A robust reputation scheme for decentralized groups management systems

Frédéric Cuppens, Nora Cuppens-Boulahia, Julien Thomas

HAL Id: hal-00426476

https://hal.science/hal-00426476

Submitted on 26 Oct 2009

A Robust Reputation Scheme for Decentralized Group Management Systems

Julien A. Thomas, frédéric Cuppens, Nora Cuppens-Boulahia

Télécom Bretagne ; LUSSI Department

Université Européenne de Bretagne

Rennes, France

October 26, 2009

Abstract

In the literature, reputation systems are used to evaluate other entities behavior and have many applications such as, for instance, the detection of malicious entities. The associated models are based on mathematic formulae, in order to formaly define elements such as the reputation evaluation and evolution and the reputation propagation between peers. Current proposals describe the behaviors of their models by examples, with few (if not no) formal analyses. In this article, we state the basic security properties such systems require and we show that current systems may not satisfy them on specific scenarios, which can be used by malicious entities to take advantage of the system. We also present a new reputation scheme, designed to satisfy these properties, and we compare it to existing research works.

Introduction

In the literature, reputation systems are used to evaluate behaviors of subjects, processes or systems and for instance detect malicious entities. The associated models [1, 2, 3], based on mathematic formulae, generally take into account two actions: the operations to perform when an incorrect behavior is detected and the operations to perform when no malicious behavior is detected during a defined interval of time. These models rely on two main variables, α and β, which respectively refer to the reputation increase and decrease rates, in case of correct (respectively incorrect) behaviors.

The global process can be summarized by the following formulae:

- when no malicious behavior is detected for a node n_i, between two reputation checkpoints, the current node n_c increases its reputation of α, which means $rep_{n_c}(n_i) = rep_{n_c}(n_i) + \alpha$.

- When a node n_i has a bad behavior, its reputation decreases of β. If several malicious behaviors are detected between two reputation checkpoints, some proposals consider them as a single bad behavior $(rep_{n_c}(n_i) = rep_{n_c}(n_i) - \beta)$, while others consider them as several ones $(rep_{n_c}(n_i) = rep_{n_c}(n_i) - nb_{detection} \cdot \beta)$.

An example of application of the reputation systems is the management of groups [4, 5], for instance in ad hoc networks[6]. In this context, groups are used to join together nodes which can then share information and communicate. Inside these groups, which are sometimes considered as communities inside undefined and potentially malicous environments, the notion of trust is important as it can be used to detect malicious nodes. Stating security properties such as *the collusion of malicious nodes must not engender an eviction of a correct node* and asserting that they will be respected is thus important. As the design of the reputation system and the values of its parameters, such as α and β, are linked to the assertion of the security properties, the system must be defined by taking these properties into account. In current research works, formal analyses of the system parameters and assertions of such security properties are not performed.

In this article, we thus propose a formal method to evaluate the system parameters, in order to define a robust reputation scheme. In section 1, we first present the notions bounded to the reputation systems and we analyze the existing approaches. We then introduce in section 2 our reputation and recommendation system, with the security properties it must satisfy. In section 3, we present our formal analyses and specify the values of our system parameters that satisfy these properties. We then present our simulations, performed on NS-2 [7], and we compare our results with existing research works. The last section concludes the article.

1   Limits of Existing Approaches

Many studies [1, 2, 3, 8, 9] have proposed reputation systems which rely on two basic systems: the reputation and recommendation based systems and the referees based reputation systems. In this section, we thus present these systems and we analyze existing proposals that rely on them in order to show their limits.

1.1   Reputation and recommendation based systems

In the recommendation and reputation system proposed by Jinshan Liu and Valérie Issarny [1], several parameters (which are summed up in the table 1) are associated with each node to evaluate other nodes' quality. Among these parameters, $SExp$ is the reputation derived from direct interactions between the current node and the analyzed one and $SRep$ is a node reputation derived

from personnal evaluations and third nodes information. Moreover, each node has information about other nodes recommendation quality ($RRep$). $RRep$ is used as a weighting coefficient in the reputation evaluations. Finally, Rec is the reputation declared by a node about a peer and is the only value shared in the network. For a correct node, $Rec_a(o) = SRep_a(o)$.

$SRep_a(o)^t$	node o's reputation, declared by a, at time t
$RRep_a(o)^t$	o's recommendation, about a, at time t
$SExp_a(o)^t$	Immediate experience of a about o
$Rec_a(o)^t$	Recommendation made by a about o, at time t.
ρ_e, ρ_c	weighting coefficient of the reputation and recommendation functions

Table 1: Recommendation and Reputation System Parameters

An important aspect of this study is the distinction between recommendation and reputation: when a node provides a correct service, it can always be used, even if its recommendations are not correct and thus can be ignored.

Reputation evolution: For a node n_c, a node's reputation is based on three parameters: its old reputation, its new reputation according to n_c (which are both represented by $SExp_a(o)^t$) and the other nodes declared reputations $RRep$, where all these parameters are weighted by credibility and freshness coefficients. The reputation of a node o, according to a node a, is defined as follows:

$$SRep_a(o)^t = \rho_e \cdot SExp_a(o)^t + (1 - \rho_e) \cdot \frac{\sum_p (RRep_a(p) \cdot Rec_p(o))}{\sum_p RRep_a(p)}$$

Recommendation evolution: For a node a, the recommendation quality of a node p relies on the differences between the recommendation made by p (Rec_p) and its personnal evaluation ($SExp_a$) for each node $o \in N$. We thus have the basic formula: $diff_1(o) = |Rec_p(o) - SExp_a(o)|$. However, the differences between the values of two nodes can be due to analyses of different data (i.e. different contexts). In order to solve this problem, they use the notion of tolerance threshold δ_a. We then have a difference evaluation $diff = \frac{1 - diff_1}{\delta_a}$.

The recommendation evolution mechanism satisfies the following principle: the recommendation of p at time t relies on the precedent recommendation at time t' and the evaluation differences in this interval $\Delta t = t - t'$:
$RRep_a(p)^t = RRep_a(p)^{t'} \cdot \rho_c^{(t-t')} + diff \cdot (1 - \rho_c^{(t-t')})$.

1.2 Referees based reputation systems

In the research work by Conrad and al. [2], the notion of reputation is studied in order to first mimic the human trust formation and secondly to have a

lightweight approach. They use the notions of subjective trust and distrust to apply their reputation system to e-services and on-line transactions, as the results are quite binary: either the result is correct, or not.

As for many studies, the reputation analysis is based on two principal components: the node which performs the evaluation and the others. The reputation function they suggest is $reputation(c) = experience(c) \cdot p + (1 - p) \cdot hearsay(c)$ where p is the value to assign to our own credibility ($p = selfConfidence(c)$).

The notion of self-experience is based on two parameters: prior experiences and immediate experiences. No weighting is made between these two parameters and we thus have $experience(c) = \frac{immediateExperience(c)+experience(c)}{2}$. Another interesting aspect in this study is the way the hearsay parameter is evaluated: contrary to the previous study, the nodes do not take into account the information from all the nodes of the network. We have a notion of referees R that are used to analyze a service reputation: $hearsay(c) = \frac{\sum_{r \in R} reputation_r(c)}{|R|}$. The choice of a correct value of $|R|$ is important: if we have a too small value, few analyses will be used and the result may not be representative while with a too big value, the reputation system becomes too slow. By performing simulations, they chose $|R| = 10$ and $selfConfidence(c) = 30\%$

1.3 Analysis of existing proposals

The differences between the reputation and the recommendation is important. A node can have a quite bad behavior in the group (due to energy problem, for instance), but always a correct recommendation. In the opposite, an attack would consist in acting well, in order to avoid attack detection mechanisms, and lying about the reputation of other nodes, in order for example to obtain privileges.

In Jinshan Liu and Valérie Issarny study and in others which are similar, such as [8, 9], the reputation and recommendation systems have some flaws: calculi are based on all the nodes of the networks. The first and most obvious issue is the scalability problem. However, a more important problem happens when the detection of malicious behaviors can be performed only in local area: when the group size increases, no significant reputation decrease may occur. Consider the following example:

- the detection of malicious behaviors can be performed only on direct neighbors, which is often the case for the lowest levels of the ISO model

- we have N nodes, and we consider that each node has k neighbors

- we assume that each node gives correct recommendations

In the figure 1a, we can see that the reputation mechanism suffers from scalability problems, when the number of nodes increases. In the referee
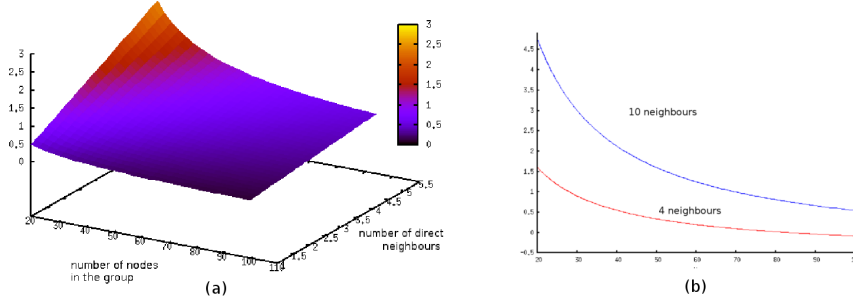
Figure 1: Reputation evolution in Jinshan Liu and Valérie Issarny approach

based approach [2], the authors suggest to take into account only the nodes that belong to the referees R. This prevents the case illustrated in the figure 1 from occurring. However, no distinction is made between the reputation and the recommendation. This study is thus relevant to detect incorrect behaviors for the services, using neighbors' cooperation, but cannot be used to detect malicious nodes inside the network.

We have seen that existing proposals fail when some conditions occur, such as when the number of nodes increases while the detection region remains the same. These problems arise because formal analyses of the system have not been performed. For instance, the notion of local region R presented in the referee-based system is not fully described: how can we evaluate R? What are the impact of the size of R on the reputation mechanism? As these questions have not been answered, flaws may be discovered in the future. We can conclude that the definition of a reputation system requires a formal analysis of the system and the environment, which is not performed in current proposals.

2 Formal Model for Reputation and Recommendation functions

As described in the previous section, reputation systems must be developed using a formal approach. In this section, we describe our reputation functions. An example of application of the reputation systems is the management of groups. We thus present the group decision principle and finally the security properties reputation systems must satisfy, based on these decisions. Formal analyses are presented in the next section.

2.1 Definitions of our reputation model

As presented in the section 1, we are able to have a scalable mechanism by using local region. However, local regions engender local reputations. In our

case, as we want to be able to take the same decisions on the whole group (property 4 of the section 2.2.2), we must have global reputations. This thus prevents nodes from waiting the acknowledgement of their decisions.

As the notion of reputation is bound to the recommendation, the way the recommendation is evaluated is also not correct for group decisions. In fact, the recommendation we need is a group recommendation, and not a node-dependent recommendation, as presented in Jinshan Liu and Valérie Issarny research work. In this study, the recommendation is defined by

$$rec_k(i) = rec_{k-1}(i) \cdot \rho_{rec} + (1 - \rho_{rec}) \cdot \frac{\sum_{j=0}^{n} diff(rep_{k-1}(j,i), rep_{k-1}(j))}{n}$$

where $rep_k(i,j)$ is the reputation of i declared by j at the step k. In this formula, the function $diff$ is used to evaluate the differences between the recommendations made by the current node and the ones made by the node i.

In order to have a global recommendation, we must evaluate the difference of the node's evaluations with all the other nodes' evaluations. Our notion of group reputation is defined as the following:

$$group\_reputation_k(i) = \frac{\sum_{j \in R_i} rec_{k-1}(j) \cdot rep_k(i,j)}{\sum_{j \in R_i} rec_{k-1}(j)} \qquad (1)$$

Using this formula, we get the following initial group recommendation:

$$rec_k(i) = rec_{k-1}(i) \cdot \rho_{rec} +$$

$$(1 - \rho_{rec}) \cdot \frac{\sum_{j=0}^{n} diff(rep_k(j,i), group\_reputation_k(j))}{n} \qquad (2)$$

Note that the recommendation function is studied in section 3.3, as it does not affect the evaluation of our reputation function in the worst cases.

Finally, the reputation is similar to the one presented in [2]:

$$rep_k(i) = \frac{100 \cdot experiences + \sum_{j \in R_i \wedge j \neq myself} rec_{k-1}(j) * rep_{k-1}(i,j)}{100 + \sum_{j \in R \wedge j \neq myself} rec_{k-1}(j)} \qquad (3)$$

2.2 Group Decisions Principle

In a group management algorithm, we can find two groups of operations for group management protocols: group operations and group agreements. The first group describes all the basic decisions, such as «a request to add a node» while the second one describes all the group decisions, such as «the group adds a node». This distinction is important as the first operations can be decided by a single node while the second ones have to be decided by the whole group.

2.2.1 Group Operations

As described above, these operations are made by a single node: depending on several parameters, a node may want to authorize a new node to join a group, or may want to evict a node from the group.

Adding a node: A node n_i sends an adding message to the group if the local reputation of the node to add is higher than or equals to $threshold_{Add}$.

Removing a node: As for adding a node, a node sends an eviction message about the node n_m if the node n_m has a reputation lower than or equal to $threshold_{Evict}$.

2.2.2 Group Agreements

An important aspect of the group agreements is to have common decisions: if a node starts a removing or adding operation at the protocol group layer, all nodes in the network must do it too. In order to have stable group decisions, we define several functional properties. They rely on the variables τ_{add}, $\tau_{eviction}$ and $minimal\_recommendation$ which respectively refer to the minimal number of nodes to take an adding message into account, the minimal number of nodes to take an eviction message into account and the minimal recommendation to consider a node's message as trustworthy. Finally, the variable τ is linked to security of the systems : τ - 1 is the maximal number of malicious nodes the system supports. Thus, we have $\tau \leq \tau_{add}$ and $\tau \leq \tau_{eviction}$.

For the group decisions, there are mainly four functional properties:

Property 1: In order to start an adding operation, a node must have received τ_{add} adding messages from distinct nodes in the network.

Property 2: In order to start an eviction, a node must have received $\tau_{eviction}$ eviction messages from distinct nodes among the network.

Property 3: A node message should be taken into account only if the node recommendation is higher than or equal to $minimal\_recommendation$.

Property 4: Upon receiving a group management operation, each node of the group must take the same decision.

2.3 Basic Security Properties

In the previous section, we have presented the functional properties our reputation system must satisfy. However, in order to develop a robust system, we must also state the security properties our system must satisfy.

The first one deals with the impact of the reputation increase rate.
Security Property 1: the collusion of malicious nodes must not engender an eviction of a correct node

For the reputation decrease scenario, two security properties are defined.
Security Property 2: A collusion of malicious nodes must not prevent a malicious node from having a decrease of its reputation.
Security Property 3: The group must be able to evict a malicious node, according to the functional properties, when its reputation exceeds a defined threshold.

Finally, in order to prevent malicious nodes from interfering with correct information about a node, their recommendation must decrease. This is expressed by the forth security property:
Security Property 4: a node recommendation must decrease if it acts maliciously.

3 Theoretical quantification of the model's parameters

In the previous section, we described our reputation and recommendation functions. In this section, we analyse the parameters of these functions and the impact of their values on the reputation system and the assessment of the security properties. In subsections 3.1 and 3.2, we introduce the global ideas about the reputation functions evaluation and our solution, which solves three principal problems: what is the value of the reputation increase rate if a node acts well? what is the value of the reputation decrease rate if a malicious node is detected? How can we define the local region R of a node? Finally, the evaluation of the recommendation function is given in section 3.3.

The evaluation of the different parameters is made by first formulating the worst cases that can occur. We then specify values that satisfy our security properties. Due to space limitation, complete demonstrations of the mathematical equations are not given in this paper but one can refer to [10].

3.1 Reputation increase assessment

3.1.1 Worst Case 1: incorrect eviction

The usual worst case is related to the eviction by malicious nodes of a node acting well. This can be represented by the following scenario: τ - 1 malicious nodes declare a reputation of 0 for this node while others increase its reputation by α.

According to the Group Agreement Property 2, the eviction of a node occurres if $\tau_{eviction}$ nodes send an eviction message. As $\tau_{eviction} \geq \tau$, this means that at least one «correct» node has to send an eviction message. Thus, to satisfy the Security Property 1, we must ensure that no correct node sends an eviction message. This can be ensured by the following requirements:

- the reputation does not go under the eviction threshold $Evic_{threshold}$ ($Req1$)

- the reputation is still able to increase ($Req2$)

To satisfy the first requirement, we must assure that there is no $i \in N$ such that $rep_i < Evic_{threshold}$. At the nth round, the reputation of the attacked node is given by (rep_0 is the initial reputation): $rep_n = rep_0 \cdot a^n + b \cdot \sum_{i=0}^{n-1} a^i$ where $a = \frac{|R|-\tau+1}{|R|}$ and $b = \alpha \cdot \frac{|R|-\tau+1}{|R|}$. Based on this formula and considering different eviction thresholds $Evic_{threshold}$, the table 2 illustrates the different values of α_{min} that satisfy $Req1$.

$Evic_{threshold}$	α_{min}	$Evic_{threshold}$	α_{min}	$Evic_{threshold}$	α_{min}	$Evic_{threshold}$	α_{min}
10	4	30	10	20	7	40	14

Table 2: Minimal value of α depending on $Evic_{threshold}$, $|R| = 4 \cdot \tau$

For the second requirement (the reputation is still able to increase), we can analyze the impacts of the reputation system parameters with several scenarios. We considered the following ones, where V_0 is the intial value of the reputation:

- $\{\alpha = 4,\ V_0 = 50,\ |R| = 2 \cdot \tau\}$ (figure 2a)

- $\{\alpha = 4,\ V_0 = 50,\ |R| = 4 \cdot \tau\ \}$ (figure 2b)

- $\{\tau = 20,\ V_0 = 50,\ |R| = 4 \cdot \tau\ \}$ (figure 3)

We can see that as τ is proportional to $|R|$, its value does not interact with the reputation increase rate. However, the way $|R|$ is evaluated does interact with the reputation increase rate. For instance, with $|R| = 4 \cdot \tau$, we manage to get a maximal reputation (i.e. 100) faster than with $|R| = 2 \cdot \tau$. The choice of $|R| = 4 \cdot \tau$ is due to several reasons. First, the increase rate is
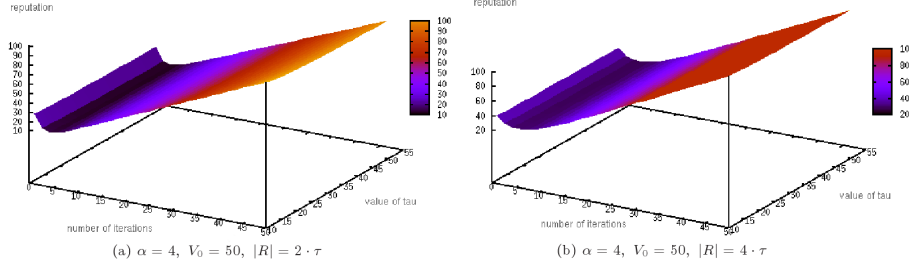
(a) $\alpha = 4$, $V_0 = 50$, $|R| = 2 \cdot \tau$ (b) $\alpha = 4$, $V_0 = 50$, $|R| = 4 \cdot \tau$

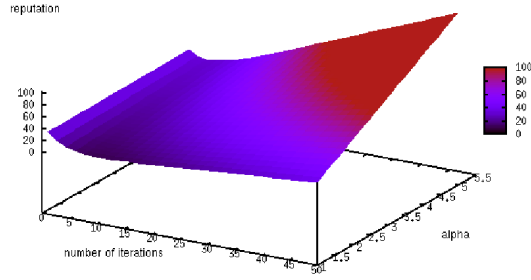Figure 2: Reputation increase - minimal rates

Figure 3: Reputation increase according to α's value, with $|R| = 4 \cdot \tau$

more important than with $|R| = 2 \cdot \tau$, which means that correct nodes will reach the maximal (and thus the best) reputation faster. Secondly, if one decide to take R such that $|R| = 6 \cdot \tau$ or $|R| = 2^\tau$, we would have better results but the size of $|R|$ would increase very quickly, which means that τ_{max} would be far less important and that nodes would have to keep a watch on more nodes. Obviously, as for R, several values for α can be taken into account. We decide to consider $\alpha = 4$, as the increase rate is correct (and $4 > \alpha_{min}$ for $|R| = 4 \cdot \tau$).

3.1.2   Worst Case 2: incorrect increase rate

Another problem occurred when malicious nodes cooperate in order to quickly increase a node's reputation: all of them decide to give a value of 100 to the reputation. This is represented by the formula $rep_k = \frac{100 \cdot (\tau - 1) + (rep_{k-1} + \alpha) \cdot (|R| - \tau + 1)}{|R|}$. In this case, we must choose a value of α which leads to a correct reputation increase. The formula can be represented by $rep_k = rep_0 \cdot a^n + b \cdot \sum_{i=0}^{n-1} a^i$, where $a = \frac{|R| - \tau + 1}{|R|}$ and $b = \frac{100 \cdot (\tau - 1) + \alpha \cdot (|R| - \tau + 1)}{|R|}$

As we can see in the figure 4, the reputation of the malicious node evolves very quickly, no matter the value of τ: with $|R| = 4 \cdot \tau$, five iterations are needed to get the maximal reputation, starting from a value of 50 while it is of three for $|R| = 2 \cdot \tau$. A solution to this problem is to find a way to decrease in all the cases the reputation of the malicious nodes.
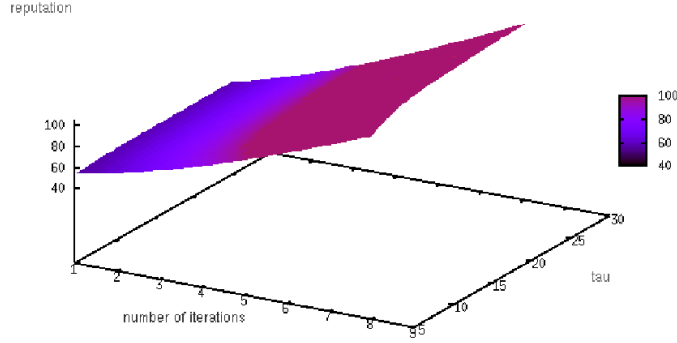
10

Figure 4: Reputation increase - maximal increase rate with $\alpha = 4$

3.1.3 Case 3: Common case

Finally, the common case is when each node increases the reputation of α. We must choose parameters values such that the increase rate is not too fast, in order to prevent malicious nodes from recovering a good reputation too quickly. In this case, the evolution formula is $rep_k = \frac{(rep_{k-1}+\alpha)\cdot|R|}{|R|} = (rep_{k-1} + \alpha)$. So, the increase is equal to α. With a value of 4 for α, 13 iterations are needed to get a maximal reputation, starting from a reputation of 50.

According to the different possible cases, we can see that a value of 4 for α and a value of $4 \cdot \tau$ for $|R|$ are interesting.

3.2 Reputation decrease assessment

3.2.1 Standard reputation decrease

The worst case of the reputation decrease scenario is the following one: all the malicious nodes cooperate in order to prevent the decreases of a malicious node reputation. They send a reputation of 100 and others decrease the malicious node's reputation of β.

In this scenario, we must choose the size of R and β so that the reputation will still decrease. Moreover, we must choose a value of β that decreases in a significant way the malicious node reputation, in order to increase the time this node requires to recover the maximal reputation (also called the recovering time).

With β a constant, we have the following worst case:

$$rep_k = \frac{100\cdot(\tau-1)+(rep_{k-1}-\beta)\cdot(|R|-\tau+1)}{|R|} \text{ , where } rep_0 = 100$$

According to the section 3.1, we can analyze several values of β, which are described in the table 5 (with $\tau = 25, |R| = 4\cdot\tau$). The main idea is to choose a correct value of β that implies a long recovering time, which tends to decrease the number of bad behaviors. For instance, with $\beta = 25$, a malicious node

11

Figure 5: Influence of β on the reputation decrease

decrease rate	β	recovering time (nb of iterations)	decrease rate	β	recovering time (nb of iterations)
10	7	3	30	43	8
20	25	5	40	60	10

has to wait for five iterations before getting its maximal reputation back. If it acts maliciously during each reputation update intervals, its reputation will decrease and it would have a reputation of 50 after 5 iterations and a reputation of 30 after 15 iterations.

However, a drawback is that we may not be able to get a reputation of $threshold_{Evict}$ for malicious nodes, depending on τ and β. Moreover, if we use usual equations, a special case cannot be taken into account: a malicious node acts badly, waits for its reputation to increase and restarts to act badly. We need a group history to take this case, namely the Moral Hazard [11] (byzantine behavior), into account. Thus, the current reputation function does not satisfy the security property 3 and has to be modified.

3.2.2 Dynamic reputation decrease

In a study made by Ba & Pavlou [12], an analysis of ebay's reputation mechanism has been made. Based on ebay's reputation results, they modelled the ebay trust system with a correlation between positive rates (PR) and negative rates (NR). It is given by the following formula: $Trust = \beta_0 + \beta_1 \cdot Log(PR) + \beta_2 \cdot Log(NR)$.

In our situation, positive rates are implicit: a node increases the reputation of the other nodes at each check, if this node does not have a bad behavior. Our current equation takes negative rates into account with the variable β, in which $\beta = f(NR)$. Using NR, we obtain $\beta(NR) = \beta_0 + f(NR) \cdot \beta_1$. The principal scheme of f is that $f(0) = 0$ and $f(NR_{max}) = \frac{100}{\beta_1} - \beta_0$ (as $\beta(NR_{max}) = 100$). A common aspect of $\beta(NR)$ would be that its value is reduced by 2 at each bad behavior. Thus, with $\beta(NR) = \frac{100}{2^{NR_{max}}} \cdot 2^{NR}$, we have a reputation decrease that satisfies the Security Property 2 and 3.

3.3 Evaluation of the recommendation functions

In existing studies, the recommendation function is the following:

$$rec_k(i) = rec_{k-1}(i) \cdot \rho_{rec} + (1 - \rho_{rec}) \cdot \frac{\sum_{j=0}^{n} diff(rep_{k-1}(j,i), X\_reputation_k(j))}{n}$$

(4)

where $X\_reputation_k(j)$ can be $group\_reputation_k(j)$ or the node reputation, for node-oriented reputation mechanisms. As for the reputation mechanism, we can see that this function is linked to the size of the group. Thus,

a simple attack from the malicious nodes would be to target a single node. As shown with the simulation results (section 4.1), we got a stabilized state where the malicious nodes' recommendation are still high (94%) while the attacked node's reputation is low. In this case, the Security Property 1 is not satisfied.

By analyzing these drawbacks of the recommendation mechanism, we first propose the function 5, which is not group size-dependent. By using the *multiply* operation instead of the *sum* one, isolated lies are not hidden and the cumulation of lies amplify the recommendation decrease.

$$rec_k(i) = rec_{k-1}(i) \cdot \rho_{rec}+$$

$$(1 - \rho_{rec}) \cdot \frac{\Pi_{j=0}^{n} diff(rep_{k-1}(j,i), group\_reputation_k(j))}{n} \qquad (5)$$

This function is thus robust against the mentionned attack. However, if we consider intelligent malicious nodes, similar drawbacks remain: in this function, the decrease rate is directly associated to the difference between what the malicious node says and the group_reputation value. Thus, by sending reputation values that are lower than the group_reputation, but not so far, malicious nodes can still lie about others' reputation and the decrease of their recommendation will not be important. Moreover, advanced attacks such as the binary state {correct, malicious} can impact the reputation mechanism. So, though attacks need to be more sophisticated, the mechanism may still be affected by the collusion of malicious nodes.

In our case, we made a strong assumption which has not been taken into account yet: we decided to choose R such that $R = 4 \cdot \tau$, with $\tau - 1$ being the maximal number of malicious nodes our system has to support. Thus, we assume that at least 75% of the nodes among R are not malicious. Moreover, the choice of R (and τ) is made such that each node among R is able to detect if a node is acting maliciously or not at the group layer. We are thus assured that most of the reputation values are correct. So, we can compare a node recommendation with the majority value, instead of the group value with the following function (in which $majority\_reputation(k)$ refers the majority value for the reputation about the node k):

$$rec_k(i) = rec_{k-1}(i) \cdot \rho_{rec} + (1 - \rho_{rec}) \cdot lieValue(i)$$

$$lieValue(i) = \begin{cases} 0 \text{ if } \exists j \in R/rep_k(j,in) \neq majority\_reputation_k(j) \\ 100 \text{ otherwise} \end{cases} \qquad (6)$$

As when a node lies its recommendation is set to 0, no matter how much incorrect information it provides, it is obvious that the Security Property 4, *a node recommendation must decrease if it acts maliciously*, is satisfied.

4 Simulations

4.1 Results and Comparisons

In the previous section, we have presented several recommendation evaluations. In order to compare them, we have used the NS-2 [7] simulator with the UM-OLSR [13] implementation of the OLSR Ad hoc routing protocol.

In the simulation, we have compared the different recommendation functions:

- $\sum 10$ and $\sum 25$ refer to the equation 4 with τ equal to 10% and 25%

- $\Pi10$ and $\Pi25$ refer to the equation 5 with τ equal to 10% and 25%

- $\Pi10b$ refers to the equation 5, with $\tau = 10\%$ and the attack which consists in alterning correct and malicious behaviors.

- $lying10$ and $lying25$ refer to the equation 6 with τ respectively equal to 10% and 25%

We compared the functions by using the following issues: when do the malicious nodes' recommendation (figure 6a) and the attacked node reputation (figure 6b) are stabilized? What are the stabilized recommendation (figure 6c) and reputation (figure 6d)? According to the security property 4,
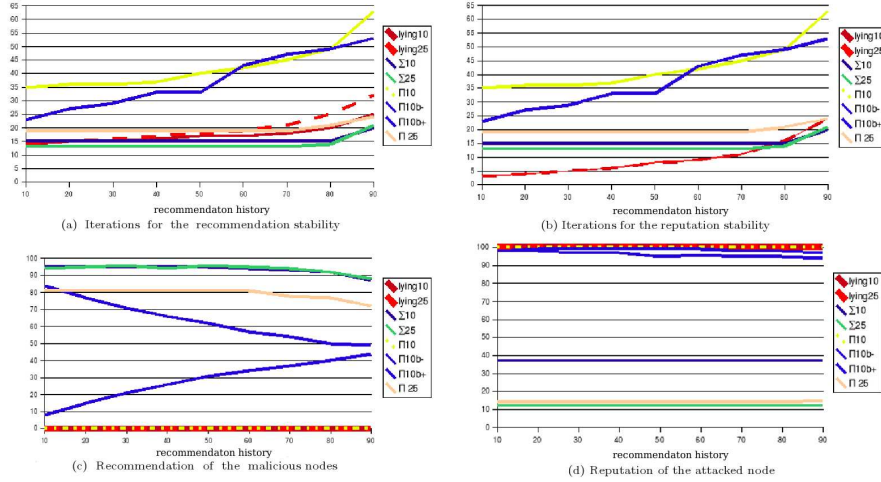

Figure 6: Comparison of recommendation evaluation functions

the recommendation value of the malicious nodes must be null. It is easy to see that this property is not assured by the standard recommendation evaluations. With the updated recommendation evaluation we suggest (equation 5), the recommendation evaluation and the reputation evaluation are correct in the case of basic malicious nodes, with $\tau = 10\%$. However, when we reach

14

the extreme case $\tau = 25\%$, the attacked node's reputation is impacted. With $\tau = 10\%$ and advanced malicious nodes, the attacked node's reputation is not really malicious and the malicious nodes' recommendations are neither considered as good nor bad. Finally, we can see that the lying method provides really good results, as malicious nodes recommendations are always null and the stabilized states are quickly reached. Thus, our proposals respect our security properties and the system stability is quickly reached, which is important in ad hoc networks.

4.2   Evaluation of the history parameter ρ_{rec}

The parameter ρ_{rec} defines the importance of the history and thus will have consequences and the system's evolution. In this case, the choice of ρ_{rec} is important. For instance, with $\rho_{rec} \sim 0$, an incorrect behavior will have immediate repercussion, whilte it is not the case with $\rho_{rec} \sim 1$.

The table 3 illustrates the importance of ρ_{rec} in several cases which are parts of the worst cases presented in section 3.1:

- $stability_1$ illustrates the recommendation decrease of a malicious node in the worst case 1 of the reputation increase

- $stability_2$ illustrates the recommendation increase rate in the common case

- $stability_3$ illustrates the recommendation decrease of correct nodes in the reputation decrease case, starting with a reputation of 100

- $stability_4$ illustrates the recommendation decrease of malicious nodes in the reputation decrease case, starting with a reputation of 100

ρ_{rec}	$stability_1$ (%)		$stability_2$	$stability_3$			$stability_4$		
	Π	lying	(%)	$\beta = 20$	$\beta = 40$	lying	$\beta = 20$	$\beta = 40$	lying
0	14	100	100	5	10	0	15	30	100
0.1	12.6	90	90	4.5	9	0	13.5	27	90
0.2	11.2	80	80	4	5	0	12	24	80
0.5	7	50	50	2.5	5	0	7.5	21	50
best	max	max	max	min	min	min	max	max	max

Table 3: influence of ρ_{rec} on the reputation mechanism, $\tau = 25\%$

With $\rho_{rec} \sim 1$, the recommendations of the malicious nodes and the attacked nodes decrease very slowly. This is the opposite in the case of no recommendation history. By choosing $\rho_{rec} = 0.2$, we limit the recommendation decreases of the correct nodes, and we also reduce the increase rate in common states, which prevents malicious nodes from alterning correct and malicious behaviors.

Conclusion

In this article, we have shown that designing a reputation and recommendation mechanism at the group layer requires to develop a reputation shared between the nodes and not a local reputation, as proposed in existing studies. This kind of system relies on many parameters, such as update rates, synchronization intervals and thresholds, which are linked together in complex ways. We have defined basic security properties (such as *the collusion of malicious nodes must not engender an eviction of a correct node*) whose enforcement requires a correct setting of the system parameters. We have analyzed the system parameters and determined values that satisfy our security properties.

Moreover, as the recommendation aspect is as important as the reputation aspect, we have studied the existing recommendation evaluation. We have shown that the basic principle *a node recommendation must decrease if it acts maliciously* is not assured in the worst cases, which may engender incorrect stabilized states. We have then proposed two modifications of the evaluation scheme: a recommendation function that improves the existing function and a new one, designed under hypotheses about the group environment, whose results are even better.

Our reputation system may be used in different contexts, such as the group management in ad hoc networks, as a reinforcement of existing proposals such as [6], and the reinforcement of routing protocol with misbehaviors detection [14].

References

[1] Liu, J., Issarny, V.: Enhanced reputation mechanism for mobile ad hoc networks. In: iTrust. (2004) 48–62

[2] Conrad, M., French, T., Huang, W., Maple, C.: A lightweight model of trust propagation in a multi-client network environment: To what extent does experience matter? In: ARES '06: Proceedings of the First International Conference on Availability, Reliability and Security (ARES'06), Washington, DC, USA, IEEE Computer Society (2006) 482–487

[3] Guha, R., Kumar, R., Raghavan, P., Tomkins, A.: Propagation of trust and distrust. In: WWW '04: Proceedings of the 13th international conference on World Wide Web, New York, ACM Press (2004) 403–412

[4] Wong, C.K., Gouda, M.G., Lam, S.S.: Secure group communications using key graphs. In: Proceedings of the ACM SIGCOMM '98 conference on Applications, technologies, architectures, and protocols for computer communication. 68–79

[5] Sherman, A.T., McGrew, D.A.: Key establishment in large dynamic groups using one-way function trees. IEEE Transactions on Software Engineering **29**(5) (2003) 444–458

[6] Cuppens, F., Cuppens-Boulahia, N., Thomas, J.A.: STGDH: An enhanced group management protocol. In: Proceedings of the CRISIS Conference, Marocco, Marrakech (July 2007)

[7] Fall, K., Varadhan, K.: The ns Manual. http://www.isi.edu/nsnam/ns/doc/

[8] Anantyalee, T., Wu, J.: Reputation-based system for encouraging the cooperation of nodes in mobile ad hoc networks. In: IEEE ICC, 2007. (24-28 June 2007)

[9] Carbone, M., Nielsen, M., Sassone, V.: A formal model for trust in dynamic networks. In Cerone, A., Lindsay, P., eds.: Proceedings of Int. Conf. on Software Engineering and Formal Methods, SEFM 2003, IEEE Computer Society (2003) 54–61

[10] Cuppens, F., Cuppens-Boulahia, N., Thomas, J.A.: Malevolence detection and reactions in ad hoc networks. Technical report (June 2007)

[11] Dembe, A.E., Boden, L.I.: The story of the moral. In: New Solutions. (2002) 257–279

[12] Ba, S., Pavlou, P.A.: Evidence of the effect of trust building technology in electronic markets: Price premiums and buyer behavior. MIS Quarterly **26**(3) (2002)

[13] Ros, F.J., Ruiz, P.M.: Implementing a New Manet Unicast Routing Protocol in NS2. Technical report, Dept. of Information and Communications Engineering University of Murcia (December 2004)

[14] Cuppens, F., Cuppens-Boulahia, N., Ramard, T., Thomas, J.A.: Misbehaviors detection to ensure availability in olsr. In: MSN, Mobile Sensor Networks. Volume 4864 of Lecture Notes in Computer Science., Springer (2007) 799–813

17