediation differentiate themselves as “decentralized” [8, 9] or “distributed” [7, 59] due to differences in how they use their TTPs; notably, they only refer to their own works as such and not the entirety of what we term Third-Party Mediation). However, there is significant confusion between the “decentralized” of Ephemeral Mesh Topology and of Proofs of Validity. An alternate term, “distributed”, also commonly refers specifically to what we term Ephemeral Mesh Topology; we distinguish these terms as “system-defined decentralized” and “user-defined decentralized”.

Between voter and votee privacy properties, a majority of papers focus on voter privacy properties. As such, Table 3, referring to the terms from the voter privacy portion of Section 3.3, features the greatest diversity of terms. In order to descriptively term these properties, as inspired by Kuhn *et al.* [40], we chose to name them (excluding Exact Reputation Blinding, for which this approach seemed less appropriate) with respect to an unlinkability between two entities. Three papers took a similar approach, and though they do not use the same terms “Voter-Vote” and “Two-Vote”, their choices (Peer-Pseudonym/Pseudonym-Pseudonym Unlinkability [4], Review-Payment/Review-Review Unlinkability [56], and Transaction-Rating/Rating-Rating Unlinkability [9]) embody a similar spirit. Aside from those, concerning Voter-Vote Unlinkability specifically, previous work was widely split between “anonymity” and “privacy”, though a few used other terms like “confidentiality” or “rating secrecy”. Where other papers did consider Two-Vote Unlinkability, they most commonly referred to it simply as “unlinkability”, though one paper confusingly referred to it as “anonymity” as well [44]. This paper, in concert with Kuhn *et al.*’s [40] wider suggestion to do so, particularly inspired our desire to avoid the word “anonymity” in the name of any of our terms.

Votee privacy properties were typically less commonly provided by systems, so it is not surprising that terminology is not as frequently developed. What terminology was developed can be observed in Table 4, referring to the terms from the votee privacy portion of Section 3.3. Reputation-Usage Unlinkability is closely related to Voter-Vote Unlinkability, and as “anonymity” was a common choice for that term, both examples we observed were modifications of anonymity (“identity anonymity” [66] and “signer anonymity” [11]). Exact Reputation Blinding is similarly obscure, and we felt that the only specific terms used previously (“reputation budget” and “cloaking of reputation scores”) were not adequately descriptive of what the property accomplished.

Table 5, referring to the terms from Section 4, is very sparse. We believe this is due to the fact that a majority of papers considered in this systematization either completely ignored reputation functions in their system design, or only worked with one specific function and did not see a need to name it. The unusual “terms” in PrivBox [6] (being actually just sets of values that can be used to vote with) come from the fact that in that paper, instead of naming these as functions, the paper makes comparison directly to the choices of values to

Table 5. Mapping of Reputation Functions

Accrue Stars	Average Stars	Gompertz function	Short-term Memory Consensus	Long-term Memory Consensus
Sum [50]	Mean [53]	Gompertz function [33]	Ordered Weighted Average [47]	N/A
(0, 1) / (0, 1, -1) [6]	(0–5) [6]			

vote with instead. These sets are then used as a proxy for the functions themselves. Again, we felt that our names were more descriptive of the actual functions they describe, with the exception of the Gompertz function, which we take directly from Huang *et al.* [33].

B The Gompertz Function

Although most of the reputation functions we describe in Section 4 are common and/or simple to describe, the Gompertz function is relatively complex. As mentioned above, it is only used as the reputation function in one system we identify, that of Huang *et al.* [33]. In this appendix, we elaborate on the Gompertz function.

As discussed above, the Gompertz function is intended to model the trust of humans in social interactions by allowing reputation to slowly increase but quickly decrease. It was specifically suggested for use in participatory sensing, where a single server, the only voter, evaluates the quality of data submitted by a set of devices, the votees. Systems that use the Gompertz function operate with respect to epochs. Votes are cast once per epoch, and more recent epochs are weighted more highly in the output of the function. Votes are also normalized per epoch; if all votees receive high ratings in one epoch, the effect is the same as if all votees receive low ratings.

In this description, the following notation is used. V represents the options of what a user may rate another user. S represents the potential output reputation ratings possible with the function. U represents the set of users involved in the system (in the case of participatory sensing, specifically the votees). T represents the set of epochs during which the system operates. $\overrightarrow{x_{u,t_k}} \in V^*$ represents an arbitrary-length list of votes assigned to a user $u \in U$ over a sequence of epochs $\langle t_i \mid t_i \in T \wedge t_1 \leq t_i \leq t_k \rangle$. $x_{u,t_i} \in \overrightarrow{x_{u,t_k}}$ represents the specific vote for user u occurring during the epoch t_i. $\widehat{x_{t_i}}$ represents the average value of a vote in epoch t_i, and is calculated as follows: $\widehat{x_{t_i}} = \frac{\sum_{u \in U} x_{u,t_i}}{|U|}$. $\widehat{x_{u,t_i}}$ represents the vote for user u during the epoch t_i, normalized against all other votes during the epoch t_i, and is calculated as follows: $\widehat{x_{u,t_i}} = \frac{x_{u,t_i}}{\widehat{x_{t_i}}}$. $\Phi : V^* \to S$ represents the Gompertz function itself.

The Gompertz function takes three parameters, $b \in \mathbb{R}^-, c \in \mathbb{R}^-, \lambda \in (0, 1]$. System designers are directed to choose these parameters such that, when real votes are input to the function, reputation values slowly increase and quickly decrease as intended. In this reputation function, $V = [0, 1]$ and $S = [0, 1]$. Φ is defined as follows:

$$\Phi(\overrightarrow{x_{u,t_k}}) = e^{be^{\left(c \sum_{i=1}^{k} \widehat{x_{u,t_i}} \lambda^{(t_k - t_i)}\right)}}$$

As mentioned above, more recent epochs are weighted more highly in the output of the function. This is accomplished by the choice of $\lambda \in (0, 1]$ and the $\lambda^{(t_k - t_i)}$ term. When λ is closer to 1, less priority is given to more recent epochs, and when λ is closer to 0, more priority is given to more recent epochs.

Further, votes being normalized per epoch is accomplished by the $\widehat{x_{u,t_i}}$ term, and is a fairly straightforward result from the definition of this term.

The specific response of the function to high and low reputation scores, increasing and decreasing at appropriate rates, is determined by choice of b and c. This function generates a logistic curve. The choice of c of may "tighten" or "loosen" the curve; a c with greater absolute value makes the function reach its asymptotes at $\Phi(\overrightarrow{x_{u,t_k}}) = 0$ and 1 more quickly with smaller change in vote values. The choice of b shifts the curve; a b with greater absolute value moves the curve's inflection point to happen only with greater vote values.