

Naouel Ben Salem^a

Naouel.BenSalem@epfl.ch

Jean-Pierre Hubaux^a

Jean-Pierre.Hubaux@epfl.ch

Markus Jakobsson^b

Markus@indiana.edu

^aLaboratory of Computer Communications and Applications (LCA), EPFL, Switzerland

^bSchool of Informatics Indiana University at Bloomington, IN 47406, USA

Wi-Fi networks have a very strong potential: They are easy to deploy, they use unlicensed frequencies and they provide Internet connectivity that is several times faster than by cable modem. However, two major problems still need to be solved: the lack of a seamless roaming scheme and the variable quality of service experienced by the users. The reputation-based solution presented in this paper solves both problems: It allows a mobile node to connect to a foreign Wireless Internet Service Provider (WISP) in a secure way while preserving its anonymity and it encourages the WISPs to provide the users with good QoS. We analyze 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. We also propose a simple mechanism that allows the WISPs to predict the QoS they are able to offer to the (mobile) clients.

I. Introduction

The rapid growth of WiFi networks over the past years is due primarily to the fact that they solve several of the intrinsic drawbacks of cellular data services such as GSM/GPRS. These drawbacks are mainly the relatively low offered bitrates and the slow deployment of new features due to several factors such as the large size and the oligopolistic behavior of the operators, their willingness to provide homogeneous service, and the huge upfront investment. Therefore, 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, deployment based possibly on local initiative, higher competition, and much shorter time-to-market for new features. This may, in turn, pave the way for new types of services.

In recent years, wireless Internet service providers (WISPs) have established thousands of WiFi hot spots notably in cafes, hotels and airports. 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 lack of a good quality of service guarantee for the users.

This paper provides a response to these challenges. By appropriately unbundling the major functions of the network, our solution institutes a virtuous cycle of deployment and usage: Each WISP will be encouraged to deploy its network and will be confident that mobile users registered with other WISPs will pay for the service it provides them; 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 provide a detailed threat analysis and we show

*The work presented in this paper was supported (in part) by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005 – 67322

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

that the proposed protocols can thwart rational attacks and detect malicious attacks (we define these terms in Subsection III.B).

This paper is organized in the following way: In Section II, we present the state of the art and in Section III we present the system and trust models. In Section IV, we give an overview of the proposed solution and describe the details of the protocols. We study the security of the protocols in Section V and evaluate the overhead in Section VI. In Section VII, the reputation system is evaluated by means of simulations. Finally, we discuss the prediction the QoS in Section VIII and we conclude in Section IX.

II. 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 has been widely studied in various fields and with different approaches. Studies such as [13], [23] and [24] consider the effect of *online* reputation systems [9] on e-marketing and trading communities such as eBay. Reputation mechanisms are also used to foster cooperation in peer-to-peer networks [10] or in ad hoc networks [6, 19].

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 (this is also called the *fragmentation* problem [21]): A client that has an account with a WISP A cannot connect to a hot spot managed by a WISP B . This solution, however, 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 on international scale (e.g., between the British BT and the American Airpath).

Another solution would be to use the service of a *WiFi roaming operator* such as *Boingo Wireless* [14]. Such an operator tries to solve the roaming problem by having agreements with as many WISPs as possible. It then 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 [22], 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 of [22] 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. The same problem exists in the solution proposed by Zhang et al. [26].

In [11], Efstathiou and Polyzos present a Peer-to-Peer Wireless Network Confederation (P2PWNC) where the roaming problem is considered as a peer-to-peer resource sharing problem. They propose a solution where a WISP has to allow the foreign users to access its hot spots in order to allow its own users to connect to foreign WISPs' hot spots. This solution however presents the same problem as for [22], i.e., there is no guarantee of a good QoS provision.

In [3], we also considered the problem of interoperability between the WISPs and we used a reputation system to foster good QoS provision. But, the solution proposed in [3] differs in two main points from the solutions we present in [4] and in this paper. The first difference is the trust model: In [3], we consider that even if the home network H is itself a WISP, it plays only the role of a home network and it is trusted by all other parties; on the contrary, the selected WISP S is considered to be rational (i.e., it can cheat if it is beneficial). We think that this assumption should be relaxed 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 is inappropriate. The second difference is the definition and the evaluation of the solution: Compared to [3], here we present the details of the protocols, we offer a detailed security analysis of the solution and we evaluate the reputation system. Compared to [4], we added in this paper a section in which we propose a mechanism that can be used by the WISPs to predict the QoS they can offer to the mobile nodes.

III. System and Trust Models

III.A. 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 WISP that we denote by S (see Figure 1). MN is affiliated with

its home WISP² H 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 , however, we ensure MN 's anonymity with respect to a foreign WISP S (note that it is possible to have $S = H$).



Figure 1: System model.

In our model, all WISPs are registered with the trusted central authority (TCA) that creates for each of them a public/private key pair and a certificate of their public key and of their identity. We assume that TCA 's public key is known by all other entities. In a “grassroots” vision, TCA would be a federation of WISPs, who join forces to centralize a few strategic functions. In a more conventional vision, TCA can be under the control of a world-wide organization such as a quality control company, a certification company, or a global telecommunications operator. TCA servers can be distributed to avoid bottlenecks.

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. In our model, each WISP 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 VII.

III.B. Trust and Adversarial Model

We consider an attacker \mathcal{A} that wants to perform an attack against our protocols (see Section V for the list of attacks). \mathcal{A} can be a mobile node or a WISP. We assume that (i) TCA never cheats and is trusted by the other parties for all the actions it performs; (ii) the WISPs (here S and H) are rational and therefore they cheat (i.e., perform one of the attacks presented in Section V) only if it is to their advantage (e.g., in terms of money); and (iii) MN may be malicious and therefore it can cheat even if there is no gain from cheating

(this implicitly assumes that MN can also perform rational attacks). We also assume that MN trusts H to manage its account and that 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 such as DoS attacks based on jamming. However, we do design the protocols with DoS in mind, making sure that we do not expose protocol participants to unnecessary risks by relying on heavyweight operations.

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.

IV. Proposed Solution

IV.A. Rationale of the Solution

Our solution consists of four phases: *Session Setup*, *Service Provision and Payment*, *Session Closing* and *Reputation Update*.

Session Setup: When MN wants to connect to the Internet, it contacts all the neighboring WISPs³ and selects the WISP S that presents the best offer. The decision making is based, among other criteria, on the reputation records of the WISPs (see Subsection IV.C.1). Then, MN and S establish a secure session by setting up a symmetric key k_{MS} .

Service Provision and Payment: 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. In order to make sure that the mobile nodes pay for the service they receive, we use a credit-based micro-payment scheme: the PayWord scheme [25] (see Subsection IV.B.1).

Session Closing: At the end of the connection, the session is closed and MN reports on the QoS it received to TCA .

Reputation Update: TCA collects the feedback about the different WISPs, updates periodically the

²The solution works even if H does not operate hot spots itself.

³Note that we refer to the access points using the identities of the WISPs that are managing them.

reputation records according to the collected information, and provides the WISPs with their new reputation records.

IV.B. Basic Mechanisms

IV.B.1. Micro-payment scheme

As already mentioned in Section III, the payment scheme we use in this paper is the PayWord scheme [25]:

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 secure 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 session, *S* sends the last payment (w_ℓ, ℓ) it received to *H*. *H* verifies the validity of w_ℓ , pays *S* the amount corresponding to ℓ paywords and charges *MN* for that amount by updating its billing account.

We use this micropayment scheme because it allows for an offline verification of the payment proofs and because of its low computational and storage costs for the mobile nodes.

IV.B.2. Authentication of MN by H

As stated in Section III, all communication between *MN* and *H* goes through *S*. Therefore, in order to preserve the anonymity of *MN* with respect to *S*, we use the following authentication mechanism, which is commonly used in the industry (e.g., SecurID [16]): When *MN* gets affiliated with *H*, the two parties share a random seed s that represents the input to a pseudo-random 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.

IV.C. Details of the Protocols

IV.C.1. Session Setup Phase

This phase consists of three steps (see Figure 2): *Selection of the WISP*, *Authentication of MN* and *Secure session establishment*.

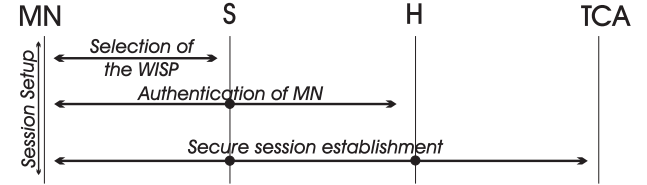


Figure 2: Session setup

Selection of the WISP: When *MN* wants to obtain Internet access, it scans the spectrum, contacts the neighboring WISPs and asks for 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 $Offer_W$:

$$W \rightarrow MN : Offer_W, S_{pk_W}(Offer_W, OfferReq) \quad (2)$$

where

$$Offer_W = [W, RR_W, AQ_W, P_W, Cert(W), n_W]$$

RR_W is the most recent *reputation record* of *W* (signed by *TCA*), AQ_W is the QoS it advertises⁴, P_W is the price it is requesting for each part of the service (see Subsection IV.C.2), pk_W is its private key and $Cert(W)$ is the certificate of its public key PK_W .

For each offer $Offer_W$, *MN* verifies the freshness of n_W and computes a value⁵ $Decision_W = RR_W^\alpha \cdot AQ_W^\beta \cdot P_W^{-\gamma}$, where the exponents α , β and γ are parameters that depend on the application *MN* is running⁶. Then, *MN* determines $Decision_S = \max_W \{Decision_W\}$, selects the WISP *S* and verifies its certificate and the

⁴The estimation of the QoS offered by *W* is discussed in Section VIII.

⁵The decision function given here is an example; it can be any function $f(RR_W, AQ_W, P_W)$.

⁶We can have for instance $(\alpha, \beta, \gamma) = (2, 1, 3)$ for chat applications to put the emphasize on low price offers and $(\alpha, \beta, \gamma) = (2, 2, 1)$ for file transfer applications to put the emphasis on QoS. The decision function being exponential amplifies the difference between these two cases.

signature of its offer. If the verification is incorrect, MN checks the second best offer and so on. We denote the selected WISP by S .

Authentication of MN : 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 IV.B.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 key⁷ k_{HM} , the identities of MN and S , the tag 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 IV.B.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 indeed the WISP 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 (i) in clear, a fresh temporary identifier TID for MN (TID will be used during service provision), (ii) TID , n_M , and k_{MS} encrypted using the symmetric key k_{HM} , and (iii) 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 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).

Secure Session Establishment: MN generates a long hash chain of $n + 1$ elements, computed from a randomly chosen seed w_n as described in Subsection IV.B.1. Then MN generates a contract $C = [CID, w_0, RR_S, AQ_S, P_S]$ 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:

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

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

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

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

(9) MN generates a fresh symmetric key k_{MT} that MN will use later to encrypt data for TCA (see Subsection IV.C.3). Then, MN encrypts⁸ C and k_{MS} using

⁷ 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.

⁸In order to prevent the key retrieval by an attacker, MN can use a probabilistic encryption algorithm, e.g. RSA-PSS [8], RSA-OAEP [2], or ElGamal [12].

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

(10) S verifies the MAC 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 TCA 's signature and forwards the MAC to MN .

IV.C.2. Service Provision and Payment

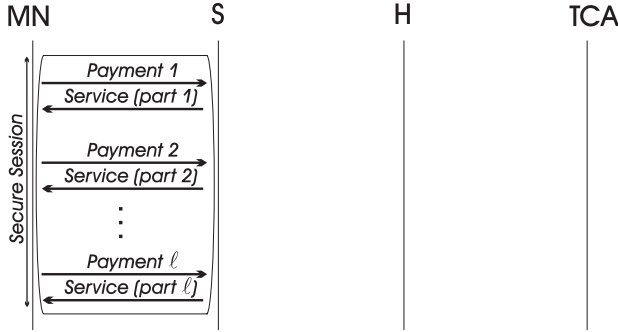


Figure 3: Service provision and payment

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

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

$$S \rightarrow MN : i^{th} \text{ part of the service} \quad (14)$$

(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. Note that the data packets corresponding to the i -th service are cryptographically protected using the key k_{MS} (e.g., the key is used to encrypt the packets if privacy is required and to compute a MAC if authentication is required).

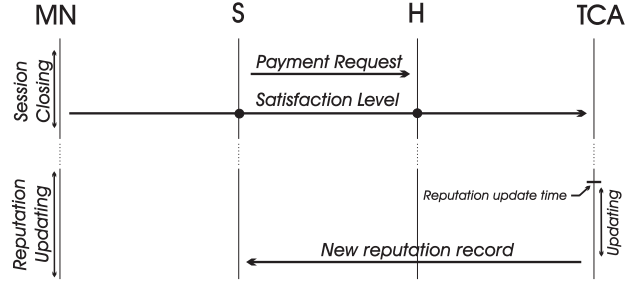


Figure 4: Session closing and the reputation update

IV.C.3. Session Closing and Reputation Update

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 CID , the identity of S so that H is able to retrieve the symmetric key k_{HS} .

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

Upon receipt of PR , H verifies the validity of w_ℓ as explained in Subsection IV.B.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 IV.D).

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

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

$QoSEval_{S,CID}$ is MN 's estimate of the compliance of the obtained QoS with the announced one and k_{MT} is the key MN shares with TCA .

Then, MN sends its *satisfaction level* to TCA :

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

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

(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 $QoSEval$ 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.

TCA collects the information about the satisfaction levels for a given period and then, at the *reputation update time*, it 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 Subsection VII.

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 IV.C.1). 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.

IV.D. Charging and Rewarding Model

In this subsection, we provide additional details regarding the charging and rewarding model:

- 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 $X = P_S * \ell$ corresponding to the ℓ parts of the service and rewards S , using a well-established e-payment technique, the amount⁹ $X - \varepsilon$. 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 or if MN pays a specific amount per session.

⁹As already mentioned in Subsection IV.C.3, ε is the reward MN receives if it reports on its satisfaction level to TCA .

V. Security Analysis

In this section, we analyze the robustness of our protocols against various attacks against our protocols (see Subsection III.B for the trust and adversarial model). We identify eight attacks that are specifically targeted against our solution: *Publicity*, *Selective Publicity*, *Denigration*, *Flattering*, *Report Dropping*, *Service Interruption*, *Refusal to Pay* and *Repudiation* attacks. We also consider more general attacks such as *Packet Dropping*, *Filtering* and *Replay* attacks.

V.A. Specific Attacks

Publicity Attack: In this attack, S advertises a QoS that is higher than the real QoS it can offer. As a reaction, MN will send a negative report to TCA at the end of the session. 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: In this attack, S attempts to perform the Publicity attack with a specific MN . However, 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: In this attack, MN receives a good QoS from S but pretends the contrary by sending a negative report or no report at all.

If no report is sent, H will not give MN the ε reward and TCA will consider the absence of feedback as negative feedback. Therefore, this attack is not rational for MN . Therefore, 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 feedback. 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 MNs . This attack is rational if the MNs belong to a competitor that wants to affect the WISP's

reputation. However, *TCA* can statistically detect it if the following events happen frequently:

1. The *MNs* affiliated with *H* always pretend that they received a bad QoS (i.e., lower than the advertised QoS) from a given WISP, whereas many other *MNs* report on a good QoS from that very WISP. As the selective publicity attack is not possible, this situation is suspect and *TCA* may punish *H*, e.g., by downgrading its reputation record.

2. *TCA* never receives reports from *MNs* affiliated with *H* about the sessions they established with *S*.

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 hot spots) - if the attacker pays the price - but it is much too costly against WISPs with hundreds of hot spots.

Flattering Attack: In this attack, *MN* sends systematically a good feedback about *S*'s behavior to *TCA*. This attack makes sense particularly if $S = H$; it significantly 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 resides 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: In this attack, *MN* sends the report but *S* does not transmit it to *TCA* (e.g., because *S* expects a negative feedback). However, as the absence of feedback counts as the lowest possible feedback, this dropping does not help *S*: 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: In this attack, *S* receives the *i*-th payment proof from *MN* but does not provide the corresponding part of the service. *MN* will then 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 also affects the satisfaction level of *MN*. If nevertheless, we want to prevent *S* from receiving the *i*-th payment without providing the *i*-th service, we can use the solution proposed in [7].

Refusal to Pay Attack: In this attack, *MN* does not send the *i*-th payment to *S*. Then, *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: In this attack, *S* or *MN* retracts the agreement it has with other party (e.g., *S* asks for higher price than agreed upon when the contract *C* was established). This attack is not efficient 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 incorrect information to *H*, we can also require a response from *H* to establish the session.

V.B. General Attacks

Packet Dropping Attack: In this attack, *A* drops a message it is asked to forward or discards a message it is asked to generate and send. If this is done during session setup, the secure session will not be established. If $A = MN$ (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 $A = S$, it will not provide 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 session is closed.

If *S* does not forward the message *SI* of *MN*, it is equivalent to the denigration attack (see Subsection V.A).

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: In this attack, *A* modifies a packet it is asked to forward or generate. However, the messages exchanged between the different parties in our protocols are cryptographically protected, using MAC computations or digital signatures. There-

fore, 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: In this attack, \mathcal{A} replays a valid message that was exchanged between two parties.

During session setup, the messages exchanged between the different entities (Messages (2) to (6)) are protected using nonces; delayed messages are easily 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 and the satisfaction level (Messages (15), (17) and (18)) are expected only once; a replay is immediately detected and discarded.

VI. 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 address all the wireless aspects of the communications.

VI.A. Computation Overhead

During the different phases of our protocols, we use symmetric and public key cryptography primitives to secure the message exchange and to 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.

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 cryptography primitives: $5 + 2\ell$ MAC operations (ℓ being the total number of service parts), 2 encryptions and 1 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 convert 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 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} .

VI.B. Communication Overhead

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

Field Name	ReqID	IDs	$n_{M, \text{pad}}$	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 16 bytes long (assuming an IPv6 format for example). The paywords w_i are 20 bytes long (e.g., assuming SHA) and the QoS (AQ and $QoSEval$), the reputation R and the price P are encoded on 1 byte each (which is enough to encode values between 0 and 100). The symmetric keys k_{HM} , k_{HS} , k_{MS} and k_{MT} are 16 bytes long (128 bits) and the public keys are 150 bytes long (e.g., assuming RSA, see [18]). We encode the nonces (n_M and n_W) and the pads on 20 bytes, the *tag* on 6 bytes (see Subsection IV.B.2) and the MACs on 16 bytes. Finally, we use a sequence number ℓ that is 2 bytes long to support long sessions.

We consider the example where MN is downloading a 1 MB file. We assume that 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).

VII. Reputation System Assessment

Our solution motivates the different players to participate in the reputation mechanisms. Indeed (i) S 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 V.A), (ii) MN is motivated to report on its interaction with S because it receives a refund ε and (iii) S is motivated to forward the report (see the analysis of the report dropping attack in Subsection V.A).

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 the ns-2 simulator [15].

VII.A. Simulation Environment

We consider a static¹⁰ network of 5 WISPs, numbered from 1 to 5, and 50 MNs. Each WISP is a home WISP for 10 MNs. Each WISP W is characterized by a triplet (AQ_W, RQ_W, P_W) where AQ_W is the QoS advertised by W , RQ_W is the real QoS provided by W and P_W is the price W is asking for. We consider that a WISP W is *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., $RQ_W < AQ_W$) and *modest* if it advertises a QoS that is lower than the real QoS it is offering (i.e., $RQ_W > AQ_W$).

We initialize the reputation of the WISPs to $maxRR = 100$. At the end of each session, MN sends to TCA its *satisfaction level* $Sl = [E_{k_{MT}}(CID, QoSEval_{W,CID}, w_\ell, \ell)]$, where $QoSEval_{W,CID} = \frac{RQ_W}{AQ_W}$.

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

$$RR_W(t+1) = \lambda \cdot RR_W(t) + (1 - \lambda) \cdot \frac{feedback_W}{nbS_W}$$

where $RR_S(t)$ is the current reputation of W , nbS_W is the number of sessions established by W (and already closed) during the last 2000 seconds and $feedback_W$ is the sum of all $QoSEval_{W,CID}$ received over all these sessions (the absence of feedback is considered as $QoSEval_{W,CID} = 0$). 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., $AQ_W < RQ_W$), we will have $QoSEval_W > maxRR$, which may lead to a new reputation that is also higher than $maxRR$. If it is the case, TCA keeps $RR_W(t+1)$ as it is in its database but sends to S a new reputation record equal to $maxRR$.

VII.B. Studied Scenarios

We want to study the reaction of the network to the co-existence of honest, misbehaving and modest WISPs in the network: WISPs 1, 2, 3, 4 and 5 advertise

¹⁰All MNs are within the power range of all WISPs, it is therefore useless to consider mobility in this case.

$AQ=60, 70, 80, 90$ and 99 , respectively; but all of them offer $RQ = 80$. We assume that the values of AQ and RQ remain constant and are independent from the number of MNs that are simultaneously connected to the WISPs¹¹. We consider the two following scenarios:

Scenario 1: All the WISPs ask for the same price¹² $P = 15$. 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: The WISPs ask for prices that are equal to their QoSs ($P_W = AQ_W$). We expect the choice of the application to have an effect on the results, so we run 2 sets of simulations; one where all the nodes run a chat application i.e., $(\alpha, \beta, \gamma) = (2, 1, 3)$ and the other where they run a file transfer application, i.e., $(\alpha, \beta, \gamma) = (2, 2, 1)$ (see Subsection IV.C.1).

VII.C. Simulation Results

We run 10 simulations for each of the scenarios and plot the average value. The results for Scenario 1 show that if all the WISPs ask for the same price, most of the users select the WISP with the advertised QoS that corresponds best to the real QoS it offers (WISP 3 in Figure 5). Due to their good reputation, modest WISPs (here WISPs 1 and 2) perform better than misbehaving WISPs (here WISPs 4 and 5) but are still selected much less often than the honest WISP. Indeed, among the WISPs that have good reputations (WISPs 1, 2 and 3), WISP 3 is the one offering the best QoS and thus is selected more often. Therefore, the best strategy for the WISPs is to be honest.

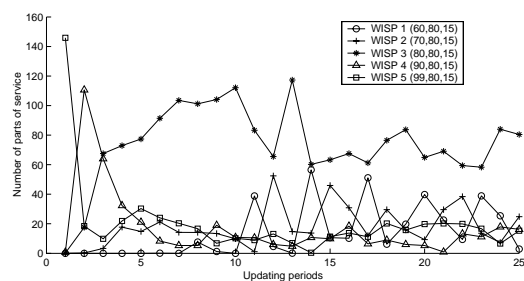


Figure 5: Results for Scenario 1.

The results for Scenario 2 show that almost all the nodes that run the chat application (see Figure 6)

¹¹The case where these values vary is studied in Section VIII.

¹²MN selects the WISP according to the decision formula presented in Subsection IV.B.2. In this formula, if all the WISPs ask for the same price, the weight of the price parameter is the same for all WISPs. The value assigned to the price is therefore not important; we arbitrarily set it to 15.

choose WISP 1, which asks for the lowest price and at the same time has a very good reputation. The majority of the nodes running a file transfer application (see Figure 7) choose WISP 3 because it offers the best real QoS: Therefore, WISPs offering different QoSs can co-exist in the same network because nodes with different needs (i.e., different α , β and γ coefficients) choose different WISPs.

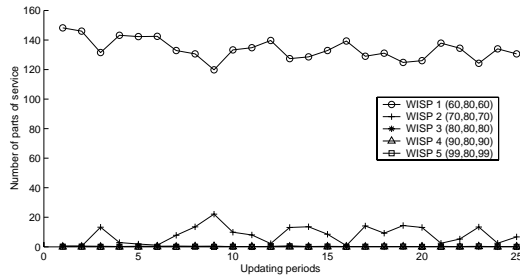


Figure 6: Results for Scenario 2 (chat).

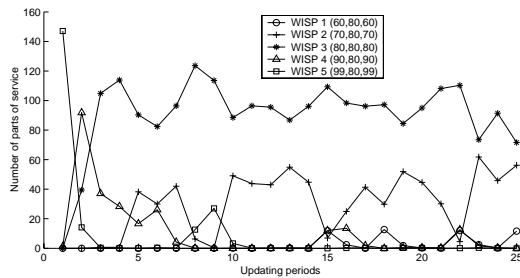


Figure 7: Results for Scenario 2 (file transfer).

VIII. Prediction of the QoS Offered by the WISP

The results of Scenario 1 show that the WISPs are encouraged to be honest. However, this requires each WISP to accurately “predict” the QoS it can offer to its clients. This prediction depends on several parameters such as the number of neighboring WISPs, the number of clients that are simultaneously connected in the neighborhood, the clients’ arrival rate, etc.

In this section, we propose the following simple prediction mechanism that consists of three main steps: (i) the estimation of the number of clients expected in the network during the next period of time, (ii) the computation of the total throughput expected in the network during the next period of time, and (iii) the definition of the prediction strategy.

VIII.A. Estimation of the Number of Clients

During this phase, a WISP W has to estimate, for the next period of time, the number of mobile clients that will be served in its neighborhood. This estimation has to take into consideration three main parameters:

(i) The length of the estimation period, i.e., the period of time for which the estimation is done (e.g., the next 15 minutes, the next hour).

(ii) The period of the day (e.g., peak hours, etc.) or of the year (e.g., working day, week-end, holidays, etc.) during which the estimation period is considered. This parameter gives an idea about the expected traffic.

(iii) The length of the history maintained by the WISPs. Indeed, while it operates, each WISP maintains the history of the number of clients simultaneously served in the neighborhood, the duration of the connections, the clients’ arrival rate, the duration of the connections, etc. A longer history leads to a better estimation.

VIII.B. Computation of the Total Throughput

During this phase, W computes the total throughput expected in the network during the next estimation period. This value can be computed using the number of clients simultaneously served in the neighborhood (estimated in the previous phase) and Bianchi’s throughput performance evaluation formula [5] (see Figure 8).

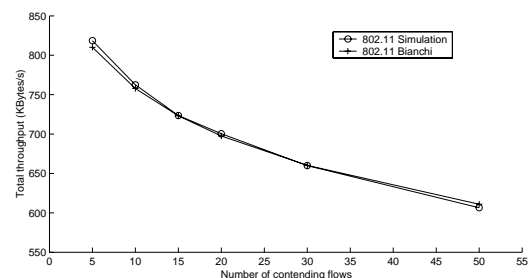


Figure 8: The total throughput obtained using Bianchi’s throughput performance evaluation formula [5]. Bianchi’s throughput is very close to the throughput we get by means of simulations.

VIII.C. Definition of the Prediction Strategy

Each WISP considers the value of the total throughput computed during the previous phase and decides the QoS it will advertise and to what extent it wants to “overbook” itself. The efficiency of a given strategy

depends on several parameters such as the duration of the connections and the clients' arrival rate. We cannot compare these strategies as they may perform differently in different circumstances: A strategy that performs well in case of short and frequent connections may perform poorly when the connections become long and sporadic. Therefore, the WISPs may consider using different strategies according to the situation.

IX. Conclusion

In this work, we describe a simple solution that enables a mobile node to connect to a foreign WISP in a secure way while preserving its anonymity and encouraging the WISPs to provide the mobile users with a good QoS. Our solution takes into account the fact that the mobile clients are resource restrained mobile device and therefore have much less computing and storage resources than *TCA*, *H* or *S*.

We have analyzed the robustness of our solution against different attacks and we have shown that our protocols thwart rational attacks, detect malicious attacks and can help identify 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 that they can really offer.

Acknowledgement

The authors would like to thank Prof. Yves Pigneur for his helpful discussions and comments, Dr. Imad Aad for helpful discussions on admission control in 802.11 and Márk Félegyházi for useful feedback.

References

- [1] 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.
- [2] M. Bellare and P. Rogaway. Extended abstract in Advances in Cryptology. *Proceedings of EUROCRYPT*, 1995.
- [3] N. Ben Salem, J.-P. Hubaux, and M. Jakobsson. Fuelling WiFi deployment: A reputation-based solution. In *Proceedings of WiOpt*, 2004.
- [4] N. Ben Salem, J.-P. Hubaux, and M. Jakobsson. Reputation-based Wi-Fi Deployment - Protocols and Security Analysis. In *Proceedings of WMASH*, 2004.
- [5] G. Bianchi. Performance Analysis of the IEEE 802.11 Distributed Coordination Function. In *IEEE Journal on Selected Areas in Communications*, 18(3), 2000.
- [6] 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*, 2002.
- [7] L. Buttyán. Removing the Financial Incentive to Cheat in Micro-payment Schemes. *IEE Electronics Letters*, January 2000.
- [8] J.-S. Coron. Optimal security proofs for PSS and other signature schemes. *Proceedings of EUROCRYPT*, 2002.
- [9] C. Dellacrocas and P. Resnick. Online Reputation Mechanisms - A Roadmap for Future Research. In *1st Interdisciplinary Symposium on Online Reputation Mechanism*, 2003.
- [10] Z. Despotovic and K. Aberer. Trust and Reputation in P2P networks. In *1st Interdisciplinary Symposium on Online Reputation Mechanism*, 2003.
- [11] E.C. Efstathiou and G.C. Polyzos. A Peer-to-Peer Approach to Wireless LAN Roaming. In *Proceedings of WMASH*, 2003.
- [12] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *IEEE Transactions on Information Theory*, 31(4), 1985.
- [13] D. Houser and J. Wooders. Reputation in Auctions: Theory, and Evidence from eBay. Working Paper 00-01, University of Arizona, 2001.
- [14] <http://www.boingo.com/>.
- [15] <http://www.isi.edu/nsnam/ns/>.
- [16] <http://www.rsasecurity.com/products/securid/>.
- [17] IEEE 802.11 WG, Draft Supplement to Standard for Telecommunications and Information Exchange between Systems-LAN/MAN Specific Requirements- Part 11: Wireless LAN MAC and Physical Layer (PHY) Specifications: Medium Access Control (MAC), Enhancements for QoS, 802.11e Draft 4.1, February 2003.
- [18] A. K. Lenstra and E. R. Verheul. Selecting Cryptographic Key Sizes. *Journal of Cryptology: the journal of the International Association for Cryptologic Research*, 14(4), 2001.
- [19] 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*, 2002.
- [20] A. Miu and P. Bahl. Dynamic Host Configuration for Managing Mobility between Public and Private Networks. In *The 3rd Usenix Internet Technical Symposium*, 2001.
- [21] Boingo Wi-Fi Industry White Paper. Towards Ubiquitous Wireless Broadband. <http://www.boingo.com/wi-fi-industry-basics.pdf>, 2003.
- [22] B. Patel and J. Crowcroft. Ticket based Service Access for the Mobile User. In *Proceedings of MobiCom*, 1997.
- [23] 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*, 2001.
- [24] P. Resnick, R. Zeckhauser, J. Swanson, and K. Lockwood. The Value of Reputation on eBay: A Controlled Experiment. In *ESA Conference*, 2002.
- [25] R. Rivest and A. Shamir. PayWord and MicroMint: Two simple micro-payment schemes. Technical report, MIT Laboratory for Computer Science, 1996.
- [26] J. Zhang, J. Li, S. Weinstein, and N. Tu. Virtual Operator Based AAA in Wireless LAN Hot Spots with Ad Hoc Networking Support. *Mobile Computing and Communications Review*, 6(13), 2002.