

Reputation-based Wi-Fi Deployment: Protocols and Security Analysis

Naouel Ben Salem*, Jean-Pierre Hubaux*, Markus Jakobsson†

Abstract

In recent years, wireless Internet service providers (WISPs) have established thousands of WiFi hot spots in cafes, hotels and airports in order to offer to travelling Internet users access to email, web or other Internet service. However, two major problems still slow down the deployment of this kind of networks: the lack of a seamless roaming scheme and the variable quality of service experienced by the users. This paper provides a response to these two problems: We present a solution that, on the one hand, allows a mobile node to connect to a foreign WISP in a secure way while preserving its anonymity and, on the other hand, encourages the WISPs to provide the users with good QoS. We analyse the robustness of our solution against various attacks and we prove by means of simulations that our reputation model indeed encourages the WISPs to behave correctly.

1 Introduction

Wireless data services based on cellular networks, such as GSM/GPRS, provide users with very good coverage. However, they have several intrinsic and well-known drawbacks: the offered bitrates are relatively low (and this is unlikely to change with the Third Generation), and the deployment of new features is hampered by several factors such as the large size and oligopolistic behavior of the operators, their willingness to provide homogeneous service, and the huge upfront investment; in addition, very often, a user located in his home country is not allowed to obtain service from the competitors of his home network.

The deployment of wireless networks such as WiFi in unlicensed frequencies makes it possible to envision a substantial *paradigm shift*, with very significant benefits: much higher bandwidth network, deployment based possibly on local initiative, higher competition and much faster time-to-market for new features.

This may, in turn, pave the way for new types of services, whether these require higher bandwidth, lower per-bit costs, reduced energy consumption for the mobile nodes or higher reliance on fast-changing and locally provided content.

The current, rapid deployment of hot spots reveals the strong potential of this approach. However, two major problems still need to be solved. The first problem is the provision of a seamless roaming¹ scheme that would encourage small operators to enter into the market. This is a fundamental issue for the future of mobile communications. Indeed, without an appropriate scheme, only large stakeholders would be able to operate their network in a profitable way, and would impose a market organization very similar to the one observed today for cellular networks; one of the greatest opportunities to fuel innovation in wireless communications would be missed. The second problem is the guarantee of a good quality of service provision to the users.

This paper provides a response to these two challenges. By appropriately unbundling the major functions of the network, it institutes a virtuous cycle of deployment and usage: Wireless Internet Service Providers (WISP) will be encouraged to deploy their network and will be confident that mobile users registered with other WISPs² will pay for the service they receive; likewise, users will be assured that the WISPs are under the scrutiny of all the other users (including the roaming ones), and that they will be informed about their degree of satisfaction.

As we will see, the solution is relatively simple, provided that the roles of the different entities are clearly defined. We describe these entities in detail, along with the security protocols and the charging mechanism. In order to facilitate user acceptance, the proposed solution minimizes user involvement: once the mobile device has been initialized, it can make all decisions autonomously.

One of the major goals of this work is to build up trust between mobile users and WISPs. For this reason, we pro-

*Laboratory of Computer Communications and Applications (LCA), Swiss Federal Institute of Technology – Lausanne (EPFL), Switzerland ({naouel.bensalem, jean-pierre.hubaux}@epfl.ch)

†RSA Laboratories, 174 Middlesex Turnpike, Bedford, MA 01730, USA (mjakobsson@rsasecurity.com)

¹Note that by roaming we designate the operation of obtaining service from different operators, and not the handoff between access points (managed by the same provider or by two different providers). The hand-off problem is out of the scope of this paper.

²Unlike [1], we require the mobile node to be registered with a home WISP.

vide a detailed threat analysis and we show that the proposed protocols can thwart rational attacks and detect malicious attacks (we define these terms in Subsection 2.2).

The rest of the paper is organized in the following way: In Section 2 we present the system and trust models and we give an overview of the proposed solution. In Section 3, we describe the details of the protocols. We study the security of the protocols and analyse some interesting aspects of the solution in Section 4. In Section 5, the simulations are described and the results are analyzed. Finally, we present the state of the art in Section 6 and we conclude in Section 7.

2 System Model

In this paper, we consider a mobile node MN that wants to connect to the Internet via a neighboring hot spot (i.e., a hot spot that is within its power range); we assume the hot spot to be managed by a Wireless Internet Service Provider (WISP) that we denote by S (see Figure 1). MN is affiliated with its home WISP H^3 with whom it has an account and shares a symmetric key k_{HM} . We assume that all the messages exchanged between MN and H go through S . Note that it is possible to have $S = H$.

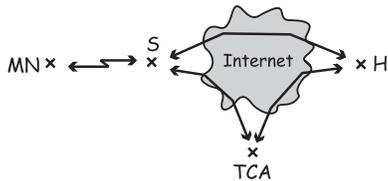


Figure 1: The mobile node MN is affiliated with a home WISP H and chooses to connect to the Internet via a hot spot managed by the WISP S (it is possible to have $S = H$). S and H are registered with the trusted central authority TCA .

All WISPs in our model are registered with the trusted central authority⁴ TCA that creates for each of them a public/private key pair and a certificate of their public key and of their identity.

In this paper, we present a reputation based mechanism that, on the one hand, allows MN to evaluate the behavior of the WISPs and, on the other hand, encourages the WISPs to provide the users with good QoS. Each WISP in our model has what we call a *reputation record* that represents an evaluation of its behavior and that is generated and signed by TCA . The choice of the initial reputation record of a WISP is discussed in Section 5.

³The solution works even if H does not operate access points itself.

⁴In a “grassroots” vision, the TCA would be a federation of WISPs, who join forces to centralize a few strategic functions. In a more conventional vision, the TCA can be under the control of a world-wide organization much as a quality control company, a certification company, or a global telecommunications operator. TCA can be distributed, as certification companies are, to avoid being a bottleneck.

In order to make sure that the mobile nodes pay for the service they receive, we also propose a credit-based micro-payment scheme (see Subsection 3.1.1) that is highly inspired from the PayWord scheme [19]. Our solution takes into account the fact that MN is a resource-restrained mobile device and therefore has much less computing and storage resources than TCA , H or S .

2.1 Assumptions

We make the following assumptions in this paper:

- The public key of TCA is known by all other entities.
- H and S can use their public keys to establish a temporary symmetric key k_{HS} . We assume that this key is generated prior to the execution of our set of protocols.
- S is able to predict the QoS it can offer to a mobile node that is willing to connect to one of its hot spots. We will discuss this issue more in detail in Subsection 5.3.
- The backbone is a commodity; the rewarding of the backbone operator should follow already established practices and techniques, and will not be addressed in this paper (we assume that S , H and TCA have an appropriate agreement to have connectivity - e.g., a flat rate subscription).

2.2 Trust and adversarial model

We consider an attacker \mathcal{A} that wants to perform an attack against our protocols (see Subsection 4.1 for the list of attacks). \mathcal{A} can be a mobile node or a WISP. We assume that:

- TCA never cheats and is trusted by the other parties for all the actions it performs.
- The WISPs (here S and H) are rational and therefore they cheat (i.e., perform one of the attacks presented in Subsection 4.1) only if it is to their advantage (i.e., they gain something from cheating). This assumption is reasonable because a WISP is typically stationary and therefore it is possible to shut it down if it cheats; the WISP is thus likely to be motivated by economic incentives, and would not be inclined to disrupt the communication of mobile nodes (who could simply choose another WISP if this were to occur).
- MN may be malicious and therefore it can cheat (i.e., perform one of the attacks presented in Subsection 4.1) even if there is no gain from cheating (this implicitly assumes that MN can also perform rational attacks).

- *MN* trusts *H* for managing its account.
- Several attackers can collude and share information (possibly their secret keys) to perform more sophisticated attacks.

Confidentiality of data is not an issue in our case, so we do not consider passive attacks where the attacker eavesdrops the data exchanges between two parties. Note that this is an orthogonal issue that is easily addressed using standard security techniques.

We consider exclusively attacks performed against the different phases of our protocols, meaning that we do not consider other arbitrary attacks like DoS attacks based on jamming for example.

In this paper, we want to study the effect of rational and malicious attacks on our set of protocols. Our goal is to make sure that our solution thwarts rational attacks, detects malicious attacks and, if possible, identifies the attacker.

2.3 Rationale of the solution

When *MN* wants to connect to the Internet, it identifies the neighboring WISPs⁵ and contacts them (see Figure 2). Each WISP sends to *MN* an offer that contains its reputation record, the QoS it proposes and the price it asks for. Then, *MN* selects the WISP *S* that proposes the best offer and verifies its identity. *S* also verifies, with the help of *H*, that *MN* is a valid node. *MN* and *S* establish a contract, inform *TCA* and *H* about it and establish a secure session by setting up a symmetric key k_{MS} .

This secure session is divided into parts. During the i -th part, *MN* sends a payment proof for the i -th part of the service and *S* provides that part of the service. The payment proofs and the services are secured using the shared key k_{MS} .

At the end of the connection, *MN* assesses the QoS it received, compares it to the QoS advertised by *S* during the session setup and informs *TCA* about its *satisfaction level*. *S* also sends the payment proof(s) to *H* which charges *MN* and remunerates *S* according to the received information.

TCA collects the feedback about the different WISPs, updates periodically the reputation records according to the collected information and provides the WISPs with their new reputation records.

⁵Note that we refer to the access points using the identities of the WISPs that are managing them.

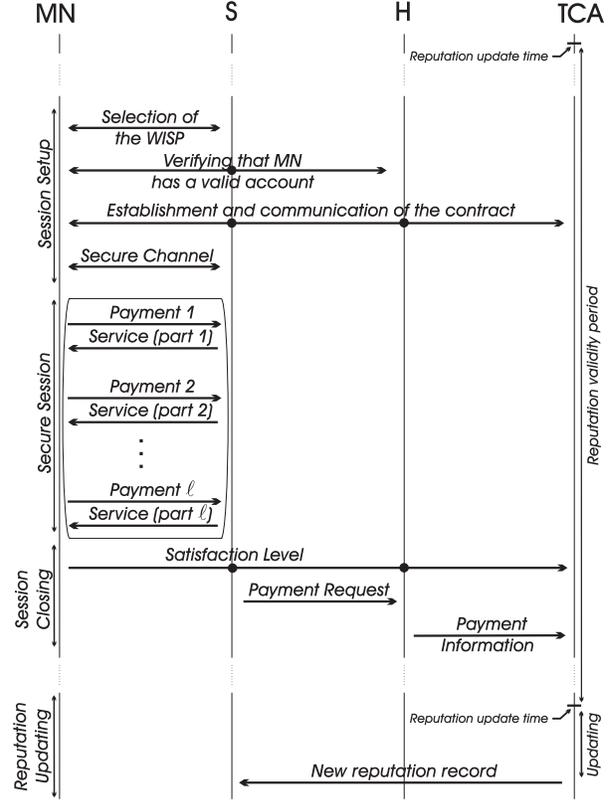


Figure 2: The proposed solution

3 Proposed Solution

3.1 Basic mechanisms

3.1.1 Micro-payment scheme

As already mentioned in Section 2, the payment scheme we use in this paper is highly inspired from the PayWord scheme [19]: During the session setup, *MN* generates a long fresh chain of paywords w_0, w_1, \dots, w_n by choosing w_n at random and by computing $w_i = h(w_{i+1})$ for $i = n-1, n-2, \dots, 0$, where h is a one-way hash function and n is the maximum number of payments that *MN* can send to *S* during the session. Then, *MN* reveals the root w_0 of the payword chain (which is not considered as a payword itself) to *S*, *H* and *TCA*.

During the session, *MN* sends (w_i, i) to *S* as a payment proof for the i -th part of the service. *S* can easily verify w_i using w_{i-1} that is known from the previous micro-payment or from w_0 if $i = 1$. At the end of the service provision, *S* contacts *H* and presents the last payment (w_ℓ, ℓ) it received. *H* verifies the validity of w_ℓ , pays *S* the amount corresponding to ℓ paywords and charges *MN* for that amount by manipulating its billing account.

We use this micropayment scheme because it allows an offline verification of the payment proofs and because

of its low computational and storage cost for the mobile nodes.

3.1.2 Authentication of MN by H

As stated in Section 2, all communication between MN and H goes through S . Therefore, in order to preserve the anonymity of MN regarding S , we use the following authentication mechanism, that is commonly used in the industry (e.g., SecurID [11]): When MN gets affiliated with H , the two parties share a random seed s that represents the input to a pseudorandom generator. The output is a random number tag that is 30 to 50 bits long. H keeps a small window (e.g., 50 entries) of upcoming tags for each mobile node and maintains the pairs (tag ; $node$'s $identity$) in a sorted database. Upon receipt of a given tag , H searches its database, retrieves the pair (tag ; $identity$) and identifies MN . In case of collision (i.e., more than one pair contains the random number tag), H asks MN to send the next tag value.

3.2 Details of the protocols

3.2.1 Selection of the WISP

When it wants to obtain Internet access, MN scans the spectrum, identifies the neighboring WISPs and asks them an offer by broadcasting the following request message:

$$OfferReq = [ReqID, n_M] \quad (1)$$

where $ReqID$ is the request identifier and n_M is a nonce generated by MN . Each WISP W willing (and able) to provide service at that time responds by a signed offer $Offer_W$:

$$W \rightarrow MN : Offer_W, S_{pk_W}(Offer_W, n_M) \quad \text{where} \\ Offer_W = [W, RR_W, AQ_W, P_W, Cert(W)] \quad (2)$$

where RR_W is the most recent *reputation record* of W (signed by TCA), AQ_W is the QoS it advertises⁶, P_W is the price it is demanding for each part of the service (see Subsection 3.2.4), pk_W is its private key and $Cert(W)$ is the certificate of its public key PK_W .

Upon receipt of the offers, MN verifies the freshness of n_W and identifies the best offer. This choice depends on the relative importance that MN gives to the parameters RR_W , Q_W and P_W (as shown in Section 5, these parameters can depend on the application MN intends to run) and should be made by a software agent to automate the process and avoid human involvement. More sophisticated schemes (e.g., auctioning) can be envisioned in order to select the best offer.

⁶ W may advertise a QoS that is higher than the real QoS (RQ_W) it is able to offer to MN . The consequences of such a behavior are studied in Section 5.

Then, MN verifies the certificate and the signature of the WISP that proposed the best offer. If the verification is incorrect, MN checks the second best offer and so on. We denote the selected WISP by S .

3.2.2 Verifying that MN has a valid account

Before starting the session, S has to make sure that MN is a valid mobile node that is registered with a valid home WISP. As we want to preserve the anonymity of MN , the verification of MN 's identity involves H and uses the authentication mechanism described in Subsection 3.1.2. We have thus the following messages exchanged:

$$MN \rightarrow S : \mathcal{M} = [H, tag, n_M, \\ E_{k_{HM}}(MN, S, tag, n_M)] \quad (3)$$

$$S \rightarrow H : S, n_S, \mathcal{M}, MAC_{k_{HS}}(S, \mathcal{M}) \quad (4)$$

$$H \rightarrow S : TID, E_{k_{HM}}(TID, n_M, k_{MS}), \\ E_{k_{HS}}(TID, n_S, k_{MS}) \quad (5)$$

$$S \rightarrow MN : TID, E_{k_{HM}}(TID, n_M, k_{MS}) \quad (6)$$

(3) MN sends to S a message \mathcal{M} containing, in clear, the identity of H , its current tag and a freshly generated nonce n_M . \mathcal{M} also contains, encrypted using the symmetric key k_{HM} , the identities of MN and S , the tag and the nonce n_M .

(4) S sends to H its identity, a freshly generated nonce n_S , the message \mathcal{M} and a MAC computed on both items using the key k_{HS} .

(5) H searches its sorted database, identifies MN using the tag sent in clear (as explained in Subsection 3.1.2), looks up the symmetric key it shares with MN and uses it to decrypt the rest of the message. Then, H re-checks the identity of MN (the identity corresponding to the tag should also correspond to the identity MN encrypted in the message) and verifies that the WISP with which MN intends to interact is the one that sent the message.

If the message is not correct, H informs S that MN is not affiliated with it by sending a negative acknowledgement. If, on the contrary, the message verifies correctly, H generates a symmetric key k_{MS} that MN and S will use later as a session key (i.e., all the messages exchanged between MN and S during the session are secured using k_{MS}). Then, H constructs a message containing:

- in clear, a fresh temporary identifier TID for MN (MN will use this identifier later during its interactions with S),
- TID , n_M , and k_{MS} encrypted using the symmetric key k_{HM} , and
- TID , n_S , and k_{MS} encrypted using the symmetric key k_{HS} ,

and sends this message to S . H maintains a table containing the correspondence between the temporary identifiers and the identities of the nodes; given TID , H can positively identify the correspondent MN .

(6) S decrypts $E_{k_{HS}}(TID, n_M, k_{MS})$, verifies that the temporary identifier in the decrypted part corresponds to the one sent in clear and compares the nonce in the decrypted part with the one generated by MN . If these verifications are correct, S removes $E_{k_{HS}}(TID, n_M, k_{MS})$ from the message and forwards the rest to MN .

MN decrypts $E_{k_{HM}}(TID, n_H, k_{MS})$ and verifies the temporary identifier and the nonce as S did. If everything is correct, MN maintains TID in memory.

Note that if $S = H$, MN sends message (3) to H and H responds with message (6).

3.2.3 Contract establishment and communication

During this phase, MN generates a long hash chain of $n + 1$ elements, computed from a randomly chosen seed w_n as described in Subsection 3.1.1. Then MN generates a contract C as follows:

$$C = [CID, w_0, R_S, A Q_S, P_S]$$

where $CID = [TID, S, H]$ is the contract identifier and w_0 is the root of the hash chain.

Then MN and S inform H about the contract:

$$MN \rightarrow S : C, MAC_{k_{MS}}(C), MAC_{k_{HM}}(C) \quad (7)$$

$$S \rightarrow H : C, MAC_{k_{HM}}(C), MAC_{k_{HS}}(C) \quad (8)$$

(7) MN sends the contract C to S , together with two MACs computed on C using the symmetric keys k_{MS} and k_{HM} , respectively.

(8) S verifies C and $MAC_{k_{MS}}(C)$ and if they are correct, it computes a MAC on C using the symmetric key k_{HS} it shares with H . Then, S sends to H the contract C and the MACs computed with k_{HM} and k_{HS} . H verifies the MACs and, if they are correct, it stores the contract C .

MN and S also inform TCA about the contract:

$$MN \rightarrow S : E_{PK_{TCA}}(C, k_{MT}, pad), \\ MAC_{k_{MS}}(E_{PK_{TCA}}(C, k_{MT}, pad)) \quad (9)$$

$$S \rightarrow TCA : C, E_{PK_{TCA}}(C, k_{MT}, pad) \quad (10)$$

$$TCA \rightarrow S : S_{pk_{TCA}}(C), MAC_{k_{MT}}(C) \quad (11)$$

$$S \rightarrow MN : MAC_{k_{MT}}(C) \quad (12)$$

(9) MN generates a fresh symmetric key k_{MT} that MN will use later to encrypt data for TCA (see Subsection 3.2.6). In order to prevent the key retrieval by an attacker, MN uses the probabilistic encryption by appending to the key a pseudorandomly generated bitstring pad (the length on the bitstring depends on the encryption algorithm used). Then, MN encrypts C , k_{MS} and pad using

the public key of TCA , computes a MAC on this data using the key k_{MS} it shares with S and sends the encrypted data and the MAC to S .

(10) S verifies the MAC, removes it and sends C and the encrypted data to TCA .

(11) TCA decrypts the data and compares the contract C received in the encrypted data with the contract received in clear from S . If they are identical, TCA signs the contract C using its private key pk_{TCA} , computes a MAC on it using the symmetric key k_{MT} that it shares with MN , and sends the signature and the MAC back to S . TCA also maintains C and k_{MT} in its local database.

(12) S verifies the signature and if correct, it forwards the MAC to MN which verifies it and stores k_{MT} .

3.2.4 Service provision and payment

The session is subdivided into parts, depending on time or on the amount of data exchanged between MN and S . During the i -th part:

$$MN \rightarrow S : TID, w_i, MAC_{k_{MS}}(TID, w_i) \quad (13)$$

$$S \rightarrow MN : \text{i-th part of the service,} \\ MAC_{k_{MS}}(\text{i-th part of the service}) \quad (14)$$

(13) MN sends to S its temporary identity TID , the i -th PayWord w_i and a MAC computed on both items using the key k_{MS} .

(14) S verifies the validity of w_i by checking that $h(w_i) = w_{i-1}$, where h is the one-way hash function used by MN to generate the chain. If it is correct, S provides MN with the i -th part of the service.

3.2.5 Sending the payment request

At the end of the session, S sends to H a payment request PR that contains, encrypted using k_{HS} , the contract identifier CID , the last hash value w_ℓ it received from MN and the number ℓ of provided service parts. PR also contains, in clear, the identity of S so that H is able to retrieve the symmetric key k_{HS} .

$$S \rightarrow H : PR = [S, CID, w_\ell, \ell, MAC_{k_{HS}}(S, CID, w_\ell, \ell)] \quad (15)$$

Upon receipt of PR , H verifies the validity of w_ℓ as explained in Subsection 3.1.1, retrieves the price P_S from the contract, rewards S for the ℓ parts of the service, and charges MN . H is also remunerated (see details in Subsection 3.3).

3.2.6 Sending the satisfaction level

At the end of the session, MN generates a *satisfaction level* message Sl as follows:

$$Sl = [E_{k_{MT}}(CID, QoS_{Eval_S, CID, w_\ell, \ell})] \quad (16)$$

$QoS_{Eval_{S,CID}}$ is expressed by MN and compares to what extend the QoS it obtained during the session is complaint with the QoS announced by S in the offer. k_{MT} is the key MN shares with TCA .

Then, MN reports on its *satisfaction level* to TCA :

$$MN \rightarrow S : TID, Sl, MAC_{k_{MS}}(TID, Sl) \quad (17)$$

$$S \rightarrow TCA : S, CID, w_\ell, \ell, Sl, \\ S_{PK_S}(S, CID, w_\ell, \ell, Sl) \quad (18)$$

(17) MN sends to S its temporary identifier TID, Sl data and a MAC computed on both items.

(18) S verifies the MAC. If it is correct, S generates a message containing CID, w_ℓ, ℓ and Sl , signs it and sends the message and the signature to TCA .

TCA verifies the signature and retrieves the key it shares with MN (using CID). Then TCA decrypts Sl , compares the CID, w_ℓ, ℓ in the encrypted data to those received in clear from S and if they are identical, TCA considers QoS_{Eval} as a valid feedback. Then TCA informs H that it correctly received the feedback:

$$TCA \rightarrow H : Ack, S, CID, \\ S_{PK_{TCA}}(Ack, S, CID) \quad (19)$$

(19) H verifies the signature and retrieves the identity of MN (using CID). Then, H remunerates MN a small amount of money ε , which is meant to encourage the mobile nodes sending the reports.

3.2.7 Updating the reputation record

TCA collects the information about the satisfaction levels for a given period and then, at the *reputation update time*, TCA updates the reputation record of each WISP, signs them and informs the WISPs about their new records. The new reputation record depends on the old one and on the collected information. An example is given in Section 5.

TCA considers the absence of feedback as negative feedback. Indeed, TCA knows that a session has been established between MN and S and that H is the home WISP of MN (see Subsection 3.2.3). TCA is thus waiting for the report from MN about its interaction with S , and not receiving it within a “reasonable” time is considered as bad feedback.

3.3 Charging and rewarding model

In this subsection, we summarize the charging and rewarding mechanism we use in this paper:

- During session setup, MN generates a chain of pay-words w_0, w_1, \dots, w_n .

- During the secure session with S , MN sends (w_i, i) to S as a payment proof for the i -th part of the service.

- H remunerates MN a small amount ε when it receives from TCA the confirmation that MN reported on its interaction with S .

If, at the end of the session, MN moves away from S (and therefore cannot send the feedback via S), it is still possible for MN to report on its satisfaction level to TCA via another WISP W : W includes its identity in message (18) and signs the message using its own private key. TCA then verifies the signature and informs H in message (19) about the identity of W . Then H gives both MN and W a reward $\varepsilon/2$.

- At the end of the session, S sends to H the last payment proof (w_ℓ, ℓ) it received from MN . H verifies the validity of the payword w_ℓ , charges MN the amount $P_S * \ell$ corresponding to the ℓ parts of the service and rewards S , using a well-established e-payment technique, the amount⁷ $P_S * \ell - \varepsilon$. If TCA receives no report from MN , ε is handled according to some policy (e.g. it can be distributed to charity).

- The home network H is also remunerated. This can be done e.g., if MN pays a flat monthly subscription A or if MN pays an amount a per session. The two approaches are equivalent if we consider that $a = A/nbSessions$ where $nbSessions$ represents the average number of sessions established by MN during one month. For sake of simplicity, we consider the second approach in this paper. A numerical example is given in Section 5.

4 Security assessment

4.1 Attacks

In this Subsection, we identify the attacks that an attacker⁸ \mathcal{A} may want to perform against our protocols (see Subsection 2.2 for the trust and adversarial model). We identify the following attacks that are specific to our solution:

- *Publicity* attack: In the offer it sends to MN , S advertises a QoS that is higher than the real QoS it can offer.
- *Selective publicity* attack: S performs the Publicity attack with a specific MN .

⁷As already mentioned, ε is the reward MN receives if it reports on its satisfaction level to TCA .

⁸As mentioned in Subsection 2.2, \mathcal{A} can be a mobile node or a WISP.

- **Denigration attack:** MN receives a good QoS from S but pretends the contrary by sending a negative report on the satisfaction level or by not sending the report at all.
- **Flattering attack:** MN sends systematically a good feedback about S 's behavior to TCA . This attack makes sense particularly if $S = H$.
- **Report dropping attack:** MN sends the report but S does not transmit it to TCA .
- **Service interruption attack:** S receives the i -th payment proof from MN but does not provide the corresponding part of the service.
- **Refusal to pay attack:** MN does not send the i -th payment to S .
- **Repudiation attack:** S or MN retracts the agreement it has with other party (e.g., S asks for higher price than agreed on when the contract C was established).

We also consider general attacks such as:

- **Packet dropping attack:** A drops a message it is asked to forward or discards a message it is asked to generate and send.
- **Filtering attack:** A modifies a packet it is asked to forward or generate.
- **Replay attack:** A replays a valid message that was exchanged between two legitimate parties.

We do not consider the case where a MN is compromised but not duplicated (e.g., the the mobile device in stolen): Well-established mechanisms (e.g., blocking the node's account) can be used in this case.

4.2 Security Analysis

In this subsection, we will analyse the robustness of our protocols against these attacks.

Publicity attack: If S does not provide MN with the promised QoS, MN will send a negative report to TCA . If this attack is repeated, the cumulation of the negative reports will affect the future reputation records of S . If on the contrary, this attack is performed rarely, it will not affect much the reputation of S but S gains almost nothing from performing this attack; as S is rational, it will not perform this attack.

The same reasoning holds if $S=H$ with, in addition, the possibility for MN to punish H by choosing another home WISP.

Selective publicity attack: The anonymity of the mobile nodes prevents S (if $S \neq H$) from performing the Publicity attack against a specific MN . The only possible selection would be based on the home network (i.e., S performs the Publicity attack with all the MNs affiliated with a given home network). S gains nothing from this attack and thus S will not perform it.

Denigration attack: If MN does not send the report on the satisfaction level, H will not give it the ε reward and TCA will consider the absence of feedback as negative feedback. Therefore, this attack is not rational for MN .

So it is more interesting for MN to send a negative feedback instead of not sending the report at all: The effect of the attack is the same and at least MN will get paid for the sending. But this attack is still not rational. Indeed, MN gains nothing from sending a negative feedback instead of a positive one (the cost of the sending remains the same). Such behavior is thus purely malicious.

This attack is not harmful for the WISP, unless it is performed systematically and by a high number of colluding attackers. However, TCA can statistically detect it if the following events happen frequently⁹:

- The MNs affiliated with H always pretend that they received a bad QoS from a given WISP (from a given hot spot managed by that WISP), whereas many other MNs report on a good QoS on that very WISP¹⁰. As the selective publicity attack is not possible, this situation is suspect.
- H never receives reports from MNs affiliated with H about the sessions they established with S .
- The MNs affiliated with H pretend that the QoS was bad but at the same time the duration of the session and the amount of data exchanged prove that the QoS was good¹¹.

Note that this attack comes with an important cost: if an attacker A wants to alter the reputation of S by parking misbehaving nodes close to the hot spots managed by S , A should own many devices and devote them to the attack. Note also that this colluding attack may harm very small WISPs (with few number of hot spots) - if the attacker pays the price - but it is much too costly against WISPs with hundreds or thousands hot spots.

⁹The higher the number of events is, the more accurate the detection is. Note that statistical detection techniques do not hold if the majority of the nodes are misbehaving, which is not likely to be the case in WiFi networks.

¹⁰In order to have more accurate detection, TCA can consider each access point of the WISP separately.

¹¹ TCA knows the root w_0 of the hash chain from the contract and knows w_{ell} from the report; it can therefore estimate the amount of data exchanged between MN and S .

Flattering attack: It is not rational for MN to send a positive feedback if it receives a bad QoS from S , unless it has an incentive to do so (e.g., S remunerates MN for the reports).

This attack improves the reputation of the targeted WISP only if it is performed systematically and by a high number of colluding attackers. The detection mechanism can be similar to the one proposed for the Denigration attack.

However, a specificity of this attack resided in the fact that H can create “virtual” MNs (i.e., MNs that have an account but are not necessarily real devices), emulate connections with them and make them systematically send positive feedback. This leads to a cost that is much lower than the cost of the Denigration attack but TCA can detect it if (i) the MNs affiliated with H rarely connect to foreign WISPs (or at least much less than average) or if (ii) H is not rewarded for the connections it established with a high number of MNs affiliated with it (if we assume that this information is available to TCA).

Report dropping attack: If S expects a negative feedback, it may want to drop the report on the satisfaction level instead of transmitting it to TCA . But as the absence of feedback counts as negative feedback, this dropping does not help S . Furthermore, the report may be positive: Assuming that the feedback is defined between values $minRep$ and $maxRep$, not receiving the report corresponds to a feedback of $minRep$. This attack is therefore not rational for S .

Service interruption attack: If S refuses to provide the i -th part of the service, MN will keep asking for it (by sending again the i -th payment). After a predefined number of retransmission requests, MN will end the session, which prevents S from providing more service parts (and thus earning more money) and at the same time affects the satisfaction level of MN .

If nevertheless, we want to prevent S from receiving the i -th payment without providing the i -th part of the service, we can use the payment system presented in [5].

Refusal to pay attack: If MN does not send the i -th payment, S will not provide the i -th part of the service and the session will end (after a predefined number of retransmission requests). This attack is then not rational: It prevents MN from receiving the service part but does not harm S .

Repudiation attack: This attack is not possible because H and TCA receive the contract C from both MN and S (Messages 8 and 10). The two copies should be identical,

otherwise TCA will not send the message 11 and the session setup will not terminate. Therefore, once the session is established, MN and S cannot retract their agreement.

To prevent S or MN from sending a correct information to TCA but not to H , we can also require a response from H to establish the session.

Packet dropping attack: If a message is not generated or is dropped during session setup, the secure session will not be established. If $\mathcal{A} = MN$ (i.e., MN does not generate messages 1, 3, 7 or 9), it will not be able to connect to the Internet but does not harm S . If $\mathcal{A} = S$, it will not provide the part of the service to MN ; MN will select another WISP and S would lose an opportunity for revenue.

If during the secure session, the payment proof or the part of the service is not generated or is dropped, the entity that is waiting for it asks for retransmissions (if needed several times). If it does not receive the message, the secure session is closed.

If S does not forward the satisfaction level of MN , it is equivalent to the denigration attack (see Subsection 4.2).

If S does not generate the payment request and sends it to H (Message 15), it will not get rewarded for the service parts it provided to MN .

Filtering attack: The messages exchanged between the different parties in our protocols are cryptographically protected, using MAC computations or digital signatures. Therefore, any modification of a message will be detected at the receiver. Therefore, tampering with a message is equivalent to not sending the message at all (an incorrect message is discarded) and it is treated in the same way (see the *Packet dropping* attack).

Replay attack: During session setup, the messages exchanged between the different entities (Messages (2) to (6)) are protected using nonces; the delayed messages are detected and discarded.

During the secure session: the payment proofs and the parts of the service arrive in sequence; a replay is immediately detected and discarded.

During session closing, the payment request (Message (15)) and the satisfaction level (Messages (17) and (18)) are expected only once; a replay is immediately detected and discarded.

4.3 Overhead

In this subsection, we evaluate the computation and communication overhead of our solution for a mobile node. We consider only the mobile node because it is the only entity that is severely resource restrained and because in this way we cover all the wireless communications.

4.3.1 Computation overhead

During the different phases of our protocols, we use symmetric key and public key cryptography primitives to secure the message exchange and to correctly authenticate the different parties involved in the communication. We minimize however the use of public key cryptography, especially by the mobile nodes, to reduce the computation cost of our solution.

Hence, MN uses public key primitives only for two messages: it verifies the certificate, the signature and the reputation of the WISP it selects (Message 2) and it encrypts a message for TCA (Message 9). For all other messages, MN uses symmetric key cryptography primitives: $5 + 2\ell$ MAC operations (ℓ being the total number of service parts), 2 symmetric key encryptions and 1 symmetric key decryption.

Public key operations are also used in the message exchange between TCA and the two WISPs S and H (Messages 11, 18 and 19). It is however possible to commute them into symmetric key operations, if we assume that S and TCA establish a symmetric key when they first begin their interaction.

Note that the existence of a tamperproof resistant hardware at MN is not necessary for the good functioning of our protocols, but it may be a good solution for protecting the long term symmetric key k_{HM} that MN shares with H .

4.3.2 Communication overhead

Table 1 provides reasonable values of the size of the different fields appearing in our protocol.

Field Name	ReqID	IDs	n_M, pad	w_i	ℓ
Size (bytes)	4	16	20	20	2
Field Name	MAC	PK	QoS, P, R	k	tag
Size (bytes)	16	150	1	16	6

Table 1: Size of the fields used in our protocol

$ReqID$ is encoded on 4 bytes to reduce the risk of using the same identifier for two different requests. The identifiers of the WISPs and the nodes (W , H , S , MN and TID) are encoded on 16 byte (assuming e.g. an IPv6 format). The paywords w_i are encoded on 20 bytes (assuming e.g. SHA) and the QoS (AQ and $QoSEval$), the reputation R and the price P are encoded on 1 byte (which is enough to encode values between 0 and 100). The symmetric keys k_{HM} , k_{HS} , k_{MS} and k_{MT} are encoded on 16 bytes (128 bits) and the public keys are encoded on 150 bytes (assuming e.g. RSA, see [13]). We encode the nonce n_M and the pad on 20 bytes, the tag on 6 bytes (see Subsection 3.1.2) and MAC on 16 bytes. Finally, we encode ℓ on 2 bytes to support long sessions.

We consider the example where MN is downloading a 1 MB file. The file is divided into 1 KB packets and each 50 packets represent a part of service ($\ell = 20$ parts of service in total). Using the values of Table 1, an end-to-end session between MN and S represents an overhead, for MN , of 18337 bytes, which represents an overhead per packet of around 18 bytes (i.e., less than 2% of the packet size).

5 Reputation mechanism assessment

Our solution motivates the different players to participate in the reputation mechanisms. Indeed:

- W is motivated to provide MN with the QoS it promised because otherwise the feedback of MN will be negative (see the analysis of the *Publicity* attack in Subsection 4.2).
- MN is motivated to report on its interaction with W because it receives a refund ε .
- W is motivated to forward the report (see the analysis of the *Report dropping* attack in Subsection 4.2).

However, we want also to study the effect of the reputation mechanism on the behavior of the WISPs, i.e., the QoS they effectively offer to the mobile users. We therefore implemented our set of protocols using ns-2 simulator [10].

Using these simulations, we want to verify that:

- The WISPs are encouraged to provide the MNs with a good QoS;
- The WISPs are discouraged from advertising a QoS that is different from the QoS they can really offer;
- It is possible for a WISP that has a bad reputation record to improve its reputation.

5.1 Simulations setup

5.1.1 Decision making at MN

During the WISP selection phase, MN receives several offers from the WISPs. For each offer $Offer_W$, MN computes a value $D_W = Rep_W^\alpha \cdot AQ_W^\beta \cdot P_W^{-\gamma}$. It then determines $D_S = \max_W D_W$ and selects the WISP S .

- Rep_W is the reputation of the WISP W : It is a value between $minRep=0$ and $maxRep=100$.
- AQ_W is the QoS advertised by W : For sake of simplicity, we also assume that it is a value between $minQoS=0$ and $maxQoS=100$.

- P_W is the price W is demanding for each part of the service.
- The exponents α , β and γ are parameters that depend on the application MN is running and that are used to emphasize the importance of the variables (Rep_W , AQ_W or P_W). We consider as an example the two following applications:
 - Chat: The user is most likely to choose the WISP that asks for the lowest price. Therefore, we set α , β and γ to 2, 1 and 3, respectively.
 - File transfer: The user is most likely to choose the WISP that offers the highest QoS. Therefore, we set α , β and γ to 2, 2 and 1, respectively.

Note that:

- In order to minimize the human involvement, the user should set the parameters α , β and γ , for each family of applications, once and for all. However, he should have the possibility to modify them if needed.
- The traffic model (i.e., the frequency at which the packets are sent from S to MN) is the same for the three applications. The only difference is in the choice of the parameters α , β and γ .
- If two (or more) WISPs have the same D_W , one of them is selected at random.

More sophisticated utility functions can include criteria such as the minimum QoS MN is expecting or the maximum price it is willing to pay.

5.1.2 Service provision and QoS

The real QoS RQ_S received by MN can be different from AQ_S (the QoS advertised by S during session setup).

During the implementation of our set of protocols, we represented the behavior of S whose real QoS is RQ_S , $0 \leq RQ_S \leq 100$ as follows¹²: Each time S has to provide a “part of service”¹³ to MN , it sends it with a probability $RQ_S/100$. If MN does not receive the packet, it sends a retransmission request to S . After 4 unsuccessful retransmission requests, MN closes the session with S . The time during which MN is waiting for the packets and asking for retransmissions represents a delay that justifies the decrease of the QoS offered by S .

¹²As mentioned in Subsection 5.1.1, we assume that $minQoS=0$ and $maxQoS=100$.

¹³For sake of simplicity of explanation, we consider in our implementation that the provider sends one part of service per packet.

5.1.3 Satisfaction level report

At the end of each session, MN evaluates the real QoS it received from S . There can be different levels of satisfaction for this evaluation. We provide here a simple example based on packet counting:

$$RQ_S = \max(0, \frac{nbPkts - nbRetReq}{nbPkts} \cdot \maxQoS)$$

where $nbPkts$ is the total number of packets it received¹⁴ from S and $nbRetReq$ is the number of retransmissions it had to request.

Then, MN compares RQ_S to AQ_S by computing:

$$QoSEval_{S,CID} = \frac{RQ_S}{AQ_S}$$

5.1.4 Reputation records update

TCA updates the reputation records every 2000 seconds. The new reputation $newRep_S$ of S is computed as follows:

$$newRep_S = \lambda \cdot Rep_S + (1 - \lambda) \cdot \frac{\sum_{CID} QoSEval_{S,CID}}{nbSessions_S}$$

where Rep_S is the current reputation of S , $nbSessions_S$ is the number of sessions established by S (and already closed) during the last 2000 seconds and $feedback_S$ is the sum of all $QoSEval_S$ received over all these sessions (the absence of feedback is considered as negative feedback, i.e., $QoSEval_S = 0$). λ represents the “weight of the past” and is set to 1/2 in our simulations.

Note that if S advertises a QoS that is lower than the real QoS it offers (i.e., $AQ_S < RQ_S$), we will have $QoSEval_S > maxRep$, which may lead to a new reputation that is also higher than $maxRep$. If it is the case, TCA keeps $newRep_S$ as it is in its database but sends to S a new reputation record equal to $maxRep$.

5.1.5 Simulation environment

We consider a network of 5 WISPs and 50 MNs. The WISPs are numbered from 1 to 5 and for each WISP, we define the advertised QoS, the real QoS and the price it asks for each part of the service. We initialize the reputation of the WISPs to $maxRep = 100$. MNs and WISPs are static¹⁵ and each WISP is a home WISP for 10 MNs. Each simulation lasts for 50000 seconds and the reputation updates are made every 2000 seconds.

We consider that a WISP W is:

¹⁴In the special case where $nbPkts = 0$ (i.e., MN receives no packet from S), we have $RQ_S = 0$.

¹⁵All MNs are within the power range of all WISPs, it is therefore useless to consider mobility in this case.

- “honest” if it advertises the real QoS it is offering (i.e., $RQ_W = AQ_W$),
- “misbehaving” if it advertises a QoS that is higher than the real QoS it is offering (i.e., $RQ_W < AQ_W$),
- “modest” if it advertises a QoS that is lower than the real QoS it is offering (i.e., $RQ_W > AQ_W$).

We conducted three sets of simulations to study three aspects of our solution:

Set 1: We want to study the reaction of the network if all the WISPs are honest but offer different QoSs: WISPs 1, 2, 3, 4 and 5 advertise and offer QoS = 60, 70, 80, 90 and 99, respectively¹⁶. We consider the two following scenarios:

Scenario 1.1: All the WISPs ask for the same price. At the beginning of a simulation, we assign to each MN , with equal probability, one of the two following applications: chat or file transfer (see Subsection 5.1.1).

Scenario 1.2: The WISPs ask for prices that are proportional to their QoSs ($P_W \sim RQ_W$). We expect the choice of the application to have an effect on the results, so we run 3 sets of simulations; one for each kind of application (i.e., all the nodes run that application).

Set 2: We want to study the reaction of the network to the presence of misbehaving WISPs and modest WISPs: WISPs 1, 2, 3, 4 and 5 advertise $AQ = 60, 70, 80, 90$ and 99, respectively; but all of them offer $RQ = 80$. We consider the two following scenarios:

Scenario 2.1: All the WISPs ask for the same price. At the beginning of a simulation, we assign to each MN , with equal probability, one of the following applications: chat or file transfer.

Scenario 2.2: The WISPs ask for prices that are proportional to their QoSs ($P_W \sim RQ_W$). We expect the choice of the application to have an effect on the results, so we run 3 sets of simulations; one for each kind of application (i.e., all the nodes run that application).

Set 3: We assume that all the WISPs are honest, offer the same QoS and ask for the same price. At the beginning of a simulation, we assign to each MN , with equal probability, one of the following applications: chat or file transfer. We want to study the effect of the initial reputation of a WISP that opens its service. We assume that the newcomer is WISP 1 and we consider the three following scenarios:

Scenario 3.1: The initial reputation of WISP 1 equals the one of the other WISPs ($Rep_1 = maxRep = 100$ because the WISPs are honest).

Scenario 3.2: The initial reputation of WISP 1 is lower than the one of the other WISPs ($Rep_1 = 50$).

Scenario 3.3: The initial reputation of WISP 1 is lower than the one of the other WISPs ($Rep_1 = 50$) but WISP 1 asks for a lower price.

5.2 Simulation Results

We run 10 simulations for each of the scenarios listed in Subsection 5.1.5. Each WISP W is characterized by the triplet (AQ_W, RQ_W, P_W) (See the legend in Figures 3 to 11). The results are the following:

Set 1: The results of Scenario 1.1 show that if all the WISPs ask for the same price, almost all the users select the WISP that offers the best QoS (WISP 5 in Figure 3). The other WISPs (mainly WISP 4) can occasionally have some clients because the randomness introduced for the service provision at the WISPs (see Subsection 5.1.2) may lead to a slight decrease in WISP 5’s reputation.

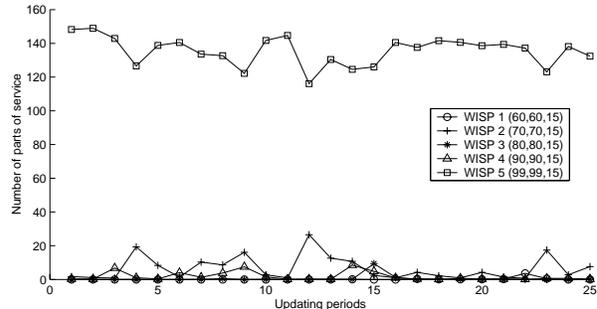


Figure 3: Scenario 1.1: All the WISPs are honest and ask for the same price. Therefore, WISP 5, which offers the highest QoS, eventually gets most of the users.

The results of Scenario 1.2 show that if all the WISPs offer different QoSs and ask for different prices, the choice of the users depends on the application they are running; e.g., if the nodes run a chat application (see Figure 4), the majority of the nodes choose the WISP 2 whereas if the nodes run a file transfer application (see Figure 5), the majority of the nodes choose the WISP 5 that offers the best QoS.

Note that in Scenario 1.2, nodes running the chat application do not choose WISP 1 even if it offers a lower price than WISP 2. By analyzing the data, we realized that this is because the reputation of WISP 2 is significantly higher

¹⁶We do not consider the case where $AQ = 100$ because such a perfect case is probably not possible in real life conditions.

than the one of WISP 1, which is caused by the randomness introduced, for the service provision, at the WISPs (see Subsection 5.1.2).

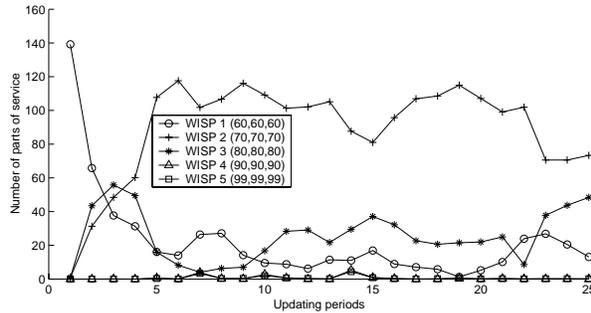


Figure 4: Scenario 1.2: All the nodes run a chat application. They choose WISP 2 which asks for a low price and at the same time has a good reputation.

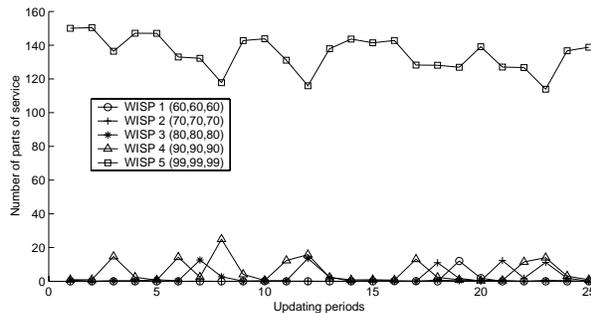


Figure 5: Scenario 1.2: All the nodes run a file transfer application. They choose WISP 5 because it offers the best QoS.

These results clearly prove that:

- the WISPs are encouraged to provide a good QoS and
- honest WISPs offering different QoSs can co-exist in the same network.

Set 2: The results of Scenario 2.1 show that if all the WISPs ask for the same price, most of the users select the WISP that offers the best real QoS (WISP 3 in Figure 6). Modest WISPs (here WISPs 1 and 2) and misbehaving WISPs (here WISPs 4 and 5) are selected much less often.

Note that the mobile nodes have no direct indication on the real QoS of the WISPs. They are however able to correctly evaluate the behavior of the WISPs because the correspondence between the advertised QoS and the real QoS is taken into consideration in the updating of the reputations.

The results of Scenario 2.2 show that almost all the nodes that run the chat application (see Figure 7) choose WISP 1 that offers the lowest price and at the same time

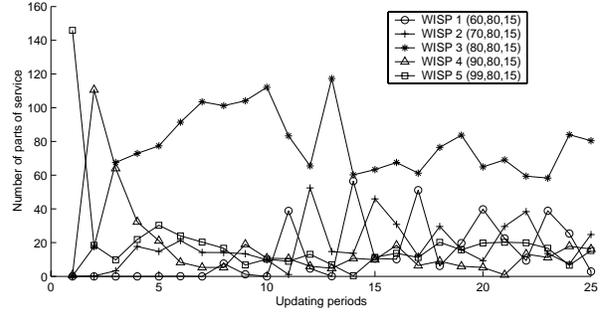


Figure 6: Scenario 2.1: All the WISPs ask for the same price. The only honest WISP, here WISP 3, eventually gets most of the users.

has a very good reputation. The majority of the nodes running a file transfer application (see Figure 8) choose WISP 3 because it offers the best real QoS.

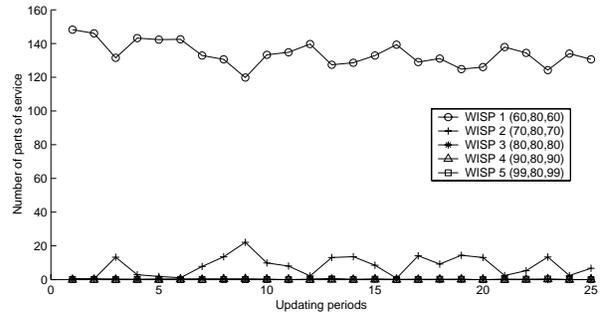


Figure 7: Scenario 2.2: All the nodes run a chat application. They choose WISP 1 because it asks for the lowest price and at the same time has a good reputation.

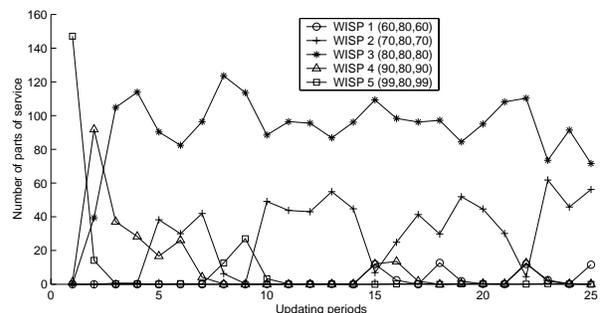


Figure 8: Scenario 2.2: All the nodes run a file transfer application. They choose WISP 3 because it offers the best real QoS.

These results clearly prove that the WISPs are discouraged from misbehaving (i.e., to advertise a QoS that is higher than the real QoS they can offer) and from being modest (i.e., advertising a QoS that is lower than the real QoS they can offer).

Set 3: In Scenarios 3.1 and 3.2, all the WISPs offer the same QoS and ask for the same price.

The results of Scenario 3.1 show that if WISP 1 has, when it opens its service, the same reputation as the other WISPs, it has more or less the same probability to get clients as others do (see Figure 9).

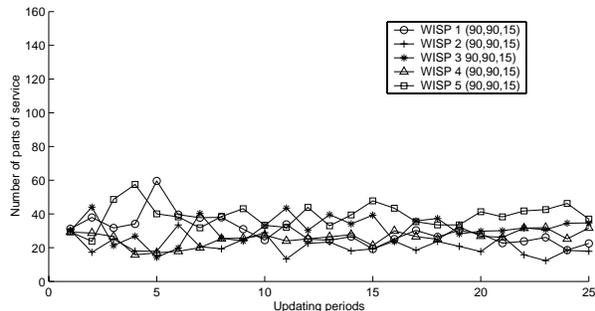


Figure 9: Scenario 3.1: WISP 1 has, when it opens its service, the same reputation as the other WISPs ($Rep = 100$); it has more or less the same probability to get clients as others do.

The results of Scenario 3.2 show if WISP 1 has, when it opens its service, a reputation that is lower than the reputation of all other WISPs, it has no chance to get clients. (see Figure 10).

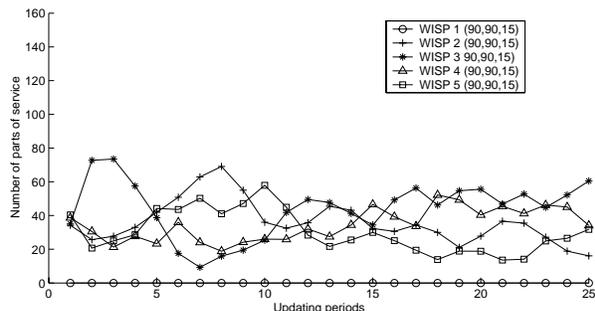


Figure 10: Scenario 3.2: WISP 1 has, when it opens its service, a lower reputation than for the other WISPs; it has no chance to get clients.

In Scenario 3.3, all the WISPs offer the same QoS and all of them, except WISP 1, ask for the same price; WISP 1 asks for a much lower price (3 times less than for the others). The results show that by decreasing the price it is asking for, WISP 1 can “reintegrate” the network and get the clients.

Note that even if according to the results WISP 1 gets almost all the clients, it is not interesting for it to keep the price very low because it will probably not cover its expenses; lowering the prices can therefore be considered a way of “launching” (if the initial reputation is not $maxRep$) or of “redemption” (if the WISP damaged its own reputation because it misbehaved).

These results clearly prove that:

- the initial reputation of the WISPs should be set to

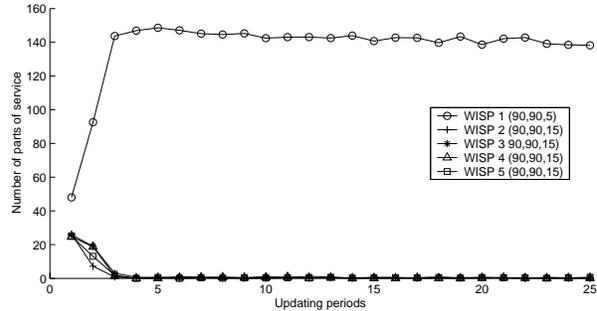


Figure 11: Scenario 3.3: WISP 1 has, when it opens its service, a lower reputation than for the other WISPs but it asks for much lower price; it eventually gets all the clients.

$maxRep$, not to oblige them to lower their prices¹⁷. If afterwards they do not offer a good QoS or if they misbehave, they will be punished as we showed in the previous scenarios.

- if the reputation of a given WISP decreases because it misbehaves, this WISP is still able to reintegrate the network. However this reintegration comes with a cost (i.e., asking for a price that is much lower than usual).

5.3 Prediction of the QoS offered by the WISP

In Subsection 2.1, we assume that S is able to evaluate the QoS it provides to the mobile nodes; in the simplest implementation, this QoS would be limited to the mean bitrate; more sophisticated solutions would consider additional parameters such as the provided peak rate, the maximum delay, and the maximum delay jitter; this would be notably the case with IEEE 802.11e [12]. Indeed, the proper operation of our protocols requires S to be able to predict the QoS that it will be able to offer (see the results of the second set of simulations in Subsection 5.2).

To the best of our knowledge, there is no well-established QoS “prediction” technique in CSMA/CA network. We propose the following, statistics-based solution: while it operates, S maintains:

- the history of its connections with the mobile nodes,
- the QoS it was able to offer to them, and
- the conditions under which this QoS provision was possible, such as (i) the number of MN s served simultaneously per hot spot; (ii) the number of neighboring access points (i.e., taking interference into account); (iii) the period of the day (e.g., peak hours,

¹⁷A WISP trying to cheat by changing its identity would be detected by the TCA (because it has to register with it each time).

etc.); (iv) the period of the year (e.g., working day, week-end, holidays, etc.).

Using this information, S predicts the QoS it can offer. It can then for example decide to what extent it wants to “overbook” itself. This QoS prediction can be combined to the use of a Differentiated Bandwidth Allocation similar to the one proposed in the CHOICE architecture [2].

6 State of the art

Reputation-based systems: These systems are mainly used to build trust and foster cooperation among a given community. The efficiency of reputation mechanisms have been widely studied in various fields and with different approaches. Studies such as [8, 17, 18] consider the effect of *online* reputation systems [6] on e-marketing and trading communities like e-Bay. Reputation mechanisms are also used to foster cooperation in peer-to-peer networks [7] or in ad hoc networks [4, 14].

But, from all these studies, we cannot draw a clear conclusion about the efficiency of reputation systems; each of these mechanisms should thus be analyzed on a per-case basis.

Roaming in WISPs: The deployment and success of WiFi networks is slowed down by the lack of interoperability between WiFi providers (also called *fragmentation* problem [15]): A client that has an account with a WISP A cannot connect to a hotspot managed by a WISP B . However, the situation is changing and more and more WISPs are establishing roaming agreements (similar to what is done for cellular networks). The roaming can be between providers within the same country (e.g., T-Mobile and iPass in the US) or international (e.g., between the British *BT* and the American *Airpath*).

Another solution would be to use the service of a *WiFi roaming operator* like *Boingo Wireless* [9]. Such an operator tries to solve the roaming problem by having agreements with as many WISPs as possible. Then it aggregates all the hot spots managed by these WISPs into a single (seamless) network. However, Boingo does not consider the problem of the variable QoS in WiFi networks.

In [16], Patel and Crowcroft propose a ticket based system that allows mobile users to connect to foreign service providers: The user contacts a *ticket server* to acquire a ticket, requests a service from a *service server* and uses the ticket to pay for that service. However, unlike the solution we present in this paper, the authors do not question the honesty of the

service providers i.e., they assume that the service providers provide the users with a good quality of service, which is far from being guaranteed in WiFi networks.

In [3], the authors consider also the problem of interoperability between the WISPs and use a reputation system to foster good QoS provision. However, their solution differs from ours in two main points. The first difference is the trust model: The authors consider that even if H is itself a WISP, it plays only the role of a home network and is trusted by all other parties. On the contrary, S is considered as rational (i.e., it can cheat if it is beneficial). We think that this assumption is inconsistent because H can be a home WISP for some nodes but, at the same time, a foreign WISP for other nodes; assuming that it will be rational and honest at the same time makes no sense. The second difference is in the content of the paper: In [3], the authors (i) present the rationale of the solution but do not present the details of the protocols; (ii) due to the absence of the security details, they present only a rough analysis of the security offered by their solution; finally, (iii) they do not evaluate their reputation system.

7 Conclusion

The work presented in this paper describes a simple solution that enables a mobile node to connect to a foreign wireless Internet service provider in a secure way while preserving its anonymity and meanwhile discouraging the WISPs from intentionally providing the mobile users with a bad QoS.

We have analyzed the robustness of our solution against different attacks and we have shown that our solution thwarts rational attacks, detects malicious attacks and identifies the attacker.

We have proved by means of simulations that the WISPs are encouraged to provide the MNs with a good QoS and, at the same time, discouraged from advertising a QoS that is different from the QoS they can really offer.

In terms of future work, we plan to study more in detail the prediction of the QoS the WISPs can offer to their clients and the cheating detection techniques. We also plan to investigate the feasibility of a “multi-hop WiFi network” (i.e., a WiFi network that is extended using multi-hop communications) in terms of network performance and security.

References

- [1] A. Acharya, C. Bisdikian, Y.-B. Ko, and A. Misra. ts-PWLAN : A Value-added System for Providing Tiered Wireless Services in Public Hot-spots. In *Proceedings of ICC*, 2003.
- [2] P. Bahl, A. Balachandran, A. Miu, W. Russell, G. Voelker, and Y.M. Wang. PAWNs: Satisfying the Need for Ubiquitous Connectivity and Location Services. *IEEE Personal Communications Magazine*, 9(1), 2002.
- [3] N. Ben Salem, J.-P. Hubaux, and M. Jakobsson. Felling WiFi deployment: A reputation-based solution. In *Proceedings of WiOpt*, 2004.
- [4] S. Buchegger and J.-Y. Le Boudec. Performance Analysis of the CONFIDANT Protocol: Cooperation Of Nodes - Fairness In Distributed Ad Hoc Networks. In *Proceedings of MobiHOC*, Lausanne, CH, June 2002.
- [5] L. Buttyán. Removing the Financial Incentive to Cheat in Micropayment Schemes. *IEE Electronics Letters*, January 2000.
- [6] C. Dellacrocas and P. Resnick. Online Reputation Mechanisms - A Roadmap for Future Research. In *1st Interdisciplinary Symposium on Online Reputation Mechanism*, 2003.
- [7] Z. Despotovic and K. Aberer. Trust and Reputation in P2P networks. In *1st Interdisciplinary Symposium on Online Reputation Mechanism*, 2003.
- [8] D. Houser and J. Wooders. Reputation in Auctions: Theory, and Evidence from eBay. Working Paper 00-01, University of Arizona, 2001.
- [9] <http://www.boingo.com/>.
- [10] <http://www.isi.edu/nsnam/ns/>.
- [11] <http://www.rsasecurity.com/products/secured/>.
- [12] IEEE 802.11 WG, Draft Supplement to Standard for Telecommunications and Information Exchange between Systems-LAN/MAN Specific Requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC), Enhancements for Quality of Service (QoS), 802.11e Draft 4.1, February 2003.
- [13] Arjen K. Lenstra and Eric R. Verheul. Selecting Cryptographic Key Sizes. *Journal of Cryptology: the journal of the International Association for Cryptologic Research*, 14(4), 2001.
- [14] P. Michiardi and R. Molva. Core: A Collaborative Reputation Mechanism To Enforce Node Cooperation In Mobile AD HOC Networks. In *Proceedings of The 6th IFIP Communications and Multimedia Security Conference*, Portoroz, Slovenia, September 2002.
- [15] Boingo Wi-Fi Industry White Paper. Towards Ubiquitous Wireless Broadband. http://www.boingo.com/wi-fi.industry_basics.pdf, September 2003.
- [16] B. Patel and J. Crowcroft. Ticket based Service Access for the Mobile User. In *Proceedings of MobiCom*, 1997.
- [17] P. Resnick and R. Zeckhauser. Trust Among Strangers in Internet Transactions: Empirical Analysis of eBay's Reputation System. In *NBER workshop on empirical studies of electronic commerce*, January.
- [18] P. Resnick, R. Zeckhauser, J. Swanson, and K. Lockwood. The Value of Reputation on eBay: A Controlled Experiment. In *ESA Conference*, June.
- [19] R. Rivest and A. Shamir. PayWord and MicroMint: Two simple micro-payment schemes. Technical report, MIT Laboratory for Computer Science, 1996.