# Formal Verification of e-Reputation Protocols[⋆]

Ali Kassem[2], Pascal Lafourcade[1], and Yassine Lakhnech[2]

[1] Université d'Auvergne, LIMOS
[2] Université Grenoble Alpes, CNRS, VERIMAG, France

**Abstract.** Reputation systems are often useful in large online communities in which most of the users are unknown to each others. They are good tools to force the users to act in truthfulness way. However, for reputation systems to work effectively users have to be willing to provide rates. In order to incentivize the users to provide honest rates, a reputation system have to ensure their privacy and anonymity. In the other hand, users are also concerned about verifying the correctness of the reputation score. In the applied pi-calculus, we define a formal framework and several fundamental privacy, authentication, and verifiability properties suitable for the security analysis of e-reputation protocols. As proof of concept, using ProVerif, we analyze a simple additive decentralized reputation protocol proposed to ensure rate privacy if all users are honest.

**Keywords:** Reputation Protocols, Formal Verification, Privacy, Authentication, Verifiability, Applied Pi-Calculus, ProVerif

## 1 Introduction

Electronic reputation (in short, *e-reputation*) systems are tools for the users to quantify the trust between each other. Electronic commerce, social news, peer-to-peer routing, and collaborative environments are examples of applications highly benefit from using e-reputation systems. Indeed, similar to the word-of-mouth reputation, an e-reputation system allows users to form an opinion on the behavior of an unknown user service provider through a reputation score. A reputation score is a mathematical value[3] (*e.g.,* the number of all users that provided a positive feedback, or the percentage of those provided a positive one from all the users that provided feedback) computed from the opinions of users that have been interacted with the service provider. For example, a user may rate a service provider on eBay or a restaurant on Yelp as useful, and these ratings allow others to identify more easily the best service provider (or product). In the following, whatever the type of parties (clients, service providers, nodes, website, etc.)

[⋆] This research was conducted with the support of the "Digital trust" chair from the Foundation of the University of Auvergne.

[3] Note that, the reputation score could be a vector computed from detailed opinions of the users, where each value reflects the quality of a specific task such as item describing accuracy, shipping speed,etc.

involved in a reputation system we call them *users*, and the opinions or feedbacks they provide about each other are called *rate*s. We explicitly distinguish the user that is currently receiving a rate by calling him *target user*.

E-reputation protocols are becoming an important tool in online communities. However, many adversarial behaviors can affect such systems [8]. Users may collude to collectively subvert the system either by giving a negative (unfair) rates on the victim in order to destroy its reputation–what is called *bad-mouthing* attack [10], or by advertising the quality of service of a certain user more than its real value to increase his reputation–what is called *ballot-stuffing* attack [10]. Moreover, reputation systems may have to face *Sybil attack*, that is users that pollute the system by creating numerous fake identities [13]. In the worst case, some parties (*e.g.,* authority, or users) may act dishonestly and modify the rates to obtain a wrong reputation score. To limit such behavior, users have to be able to check for irregularities, or to prove their absence. Target user may wish to ensure verifiability too. He may want to be transparent to inspire trustworthiness in his reputation score.

For reputation systems to be effective, in addition to their correctness, users have to be willing to provide rates. In order to incentivize the users to provide honest rates, e-reputation protocols have to ensure their privacy and anonymity. Usually, users wish to remain anonymous as they would like to be sure that the rate they provide cannot be used in a way that can affect them in the future. Actually, it has been shown that preserving the privacy of users encourage them to feed the reputation systems with honest ratings without fearing retaliation [22].

*Contributions:* In this paper, we provide the means to analyze the privacy of e-reputation protocols and verify their correctness. Precisely, we model e-reputation protocols in the applied pi-calculus [1], we define four privacy properties: *Rate Privacy*, rates provided by the users are kept secret; *Rate Anonymity*, an attacker cannot relate a certain rate to the user provided it; *Receipt-Freeness*, a user cannot prove to an attacker that he provided a certain rate about the target user; *Coercion-Resistance*, even when interacting with a coercer, the user can still provide a rate of his choice. We also give two authentication properties: *Rate Integrity*, the rate is recorded as sent (casted) by the user unmodified; *User Eligibility*, only eligible users (those who accomplish a successful interaction) can cast rates, and two verifiability properties: *Reputation Score Verification*, any one can verify the validity of a certain user score; *User Eligibility Verification*, any one can verify that every counted rate is casted by an eligible users (those who accomplish a successful interaction). Finally, we validate the effectiveness of the proposed approach by analyzing the security of an e-reputation protocol [21] using ProVerif [7].

The proposed properties still cannot detect attacks such as bad-mothing and ballot-stuffing attacks, but properties such as: *User Eligibility Verification*, *Receipt-Freeness*, and *Coercion-Resistance* can minimize them. As *User Eligibility Verification* imposes some constraints on the users in order to provide a rate (*e.g.,* accomplish a successful interaction); *Receipt-Freeness* can limit the bribing and positive rate exchange between users as they cannot provide a proof that they provided a certain rate; similarly does *Coercion-Resistance* even when interacting with a coercer.

*Related Work:* As an example of simple reputation system is the one used by eBay. After the exchange is accomplished between users (a client and a service provider), they can rate each other with a rating, -1, 0, or +1 in accordance with their satisfaction. A more sophisticated reputation systems have been recently proposed. Androulaki *et al.* propose an reputation protocol relies on trusted central authority to demonstrate the validity of rates in [4]. Pavlov *et al.* [21] were the first ones to propose a decentralized reputation protocol to preserve users privacy. Others try to have privacy preserving protocols [2, 24, 25] by dealing with *unlinkability* that is to ensure that an attacker cannot distinguish whether the same user is involved in two interactions or not. Bethencourt *et al.* [6] formally define the anonymity of both the users and the target user, and propose a protocol argued informally to satisfy those definitions. Anceaume *et al.* extend their work in [3] to handle non-monotonic ratings and mention additional security properties concerned in reputation scores correctness.

However, to best of our knowledge no general formal framework that allows the verification of the security properties in e-reputation protocols have been given. In some related domains, there are numerous papers presenting the formalization and verification of the security properties, for instance in e-voting [5, 9, 16, 17], in e-auction systems [12, 15, 18], and in e-exams [14]. Some of the security properties therein studied seem to relate with those we present for e-reputation protocols. For instance, user eligibility is analogous to voter, bidder, and student eligibility. Rate privacy reminds vote, bid, and mark privacy.

However, still there are fundamental differences. In voting, the candidates and the voters (and thus the maximum possible number of votes) are already known, and after voting process the total number of votes and that taken by each candidate will be publicly available. Thus, there is a certain leakage of information. For example, if a candidate does not receive any vote, the attacker can exclude this previously possible option. While in reputation all users can rate each others playing two rules at the same time (rate provider and target user). Also, the number of provided rates is not (necessarily) publicly known. Thus, having a reputation score does not give us any information about the number of the rates provided for the other users. Also, in protocols that support both negative and positive rates, a score zero does not means zero rates. Even in case of only positive rates, a score zero could due to the fact that this user did not make any interaction with the others yet, not necessarily means that users provides their rates to other target users like in voting. Actually, in reputation a user do not have to choose between different target users as he can provides rates for all users he interacted with. Note that, providing a rate for a certain user is not always good like when you vote for a certain candidate in voting, as the rate could be a bad one. Hence, reputation systems have many differences from voting systems.

Furthermore, the threat models for reputation, voting, auctions and exams are different: in voting collusion between the voters and the candidates (bribing) aims to see a candidate win; in reputation collusion between the users by bribing, by exchanging good rates or even making fake interaction aims to increase their reputation scores; similarly in exams collusion between the student and the examiner aims to get the highest possible mark; it is different in auction since bidders want to win with the lowest possible price, but seller want to sell with the highest possible one.

*Outline:* In Section 2, we model e-reputation protocols in the applied pi-calculus. We formally express the security properties in Section 3. In Section 4, we validate our framework by analyzing the security of an e-reputation protocol [21]. Finally, we conclude and outline the future work in Section 5.

## 2 Modelling

We model e-reputation protocols in the applied pi-calculus [1], a process calculus designed for the verification of cryptographic protocols. To perform the automatic protocol verification, we use ProVerif [7]. This tool uses a process description based on the applied pi-calculus, but has syntactical extensions and is enriched by events to check reachability and correspondence properties. Besides, it can check equivalence properties. We use the *labeled bisimilarity* ($\approx_l$) to express the equivalence between two processes [1]. Informally, two processes are equivalent if an observer has no way to distinguish between them.

Precisely, honest parties are modeled as processes in the applied pi-calculus. These processes can exchange messages on public or private channels, create keys or fresh random values and perform tests and cryptographic operations, which are modeled as functions on terms with respect to an equational theory describing their properties.

The Dolev-Yao attacker [11] has complete control of the network, except the private channels: he can eavesdrop, remove, substitute, duplicate and delay messages that the parties are sending one another, and insert messages of his choice on the public channels. To capture threats due to collusions and coercions, we assume dishonest parties. They cooperate with the attacker, revealing their secret data (*e.g.,* secret keys) to him, or taking orders from him (*e.g.,* what rate to provide). We model such dishonest parties as in Definition 10 and 15 from [9]: if the process $P$ is an honest party, then the process $P^{c_1}$ or $P^{c_1,c_2}$ is its dishonest version. The process $P^{c_1}$ is a variant of $P$ which shares with the attacker channels $c_1$. Through $c_1$, $P^{c_1}$ sends all its inputs and freshly generated names (but not other channel names). The second process $P^{c_1,c_2}$ does not only reveal the secret data on channel $c_1$, but also takes orders from the attacker on the channel $c_2$ before sending a message or branching. This models a completely corrupted party. To hide the outputs of an extended process (a process which may contain an active substitutions $\{^m/_x\}$) on a certain channel, we use the Definition 11 from [9]: if $A$ is an extended process, then the process $A^{\backslash out(ch,\cdot)}$ is a variant of $A$ that hides all the outputs on channel $ch$, it is defined by $\nu ch.(A|!in(ch,x))$. For more details about the applied pi-calculus, its standard results and all the definitions used in this paper, we refer to the papers [1,9].

Reputation protocols have some important differences. However, a large class of them can be represented as follows.

**Definition 1. (Reputation Protocol).** *A* reputation protocol *is defined by a tuple $(U, T, A_1, \ldots, A_l, \tilde{n}_p)$, where $U$ is the process executed by the users, $T$ is the process executed by the target user; the one we looking for his reputation score, $A_i$'s are the processes executed by the authorities, and $\tilde{n}_p$ is the set of private channel names.*

A reputation protocol involves users who provide the rates about the target user, and the protocol authorities who often handle the rates, calculate the reputation score, distribute the scores, etc. Note that all users execute the same process, but with different variable values, *e.g.,* keys, identities, and rates. In some protocols, especially decentralized ones such as [21], reputation scores are computed upon request by a certain user; the one looking to know the reputation score of the target user (due to a potential interaction with him), this user is represented as on of the authorities as usually his task involve some organizational work.

To reason about privacy, we talk about *reputation processes*; instances of a reputation protocol.

**Definition 2. (Reputation Process).** *Given a reputation protocol a* reputation process *is a closed process $\nu\tilde{n}.(U\sigma_{id_1}\sigma_{r_1}, \ldots, U\sigma_{id_n}\sigma_{r_n}, T, A_1, \ldots, A_m)$ where $\tilde{n}$ is the set of all restricted names, which includes the set of the protocol's private channels; $U\sigma_{id_i}\sigma_{r_i}$'s are the processes run by the users, the substitution $\sigma_{id_i}$*

*specifies the user's identity and $\sigma_{r_i}$ specifies the rate given by the user $id_i$; $T$ is the process runs by the target user; and $A_1, \ldots, A_m$ are the processes run by the authorities.*

As a notation, we use what in applied pi-calculus is called "context". The context $RP_I[\_]$ is the process $RP$ without the processes whose identities are in the set $I$; they are replaced by "holes". We use this notation when we need to specify exactly, for instance, the processes of the users $id_1$ and $id_2$ without repeating the entire reputation process. This is done by rewriting $RP$ as $RP_{\{id_1,id_2\}}[U\sigma_{id_1}\sigma_{r_1}|U\sigma_{id_2}\sigma_{r_2}]$.

## 3 Security Properties

In the following, we formally define our security properties.

### 3.1 Privacy Properties

We model our four privacy properties; *Rate Privacy*, *Rate Privacy*, *Receipt-Freeness*, and *Coercion-Resistance* as observational equivalence, a standard choice for such kind of properties [23]. We use the labeled bisimilarity to express the equivalence between two processes.

The first property, *Rate Privacy*, says that user rates have to be secret. Keeping the rates secret gives the users more incentive to provide honest rates.

**Definition 3 (Rate Privacy).** *A reputation protocol ensures* Rate Privacy *if for any reputation process $RP$, any user $id$, and any two rates $r$, $r'$, we have that:* $RP_{\{id\}}[U\sigma_{id}\sigma_r] \approx_l RP_{\{id\}}[U\sigma_{id}\sigma_{r'}]$.

*Rate Privacy* states that two processes with different rates have to be observationally equivalent. Note that, such a property can be defined as a reachability property: an attacker can not reach a state where the rate $r$ is in his knowledge. However, modeling it as an equivalence property is stronger, as this prevents the attacker from obtaining any information about the rate.

Here, we can consider dishonest target user, as he might be interested in knowing the rates provided by the others users about him. We can do this by replacing honest $T$ with dishonest one. If we assume that $T$ has an identity $id_t$, we obtain $RP_{\{id,id_t\}}[U\sigma_{id}\sigma_r|T^{c_1,c_2}] \approx_l RP_{\{id,id_t\}}[U\sigma_{id}\sigma_{r'}|T^{c_1,c_2}]$. We can also add another dishonest users using the same technique, however the user who provides the two different rates has to be honest. Otherwise the property can be trivially violated by him revealing his rate to the attacker. This technique can be used in all the following properties if we want to add some dishonest parties.

The previous definition of *Rate Privacy* ensures that the attacker cannot know the rates of the users. However, in practice the rates could be publicly available, *e.g.,* in some sites we could find that $n$ users provide a certain rate without mentioned the identities of these users. Another variant of *Rate Privacy* is *Rate Anonymity*, *i.e.,* the attacker might know the list of all rates, but is unable to associate a certain rate to its corresponding user.

**Definition 4. (Rate Anonymity).** *A reputation protocol ensures* Rate Anonymity*, if for any reputation process $RP$, any users $id_1$, $id_2$, and any rates $r_1$, $r_2$, we have that:* $RP_{\{id_1,id_2\}}[U\sigma_{id_1}\sigma_{r_1}|U\sigma_{id_2}\sigma_{r_2}] \approx_l RP_{\{id_1,id_2\}}[U\sigma_{id_1}\sigma_{r_2}|U\sigma_{id_2}\sigma_{r_1}]$.

This definition states that the process where user $id_1$ provides a rate $r_1$ and user $id_2$ provides a rate $r_2$ is equivalent to the process where $id_1$ provides a rate $r_2$ and $id_2$ provides a rate $r_1$. This prevents the attacker from obtaining the identity of the user who provides a certain rate.

A protocol that ensures *Rate Privacy* also ensures *Rate Anonymity*. We have $RP_{\{id_1,id_2\}}[U\sigma_{id_1}\sigma_{r_1}|U\sigma_{id_2}\sigma_{r_2}] \approx_l RP_{\{id_1,id_2\}}[U\sigma_{id_1}\sigma_{r_2}|U\sigma_{id_2}\sigma_{r_2}]$ using *Rate Privacy* since only the rate provided by the user $id_1$ is changed from $r_1$ in the left side to $r_2$ in the right one. Similarly, by changing the rate of $id_2$, we have: $RP_{\{id_1,id_2\}}[U\sigma_{id_1}\sigma_{r_2}|U\sigma_{id_2}\sigma_{r_2}] \approx_l RP_{\{id_1,id_2\}}[U\sigma_{id_1}\sigma_{r_2}|U\sigma_{id_2}\sigma_{r_1}]$.

For *Receipt-Freeness* and *Coercion-Resistance* we follow the definitions introduced in [9].

A protocol is receipt-free, if an attacker cannot distinguish between a situation where a user $id_1$ provides a rate $r_c$ according to the attacker's wishes and reveals his data on a channel $ch$, and a situation where $id_1$ actually provides a rate $r_1$ of his choice and pretends to reveal his secret data (this is modeled by process $U'$). The process $U'$ is a process in which user $id_1$ provides a rate $r_1$, but communicates with the attacker (coercer) to trick him by saying that his desired rate $r_c$ is provided. This can be done by providing the attacker a fake receipt, *e.g.,* using a trapdoor to generate a different opening key. Note that, user $id_2$ swaps the rates with $id_1$ to avoid the the case where the attacker can distinguish the situations by counting the rates, if possible.

**Definition 5 (Receipt-Freeness).** *A reputation protocol ensures* Receipt-Freeness *if for any reputation process $RP$, any users $id_1$, $id_2$, and any rates $r_1$, $r_c$, there exists a closed plain process $U'$ such that:*

– $U'^{\backslash out(ch,.)} \approx_l U\sigma_{id_1}\sigma_{r_1}$, *and*
– $RP_{\{id_1,id_2\}}[(U\sigma_{id_1}\sigma_{r_c})^{ch}|U\sigma_{id_2}\sigma_{r_1}] \approx_l RP_{\{id_1,id_2\}}[U'|U\sigma_{id_2}\sigma_{r_c}]$.

*Coercion-Resistance* is a stronger property than *Receipt-Freeness*, as the attacker can not only ask for a receipt, but is also allowed to interact with the user during the rating process and to provide the messages the user should send.

**Definition 6 (Coercion-Resistance).** *A reputation protocol ensures* Coercion-Resistance *if for any reputation process $RP$, there exists a closed plain process $U'$ such that for any $\sigma_?$, and context $C = \nu c_1.\nu c_2.(\_|P)$ satisfying $\tilde{n} \cap fn(C) = \emptyset$ and $RP_I[C[(U\sigma_{id_1}\sigma_?)^{c_1,c_2}]|U\sigma_{id_2}\sigma_{r_1}] \approx_l RP_I[(U\sigma_{id_1}\sigma_{r_c})^{ch}|U\sigma_{id_2}\sigma_{r_1}]$, where $I = \{id_1, id_2\}$, we have that:*

- *$C[U']^{\backslash out(ch,.)} \approx_l U\sigma_{id_1}\sigma_{r_1}$, and*
- *$RP_I[C[(U\sigma_{id_1}\sigma_?)^{c_1,c_2}]|U\sigma_{id_2}\sigma_{r_1}] \approx_l RP_I[C[U']|U\sigma_{id_2}\sigma_{r_c}]$.*

Here, the context $\mathcal{C}$ models the attacker's behaviors which tries to force the user to provide the rate $r_c$. Note that, no matter what rate the user $id_1$ intends to provide ($\sigma_?$), the attacker will force him to provide the rate $r_c$.

## 3.2 Authentication Properties

We model the authentication properties as correspondence properties, a well-known approach [23]. Correspondence properties capture relationships between events, which may have the same structure *"if a event $e$ is executed the event $e'$'s has been previously executed"*. Events are annotations that do not change a process behavior, but are inserted at precise locations to allow reasoning about the authentication properties.

To define our two authentication properties, we use the following events:

- Event $sent(id_u, id_t, r)$ emitted when the user $id_u$ send a rate $r$ (or sum of rates) for the authority (or responsible party) to evaluate the target user $id_t$. This event is emitted just before sending the message containing the rate.
- Event $record(id_u, id_t, r)$ emitted when the rate $r$ (or sum of rates) from the user $id_u$ provided about the target user $id_t$ is received by the authority (or the intended party). This event is placed after receiving the rate and perform the required checks before accepting it, if any.
- Event $eligible(id_u, id_t)$ emitted when the user $id_u$ is certified as an eligible user to provide a rate about the target user $id_t$. It is placed just before providing the credential by the responsible party.

After placing these events inside the reputation process, the authentication properties can then defined as follows:

First, *Rate Integrity*, which ensures that the rate is not altered and received by the responsible party as it provided by the user.

**Definition 7 (Rate Integrity).** *A reputation protocol ensures* Rate Integrity*, if for every reputation process $RP$ each occurrence of the event $record(id_u, id_t, r)$ is preceded by a distinct occurrence of the corresponding event $sent(id_u, id_t, r)$ on every possible execution trace.*

The second property is *User Eligibility*, which ensures that only the users have a certain credential can provide a rate. The credential can be a certificate from an authority, a proof that he has been interacted with the target user, or marked in some database as interacted with the target user. It is defined using the following two events:

**Definition 8 (User Eligibility).** *A reputation protocol ensures* User Eligibility*, if for any reputation process $RP$ each occurrence of the event $record(id_u, id_t, r)$ (for any $r$) is preceded by a distinct occurrence of the corresponding event $eligible(id_u, id_t)$ on every possible execution trace.*

### 3.3 Verifiability Properties

Similar to [15], we define a verification test as an efficient terminating algorithm that takes as input the data visible to a participate of the reputation protocol and returns a boolean value ($true$ or $false$). To reason about verifiability properties, we extend the reputation process with the following functions and variables:

- $rs$ is a variable referring to the reputation score assigned to the target user;
- $L : List(ERate)$ is a list of (encrypted or anonymized) rates provided by the users about the target user;
- $getRate : ERate \mapsto Rate$ is a function that maps the (encrypted or anonymized) rates to a regular rates (*e.g.,* an integer). This function does not need to be computable for any party, as it is only used to define the verification test;
- $compRep$ is a function that computes the reputation score given a list of rates. This might be simply the summation of all the rates, but there may be more complex operations to determine this value;
- $isEligible : ID \times ID \mapsto \{true, false\}$ is a function which takes a rate and returns $true$ if the rate was provided by a certified user (user which interacted with the target user).

We write $getRate(L)$ for $getRate(L[1]), \ldots, getRate(L[n])$, where $L[i]$ is the $i^{th}$ entry of the list $L$.

Users would like to verify the validity of the reputation score of the target, this is ensured by *Reputation Score Verification*.

**Definition 9 (Reputation Score Verification).** *A reputation protocol ensures* Reputation Score Verification*, if we have a test $RSV$ respecting the following:*

- *Soundness: $RSV = true \Rightarrow rs = compRep(getrate(L))$;*
- *Completeness: if all participants follow the protocol correctly, then $rs = compRep(getrate(L)) \Rightarrow RSV = true$.*

Another interesting verifiability property is *User Eligibility Verification* which allows anyone to check if every counted rate is provided by an eligible user.

**Definition 10 (User Eligibility Verification).** *A reputation protocol ensures* User Eligibility Verification*, if we have a test $UEV$ respecting the following conditions:*

- *Soundness: $UEV = true \Rightarrow \forall r \in L, isEligible(r) = true$;*
- *Completeness: if all participants follow the protocol correctly, then $\forall r \in L, isEligible(r) = true \Rightarrow UEV = true$.*

Note that, this property is different from the *User Eligibility* presented before. A protocol satisfying *User Eligibility* means that it authenticates the users and only allows eligible ones to provide rates, but this does not necessarily means that anyone (*e.g.,* another user) can himself verify that the rates were provide by an eligible users as the protocol might not provide a proof for that. In the protocols that satisfy *User Eligibility* but not *User Eligibility Verification*, users usually have to trust a certain authority about the eligibility of the users, however in those satisfying *Use Eligibility Verification* a user can himself verify the eligibility of the users and thus detecting any error or cheating, and does not have to trust any authority concerning this point as the authority itself might act dishonestly.

## 4 Case Study

Using ProVerif [7], we applied the previously explained definitions on the first protocol proposed by Pavlov *et al.* in [21]. This protocol was designed to provide privacy of the rates in decentralized additive reputation systems, if all users flow the protocol correctly *i.e.,* all users are honest. It also assumes that all users provide honest rating about the target user (*i.e.,* rating that correctly reflects their satisfaction), as the protocol cannot prevent the users from providing an unfair rating to increase (or decrease) the reputation score of the target user more (or less) than its real value. The authors argue that the protocol preserves the secrecy of all the rates if the users do not collude with each others.

*Informal Description.* The basic idea behind this protocol is to consider the rate provided by each user to be his secret information. The rates are cumulatively added to each other, without revealing them, to obtain a sum represents the reputation score of the target. The protocol is initiated by a querying agent $A_q$ looking to know the reputation score of the target user, and proceeded as follows:

1. Initialization Step: the querying agent $A_q$ orders the users in a ring: $A_q \rightarrow U_1 \rightarrow, \dots, \rightarrow U_n \rightarrow A_q$, and sends to each user $U_i$ the identity of his successor in the ring, *i.e.,* for $i \in \{1, \dots, n-1\}$ it sends the identity of $U_{i+1}$ to $U_i$, and for $U_n$ it sends its identity.
2. $A_q$ chooses a random number $r_q \neq 0$ and sends it to $U_1$.
3. Upon reception of $r_p$ from his predecessor in the ring, each user $U_i$ for $i \in \{1, \dots, n\}$ calculates $r_p + r_i$ and sends the obtained value to his successor in the ring, where $r_i$ is the reputation rate of the user $U_i$ about the target user.
4. Upon reception of the feedback from $U_n$, $A_q$ subtracts $r_q$ from it in order to obtain the reputation score of the target user represented by the sum of all rates.

To ensure secrecy and authentication, the designers of the protocol assume an authenticated secure channel between every two users.

*Formal Model.* Generally, it is difficult to model arithmetic operations in formal protocol provers such as ProVerif. However, we build a simple equational theory which handles the required arithmetics to verify the protocol for the case where we have two users $U_1$ and $U_2$ in addition to the querying agent.

We modeled the protocol in ProVerif using a standard equational theory for symmetric encryption (functions senc and sdec), $\texttt{senc}(\texttt{m}, \texttt{k})$ represents the symmetric encryption of the message $m$ with the key $k$ and sdec represents the decryption function; and an equational theory for arithmetic addition and subtraction (functions sum and sub). The function sum takes two values and return their sum. Having x or y we can obtain the other one from $\texttt{sum}(\texttt{x}, \texttt{y})$ using the function sub. Similarly, we can obtain $\texttt{sum}(\texttt{y}, \texttt{z})$ and $\texttt{sum}(\texttt{x}, \texttt{z})$ from $\texttt{sum}(\texttt{sum}(\texttt{x}, \texttt{y}), \texttt{z})$ having x and y respectively. Note that with this equational theory, having x or y, one cannot obtain $\texttt{sum}(\texttt{z}, \texttt{y})$ or $\texttt{sum}(\texttt{z}, \texttt{x})$ from $\texttt{sum}(\texttt{z}, \texttt{sum}(\texttt{x}, \texttt{y}))$. Solving this requires two additional equations similar to the last two equations presented below. One could also says why not modeling sum as a commutative function, *i.e.,* $\texttt{sum}(\texttt{x}, \texttt{y}) = \texttt{sum}(\texttt{y}, \texttt{x})$, which solves this problem and moreover allows us to represent the subtraction using only two equations instead of four. This is due to a ProVerif problem as both solutions cause non termination. Actually, it is well known that ProVerif has difficulties with commutative operations. However, we overcome this weakness by giving the value $r_p$ received by the user process $U_i$

as the first argument to the function $\mathtt{sum}$, and his rate $r_i$ as the second argument. Thus, such a term $\mathtt{sum}(\mathtt{z}, \mathtt{sum}(\mathtt{x}, \mathtt{y}))$ does not appear between the messages involved in the protocol.

$$\begin{aligned}
\mathtt{sdec}(\mathtt{senc}(\mathtt{m}, \mathtt{k}), \mathtt{k}) &= \mathtt{m} \\
\mathtt{sub}(\mathtt{sum}(\mathtt{x}, \mathtt{y}), \mathtt{x}) &= \mathtt{y} \\
\mathtt{sub}(\mathtt{sum}(\mathtt{x}, \mathtt{y}), \mathtt{y}) &= \mathtt{x} \\
\mathtt{sub}(\mathtt{sum}(\mathtt{sum}(\mathtt{x}, \mathtt{y}), \mathtt{z}), \mathtt{x}) &= \mathtt{sum}(\mathtt{y}, \mathtt{z}) \\
\mathtt{sub}(\mathtt{sum}(\mathtt{sum}(\mathtt{x}, \mathtt{y}), \mathtt{z}), \mathtt{y}) &= \mathtt{sum}(\mathtt{x}, \mathtt{z})
\end{aligned}$$

An alternative equational theory could be modeling the sum of two numbers $x$ and $y$ as their exclusive-or (XOR), *i.e.,* $\mathtt{xor}(\mathtt{x}, \mathtt{y})$ instead of $\mathtt{sum}(\mathtt{x}, \mathtt{y})$. In this case the subtraction will represented by another XOR application, *i.e.,* $\mathtt{xor}(\mathtt{xor}(\mathtt{x}, \mathtt{y}), \mathtt{y}) = \mathtt{x}$. ProVerif, originally does not handel the XOR operator, however Küsters *et al.* show how to reduce the derivation problem for horn theories with XOR to the XOR-free case in [20]. Their reduction allows one to carry out the analysis of the protocols that involve XOR operator using tools, such as ProVerif. We belive that their result allows us to carry out the analysis (possiblly for more than two users) with ProVerif, using the XOR operator to model the summation and subtraction. However, we kept this for future work.

All the parties (querying agent and users) are modeled as honest parties. To model the authenticated secure channel, all the messages are exchanged encrypted with a symmetric key shared between all the users, for a secure communication. For authentication, the unique identities of the sender and the receiver are included in the message to authenticate the sender, and ensure that only the intended receiver will receive the message. Note that, the attacker can still block, or re-play the messages.

*Analysis.* The results of our analysis are detailed below.

*Rate Privacy:* In case of only one user, it is clear that we do not have *Rate Privacy* as the reputation score will be equal to the rate of this user, and thus the querying agent can knows this rate. We show, using ProVerif, that the protocol ensures *Rate Privacy* in the case of two users (other than the querying agent). As we mentioned, to model the authenticated channel we add the identities to the messages. Note that, if the identity of the receiver is removed from the messages, the attacker can re-direct the message sent by the first user, which contains $sum(r_q, r_1)$, to the querying agent instead of the second user, and thus the rate of the first user will be enclosed to the querying agent even if he acts honestly. A similar attack will enclose the rate of the last user (herein user two) if the attacker re-direct the first message by querying agent, which contains $r_q$ to the last user. To make the attack on the intermediate users (in case of more than two users) the attacker needs two sessions to be initiated by the same querying agent. Note that, in all these attacks the rate will only enclosed to the querying agent and not to the attacker unless that the obtained reputation score is published by the querying agent.

*Rate Anonymity:* ProVerif was able to show that the protocol ensures *Rate Anonymity* in the case of two users (other than the querying agent).

*Receipt-Freeness:* The protocol does not ensure Receipt-Freeness since the shared key $k$ can be used as a receipt which allows the attacker to enclose the value of the message received by the victim user and that sent by him, then subtract them from each other to check whether the difference is equal to the rate $r_c$ he wants, or not. Note that, the user cannot lie about the key by giving the attacker an different key $k'$ since, for any keys $k' \neq k$ and rates $r_c \neq r_i$, we cannot have that $r_c = sub(sdec(senc(sum(r_p, r_i), k), k'), sdec(senc(r_p, k), k'))$ .

*Coercion-Resistance:* As *Receipt-Freeness* does not hold, then *Coercion-Resistance* also does not hold since by Proposition 18 of [9], if a protocol ensures *Coercion-Resistance* then it also ensures *Receipt-Freeness*.

*Rate Integrity:* We check the integrity of the messages exchanged between the parties (querying agent and users). Thus events are placed in both querying agent and user processes. ProVerif shows that the integrity of the messages is preserved if the correspondance is modeled without injectivity, and terms types (sorts) are respected. Note that, the injectivity does not hold since the attacker can re-play the message several times, and thus we will have several emission of the event *record* preceded by only one emission of event *sent*. Note also that a flaw attack exists if we ignore the types of the terms: for example, the first message sent by the querying agent to the first user to inform him about the identity of his successor, which has a type `ID`, might be received by this user as a rate $r_q$, which has another type `Rate`. Thus, the event *record* will emitted but not the event *sent*.

*User Eligibility:* We show with ProVerif that *User Eligibility* is ensured by the protocol. We assume that the users included in the ring are eligible as they are registered in a certain database. Thus, the event *eligible* is emitted when the querying agent chooses the user as a node in the ring. The event *record* emitted when the rate is received by his successor in the ring (actually the summation of all privious rates is received).

*Reputation Score Verification:* The users can only get the reputation score, however they have no means to check if this score is calculated correctly from the users rates. However, if the rates and the score are published encrypted with the shared key in a Bulletin Board, then we can design a test that allows the users to verify the reputation score. The test takes users rates and the reputation score of the target user, and simply checks if the summation of all rates is equal to the score. Note that, verifability could destroys the privacy of the rates. We show using ProVerif that the test is sound and complete.

*User Eligibility Verification:* Users that provides rates are not publicly known. Also, users do not provide any proof (*e.g.,* certificate from authority) which could allow us to verify their eligibility. Note that, as user eligibility is ensured, according to the definition of *User Eligibility Verification* the test that always gives true is sound and complete. However, this test is dummy as it does not provide any information for the users to remove their doubts, trusting the authority is all what they have to do instead.

## 5 Conclusion and Future Work

We set the first research step on the formal understanding of e-reputation systems, and establishes a framework for the automatic analysis of their security requirements. In particular, we show how to model reputation protocols in the applied pi-calculus, and how security properties such as privacy, authentication, and verifiability properties can be expressed.

We validate our model and definitions by analyzing, using ProVerif, the security of an e-reputation protocol, the protocol by Pavlov *et al.* [21]. It has been informally argued to preserve rate privacy. Our analysis shows that it ensured *Rate Privacy*, *Rate Anonymity*, and *User Eligibility*. It fails to satisfy *Receipt-Freeness*, *Coercion-Resistance*, and *User Eligibility Verification*, and presents some weakness concerning *Rate Integrity*, and *Reputation Score Verification* but satisfies them with some assumptions.

As a future work we intend to analyze more e-reputation protocols. Several e-reputation protocols are highly depends on algebraic properties such as arithmetic operations and homomorphic encryptions, *e.g.* [3,6,19,21]. Developing automatic tools that can deal with these properties is still a real challenge for the community. However, researches goes some way in the direction of finding solutions for such a problem, for instance the result obtained by Küsters *et al.* [20] allows us to analyze protocols with XOR operator using ProVerif. Note that, this result could help us in analyzing protocols with arithmetic operations, which is the case of many e-reputation protocols, if used to model summation and subtraction.

Other interesting research works include the study of the relation between our security properties as well as the definition of novel properties such as correctness of the reputation score, prevent double rating, accountability, reliability (*e.g.,* to prevent Sybil attacks), and addressing false unfair rates.

## References

1. M. Abadi and C. Fournet. Mobile values, new names, and secure communication. In C. Hankin and D. Schmidt, editors, *POPL*, pages 104–115. ACM, 2001.
2. E. Anceaume, G. Guette, P. Lajoie-Mazenc, N. Prigent, and V. V. T. Tong. A privacy preserving distributed reputation mechanism. In *ICC*, pages 1951–1956. IEEE, 2013.
3. E. Anceaume, G. Guette, P. Lajoie-Mazenc, T. Sirvent, and V. V. T. Tong. Extending signatures of reputation. In M. Hansen, J.-H. Hoepman, R. E. Leenes, and D. Whitehouse, editors, *Privacy and Identity Management*, volume 421 of *IFIP Advances in Information and Communication Technology*, pages 165–176. Springer, 2013.
4. E. Androulaki, S. G. Choi, S. M. Bellovin, and T. Malkin. Reputation systems for anonymous networks. In N. Borisov and I. Goldberg, editors, *Privacy Enhancing Technologies*, volume 5134 of *Lecture Notes in Computer Science*, pages 202–218. Springer, 2008.
5. M. Backes, C. Hritcu, and M. Maffei. Automated verification of remote electronic voting protocols in the applied pi-calculus. In *CSF*, pages 195–209, 2008.
6. J. Bethencourt, E. Shi, and D. Song. Signatures of reputation. In R. Sion, editor, *Financial Cryptography*, volume 6052 of *Lecture Notes in Computer Science*, pages 400–407. Springer, 2010.
7. B. Blanchet. An efficient cryptographic protocol verifier based on prolog rules. In *14th IEEE Computer Security Foundations Workshop (CSFW-14 2001), 11-13 June 2001, Cape Breton, Nova Scotia, Canada*, pages 82–96. IEEE Computer Society, 2001.
8. E. Carrara and G. Hogben. Reputation-based systems: a security analysis, 2007.
9. S. Delaune, S. Kremer, and M. Ryan. Verifying privacy-type properties of electronic voting protocols. *Journal of Computer Security*, 17(4):435–487, 2009.
10. C. Dellarocas. Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior. In *EC*, pages 150–157, 2000.
11. D. Dolev and A. C. Yao. On the security of public key protocols. *Information Theory, IEEE Transactions on*, 29(2):198–208, 1983.
12. N. Dong, H. L. Jonker, and J. Pang. Analysis of a receipt-free auction protocol in the applied pi calculus. In *Proc. 7th Workshop on Formal Aspects in Security and Trust (FAST'10)*, volume 6561 of *LNCS*, pages 223–238. Springer, 2010.
13. J. R. Douceur. The sybil attack. In *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, IPTPS '01, pages 251–260, London, UK, UK, 2002. Springer-Verlag.
14. J. Dreier, R. Giustolisi, A. Kassem, P. Lafourcade, G. Lenzini, and P. Y. A. Ryan. Formal analysis of electronic exams. In P. Samarati, editor, *SECRYPT 2014 - Proceedings of the 11th International Conference on Security and Cryptography, Vienna, Austria, 28-30 August, 2014*. SciTePress, 2014.
15. J. Dreier, H. Jonker, and P. Lafourcade. Defining verifiability in e-auction protocols. In K. Chen, Q. Xie, W. Qiu, N. Li, and W.-G. Tzeng, editors, *ASIACCS*, pages 547–552. ACM, 2013.
16. J. Dreier, P. Lafourcade, and Y. Lakhnech. Vote-independence: A powerful privacy notion for voting protocols. In *FPS*, pages 164–180, 2011.
17. J. Dreier, P. Lafourcade, and Y. Lakhnech. A formal taxonomy of privacy in voting protocols. In *ICC*, pages 6710–6715, 2012.

18. J. Dreier, P. Lafourcade, and Y. Lakhnech. Formal verification of e-auction protocols. In *POST*, pages 247–266, 2013.
19. O. Hasan, L. Brunie, E. Bertino, and N. Shang. A decentralized privacy preserving reputation protocol for the malicious adversarial model. *IEEE Transactions on Information Forensics and Security*, 8(6):949–962, 2013.
20. R. Küsters and T. Truderung. Reducing protocol analysis with xor to the xor-free case in the horn theory based approach. *J. Autom. Reasoning*, 46(3-4):325–352, 2011.
21. E. Pavlov, J. S. Rosenschein, and Z. Topol. Supporting privacy in decentralized additive reputation systems. In C. D. Jensen, S. Poslad, and T. Dimitrakos, editors, *Trust Management, Second International Conference, iTrust 2004, Oxford, UK, March 29 - April 1, 2004, Proceedings*, volume 2995 of *Lecture Notes in Computer Science*, pages 108–119. Springer, 2004.
22. P. Resnick and R. Zeckhauser. Trust among strangers in internet transactions: Empirical analysis of ebay's reputation system. In M. R. B. (ed.), editor, *The Economics of the Internet and E-commerce (Advances in Applied Microeconomics)*, volume 11, pages 127–157. Emerald Group Publishing Limited, 2002.
23. M. Ryan and B. Smyth. Applied pi calculus. In *Formal Models and Techniques for Analyzing Security Protocols*, chapter 6. IOS Press, 2011.
24. S. Steinbrecher. Design options for privacy-respecting reputation systems within centralised internet communities. In S. Fischer-Hübner, K. Rannenberg, L. Yngström, and S. Lindskog, editors, *SEC*, volume 201 of *IFIP*, pages 123–134. Springer, 2006.
25. S. Steinbrecher. Enhancing multilateral security in and by reputation systems. In V. Matyás, S. Fischer-Hübner, D. Cvrcek, and P. Svenda, editors, *FIDIS*, volume 298 of *IFIP Advances in Information and Communication Technology*, pages 135–150. Springer, 2008.