

Who said that? Privacy at link layer.

Frederik Armknecht and Joao Girao
NEC Europe Ltd.

{frederik.armknecht, joao.girao}@netlab.nec.de

Alfredo Matos and Rui L. Aguiar

Institute of Telecommunications, University of Aveiro
alfredo.matos@av.it.pt, ruilaa@det.ua.pt

Abstract—Wireless LAN and other radio broadcast technologies are now in full swing. However, the widespread usage of these technologies comes at the price of location privacy, be it by observing the communication patterns or the interface identifiers. Although a number of network level solutions have been proposed, this paper describes a novel approach to location privacy at the link layer level. We present a generic mechanism and then map it to a real protocol, IEEE 802.11. The work also provides an analysis of the protocol in terms of privacy and performance considerations.¹

I. INTRODUCTION

Wireless networks, in particular WLAN, have become popular in our homes and offices. The term ‘hotspot’ is now widespread as we walk the path to the ‘always connected’ paradigm. However, there are drawbacks. In this paper we address the loss of privacy that stems from the fact that we are always connected to the network, as we change locations, and can therefore be tracked. Since this is a problem which has only recently been identified, not much work has been done specifically in this area, although many of the new network architectures have support for location privacy by either using pseudonyms, making use of Network Address Translation (NAT) or other solutions (Section II). Network approaches prevent a node from tracking the location of peers beyond the link scope, but leave open the problem of two nodes communicating on the same link layer domain. This problem is most evident in broadcast mediums, mostly wireless technologies, where an attacker does not even have to participate in the communication to monitor all the necessary information. Finally, we would like to remark that our approach is best combined with other network layer approaches to location privacy in order to extend the protection to attacks originating outside the link layer cloud.

We provide a generic mechanism to deal with location privacy issues at the link layer. We further analyze the feasibility of a real case realization with an instantiation on 802.11, in light of current technology and standards. Since our approach is between the physical and link layers, we also study what we can and cannot protect in contrast with the standard.

We begin by presenting a survey of techniques used at different layers of the network stack for location privacy in Section II. In Section III we formalize the description of our network model, analyze the attacker model and provide the security objectives for our scheme along with an analysis

¹The work described in this paper is based on results of IST FP6 Integrated Project DAIDALOS. DAIDALOS receives research funding from the European Community’s Sixth Framework Programme. Apart from this, the European Commission has no responsibility for the content of this paper. The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

on leaked information by using 802.11. In Section IV we present our scheme and instantiate the model in which we apply our ideas. Section V provides some considerations on performance, and studies the feasibility of implementing the described scheme. We conclude our work in Section VI and open the door to future research directions.

Note that throughout this paper certain operations are replaced by symbols summarized in Table I.

Symbols	Description
$c = E_k(m)$	encryption of m using key k , obtaining the ciphered text c
$m = D_k(c)$	decryption of c using key k , obtaining plaintext message m
$a b$	concatenation of a with b
$ m $	size of message m

TABLE I

NOMENCLATURE USED THROUGHOUT THE ARTICLE

II. RELATED WORK

Most of the current solutions for location privacy operate at the network level, even though location privacy issues also afflict the MAC Layer. The privacy schemes at the network layer usually separate the concepts of identifier and locator, currently aggregated in IP addresses (eg. [7], [8]).

From a mobility standpoint, Mobile IP [4], [11] can provide location privacy through sub-optimal routing. However, the performance cost of using the Home Agent for all the traffic, without Route Optimization, is too big for most real-time applications.

Some architectures attempt to create new paradigms for routing and addressing, like FARA[3], which uses a completely new architecture using new abstractions that decouple Locators and Identifiers. While not directly aiming at location privacy, the logical abstractions provide support for it.

Overlay networks, such as I3 [13], Triad [2] and HI3 [9] can provide location privacy. In I3 and HI3, this is obtained by introducing a rendezvous point that allows nodes to reach their peers without knowing their locators. This approach can impact performance, since routing is not guaranteed to be optimal. Triad creates an overlay network to protect application content, and so has a different scope, but still provides an interesting approach defining realms and relay points.

A generic framework for location privacy is presented in [6]. This proposes a hierarchical approach to location privacy, addressing anonymity, pseudonymity and unlinkability. While this generic framework provides several improvements on privacy aspects, it does not protect against MAC layer tracking, and the unlinkability and pseudonymity it provides can be broken by eavesdropping on a wireless link.

All the presented solutions work either on or above the network layer. None of them address location privacy concerns

at the Link Layer (L2), and are therefore vulnerable to L2 attacks.

Note that, although our work addresses L2 attacks, there are highly specialized physical attacks which are not covered by our approach. In radio based technologies attacks may rely on the physical characteristics of the radio channel. Such attacks include: finding the nearest station and triangulation/trilateration, by analyzing the signal strength, signal to noise ratio (SNR), time difference of arrival (TDOA) or received signal strength indication (RSSI) and radio-frequency (RF) fingerprinting. These attacks may erode the protection we offer in this paper and therefore our approach should be taken in conjunction with techniques which also protect the physical layer (PHY), even though these attacks usually require non-standard equipment.

III. COMMUNICATION MODEL

A. Network Model

In our model we consider the last hop of an access network composed by one Access Point, AP and n terminals/nodes, N_i , with $i \in 1, 2, \dots, n$. We further extrapolate the individual links between each of the terminals N_i with the AP and term them as channels, where C_i refers to the channel between node N_i and the AP . Each channel has only two endpoints, which are the address of N_i , MAC_i , and of the AP , MAC_{AP} . Our assumption is that, even though communication occurs only between the individual N_i and the AP , the medium is still broadcast and all the nodes in the group can listen to all of the messages being sent over any C_i .

The model only considers communication between N_i and AP , which is in line with the traffic pattern at link layer in, for example, managed WLAN.

B. Attacker and Threat Model

In Section III-A we introduced a network model which we believe reflects typical scenarios. It seems clear that the first threat to location privacy at the link layer is that of the attacker having access to all the packets exchanged in all channels C_i .

An attacker may track a device from one network to the other by moving inside the same link layer cloud and mapping the unchanging MAC address. By associating link and network layer addresses (e.g. MAC and IP) he will further be able to circumvent any layer 3 location privacy protections. A pseudonymity approach will protect the identity of the user but not his location: passive attackers can still detect whether the MAC pseudonym which maps to a specific layer 3 identifier was already being used in the same link layer cloud.

Another issue is the tracking of origin and destination of link layer messages. Currently it is easy for an outsider to determine traffic patterns and traffic direction. This information can later be used to pinpoint a user, or correlated with other information to discover the user's identity (e.g. periodically checking an IMAP server). Even if the payload is encrypted, this information allows an attacker to perform selective Denial of Service (DoS) attacks.

Due to the nature of the scenario, DoS is a problem we also plan to tackle inside our proposal. Our proposal must be scalable and immune to DoS attacks.

Also, for DoS prevention reasons, it is important for the AP to be able to distinguish valid, authorized users even from each other, even at registration time. This is to prevent multiple

registrations and over consumption of the AP's resources. Although the problem of anonymously linking identity to a form of certification is outside the scope of this paper, we provide the mechanisms which allow the linkage of, for example, participation certificates which can be checked against a Public Key Infrastructure (PKI) and the Authentication, Authorization and Accounting (AAA) servers for validity and uniqueness. Such a combination would thwart attempts of multiple registrations on the behalf of the same user.

C. Security Objectives

Based on our network and attacker models, we list below the security objectives a successful location privacy approach for link layer should achieve:

- Avoid using a unique link layer identifier: Using the same identifier allows an attacker to track the user's location by testing the user's presence in different link layer clouds.
- Prevent linking network layer location with link layer identifiers.
- Protect communication peer identities and pseudonyms from traffic and header analysis.
- Protect users' traffic from direction inference: distinguishing traffic direction (from the AP to the terminal or vice-versa) allows an attacker to infer which service is being used and possibly the user's identity.
- Should support link layer protocol operations to minimize changes to standards and implementation costs: The feasibility of our approach depends on the intrusiveness into the link layer protocol.
- Ideally, when presented with several packets in the network, the attacker should not be able to link them or even distinguish anything other than the fact that they are disjoint packets.

D. Privacy Leakage

In most widely used protocols, Layer 2 addresses used to identify the nodes are sent in every packet. Furthermore, a channel identifier is sometimes used and, although it cannot be used to identify the node, it aids in tracking connections. Other potentially leaked information includes sequence numbers, acknowledgement frames and round-trip times, all of which can be correlated, hence tracking the connection and the user.

Each protocol requires careful analysis in order to determine if we can hide or otherwise obfuscate the offending fields. In our study case, IEEE 802.11 [1] requires that information is both hidden and obfuscated. It carries the source and destination addresses in the header, requiring procedure for association and authentication, that identify the stations. Also, some mechanisms identify the origin of the packet, infrastructure or not, and the destination, regular stations, as is the case of power management bits. Sequence numbers are not reset, so they enable to track nodes, neglecting the addresses. But, some fields cannot be hidden to support mechanisms such as the Network Allocation Vector. These types of fields must be world readable.

IV. WHO SAID THAT?

Our privacy proposal defines a novel transport that protects the data and management frames against the described attacker model, assuming that keys have previously been agreed between N_i and AP. When used in parallel with classical

networks, the node might obtain this key by, for example, contacting his home network. A key agreement phase should only be necessary if the terminal, N_i , does not have another secure way of agreeing on a key with the AP .

A. Transport

In our approach the identification of a channel C_i is given by the key K_i shared between the terminal N_i and the AP , as shown in Fig. 1. Thus the MAC is sent encrypted.

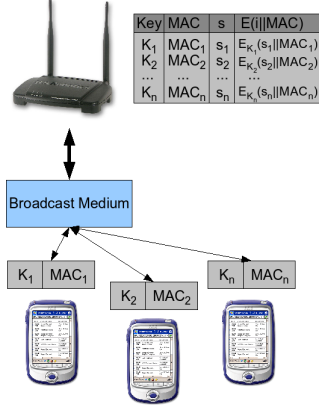


Fig. 1. Using keys as channel identifiers in a broadcast medium.

Even with encryption, if the encryption primitive does not provide randomization by itself, the encryption of the same plaintext results in the same ciphered text. For this reason we propose the use of a synchronized initialization vector (iv). The fact that the value of iv is synchronized on both end-points allows for fast determination algorithms of the origin and/or destination of the packet since both end-points can pre-compute the encrypted values. Also, with this vector iv , the encryption of the same plaintext results in different ciphertexts depending on the value of iv .

The main problem lays in the synchronization of both end-points, which is an expensive operation. To this end, we propose iv to be a sequence number, s_i , which is unique to a certain channel C_i . Since messages can be lost and this will affect synchronization, we further propose a mechanism to recover from such cases. This recovery mechanism also allows any N_i or the AP to re-seed the iv at any moment during the communication.

1) *Encryption Process*: The current value of s_i is appended at the end of every sent message. Encryption is then performed from the end to the beginning using the key K_i to the corresponding channel C_i , with the exception of the fields for fast determination. These fields contain the known plaintext values which are used to determine whether the packet should be processed or not, by a node which receives the packet (in most cases this field corresponds to the source or destination address). This field is encrypted in parallel by appending it to the value of s_i , padding the value to the block of the cipher and encrypting. We then discard the encrypted blocks which contain the value of s_i and insert the encrypted field back into the packet².

²It is important to note that, for this mechanism to work, we always encrypt and decrypt the packet from the end to the beginning. This is so the value of s_i affects all the packet. We assume that either a stream cipher, which is re-initialized for every packet with a known vector, or a block cipher with a mode, such as for example RC5-CBC, is used.

In this example the address of node N_i , MAC_i , must be encrypted independently because it is pre-computed at the other side. The node encrypts MAC_i by applying $E_{K_i}(s_i||padding||MAC_i)$, where padding refers to the fact that s_i should be expanded to the block size of the cipher. After encryption, the encrypted section of s_i is truncated and only the encrypted MAC_i is added to the packet. The same encryption process can also be applied on the receiving side since s_i is synchronized.

If this protocol is employed, all unicast packets in the network are indistinguishable from each other. In fact, an attacker will be unable to link two different packets by using link layer information.

2) *Re-synchronization Mechanism*: If a packet is lost, the value of s_i is no longer synchronized on the end points. We detect such an event by the inability of the node to find any pre-computed value for the field. In order to re-synchronize, the node must use all its keys to attempt to decrypt the packet and use known values as a check to determine if the decryption was successful. If it is not able to decrypt the packet, the packet is discarded. If it is successful, then the new value of s_i contained in the packet is used to update the local copy of the value.

3) *Transport Header*: When encryption occurs, it's likely that the resulting message differs from the plaintext message not only in content, but also in size. Therefore, we must append the size of the original message to be able to distinguish between the actual content of the original packet and the padding.

Also, depending on the technology we apply our concepts to, we might need to transmit information which is removed from the packet to avoid information leakage, but must be returned before delivering it to the higher layers.

We propose the use of a header which should be appended, if needed, before encryption, to all packets and provide information on the issues discussed above. This header should contain the values which were removed from mandatory cleartext fields, the original length of the packet and a termination with the value of s_i . Since encryption and decryption are performed from end to beginning, we are sure that the variability in the ciphering caused by the changing s_i affects the whole packet encryption.

4) *Sending a Packet*: Algorithm 1 describes the process a node must go through to send a unicast packet. Broadcast packets are still sent in the same manner since, unless the source wishes to be anonymous, the information of who is the source must be given to all nodes and hiding the identity of the source becomes a contradiction.

Algorithm 1 Sending a unicast packet.

```

Intercept the message sent from MAC to PHY
Determine which  $\langle K_i, s_i \rangle$  to use
if is not  $AP$  then
    Use stored  $\langle K_i, s_i \rangle$ 
else
    Use table to map  $N_i$  address,  $MAC_i$ , to  $\langle K_i, s_i \rangle$ 
end if
Insert transport header (Section IV-A.3)
Encrypt packet (Section IV-A.1)
 $s_i \leftarrow s_i + 1$  and update pre-computed values
Send the encrypted packet using the PHY mechanisms

```

5) *Receiving a Packet*: Algorithm 2 describes the process a unicast packet undertakes upon reception. In a similar way

to the algorithm for sending, receiving a broadcast packet is handled in the usual method proposed by the technology without any modifications.

Algorithm 2 Receiving a unicast packet.

```

Intercept the message sent from PHY to MAC
Apply the determination mechanism as follows:
if is not AP then
  Use stored  $\langle K_i, s_i \rangle$ 
else
  for all Stored  $E_{K_i}(s_i || padding || MAC_i)$  do
    if Matches the field in the packet then
      Exit the loop and use  $\langle K_i, s_i \rangle$ 
    end if
  end for
  if No  $\langle K_i, s_i \rangle$  was found then
    for all Stored  $\langle K_i, MAC_i \rangle$  do
      Decrypt the last block with  $K_i$  and retrieve  $s_i$ 
      Use  $\langle K_i, s_i \rangle$  to decrypt the address block
      if  $MAC_i$  equals address block then
        Exit the loop and use  $\langle K_i, s_i \rangle$ 
      end if
    end for
  if No  $\langle K_i, s_i \rangle$  was found then
    Proceed to Key Agreement and EXIT
  end if
end if
  Decrypt the packet using  $K_i$ 
  Remove transport header, update fields and deliver to MAC
if MAC did not detect errors then
   $s_i \leftarrow s_i + 1$  and update pre-computed values
end if

```

B. Who in 802.11 said that?

In the previous section we presented a generic approach for link layer location privacy which we will now map to the link layer protocol IEEE 802.11 [1].

Duration/AID field: As identified during the information leakage analysis, this field must be sent in plaintext so all stations can update their NAV. This poses an additional problem since, without knowing the packet type, stations would also not be able to distinguish a duration field from an AID. To protect the protocol against this problem we propose that all packets contain the value for duration in this field, which is possible to compute at all times. In packets where the AID should be sent, the AID is added to the transport header and encrypted. Before the packet is passed on to the higher layers, and after decryption, the AID will be copied on top of the duration field. Using this mechanism ensures the AID is never revealed and stations are able to update their NAV using any packet.

Beacons: In our approach the beacon must be modified to prevent attacks on the TIM.

We address this issue by encrypting each position of the bitmap individually to each of the stations. Each relevant bit will be ciphered by taking the bit at position j from $E_{K_i}(s_i || b, Padd)$, where s_i is the station's current sequence number, and b is the original TIM bit value. j is the most significant different bit when comparing the encryption for $b = 0$ and $b = 1$. When station i receives a beacon, it repeats the encryption process and uses the index j to compare the encrypted values with the value in the TIM at the expected position. This process can determine if packets for that station are queued at the AP.

Registration: In order to minimize the number of packets in the network, we re-use packets, such as the *Association*

Request and the *Association Response*, to perform the registration procedure. As such, we use the beacon to carry the puzzles, the *Association Request* to transport the solutions to the puzzle together with N_i 's part of the DH key agreement (g^a) and the *Association Response* as the DH part of the *AP* (g^b). The first data packet, management or control frame from N_i can serve as confirmation the procedure was carried out correctly.

Transport Header: The transport header serves two purposes: the first is to carry the correct length of the decrypted packet and to solve the problem of the packets which contain an AID which would identify the station. The second is to transport the value of s_i used to seed the encryption of the packet in order to allow for re-synchronization. Fig. 2 depicts the fields and added options in the standardized 802.11 packet header.

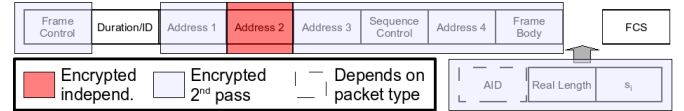


Fig. 2. 802.11 header with transport header options.

Looking at the header we can see that all fields, with the exception of the duration field, are encrypted. Also, we observe that one of the address fields is encrypted independently. In the case of an N_i sending a packet, this field will correspond to the source address. When it is the *AP* that sends a packet the destination field is used in this way.

The end result is that all packets are indistinguishable from each other, with the exception of broadcast packets. The attacker will have no way to infer the source or destination, or to correlate two unicast packets by solely analyzing the traffic.

V. PERFORMANCE EVALUATION

Performance depends on the cipher being used. When choosing a cipher for our scheme, we must ensure that it fits the operations described and, since this cipher will be used in every packet and during time-critical events, that it is also efficient. The small block size and highly efficient duty-cycle of RC5 [12] make it a perfect candidate. The block size fits the minimum encryption unit required in our scheme, which is 32 bits, which reduces the need for padding, since packets are usually 32-bit aligned.

According to [5], RC5 encryption and decryption both take 19 clock cycles (cc) per byte in a Pentium III. This is an acceptable platform for the *AP*, considering that for a real deployment crypto primitives should be implemented in hardware. In this performance section we assume that this is the cipher used, with this implementation, on a Pentium III 600 Mhz.

For our test scenario we consider one *AP*, one correspondent node (CN), which is the destination for all communications in the wireless channel, and an increasing number of nodes (N_i). Each added node N_i increases the load on the network and reduces the opportunities of another node to find the medium free.

We have performed all our simulations in NS-2 [10] 2.29, with the node number (NN) varying between 1 and 20, transmitting UDP packets of 178 bytes, at the rate 67.8 Kb/s. For each simulation we perform 10 runs of 60 seconds each.

To perform our simulations we inserted a computation delay at the *AP* which depends on whether or not the *AP* is

synchronized with this node. The mechanism used to check whether the node is not synchronized with the AP is based on whether the MAC layer has re-transmitted a packet due to collision. In cases where a re-transmission has occurred, the AP will take the average time of finding an entry in a table which is of size $NN/2$ (where NN corresponds to the number of simulated nodes)³. Once the key is found, we assume the node to be synchronized once again with the AP.

The performed simulations cover three different cases. The first, for comparison purposes, is a plain 802.11b simulation. The two other scenarios implement higher processing delays at the nodes, with one and two queue variants.

1) *Impact on Real-Time Traffic:* In this scenario we are interested in the behavior of real-time applications, such as audio and video, and make use of UDP with constant bitrate (CBR) traffic. We are interested in how our scheme affects both end-to-end delay and jitter, where each node transmits at 67.8 Kb/s, with 178 byte packets. This simulates a 64Kb/s voice call and the RTP overhead.

Figures 3 and 4 show the end-to-end delay and jitter for the real-time traffic. They show that the saturation point of the 802.11b network is located at 7 nodes per base station, where both the delay and jitter begin escalate rapidly, as further detailed in the figure. The saturation point remains the same in all scenarios.

The slight delay increase is consistent with the encryption and decryption times. With few collision below the saturation point, single and double queues have a similar performance. Above the saturation point the second queue shows greater value by providing a significantly smaller delay than the single queue, because the collision/retransmission frequency increases due to network congestion. However, above the saturation point, the bottleneck is the wireless access and therefore the delay presents variations regardless of the scenario.

Figure 4 present the jitter values for all the tested scenarios. For up to 6 nodes, the jitter increases slowly and steadily. Afterwards we notice a rapid increase, due to network congestion. The observed jitter values have similar results as the delay, and lead to the conclusion that the extra processing has little impact on performance.

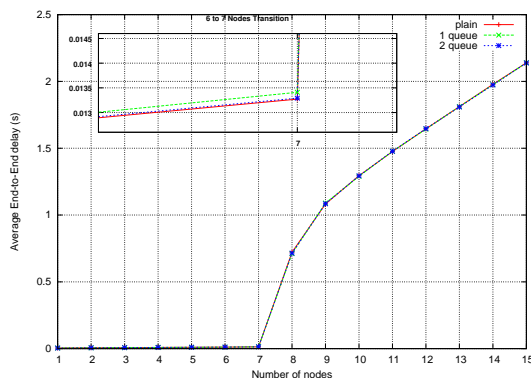


Fig. 3. End-to-end delay, CBR 67.8 Kb/s per node from 1 to 15 nodes, zoomed is the saturation point

VI. CONCLUSION

We have presented a solution to link layer location privacy and proved its feasibility under the example of a well known

³Please note that we do not assume any optimization or ordering of this table.

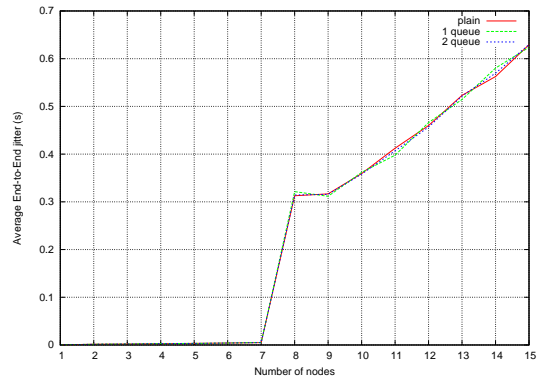


Fig. 4. End-to-end jitter, CBR 67.8 Kb/s per node from 1 to 15 nodes protocol (IEEE 802.11). Our approach should be used in conjunction with a pseudonym mechanism to prevent tracking by active communicating peers, which could be an interesting new direction for our work. Nevertheless, our approach provides privacy at the link layer without significantly undermining the performance of the network.

VII. ACKNOWLEDGEMENTS

We would like to thank Xavier Perez-Costa and Daniel Camps Mur for their comments and fruitful discussions.

REFERENCES

- [1] IEEE Standard 802.11. IEEE standard for information technology - telecommunications and information exchange between systems - local and metropolitan area networks - specific requirements, part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, 1999.
- [2] D. R. Cheriton and M. Gritter. Triad: A scalable deployable nat-based internet architecture, 2000.
- [3] David Clark, Robert Braden, Aaron Falk, and Venkata Pingali. Fara: reorganizing the addressing architecture. In *FDNA '03: Proceedings of the ACM SIGCOMM workshop on Future directions in network architecture*, pages 313–321, New York, NY, USA, 2003. ACM Press.
- [4] Charles Perkins (Ed). Ip mobility support for ipv4. Proposed Standard, August 2002.
- [5] New European Schemes for Signatures, Integrity, and Encryption NESSIE. Performance of optimized implementations of the nessie primitives, version 2.0, IST-1999-12324, 2003.
- [6] Joao Girao, Bernd Lamparter, Marco Liebsch, and Telemaco Melia. A practical approach to provide communication privacy. In *IEEE International Conference on Communications*, Istanbul, Turkey, June 2006. ICC2006.
- [7] W. Haddad. Privacy for Mobile and Multi-homed Nodes: Formalizing the Threat Model. Internet Draft (Work in Progress), February 2005.
- [8] W. Haddad. Privacy for Mobile and Multi-homed Nodes: MoMiPriv Problem Statement. Internet Draft (Work in Progress), February 2005.
- [9] P. Nikander, J. Arkko, and B. Ohlman. Host identity indirection infrastructure. In *Proceedings of The Secconf Swedish National Computer Network Workshop 2004 (SNCNW2004)*, November 2004.
- [10] ns 2. The network simulator, <http://www.isi.edu/nsnam/ns/>, as in June 2006.
- [11] C. Perkins, D. Jonhson, and J. Arkko. Mobility support in ipv6. Proposed Standard, August 2004.
- [12] Ronald L. Rivest. The rc5 encryption algorithm. In Bart Preneel, editor, *Fast Software Encryption*, volume 1008 of *Lecture Notes in Computer Science*, pages 86–96. Springer, 1994.
- [13] Ion Stoica, Daniel Adkins, Shelley Zhuang, Scott Shenker, and Sonesh Surana. Internet indirection infrastructure. In *Proceedings of ACM SIGCOMM*, 2002.