Trust Management for Selecting Trustworthy Access Points

Xavier Titi[1], Carlos Ballester Lafuente[1], Jean-Marc Seigneur[1]

**[1] University of Geneva, 7 route de Drize, Carouge,
CH1227, Switzerland**

Abstract

Both free and low-priced mobile wireless networks are expanding and its users are more numerous every day. This success is particularly due to the mobility offered to users. These networks have promoted and widespread the success and availability of mobile devices and their access technologies such as Wi-Fi, WiMAX and Bluetooth. Nevertheless, despite of their wide success, the problem is that those technologies do not match the everyday privacy and security requirements expected by the users. In this paper we focus on the Wi-Fi technology and we propose a reputation system that assesses the trust level of a Wi-Fi Access Point (AP). We obtain this trust level according to users past experiences. We have validated our proposed solution through simulation using the dynamic simulation tool AnyLogic and comparing the results of our solution to those of two previous well-known trust metrics: EigenTrust and Salem metrics.

Keywords: *Wi-Fi, Trust Management, Reputation management.*

1. Introduction

The number of APs in the world has increased significantly: they have widely spread in many locations like airports, cafes, businesses and university campuses. This fact coupled with the inherent vulnerabilities of the deployed technologies, has provoked a security breach. In addition to typical network threats, wireless networks present several challenges and specific attack types. This is due to the wide open air nature of the channel, allowing more attacks, bandwidth limitations and constant topology changes because of node mobility. Furthermore, a lot of security breaches have been registered, such as Service Set Identifier (SSID) spoofing using soft AP. To steal credit card numbers and other personal information, thieves are using a soft AP to masquerade as a legitimate wireless AP. For instance, it has been reported [1] that fake Wi-Fi networks have been set up in many airports in order to capture users' sensitive information as they surf the Web during their connection to those networks. It is important to know whether the Wi-Fi APs within range are trustworthy or not as in some locations it is not rare to find more than five potential Wi-Fi networks to connect to.

In our solution, we propose to evaluate the trustworthiness level of the AP according to previous experiences of the users. This can help users to detect which AP is the most trustworthy by taking into account the opinions of previous users as well as their friends' recommendations, in order to polish the selection of trustworthy available APs. The users will then use this trust value in order to connect to the best AP. With our trust and reputation system, the user will detect the bad APs and will isolate them by giving them a low trust value. In our simulations this reputation system, called *TrustedHotspot*, has significantly increased the chances to choose the most trustworthy wireless AP, even under a variety of bad conditions, i.e. with bad Wi-Fi networks in place and even with malicious users cooperating in an attempt to deliberately subvert the system by giving a wrong rating to an AP. We have validated our proposed solution under resource constraints through simulation, based on the dynamic simulation tool *AnyLogic*.
The rest of this paper is organized as follows: Section 2 presents the problem statement and related work. In Section 3, we present our solution and its assumptions. Section 4 presents how we have implemented our trust metrics to validate it and presents also a comparison with another trust metrics such as Salem metrics and EigenTrust metrics. Finally, Section 5 concludes the paper.

2. Problem Statement and Related Work

Section 2.1 states the problem that has motivated this paper and section 2.2 presents the related work in the same field.

2.1 Problem Statement

There is an increasing number of websites offering an assessment of the Wi-Fi networks available in a given location. This is due mainly to the great success of commercial and shared Wi-Fi access. For the user, it is important to know if the hotspot is trustworthy, and in order to do that, we have to provide the means to the users to evaluate the hotspot and at the same time our solution has to be robust against attacks. The goal of this work is to present an easy deployable solution that:

- Evaluates the level of the trustworthiness of the AP.
- Prevents the user from choosing malicious APs.
- Encourages the owner of the AP to act correctly.

2.2 Problem Statement

Salem et al. [5] proposes a reputation system that enables the user to choose the best hotspot and discourages the wireless Internet service providers (WISPs) from providing a bad quality of Service (QoS) to the mobile nodes. In this paper, they consider a mobile node (MN) that is affiliated with a home network and that wants to connect to the Internet via a hotspot managed by a wireless Internet service provider. The behavior of each WISP in their model is characterized by what they call a reputation record. This record represents an evaluation of the reputation of the WISP and it is generated and signed by a trusted central authority. The reputation mechanism is maintained by the same trusted central authority. When a WISP first enters the network, the trusted central authority provides it with an initial reputation record, that can afterwards increase (i.e., better reputation) or decrease, depending on the behavior of the WISPs. If the MN has two neighboring WISPs that propose equivalent offers, i.e., same QoS and price, the MN will choose to connect to the access point managed by the WISP that has the best reputation record. They use a micropayment scheme to make sure that the MNs will pay for the service they received. Our solution is related to this one, but the key difference is that it can work on free or paid networks. We focus our work mostly on the selection of the most trustworthy AP and on preventing the user from connecting to malicious APs.

In Nicholson et al. [6], the selection algorithm focuses on the AP's signal strength as an important metric. It presents an extensive field study conducted on three different neighborhoods in Chicago, which shows that choosing an AP based on signal strength makes the user to miss significant opportunities for Internet connectivity. They describe the design and implementation of Virgil, an automatic AP discovery and selection system which quickly associates to each AP found during a scan, and runs a battery of tests designed to discover the AP's use suitability by estimating the bandwidth and round-trip-time to a set of reference servers. Virgil also probes for blocked or redirected ports, to guide the selection in favor of preserving the application services currently in use. Their results show that Virgil finds a usable connection from 22% to 100% more often than when simply using a selection based on signal strength alone. Virgil improves both performance and accuracy for neighborhoods that the user commonly travels, by caching AP test results. Their work focuses on estimation of bandwidth and round-trip-time to assess the AP whereas our solution proposes to use the algorithm *TrustedHotspot* in order to prevent the user to connect to malicious hotspots.

The paper by Ormond et al. [7] further examines network selection decision in wireless heterogeneous networks based on a user-centric approach, which they say that allows a user to choose the network which meets their best requirements. Their network selection algorithm predicts the data rate on each interface available to the mobile node and makes a decision based on those predictions. Their approach is very interesting because they focus on the user requirements or preferences although they do not prevent the user from connecting to malicious hotspots as we do.

3. Our Solution and Its Assumptions

Section 3.1 presents the assumptions and gives an overview of our solution. In Section 3.2 we describe the trust and reputation model used and finally in Section 3.3 we present the formulas used by our solution.

3.1 Assumptions

We consider a user who has a mobile phone (MP) that integrates the Wi-Fi technology and who wants to connect to an AP as depicted in Figure 1. Our solution provides an Android application in order to rate the AP and sends this rating to our server. This application generates a public (PuK) / private (PrK) key pair in order to sign the message exchange. The MP sends to our server his public key in order to identify who is sending the rating.

In our solution, we use a trust mechanism to discourage the APs from providing a bad QoS. The trust management is used during the selection process of the best AP in accordance to the trust value of each of them. Thanks to our trust and reputation mechanism, the APs are encouraged to behave correctly: they cannot provide a bad QoS if they want to obtain a high trust value, and hence, a good reputation. By bad QoS we mean that is not good enough for the user to successfully use an application while connected to the AP. In this paper, we assume that:

- Our server is trusted by the other parties.
- The user is not able to create many identities in order to cheat, for example, using the Sybil Attacks [2].We assume that the MP has a SIM

card which unambiguously authenticates the identity of the user with a provider.

- The connection to our server is done through 3G, GPRS.
- The AP has an unlimited amount of energy and a uniform transmission range.

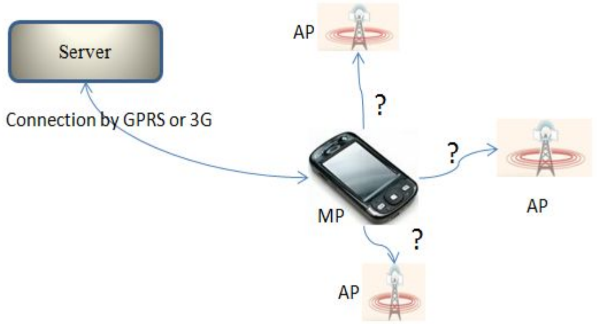


Figure 1. System Model

3.2 Trust and Reputation Model

The behavior of each AP in our model is characterized by what we call a trust value. This trust value represents the trust level of the AP, based on the previous experiences of the users with that AP, and it is signed and sent to our server. The trust value of an AP is represented in the range [0…1], being 0 untrustworthy and 1 very trustworthy. When an AP first enters the network the server provides it with an initial trust value equal to 0.5, which can afterwards increase or decrease, depending on the behavior of the AP.

The users have the possibility to ask to their friends for recommendations about a given AP**.** The user will be able to become friend with other users and they will store all their information about their friends and the APs used by them in our server. The connection to our server will be done by using GPRS or 3G. In Balasubramanian et al. [8] they have compared the energy consumption between 3G, GSM and Wi-Fi and their results showed that wireless is more energy efficient. So, if the user has the choice between 3G and Wi-Fi it will be more efficient to choose the last one. This is why 3G will only be used to retrieve the information which will help the user to choose the most trustworthy AP. The recommendations are useful when the users have not any information about an AP. We will explain how recommendations work in more detail in Section 3.3**.**

After using the AP, the user will be presented with five different rating possibilities:

$$\text{Rating}\begin{cases} 1 \; \mathit{means}\ \text{Very Low}\ \mathit{Quality} \\ 2 \; \mathit{means} \quad \mathit{Low} \quad \mathit{Quality} \\ 3 \; \mathit{means\ Normal\ Quality} \\ 4 \; \mathit{means} \quad\ \mathit{Good\ Quality} \\ 5\ \mathit{means\ Very\ Good\ Quality} \end{cases}$$

In order to aggregate the rating, it is necessary to normalize it in some manner. Otherwise, malicious users can assign an arbitrarily high rating to the AP. We define a normalized rating in the following way:

$$Rating_{normalized\,[0\ldots1]} = \frac{Rating}{5}$$

The ratings of the users are aggregated in our server. In order to make his choice when selecting an AP, the user will have information from his friends, from the aggregation of the ratings of all users and from his own previous experiences with that AP. By having information from many different sources, it will be easier for the user to choose the most appropriate and trustworthy AP.

We implement some of our functions on our server to avoid unnecessarily overloading the network with messages and to avoid consuming energy or using a lot of resources on the MP of the user.

Our solution proposes a trust/reputation management system in order to help users to choose the most trustworthy AP. Two thresholds are used: K1 and K2 with K2> K1. Using those thresholds, we have defined three cases:

- **First case**: the user will not connect to an AP if the trust value is lower than K1.
- **Second case**: the user will connect his MP to the AP when the trust value of it is between K1 and K2. In this case we assume that the user will trust the hotspot.
- **Third case:** the user will connect to an AP if K2 is lower than the trust value of it. In this case we assume that the user will fully trust the hotspot.

To apply this algorithm, our solution needs trust functions in order to compute the trust value.

3.3 Functions used by our Solution

As in any other reputation system, we must take into account the trustworthiness of the users as some of them can try to cheat by providing assessments that do not correspond to the real performance of the AP. To avoid this, each user has a trust value stored on the server side and the server will not validate ratings from users whose trust value is higher than 1. At the beginning all users will have their trust value at 0 and it will be incremented by 1 each time the user will cheat. The trust value of the user is completely different of the trust value of the AP. The trust value will be reset to 0 at the end of the week in order to prevent the user to cheat easily.

We have implemented five functions in order to compute the trust value. These functions are implemented on different elements of our trust model such as the MP, the AP and our server.

$$Trust_{Network_{by}User(AP)} = \sum_{i=0}^{n} \frac{Rating_{normalized[0...1]}}{n} \quad (1)$$

Function 1 is responsible to compute the trust value of an AP on the user side, in accordance to the feedback from the user where "n" is the number of ratings of the user on "AP". This function aggregates on the MP all the past ratings of the user and its result will be in the range [0…1].

$$Trust_{Network(AP)} = \sum_{i=0}^{m} \frac{Trust_{byUser(AP)(i)}}{m} \quad (2)$$

Function 2 is responsible to aggregate all the ratings sent by all the users who have used the AP where "m" is the number of users who have used "AP" and who have a trust value lower than 1. It computes the contextual trust value of the AP and it is implemented on the server side with its result on the range [0…1].

$$Friendshipfactor\ A \rightarrow B = \begin{cases} 1\ if\ A\ is\ a\ friend\ of\ B \\ \frac{2+nblink}{2nblink+1} \quad else \end{cases}$$
$$(3)$$

Function 3 is responsible to compute the *Friendshipfactor* between the users. We consider *nblink* as the level of friendship. As we do not know in which context two users have become friends, we need to add a weight which we call *Friendshipfactor*. The argument *nblink* defines the number of hops between two friends and it is likely to be inferior or equal to 6 according to the theory of Small World [3]. The result of this function is further used in the computation of the recommendation and it is implemented in our server.

$$Trust_{Rec(U \rightarrow AP)} = \sum_{1}^{f} \frac{Friendshipfactor * Trust(Ui \rightarrow AP)}{f} \qquad (4)$$

Function 4 is responsible to compute the recommendations coming from the user's friends and it takes into account the Friendshipfactor. We consider "*Ui*" all friends of the user "*U*" and "Friendshipfactor" the specific weight of that friendship. The argument "*f*" is the total amount of friends of the user "U" and "AP" represents the AP. This function is implemented using some of the functionality of the Facebook platform. When a user will ask for a recommendation, the Facebook platform will compute his friend list and will send it back to the server, where the function is stored and the recommendation will be computed.

$$Trust_{Value(AP)} = \frac{(1-\gamma)Trust_{Rec(U \rightarrow AP)} + (1-\text{ß}) * Trust_{Network_{by}User(AP)} + \lambda * Trust_{Network(AP)}}{(3-\gamma-\text{ß})}$$
$$(5)$$

$$With \qquad \lambda = \frac{1}{2-\Omega}$$

$$\begin{cases} if\ Trust_{Rec(U \rightarrow AP)} = 0;\ \gamma = 1 \\ \qquad else\ \gamma = 0.1 \end{cases}$$

$$\begin{cases} if\ Trust\_Network = 0;\ \text{ß} = 1 \\ \qquad else\ \text{ß} = 0 \end{cases}$$
$$\begin{cases} \Omega = 0\ if\ K1 < 0.5 \\ \quad else\ \Omega = 1 \end{cases}$$

Function 5 is responsible to compute the trust value of an AP when the user has several hotspots available. The user will use this function to compute the trust value of each AP available. The variable λ represents the weight of the trust value computed on the server side, the variable *ß* represents the weight of the trust value computed on the user side and the variable γ represents the weight of the recommendations coming from friends.

4. Implementation and Validation

We have implemented our functions and validated our solution with AnyLogic, which is a simulation tool that supports all the most common simulation methodologies: System Dynamics, Process-centric (a.k.a. Discrete Event), and Agent Based modeling. It is based on Real-time UML and Java object-oriented language.
In Section 4.1 we present the model set-up. In Section 4.2 we describe the validation methodology employed. In Section 4.3 we present the scenario and the results of the simulation with our trust metrics and finally in section 4.4 we compare the results of our solution to two well-known trust metrics: EigenTrust and Salem metrics.

4.1 Model Set-up

The basic element of an agent based model is the agent itself. By using an agent based model, we have created a new class than behaves as an AP. Each device is associated to a given agent matching its location. As the device is not static, we have modeled its mobility using X and Y random variables.
The movement and the status of our agents are controlled by a state-chart which represents the exact behavior of the device [Fig.2].

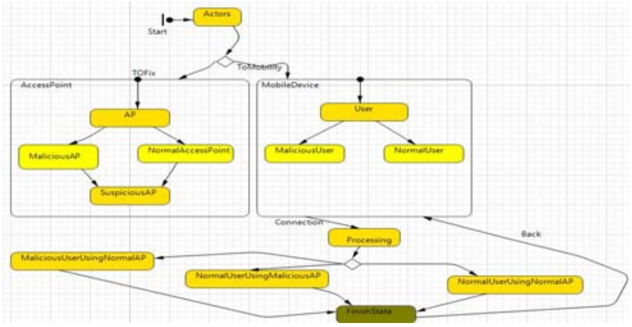


Figure 2. State-chart: Fix APs and Mobile Users

In Figure 2, each agent starts in an "Actors" state in the state-chart. A part of our agents were APs, so some of them switch to the state "AccessPoint" and the rest switch to the state "MobileDevice".

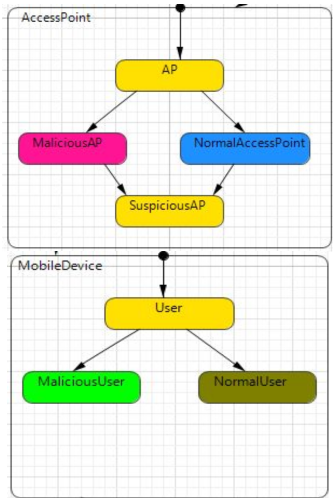


Figure 3. State-chart for Access Point and User

The two kinds of agents were placed in the map according to their GPS positions taken from Swiss hotspots reference[1]. The rest of our agents were considered as mobile nodes and each of them followed a random mobility model. The MobileDevice represents the user with his MP and it connects to an AP that is in range.

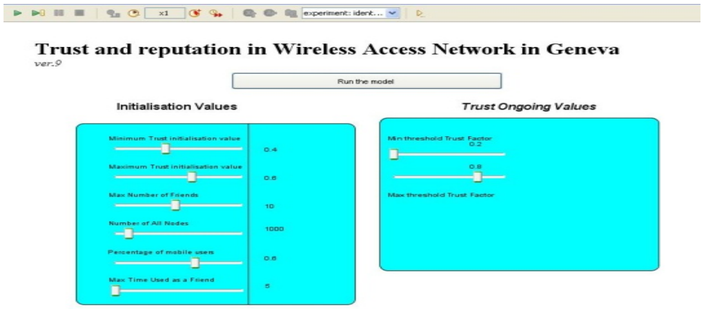
As can be seen in Figure 3, the "AccessPoint" agents can have three different states: *Malicious* which means that the AP is not good and its trust value is lower than 0.5, *Normal* which means that the AP is good and its trust value is higher than 0.5 and *SuspiciousAP* which means that the AP can change his status. The "MobileDevice" agents can have two different states: *MaliciousUser* which means that the User will cheat by giving wrong ratings meaning that the rating will not correspond to the real status of the AP and *NormalUser* which means that the User will not cheat and will give a fair rating.

4.2 Geneva APs Scenario

In our experiments, we validated our proposed solution and analyzed the extended performance under a range of various mobility scenarios. All nodes are moving over rectangular 8.69 km x 6.08 km topography. In our simulations, we considered that the APs were those taken from Geneva hotspots and the nodes were the mobile users with their MP. The coverage of the APs is limited to 100 meters and each mobile device was configured to have a maximum communication range equal to 100 meters. We deployed the APs in an incremental mode, from AP_1 to AP_n, in the exact position taken from the real GPS coordinates. Thus we estimated the impact of our solution for an existing network depending on the previous experiences of the users.

The movement pattern of the mobile clients was totally randomized, in order to comply with a real hotspot scenario. To achieve this, we used the Random WayPoint (RWP) mobility model [4] with a pause time equal to that of the time of network access and data transfer. We carried out our experiments considering thirty cases. In each studied case, we ran our simulation with different conditions.

Figure 4 shows the launching interface and Figure 5 shows the animation interface. The circles are APs and the square shapes are users whose colors correspond to their behavior. In the background of the animation, we placed the APs according to their GPS location on the map of Geneva.



Trust and reputation in Wireless Access Network in Geneva
ver.9

Initialisation Values

Minimum Trust initialisation value 0.4
Maximum Trust initialisation value 0.6
Max Number of Friends 10
Number of All Nodes 1000
Percentage of mobile users 0.6
Max Time Used as a Friend 5

Trust Ongoing Values

Min threshold Trust Factor 0.2
Max threshold Trust Factor 0.8

Figure 4. The launching interface

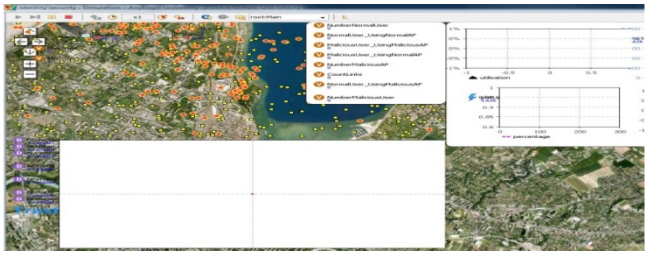


Figure 5. Animation interface

[1] *http://www.swiss-hotspots.ch/*

4.3 Scenario and Results with our Metrics

In this section we present the scenarios of our simulations. We use 230 users, 230 APs and K1=0.5, K2=0.8. We chose K1 is 0.5 because it prevents users to select malicious APs that have a trust value lower than 0.5. According to simulation tests the best value of K2 is 0.8. We use low number of users and APs because when we will first deploy our solution we will have few users and APs. In our future work, we will increase the number of users and APs in order to see the impact of high density of users and APs. We will have different kinds of simulation: one with normal and malicious APs; one with only malicious APs; one with normal, malicious and suspicious APs; one with malicious users.

- APN (AP Normal): Is an AP with trust value higher or equals to 0.5
- APM (AP Malicious): Is an AP with a trust value lower than 0.5.
- APS (AP Suspicious): Is an AP normal which can become malicious or APS is an AP malicious which can become normal.
- UN (User Normal): Is a user who rate correctly meaning if he used APN he will rate goodly and if he used APM he will rate badly.
- UM (User Malicious): Is a user who rate badly meaning if he used APN he will rate badly and if he used APM he will rate goodly.

Simulation 1

We have 230 users, 230 APs. In the 230 APs we have 207 APNs and 23 APMs, K1=0.5 and K2 =0.8. All APs at the beginning of each simulation have as trust value 0.5.

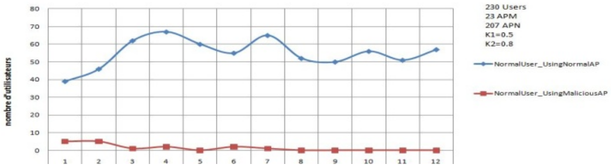


Figure 6. Result of Simulation 1

Figure 6 shows at the beginning that we have some users that use APMs. After the first round, meaning after the hotspot received for the first time the evaluation of the user, we can see that the number of users using APMs decreases and the number of users using normal APs increases. Thus, our solution helps users to choose the most trustworthy AP and prevents them to use APMs.

Simulation 2

We have 230 users, 230 APs and these APs are APMs, K1=0.5 and K2 =0.8. All APs at the beginning of each simulation have as trust value 0.5.

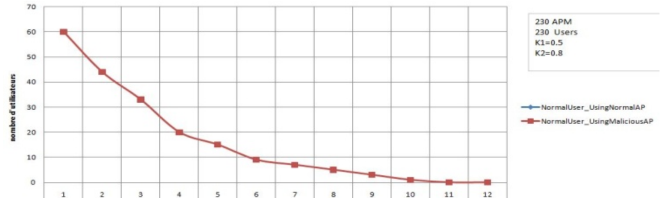


Figure 7. Result of Simulation 2

In the second simulation we have only APMs. The result shows that the number of users using APMs is decreasing every round. This happens because when the user connects to an APM, he rates it thus lowering the trust value of the APM under 0.5 which is the value of K1. When all APMs will be used by the users they will become untrustworthy. So after receiving the evaluation of users, the trust value of the APMs decreases and so it prevents future users to connect to those APM again.

Simulation 3

In this simulation we test the robustness of our solution against the attack of behavior change of AP. There are two cases of behavior change:

- When the trust value of AP is lower than 0.5 because of user evaluations, so it will start to have good behavior. In this case, K1 will prevent users to connect to this kind of AP because the trust value of this AP will be lower than K1.
- When the trust value of the AP is above 0.9 then it will start to have bad behavior.

We have 230 users, 230 APs with 19 APSs, 22 APMs, K1=0.5 and K2=0.8. All APs at the beginning of each simulation have as trust value 0.5

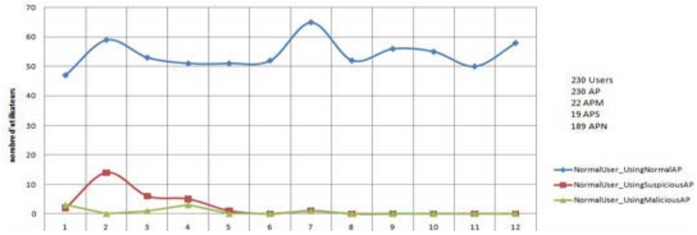


Figure 8. Result of Simulation 3

Figure 8 shows how our solution behaves when there are some APNs, APSs and APMs. After the first round, meaning after the hotspots received for the first time the evaluation of the users, we can see that the number of users using APMs decreases and that the number of users using APNs increases. Thus, our solution helps users to connect to the most trustworthy AP and prevents them to use APMs. After the first round we can also see that there are around 15 users that are using APSs, but in the next rounds we can check how the number of users connected to an APS is decreasing until reaching 0 showing that our solution can deal with this kind of behavior changing APs.

Simulation 4
We have 211 APNs, 184 UNs, 46 UMs, 19 APS,K1=0.5 and K2=0.8. All APs at the beginning of each simulation have as trust value 0.5 and all users have as trust value 0.

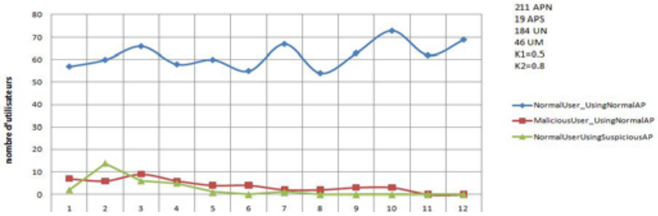


Figure 9. Result of Simulation 4

In this simulation we introduce UMs and APS at the same time. When an APS with a trust value lower than 0.5 even if it starts to act goodly no user will connect to this AP because of the threshold K1 that have as value 0.5.For the APS which have a trust value above 0.9, when they start to act badly and become malicious they receive bad rating from users, so their trust value decrease and like that the number of users using these APs decreases also. The UMs are characterized by rating in an unfairly way an APN and/or rating as good an APM.

As we can see in the first round, we have less than 10 malicious users using normal APs. The number decreases after three rounds until it reaches 0 UMs connected to APNs, because each time that a UM connects to an AP their trust value is incremented (+1). By doing that we can filter the rating coming from "trusted users" who have a trust value lower than 1. So our solution can deal with UMs that try to cheat by rating badly an APN and/or rating goodly an APM.

4.4 Results with Salem Metrics and EigenTrust Metrics

First we simulate the same scenario with Salem trust metrics in subsection 4.4.1. In subsection 4.4.2 we simulate the same scenario with EigenTrust trust metrics. We compare the results with all simulated trust metrics in subsection 4.4.3.

4.4.1 Salem

We compare our solution with the solution presented in Salem et al. [5] because it is the solution that tackles the most similar topic to ours, which is that of selecting the most trustworthy AP. So with Salem solution each AP is characterized by a triplet (AQW, RQW, PW) where AQW is the QoS advertised by W, RQW is the real QoS provided by W and PW is the price W is asking for. he considers that a WISP (wireless Internet service provider) W is honest if it advertises the real QoS it is offering (i.e., RQW = AQW), misbehaving if it advertises a QoS that is higher that the real QoS it is offering (i.e., RQW < AQW) and modest if it advertises a QoS that is lower than the real QoS it is offering (i.e., RQW > AQW). He initializes the reputation of the WISPs to maxRR = 100. At the end of each session, MN (Mobile Node) sends to TCA (Trusted Central Authority) its satisfaction level Sl = QoSEvalW where QoSEvalW = RQW/AQW

Each simulation lasts for 50000 seconds and the reputation updates are made every 2000 seconds. The new reputation RRW (t + 1) of W is computed as follows:

RRW (t + 1) = ß · RRW (t) + (1 − ß) ·feedbackW/nbSW

where RRW (t) is the current reputation of W, nbSW is the number of sessions established by W (and already closed) during the last 2000 seconds and feedback is the sum of all QoSEvalW received over all these sessions (the absence of feedback is considered as QoSEvalW = 0). ß represents the "weight of the past" and is set to 1/2 in their simulations.

- APN will have RQW = AQW
- APM will have RQW < AQW.
- APS will change between APN and APM.

Simulation 5
We have 230 users, 230 APs. In the 230 APs we have 207 APNs and 23 APMs. All APs at the beginning of each simulation have as trust value 100.

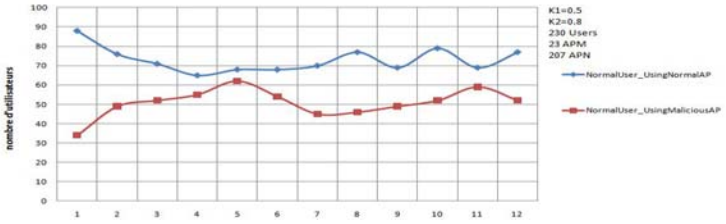


Figure 10. Result of Simulation 5

The number of normal users using APMs increases each time. This is possible because there is not any threshold to prevent users from connecting to APMs. There are more users using APNs than user using APMs, so the solution of Salem et al. promotes the selection of APNs but still the number of user using APMs is too high. By comparing it with our solution, we can notice that our solution decreases the number of users using APMs.

Simulation 6
We have 230 users, 230 APMs, K1=0.5 and K2=0.8. All APs at the beginning of each simulation have as trust value 100.

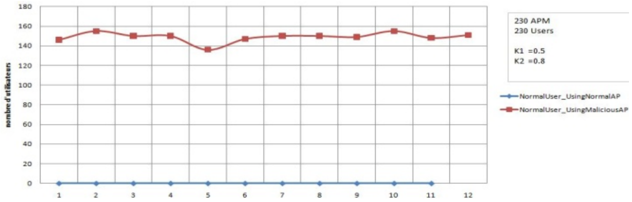


Figure 11. Result of Simulation 6

Figure 11 shows how Salem solution behaves when there are only malicious APs. As we can see, with Salem solution a lot of users use APMs because of the fact that Salem algorithm forces to use the AP who has the best reputation value among other APs even if all the APs have a low reputation value. Our solution prevents this case to happen, because the user will connect to an APM only if he is the first to use this AP after the insertion of the AP in the network. This is possible due to our threshold K1, which prevents users to connect to an AP which has a trust value lower than the threshold.

Simulation 7
We have 230 users, 189 APNs, 19 APSs, 46 APMs, K1=0.5 and K2=0.8. All APs at the beginning of each simulation has as trust value 100. In Salem solution APS will change behavior when its trust value will be lower than 50.

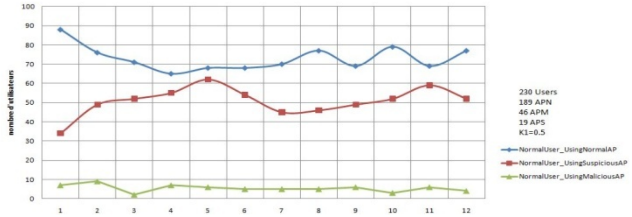


Figure 12. Result of simulation 7

Figure 12 shows how Salem solution deals with APSs and APMs at the same time. Result of simulation 7 shows that Salem solution does not prevent users from connecting to APMs and APSs. But we notice that just few users used APSs and some time the number of users using APs is close to zero. So, Salem solution is not useful from preventing users to use APSs and APMs.

4.4.2 EigenTrust

We compare also our solution with EigenTrust algorithm [10]. EigenTrust algorithm is mostly used on peer-to-peer systems. The goal of EigenTrust algorithm is to identify sources of unauthentic files and bias peers against downloading from them. EigenTrust gives to each peer a *trust value* based on its previous behavior. Each peer can ask the opinions of the people they trust and weight their opinions by their trust value. In order to apply EigenTrust some peers will be defined as APs and the remaining peers will be defined as users. EigenTrust allows peers to select the most trustworthy peer to interact with him. In our case, EigenTrust will help users to choose the most trustworthy AP. EigenTrust normalizes the trust value of peers, but EigenTrust does not distinguish between a peer with whom peer *i* did not interact and a peer with whom peer *i* has had poor experience. This is one of the weak points of their solution.

Simulation 8
We have 230 users, 230 APs. In the 1000 APs we have 207 APNs and 23 APMs. All APs at the beginning of each simulation have as trust value 0.5

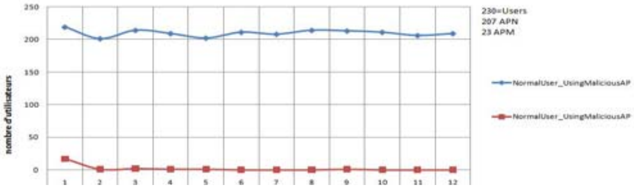


Figure 13. Result of Simulation 8

The number of normal users using APMs decreases quickly. At beginning all APs have the same trust value but after the first round the trust value of malicious APs decreases and trust value of APNs increases. Figure 13 shows that EigenTrust helps users to choose trustworthy APs in the situation where we can have malicious and normal APs.

Simulation 9
We have 230 users, 230 APMs, K1=0.5 and K2=0.8. All APs at the beginning of each simulation has as trust value 100. All APs at the beginning of each simulation have as trust value 0.5

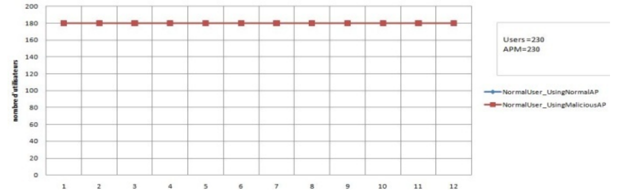


Figure 14. Result of Simulation 9

Figure 14 shows how EigenTrust solution behaves when there are only malicious APs. Unfortunately, as we can see, EigenTrust cannot prevent users from connecting to APMs when there are only APMs around. Our solution takes care of this case when there are only APMs available. This is possible due to our threshold K1, which prevents users to connect to an AP which has a trust value lower than the threshold.

4.4.3 Comparison Summary

Table 1. Synthesis of the robustness against attacks

Resistance Attacks	Our Solution	Salem Solution	EigenTrust Solution
Insertion Malicious APs	Yes	No	Yes
Change Behavior of APs	Yes	No	Yes
Insertion Malicious Users	Yes	No	Yes
All APs is Malicious	Yes	No	No

Table 2. Synthesis of the trust metrics

Trust Metrics	Our Solution	Salem Solution	EigenTrust Solution
Recommendation	Yes	No	Yes
Trust Value Normalized	Yes	No	Yes
FriendshipFactor	Yes	No	Yes
Thresholds	Yes (K1&K2)	No	No
Negative Rating	Yes	Yes	No

Our solution provides a way to select trustworthy APs and to prevent users to use malicious APs by taking into account the presence of malicious APs, malicious users and APs that can change behavior, meaning that a normal AP can become malicious or that a malicious AP can become normal.

5. Conclusion

In conclusion, we can say that our solution helps to promote the selection of trustworthy APs which have a certain degree of confidence and also to prevent users from connecting to malicious APs. Thus, our solution is applicable even in an environment where there are malicious APs. It is also applicable when APs change their behavior and it reduces the number of APs which try to trick users.
Comparing our solution with that proposed by Salem for selecting APs and with EigenTrust algorithm, we find that our solution is more robust than the one in Salem et al. because our solution is resistant to attacks such as inserting malicious APs and inserting malicious users. Regarding EigenTrust, our solution can deal with a higher percentage of malicious APs among all APs than Eigentrust.

Acknowledgments

The research leading to these results has received funding from the EU IST Seventh Framework Programme ([FP7/2007-2013]) under grant agreement n° 224024, project PERIMETER (User-centric Paradigm for Seamless Mobility in Future Internet) and from the EU IST Seventh Framework Programme ([FP7/2007-2013]) under grant agreement n° 257418, project ULOOP (User-centric Wireless Local Loop).

References

[1] Gralla Preston: Don't fall victim to the 'Free Wi-Fi' scam. In: ComputerWorld Networking & Internet. (2007)
[2] Douceur John R.: The Sybil Attack. In: Proceedings of 1st International Workshop on Peer-to-Peer Systems. (2002)
[3] E. Gray, J.-M. Seigneur, Y. Chen, and C. D. Jensen.: Trust Propagation in Small Worlds. In: Proceedings of the First International Conference on Trust Management, LNCS 2692, Springer-Verlag, 2003.
[4] Bettstetter C. and al., Stochastic Properties of the Random Waypoint Mobility Model. In: ACM Wireless Networks vol. 10, pp. 555–567, Sept. 2004.
[5] Ben Salem Naouel, Jean-Pierre Hubaux and M. Jakobsson. : Fuelling Wi-Fi deployment: A reputation-based solution. In: Proceedings of WiOpt. (2004).
[6] Nicholson Anthony J., Yatin Chawathe and Mike Y. Chen.: Improved Access Point Selection. In: MobiSys'06. (2006)
[7] Ormond O, Perry, P. Murphy, J.: Network Selection Decision in Wireless Heterogeneous Networks. In: Proc. Of IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications. (2005)
[8] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani. Energy Consumption in Mobile Phones: A Measurement Study and Implications for Network Applications. In Proc. ACMSIGCOMM IMC,2009
[9] Ben Salem Naouel thesis, 2007. Secure Incentives to Cooperate for Wireless Networks. Thesis (PhD). Ecole Polytechnique Federale de Lausanne.
[10] S.D. Kamvar, M.T. Schlosser, H. Garcia-Molina, The Eigen-Trust algorithm for reputation management in P2P networks, Proceedings of the Twelfth International World Wide Web Conference, Budapest, May, 2003.

Xavier Titi: PhD Student at University of Geneva. 3rd Best Poster Award to FIA Conference held on 23-24 November 2009 in Stockholm, Sweden; Best Quantitative Research Paper at IADIS International Conference Information Systems 2010 held on 18-20 March 2010 in Porto, Portugal.

Carlos Ballester Lafuente: Obtained his degree in Computer Engineering on 2007 from Universidad Politecnica de Valencia (Spain). He worked for one year as IT security auditor. He holds a MSc in Security and Mobile Computing (NordSecMob) jointly issued by Norwegian University of Science and Technology (Norway) and Aalto University (Finland) on 2010. He got the 2nd best poster award at the 14th Nordic Conference on Secure IT Systems (2009). Currently he is doing his PhD in Trust Management and Reputation at the University of Geneva as a researcher for EU FP7 (Seventh Framework Programme) ULOOP project.

Dr. **Jean-Marc Seigneur:** Has obtained his PhD at Trinity College Dublin on computational trust and identity management in 2005. He has won funding from the European Union and worked on EU projects in this field at the University of Geneva since then. He is now an assistant professor and has been consulted regarding his research, especially online reputation management, by many big companies such as Amazon,Siemens,Thales Verising and Philips.