# Achieving Mobility and Anonymity in IP-Based Networks

Rungrat Wiangsripanawan[1], Willy Susilo[1], and Rei Safavi-Naini[2]

[1] Centre for Computer and Information Security Research (CCISR)
School of Computer Science and Software Engineering
University of Wollongong, Australia
rw26@uow.edu.au, wsusilo@uow.edu.au
[2] University of Calgory, Canada
rei@cpsc.ucalgary.ca

**Abstract.** Mobility and anonymity are two essential properties desirable in IP-based networks. In this paper, we aim to address the issue on how to achieve mobility and anonymity concurrently. At a glance, these two properties seem to be contradictory. This is partly due to the fact that there exists *no* single definition that clearly defines these notions. We approach this problem by firstly define these properties formally and address the problem of achieving these properties at the same time. Then, we proceed with a concrete construction based on an existing IP-based network, which is Tor. Without losing generality, our method can be applied to any other existing network, such as Morphmix or Tarzan. We highlight the difficulty of achieving mobility and anonymity concurrently although it seems trivial to merge these two properties altogether. Finally, we evaluate our proposed construction based on the definition that we have developed. Our work can be seen as the *first* attempt towards formalizing the notions of mobility, anonymity and location privacy.

**Keywords:** mobility, anonymity, location privacy, IP networks, Tor, Mobile IP.

## 1 Introduction

Consider a situation where a businessman is on his holiday. Firstly, he does not want his location to be traced by his company when he is accessing the Internet. Considering the nature of the businessman, he wants to have *mobility*. That means during his movement, he wants to have a continuous connection to the Internet. This requirement implies that if he is downloading streaming contents on a train, for example, the process should continue even if the train has enforced network movements. Furthermore, he wants his anonymity to be ensured during his trip. For instance, from time to time, the businessman would like to check the status of the stock market, etc. and he wants his identity to be protected. In this scenario, we have seen that mobility and anonymity is often desirable *at the same time*. Additionally, location privacy is an additional feature that people would like to have since the support from the Internet has made this

F. Bao et al. (Eds.): CANS 2007, LNCS 4856, pp. 60–79, 2007.

possible. Unfortunately, as we shall show in the next section, these requirements are contradictory to each other as adding mobility to an anonymous network system means that location privacy is lost.

*Our Contributions.* In this paper, we aim to address how to achieve mobility and anonymity in IP based networks *concurrently*. Additionally, we would like to provide the notion of location privacy to the users in this setting. To date, the existing works do not define precisely what they meant by anonymity and location privacy. Therefore, we firstly define these notions formally. Then, we proceed with a concrete system that will provide mobility and anonymity *at the same time*. We start our design by using the existing systems (i.e. combining Mobile IP with Tor), but unfortunately we will show that a trivial merge between the existing systems will not result in a desirable system. We note that essentially Mobile IP provides mobility and Tor provides anonymity, but a combination between these two will *not* be sufficient to achieve what is required in our scenario. We also propose our new design that can achieve the desirable system as stated in the motivating scenario. Finally, we also show that our design satisfies all the formal definitions that we put forth in the beginning.

### 1.1 Related Works

To date, there are many works in the literature that have been proposed to provide anonymity. This includes the works on low latency networks (e.g. Tor [20], Morphmix [15] and Tarzan [9]) to name a few. Furthermore, several works have also been proposed to provide mobility [22,17,11,14]. As mobility always leaks the location of the host, some works have been proposed to address this issue by adding location privacy to the existing mobility systems [8,4].

Flying Freedom [7] seems to be the only system to date that provides mobility, anonymity and location privacy at the same time. Nevertheless, this network is built on top of the architecture of the Freedom Network [10], that is no longer available[1,2] and the network itself has ceased.

Therefore, the seek for a system that provides mobility, location privacy and anonymity at the same time remains an interesting research question. A combination of the two different systems, where each system provides either mobility or anonymity, seems to be the candidate to provide the solution to this problem. Unfortunately, an inherent problem that will occur is the location privacy problem. Enhancing the system with the existing location privacy mechanisms also results in a lengthy communication path, which will lead to a very ineffective system. We will elaborate this issue in a later section. We note that we are not interested in building a new system from scratch. Instead, we will use available architectures as our building blocks.

## 2 Towards Formalizing Mobility and Anonymity Notions

In this section, we aim to clarify the notions of mobility and anonymity by firstly presenting their definition in a high level, and proceed with a formal definition to capture these notions.

### 2.1 Mobility

Roughly speaking, mobility is the ability of moving from one location to another. In the context of IP-based networks, we are interested to equip applications with the ability to move from one network to another. This definition is closely related, but different, to the concept of *roaming*. In the roaming situation, a mobile host obtains the Internet access via other networks. In the contrary, IP mobility allows the mobile host to move from one IP network to another IP network whilst enjoying to receive the upper layers' services as if the mobile host is a fixed host. In other words, the movement is *transparent* to the upper layers. That implies that the user will not be aware that the network's point of attachment has changed. More specifically, the TCP sessions should not be reset and the mobile host should always be addressed by its home network's address.

To achieve mobility in IP-based networks, essentially there are two mechanisms. The first one is to establish a special route through out the communication path between the mobile host and its correspondent node (recipient). Nevertheless, this approach is not scalable since the special route is always required throughout the entire communication path whenever the host changes its location. The second approach is to assign specific nodes that are responsible to maintain the mobile host's location. Tunneling mechanism is employed to forward packets destined to the mobile host that is away, which are "channeled" via these specific nodes.

Essentially, there are two categories in the mobility management schemes [21], namely one that handles *micromobility*, such as GSM networks, and the other that handles *macromobility*. Micromobility protocol focuses on mobility of the mobile host within a small region (usually within the same subnet). Macromobility protocol is more focussed on a broader term, that is the mobility across the regions. The examples of the latter approach include Mobile IPv4 [17] and Mobile IPv6 [11].

*Mobility vs. Location Privacy*
We observe that adding mobility to the IP-based networks will have an impact of losing the *location privacy*. The term *location privacy* refers to the case where one would like to conceal his location from anyone else. The need of mobility will enforce the need of the node attachment to *monitor* the location of the mobile host during its movement or the need of the specific route. The node attachment is the node that will ensure the connectivity of the mobile host to the IP networks, or in other words, it will provide the necessary upper layers' services to the mobile host. Therefore, the location of the mobile host is always exposed to the node attachment. Also, if messages exchanged on the communication path between the mobile host and its correspondent node are not carefully protected, an observer (one who can "listen" to the communication by observing the packets travelling through the wire) can obtain the location information either from the content of the messages or the messages' headers. Moreover, a system that allows the mobile host to update its location with its recipients directly for the sake of performance exposes the mobile host's location further.

In the following, we will firstly define the entities involved in the system. After observing what happens in the real environment, we are ready to define the system formally.

*Entities*. There are three entities involved in the IP-based mobility system: the mobile host, its communication partner and the node attachment. The mobile host entity is represented by its initial IP address that is provided from its home network eg. from the home network's DHCP server. We should stress that the location of the mobile host is *not* an entity rather than the mobile host's attribute, which is an IP address provided by each network it visits. The role of being the sender or the receiver in the communication path depends entirely on the message direction in the path. To illustrate, we refer the mobile host to be the sender and its partner is the receiver when a message is sent from the mobile host to its partner.

*Mobile Host Movement to a Different Network*. When the mobile host moves to a different network, it will firstly obtain a new location in the new network e.g. from the DHCP server of the new network. Then, it will establish a communication channel from this new location to its partner. This can be done either by creating a channel directly to the partner or a channel through the node of attachment. The aim of the adversary is to learn the mobile host's location. We consider the *adversaries* to be all the other *untrusted* entities in the path. They can simply be an observer that can only wiretap the connection or the nodes that help forwarding packets in the path, such as the node attachment. Therefore, we divide an adversary in the IP-based location privacy into three types: an observer, a mobile host's communication partner, and a node or nodes on the communication path. We note that from the adversaries' point of view, the node attachment is *not* directly related to the ongoing communication between the sender and the receiver during a communication session. Therefore, in the following communication model, we consider the simplest view of the channel by employing a single sender and receiver available in the system.

*Model of the Communication Channel*. In the following, we assume that the communication will employ a traditional point-to-point model. That is, there is a single host that sends its package via a public untrusted network, and the recipient is sitting at the other end of the network, which refer to the mobile host and its communication partner in the above scenario. As explained above, in the following definition, we shall omit the node attachment as an entity in the environment.

We consider there exists an observer (or an "adversary") who can observe the communication in the network. We assume that the mobile node has obtained its new location from the provided system such as by DHCP [6]. For a more elaborate treatment of this model, we refer the reader to [23], where we carefully analyzed the case that involves the node attachment itself. Note that in this definition we consider the direction when the mobile host is a receiver.

Intuitively, the notion of location privacy is defined as follows. Given a transcript of a message sent by a sender to a receiver in two possible locations of

the receiver, the task of the adversary is to correctly guess where the location of the receiver is. Formally, we will define location privacy using the following interaction between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. The adversary is given an access to the $\mathsf{PacketReq}^{\mathsf{LP}}$ oracle, that is, given a message $m$, a receiver's location $\mathcal{L}$ and a pair of sender-receiver, the oracle is to output the correct transcript of the communication, $\Omega$, that represents a message $m$ sent by the sender to the receiver in the location $\mathcal{L}$. The oracle $\mathsf{PacketReq}^{\mathsf{LP}}$ represents the capability of the observer (or the adversary) to request a message from a sender of his choice to be sent to a receiver that located in $\mathcal{L}$. Note that in this model the adversary can passively listen to the communication in the channel. The formal definition is as follows.

**Location Privacy Interaction:** Let $\mathcal{C}$ be the challenger and $\mathcal{A}$ be the adversary who would like to break the location privacy.

1. *Initialization.* Let $k \in \mathsf{N}$ be the security parameter. $\mathcal{C}$ is invoked with all the condition and information known in the communication channel. The information is provided to $\mathcal{C}$ by $\mathcal{A}$. In particular, the pair of sender $\mathcal{S}$ and receiver $\mathcal{R}$ is provided to $\mathcal{C}$ together with some possible locations $\{\mathcal{L}_1, \mathcal{L}_2, \cdots, \mathcal{L}_\ell\} \in \mathcal{L}$ which represent $\mathcal{R}$'s position (note that $\mathcal{L}$ is $\mathcal{R}$'s attribute).
2. *Attacking Phase*
   (a) $\mathcal{A}$ can make the $\mathsf{PacketReq}^{\mathsf{LP}}$ queries as defined as follows.
      – $\mathsf{PacketReq}^{\mathsf{LP}}$. $\mathcal{A}$ can provide a message $m_i$ and select a location $\mathcal{L}_i$ and query the $\mathsf{PacketReq}^{\mathsf{LP}}$ oracle to obtain a transcript $\Omega(m_i, \mathcal{S}, \mathcal{R}, \mathcal{L}_i)$ to denote a message $m_i$ sent by $\mathcal{S}$ to $\mathcal{R}$, which is currently located at $\mathcal{L}_i$.

   These queries can be invoked for at most $q_{PR_{\mathcal{L}}}$ times.
   (b) $\mathcal{A}$ outputs two distinct locations $(\mathcal{L}_0, \mathcal{L}_1)$ and a target message $m^* \in \mathcal{M}$ where $\mathcal{M}$ is a set of messages that have been queried before. In return, $\mathcal{C}$ outputs a transcript $\Omega(m^*, \mathcal{S}, \mathcal{R}, \mathcal{L}_i)$ where $i$ is obtained from a coin toss.
   (c) $\mathcal{A}$ can execute $\mathsf{PacketReq}^{\mathsf{LP}}$ queries for any message $m_j \neq m^*$ for any location in $\mathcal{L}$.
3. *Output Phase.* $\mathcal{A}$ outputs his guess $i$, where $\mathcal{L}_i$ is the location of the receiver who produces $\Omega(m^*, \mathcal{S}, \mathcal{R}, \mathcal{L}_i)$.

The success probability of the adversary in attacking location privacy is defined by $Succ_{\mathcal{A}}^{PR_{\mathcal{L}}} = \frac{1}{2} + \epsilon$.

**Definition 1.** *A system is said to provide* location privacy *if there is no polynomial time algorithm $\mathcal{A}$ that has a non-negligible probability in the* **Location Privacy Interaction** *defined above.*

### 2.2 Anonymity

In the Internet, anonymity can be classified into two types: *data anonymity* and *connection anonymity* [5]. The term *data anonymity* refers to the identification

of information that can be extracted from the data exchanged in a particular application. The term *connection anonymity* refers to the identities of sender and receiver during data transfer.

The ultimate goal of anonymous communication systems is to ensure that an adversary gains no information about the communication that is happening in the communication channel [12]. However, this system is unrealistic in a public network as in the Internet. It is therefore considered adequate if the system satisfies some properties of the anonymous communication system, that include the inability of the adversary to identify the sender or the receiver. We will describe this possibility formally in the following paragraph.

Assuming the same communication channel model as in location privacy is used, the main intention of the sender is to ensure that her identity is not revealed to the observer (*privacy* of the sender). Additionally, the main intention of the receiver is also to ensure that his identity will not be disclosed (*privacy* of the receiver). From the observer's point of view, his task is considered to be "successful", if he can observe the communication channel and figure out who the sender and/or the receiver is (*adversarial goal*). If the observer cannot be successful in this particular task, then we say that the network ensures *anonymity* in the system.

Based on this setting, we further divide the notion of anonymity into three different properties: i) sender anonymity, ii) receiver anonymity, and iii) unlinkability. A system that satisfies these three properties is said to be an *anonymous* system [19].

*Oracles*. Let the oracle $\mathsf{PacketReq}^{\mathsf{SA}}$, the oracle $\mathsf{PacketReq}^{\mathsf{RA}}$ and the oracle $\mathsf{PacketReq}^{\mathsf{UL}}$ represent the capability of the adversary to request a message sent by a particular sender of his choice to a receiver in the sender anonymity, receiver anonymity and unlinkability games, respectively. This is to represent the ability of the adversary to wiretap the communication channel and to select the messages learnt from the channel.

*Sender Anonymity*. This property ensures that the observer (or the adversary) cannot identify who the sender is, given a stream of packages traveling through the communication channel. Intuitively, the task of the adversary is to guess who the sender is, given a transcript that could be produced by two different senders. Formally, this property is defined using the following interaction between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. The adversary is given an access to the $\mathsf{PacketReq}^{\mathsf{SA}}$ oracle that behaves as follows: given a message $m$, a particular sender and a receiver, the oracle will return a correct transcript $\Omega$ that represents a transcript of a message $m$ that is sent by the sender to the receiver.

**Sender Anonymity Interaction:** Let $\mathcal{C}$ be the challenger and $\mathcal{A}$ be the adversary who would like to break the sender anonymity.

1. *Initialization.* Let $k \in \mathsf{N}$ be the security parameter. $\mathcal{C}$ is invoked with all the condition and information known in the communication channel. The information is provided to $\mathcal{C}$ by $\mathcal{A}$. In particular, the set of senders $\{\mathcal{S}_1, \mathcal{S}_2, \cdots, \mathcal{S}_\ell\} \in \mathcal{S}$ is provided to $\mathcal{C}$ together with a receiver $\mathcal{R}$.

2. *Attacking Phase*
   (a) $\mathcal{A}$ can make the $\mathsf{PacketReq}^{\mathsf{SA}}$ queries as defined as follows.
      – $\mathsf{PacketReq}^{\mathsf{SA}}$. $\mathcal{A}$ can provide a message $m_i$ and select a sender $\mathcal{S}_j \in \mathcal{S}$ and query $\mathsf{PacketReq}^{\mathsf{SA}}$ oracle to obtain a transcript $\Omega(m_i, \mathcal{S}_j, \mathcal{R})$ that represents a message $m_i$ sent by $\mathcal{S}_j$ to $\mathcal{R}$.
   These queries can be invoked for at most $q_{PR_S}$ times.
   (b) $\mathcal{A}$ outputs $(\mathcal{S}_0, \mathcal{S}_1)$ and a target message $m^* \in \mathcal{M}$ where $\mathcal{M}$ is a set of messages that have been queried before. In return, $\mathcal{C}$ outputs a transcript $\Omega(m^*, \mathcal{S}_i, \mathcal{R})$ where $i$ is obtained from a coin toss.
   (c) $\mathcal{A}$ can execute $\mathsf{PacketReq}^{\mathsf{SA}}$ queries for any message $m_j \neq m^*$ for any sender in $\mathcal{S}$.
3. *Output Phase.* $\mathcal{A}$ outputs his guess $i$, where $\mathcal{S}_i$ is the sender who produces $\Omega(m^*, \mathcal{S}_i, R)$.

The success probability of the adversary in attacking the sender anonymity is defined by $Succ_{\mathcal{A}}^{PR_S} = \frac{1}{2} + \epsilon$.

**Definition 2.** *A system is said to provide* sender anonymity *if there is no polynomial time algorithm $\mathcal{A}$ that has a non-negligible probability in the* **Interaction Sender Anonymity** *defined above.*

*Receiver Anonymity.* This property ensures that the observer (or the adversary) cannot identify who the receiver is, given a stream of packages traveling through the communication channel. Intuitively, the task of the adversary is to correctly guess whom the sender has sent her message to, given two possible receivers. Formally, this property is defined using the following interaction between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. The adversary is given an access to the oracle $\mathsf{PacketReq}^{\mathsf{RA}}$ that accepts a message $m$, a particular receiver and a sender, to output the correct transcript $\Omega$ that represents a transcript of a message $m$ that is sent by the sender to the receiver. The formal definition follows.

**Receiver Anonymity Interaction:** Let $\mathcal{C}$ be the challenger and $\mathcal{A}$ be the adversary who would like to break the receiver anonymity.

1. *Initialization.* Let $k \in \mathsf{N}$ be the security parameter. $\mathcal{C}$ is invoked with all the condition and information known in the communication channel. The information is provided to $\mathcal{C}$ by $\mathcal{A}$. In particular, the set of receivers $\{\mathcal{R}_1, \mathcal{R}_2, \cdots, \mathcal{R}_\ell\} \in \mathcal{R}$ is provided to $\mathcal{A}$ together with a sender $\mathcal{S}$.
2. *Attacking Phase*
   (a) $\mathcal{A}$ can make the $\mathsf{PacketReq}^{\mathsf{RA}}$ queries as defined as follows.
      – $\mathsf{PacketReq}^{\mathsf{RA}}$. $\mathcal{A}$ can provide a message $m_i$ and select a receiver $\mathcal{R}_j \in \mathcal{R}$ and query the $\mathsf{PacketReq}^{\mathsf{RA}}$ oracle to obtain a transcript $\Omega(m_i, \mathcal{S}, \mathcal{R}_j)$ that represents a message $m_i$ sent by $\mathcal{S}$ to $\mathcal{R}_j$.
   These queries can be invoked for at most $q_{PR_R}$ times.
   (b) $\mathcal{A}$ outputs $(\mathcal{R}_0, \mathcal{R}_1)$ and a target message $m^* \notin \mathcal{M}$ where $\mathcal{M}$ is a set of messages that have been queried before. In return, $\mathcal{C}$ outputs a transcript $\Omega(m^*, \mathcal{S}, \mathcal{R}_j)$ where $i$ is obtained from a coin toss.

(c) $\mathcal{A}$ can execute $\mathsf{PacketReq}^{\mathsf{RA}}$ queries for any message $m_j \neq m^*$ for any receiver in $\mathcal{R}$.

3. *Output Phase.* $\mathcal{A}$ outputs his guess $i$, where $\mathcal{R}_i$ is the receiver whom receives $\Omega(m^*, \mathcal{S}, \mathcal{R}_i)$, sent by $\mathcal{S}$ in this transcript.

The success probability of the adversary in attacking the receiver anonymity is defined by $Succ_{\mathcal{A}}^{PR_{\mathcal{R}}} = \frac{1}{2} + \epsilon$.

**Definition 3.** *A system is said to provide* receiver anonymity *if there is no polynomial time algorithm $\mathcal{A}$ that has a non-negligible probability in the* **Interaction Receiver Anonymity** *defined above.*

*Unlinkability.* This property ensures that the observer (or the adversary) cannot link two different transcripts whether they are coming from the same sender or not. Intuitively, the task of the adversary is to guess whether two transcripts are related to each other (i.e. they come from the same sender, or the same receiver). Formally, this property is defined using the following interaction between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. We note that in our definition, we assume that there exists a single receiver $\mathcal{R}$. However, without losing generality, our definition can be trivially modified to include multiple receivers, but this setting has been captured by our model. The adversary is given an access to the oracle $\mathsf{PacketReq}^{\mathsf{UL}}$ that accepts a message $m$, a sender and a receiver, to output a transcript $\Omega$ representing a transcript of a message $m$ that is sent by the sender to the receiver.

**Unlinkability Interaction:** Let $\mathcal{C}$ be the challenger and $\mathcal{A}$ be the adversary who would like to break the unlinkability.

1. *Initialization.* Let $k \in \mathsf{N}$ be the security parameter. $\mathcal{C}$ is invoked with all the condition and information known in the communication channel. The information is provided to $\mathcal{C}$ by $\mathcal{A}$. In particular, the set of senders $\{\mathcal{S}_1, \mathcal{S}_2, \cdots, \mathcal{S}_\ell\} \in \mathcal{S}$ is provided to $\mathcal{A}$ together with a receiver $\mathcal{R}$.
2. *Attacking Phase*
   (a) $\mathcal{A}$ can make the $\mathsf{PacketReq}^{\mathsf{UL}}$ queries as defined as follows.
      – $\mathsf{PacketReq}^{\mathsf{UL}}$. $\mathcal{A}$ can provide a message $m_i$ and select a sender $\mathcal{S}_j \in \mathcal{S}$ and query the $\mathsf{PacketReq}^{\mathsf{UL}}$ oracle to obtain a transcript $\Omega(m_i, \mathcal{S}_j, \mathcal{R})$ that represents a message $m_i$ sent by $\mathcal{S}_j$ to $\mathcal{R}$.

   These queries can be invoked for at most $q_{PR_S}$ times.
   (b) $\mathcal{C}$ outputs $\mathcal{S}_j$, a transcript $\Omega_1(m_1^*, \mathcal{S}_j, \mathcal{R})$, $\Omega_2(m_2^*, \mathcal{S}_k, \mathcal{R})$, and two target messages $m_1^*, m_2^* \notin \mathcal{M}, m_1^* \neq m_2^*$ where $\mathcal{M}$ is a set of messages that have been queried before and $i$ is obtained from a coin toss. In this output, $j = k$ if the output of the coin toss is 1, and $j \neq k$ otherwise.
   (c) $\mathcal{A}$ can execute $\mathsf{PacketReq}^{\mathsf{UL}}$ queries for any message $m_j \neq \{m_1^*, m_2^*\}$ for any sender in $\mathcal{S}$.
3. *Output Phase.* $\mathcal{A}$ outputs his guess 0/1 to indicate whether $\Omega_1$ and $\Omega_2$ have been produced by the same sender $S_j$ or not.

The success probability of the adversary in attacking the unlinkability is defined by $Succ_{\mathcal{A}}^{ULs} = \frac{1}{2} + \epsilon$.

**Definition 4.** *A system is said to provide* unlinkability *if there is no polynomial time algorithm $\mathcal{A}$ that has a non-negligible probability in the* **Unlinkability Interaction** *defined above.*

**Definition 5.** *A communication channel is said to be* anonymous *if it satisfies sender anonymity, receiver anonymity and unlinkability.*

**Definition 6.** *A communication channel is said to provide* mobility *and* anonymity *if it provides mobility to its mobile hosts, is anonymous and ensures location privacy.*

## 3 Review on Existing Infrastructure That Provides Mobility and Anonymity

### 3.1 Mobile IP

Mobile IP protocol was designed to provide mobility to the IP-based networks. There are two versions of Mobile IP, namely Mobile IP version 4 (MIPv4) and Mobile IP version 6 (MIPv6). MIPv4 has been designed to work on top of the IPv4 network, and MIPv6 is designed for the IPv6 network. Nonetheless, the fundamental concept is essentially the same.

Mobility in Mobile IP protocol is provided via the use of two IP addresses, namely a home address (HoA) and a care-of address. A home address is an IP address of a mobile node when the mobile node resides in its original network. The home address is used for identification purpose. A care-of address is an IP address used by the mobile node when it is away at the visiting network. This IP address is used for identifying the location of the mobile node. The communication between these two addresses is assisted by two other entities, namely a home agent and a foreign agent[1]. When the mobile node is away from its home network, firstly it obtains its care-of address from one of the following possibilities: 1) the visiting network's foreign agent (in the MIPv4 setting); 2) the stateless configuration [11] in MIPV6; or 3) the DHCP mechanism in both MIPv4 and MIPv6 settings. Then, the mobile node registers the newly-obtained care-of address to its home agent. When there are packets destined to the mobile node's home address, then the home agent will forward these packets to the mobile node's care-of address. To avoid an ingress filtering problem at the foreign agents, the Mobile IP protocol employs a reverse tunneling mechanism [16] that allows the mobile node to send packets to its corresponding node via its home agent. When route optimization is deployed, the mobile node is allowed to update its care-of address directly to its correspondent node. It is clear that Mobile IP does *not* provide location privacy protection against the observer when there is no encryption in the tunneling packets. In addition, the location is also revealed

[1] We note that there exists no foreign agent in MIPv6.

to the corresponding node in case of route optimization. Moreover, the home agent also needs to monitor the location of the Mobile IP user.

### 3.2 Tor - A Low Latency Network

Tor [20] is the second-generation of Onion Routing [18]. Tor is a distributed system that provides anonymous connections to low-latency applications, such as web browsing, secure shell and instant messaging. Similar to Onion Routing, the architecture of Tor is based on the Chaum [3] mix network model, but Tor relay node does not perform any mixing operations (i.e. batching, reordering or delaying packets). Tor is an anonymous system. Intuitively, sender anonymity, receiver anonymity and unlinkability provided by Tor are guaranteed due to the use of Tor servers (will be defined later in this section) that will act as the anonymizers in the network.

*Entities in A Tor Network.* There are three main entities in a Tor connection, which are 1) a Tor client, 2) a Tor-enabled application server[2], and 3) a group of Tor servers. When a user (or a sender, respectively) would like to establish a Tor connection to access any Tor-enabled application servers, the user is required to install a Tor client software. Then, the user's host has become one of the Tor clients in the network. The Tor client is responsible for fetching directories (of all Tor servers), establishing circuits[3] and handling connections from the user's application. The user would like to access the services provided by one of the application servers. This particular application server is known to be the *recipient* of the Tor connection. In order to allow the Tor client to reach the application server, there are several relay nodes that will be involved to establish the connection. These relay nodes are known as the Tor servers. The first node (i.e. a Tor server) in this connection is also known as the *entry node*, whilst the last node is known as the *exit node*. Currently, according to the Tor specification, the size of each circuit is set to involve 3 Tor servers.

*How Tor Works.* Firstly, a Tor client selects a number of Tor servers as members of the Tor circuit. Circuits in Tor are established preemptively. When an anonymous connection is required, the Tor client can simply select one of the already-established circuits. In contrast to the Onion routing that restricts one circuit per one TCP stream, Tor allows many TCP streams to share a single circuit. When the Tor client would like to send some data (e.g. when a user uses his browser to connect to a website), the streams of packets are divided into *fixed-size cells* and these cells are sent to the selected circuit. During the transmission, these packets are wrapped in a *layer-by-layer* fashion using session keys derived from pre-negotiated common keys. The intended purpose of this mechanism is to allow a Tor server, which will unwrap the packets, *only* to know merely its predecessor and successor nodes. There is no mixing process involved.

[2] A Tor-enabled application server is an application server that can function within a Tor network.

[3] Tor system calls a path as a *circuit*.

The incoming cells to any Tor nodes are simply placed into queues, processed and sent out in the first come first served fashion.

*Circuit Establishment in Tor.* Tor establishes and extends its circuit hop by hop until it reaches the length of the circuit. Normal Tor circuit's length is 3 hops, which comprises the entry node, the second Tor server nodes and the exit node. Suppose Alice wants to use Tor to anonymize her communication, then the description of how a circuit is established can be outlined as follows. Firstly, Alice's Tor client picks three nodes as its Tor entry node, its second node and its exit node, respectively. Without losing generality, let us assume that Alice picks $Tor_A, Tor_B$ and $Tor_{Exit}$ for a circuit path $Alice \rightarrow Tor_A \rightarrow Tor_B \rightarrow Tor_{Exit}$. To establish an onion encryption within the circuit, the Tor client and each of the Tor servers in the circuit must be equipped with a shared key. Tor uses Diffie-Hellman key exchange to accomplish this purpose. In every hop-connection, there is a circuit id used to represent a connection between any two consecutive nodes and this circuit id is known only between these two consecutive nodes. Due to page limitation, we refer the readers to [23] for more details on Tor circuits, commands and diagram.

## 4 Anonymous Communication with Mobility in IP-Based Networks

For clarity, we reiterate the ultimate goal in our scenario. Consider the situation where Bob, who is the CEO of the company ABC, is having his holiday break and he would like to make an anonymous communication for example downloading a streaming content, such as movies or video clips. In addition to being anonymous, Bob would like to have a continuous session during his trip on a train. Furthermore, he does not want his location to be revealed. We would like to provide a solution to this problem to satisfy Bob's requirements.

In summary, there are essentially three main properties required in this scenario, namely mobility, anonymity and location privacy.

Bob would like to receive a continuous session during his trip on a train. This requires *mobility* to be provided in an IP network. By allowing mobility, Bob will be given a continuous connection to the Internet application regardless his location. Bob's mobile node has to change from one network connection to another, but this movement (or also known as a *hands-off*) needs to be transparent to Bob.

Bob would also like to access the Internet applications anonymously. Bob does not want anyone to find out which services he has used. In short, Bob would like to achieve sender anonymity, recipient anonymity and unlinkability. Firstly, Bob does not want anyone to know that he is the sender of the message requesting the Internet service (sender anonymity). Secondly, he also does not want anyone to know to whom he is sending the message to (or which website Bob is currently browsing - and hence, recipient anonymity). Finally, he does not want anyone to be able to identify whom he is communicating to.

As described earlier, mobility implies exposing the location privacy. This means that Bob's location will need to be acquired by the system to allow the continuous session. Nonetheless, this will defeat Bob's requirement as he is having his holiday. Therefore, the final property that Bob would like to achieve is location privacy. As mentioned earlier, these three requirements seem to be contradictory.

In this section, we will describe how to achieve anonymity and mobility concurrently using the existing networks. We incorporate the existing IP-based networks that can provide us with anonymity or mobility, and we adjust the system so that it can satisfy our needs. We choose Mobile IP as our base system that will provide mobility. Furthermore, we also choose Tor as our building block for our anonymous system because of its rapid usage growth and availability.

Intuitively, by combining Mobile IP and Tor, we could achieve all the properties that we would like to obtain. Unfortunately, as we shall show in the next section, a trivial combination of these two systems will not provide us with a complete and good solution In particular, the new system will suffer from the location privacy feature. Then, we also present our enhancement to Tor to provide a better system. The new system represents a "better" network in terms of latency. Finally, we add the location privacy system to our hybrid system to fully satisfy our requirements.

### 4.1 Architecture MA1. Achieving Mobility and Anonymity Via Trivial Combination of Mobile IP and Tor

Without losing generality, we discuss our design and implementation using Mobile IP and Tor as our building blocks. Mobile IP is chosen to represent an IP network that is equipped with mobility, whilst Tor is chosen due to its low latency anonymity feature. Mobile IP works in network layer (layer 3) while Tor works in transport layer (layer 4).

*Basic Setting.* The scenario that we would like to consider is as follows. A user participates in a Mobile IP network. The user also installs a Tor client software

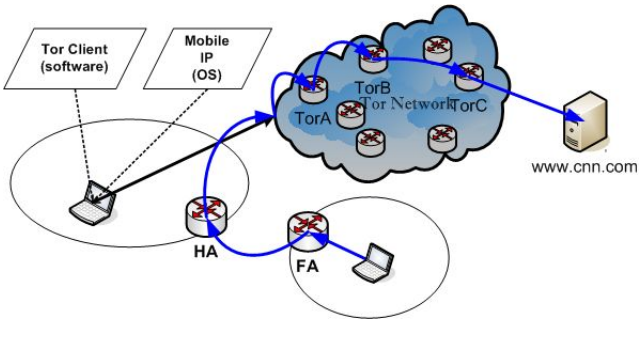


**Fig. 1.** An Illustration when a Mobile IP node would like to have a Tor connection

in his mobile node, and hence, the user is a Tor client in the Tor network. This scenario is illustrated in Figure 1.

To illustrate our idea, we start by providing a mechanism of how the system works when the mobile node is in its home network.

*Phase 1. The mobile node resides in its network.* In this phase, we start the scenario with the situation where the mobile node resides in its home network and the mobile node would like to make an anonymous communication to the remote destination (for example `http://www.cnn.com`). This situation is analogous to a static IP network that incorporates a Tor network. Suppose that a mobile node user, Alice, would like to browse the network anonymously. She starts her Internet browser by pointing its URL to `http://www.cnn.com`, and her Tor client will firstly selects a circuit to be used to route this particular Tor application. Without losing generality, suppose the Tor client picks a circuit $c1$ that consists of $Tor_A$, $Tor_B$ and $Tor_C$ as an entry node, the middle node and the exit node, respectively. The communication path between Alice and the HTTP server appears as $Alice \rightarrow Tor_A \rightarrow Tor_B \rightarrow Tor_C \rightarrow \texttt{http://www.cnn.com}$. Alice's IP address is used by $Tor_A$ as her identity. In this case, it is her home network's address.

*Phase 2. The mobile node is away.* When Alice moves to a different network (i.e. a foreign network) outside her home network, then her mobile node is away. In a typical Mobile IP scenario, the mobile node is required to report its new point of attachment, namely its care-of address, to its home agent via the registration process. This activity is assisted by the foreign agent in the foreign network in Mobile IPv4. After this process is completed, all the IP connections destined to this mobile node will be redirected to its home agent and the home agent is responsible to forward the packets to the mobile node's current location. This movement is transparent to Tor, since Tor works in the transport layer (layer 4). Hence, the communication path is $MN \rightarrow FA \rightarrow HA \rightarrow Tor_A \rightarrow Tor_B \rightarrow Tor_C \rightarrow \texttt{http://www.cnn.com}$. When route optimization is deployed, the mobile node is also required to update its location directly to its correspondent node (CN), which is $Tor_A$ in this case. Hence, the communication path is $MN \rightarrow FA \rightarrow Tor_A \rightarrow Tor_B \rightarrow Tor_C \rightarrow \texttt{http://www.cnn.com}$.

*The Drawbacks of Architecture* MA1. Firstly, we note that Mobile IP networks do not provide location privacy [13]. The home agent always knows the mobile node whereabout, a correspondent node has this knowledge when route optimization is deployed, and an observer can obtain this knowledge from the content of unprotected messages. Therefore, the architecture MA1 intrinsically inherits this problem. In a typical Mobile IP system, a proposed solution is to use forward and reverse tunneling to the home agent and then applies the ESP encryption in the inner IP packets [13]. However, this could only protect the mobile node's location from an observer. It does not prevent the home agent this knowledge. An idea that comes to mind is to add location privacy to the underlying Mobile IP, using techniques like adding a set of location proxies [8]. Nevertheless, this results in an extremely long communication path between the mobile node and

its recipient, in particular when the length of the proxy nodes are quite long. To justify this argument, let us refer to the communication path. Let $LP_i$ denote a location proxy node $i$. The whole communication path consists of the following entities: $MN \rightarrow FA \rightarrow LP_1 \rightarrow ... \rightarrow LP_n \rightarrow HA \rightarrow Tor_A \rightarrow Tor_B \rightarrow Tor_C \rightarrow$ `http://www.cnn.com` Furthermore, in this communication path, the benefit given by the low latency network, such as Tor, will be overridden by the lengthy and unnecessary communication path resulted by the location-privacy-enabled mobile IP networks.

Providing location privacy when deploying route optimization remains as an open question in Mobile IP protocol [13]. It seems odd to use the set of location proxy's technique, i.e. $MN \rightarrow FA \rightarrow LP_1 \rightarrow ... \rightarrow LP_n \rightarrow$ `http://www.cnn.com` to achieve this goal. Particularly, when route optimization is proposed to increase the system performance by allowing a direct connection between the mobile node and the correspondent node instead of tunneling through the home agent. However, the combined system benefits the Mobile IP route optimization some degrees of location privacy. That is, the mobile node's location is transparent to the Tor application's recipient (the CNN server in the above example). Nevertheless, a new problem arises. The mobile node's location is always exposed to the Tor entry node. Unlike the home agent that can be trusted to some extent, Tor nodes are not designed be trusted. Using two sets of proxies and combining them together at the Tor entry node instead of the home agent, as in a typical Mobile IP scenario, also results in the same problem, even though the path is one hop shorter.

In summary, by trivially combining a location-privacy-enabled mobility system and anonymity system seems to be insufficient to achieve mobility and anonymity concurrently. In a typical Mobile IP system scenario, when a high level of location privacy is required, this combination appears as two mix networks that are "glued" together. The first mix network aims at providing location privacy and mobility, whilst the second mix network deals with anonymity. These networks are combined by the point of attachment entity, such as the home agent. In a route optimization system scenario, the combined system seems to provide more location privacy as the mobile node's location is transparent to the Tor application's correspondent node. It instead shifts the problem to the Tor entry node and the seemingly available solution also results in a long communication path.

### 4.2 Architecture MA2. Adding Mobility to Tor

Essentially, Tor does not support mobility. When there is a change of the client's point of attachment during a Tor connection, all connections in circuits from the Tor client to its application's recipient will be required to be reset. Our architecture MA1 attempted to solve this problem by combining Mobile IP with Tor at the mobile node to add Tor's ability to provide mobility. Unfortunately, as we have shown earlier, the location privacy problem, which is an inherent problem in Mobile IP networks, will occur in the resulting architecture. Adding the location privacy to the underlying mobile IP networks will result in a different

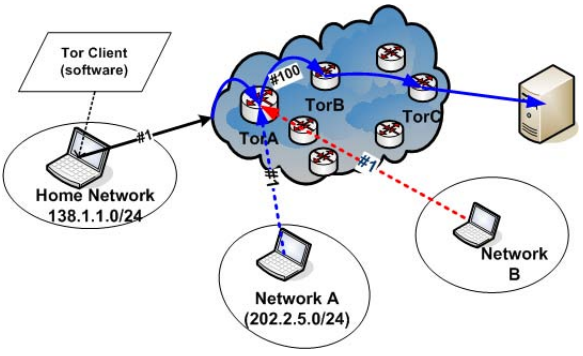


**Fig. 2.** Illustration of Mobile Node's Movement with a Single Entry Node

problem. Therefore, in this section, we are interested in taking a totally different approach, i.e. by adding the mobility capabilities to Tor instead of relying on another type of network, like Mobile IP. We will limit ourselves to the scenario where we are interested in and then describe the technique that we used to design and implement the mobility for Tor networks.

*Limited Scenario: Client-only Mobility.* We are interested to add mobility to the client in the Tor networks. As inspired by our scenario mentioned earlier, the recipient (for example `http://www.cnn.com`) can stay the same during the duration of the movement, but we allow the client to move from one network to another. The mobile node will always initiate the communication. Furthermore, without losing generality, we assume that the recipient, a low-latency application server, is always a fixed host.

*Design Strategy 1. Maintaining TCP connection of the Exit Node.* A Tor client does not have a direct connection to its recipient. The Tor client requires a series of nodes between itself and the application server, namely the Tor servers. We further note that Tor generates its circuit hop-by-hop. To illustrate this idea, let us consider the following Tor circuit that consists of four Tor connections:

- The first hop is between the Tor client and the Tor entry node
- The second hop is between the entry node and the middle node.
- The third hop is between the middle node and the exit node.
- The fourth hop is between the exit node and the recipient.

Each connection has its own underlying TCP connection. From the recipient's point of view, its "sender" is the exit node. Therefore, if a TCP connection between the exit node and the recipient can be maintained during the mobile node (i.e. the Tor client) movement, then we can preserve the sender-recipient indirect connection. That is, the change of the mobile node's point of attachment is transparent to its applications server. Therefore, our main aim is to maintain this particular TCP connection in the circuit during the mobile node movement.

*Design Strategy 2. Modification to The First-Hop Tor Connection.* By investigating the Tor circuit, we can observe that the mobile node's movement has a direct implication to the first hop of the established circuit. We note that this first hop is the TCP connection between the mobile node and the entry node. The movement of the mobile node will result in the change of the mobile node's IP address. This change will imply the failure of the first-hop TCP connection. Furthermore, since this TCP connection implies the whole Tor connection, the failure of the first hop will eventually stop the whole Tor circuit.

In order to ensure that the Tor circuit is still established, the Tor servers must provide a mechanism to allow the first-hop Tor connection to stay alive even though its underlying TCP connection is turned down and changed to the new point of attachment. For simplicity, we apply the known technique used in the TCP/IP network to allow the mobile node to acquire its new IP address during its movement, for example, DHCP [6], and to change the point of attachment by using the hands-off mechanism such as the one used in Mobile IP [17]. We do not aim to improve this technique as this is out of the scope of this paper. We illustrate our idea in Figure 2. We note that in this design, in contrast to the previous design, we modified the Tor server and client to handle mobility without relying on mobility entities of other existing mechanisms (eg. Mobile IP's home agent).

*The Detail of The Design.* Now, we discuss the situation when the mobile node moves to a new network from the above scenario. Once the mobile node obtains a new IP address from the new network, it sends an *additional* Tor-command cell, namely the *Resume* command cell, to the Tor entry node to request the Tor entry node to update its IP address with the newly acquired IP address. The cell must be encrypted with the common key between the Tor entry node and the mobile node. Tor servers need to be modified to allow a waiting period before closing its connection while its communicating partner is unavailable. This allows the whole circuit to stay alive when the mobile node moves.

To guarantee the authenticity of the Tor entry node, we employ a keyed hash function with a random number. The Tor entry node stores the up-to-date IP address of the mobile node as its sender address. We note that the initial IP address of the mobile node can be the home address of the mobile node when the system is just initialized. We also note that each Tor server must allow a longer waiting time period when the host or network unavailability is detected.

The *Resume* command cell consists of the command *Resume*, the circuit id, an encrypted value of the new IP address, the old IP address, a random number and the hash value of the old IP address and the random value. Due to page limitation, we refer the readers to [23] for more details.

*Existing Drawbacks: Location Privacy.* As in other mobility systems, our system also exposes the mobile node's location to the Tor entry node. Therefore, the problem of location privacy seems to be *inherent* whenever mobility is added to the network. The solution like the set of proxies is not appropriate as previously described in MA1.One could propose that the Tor entry node must be a trusted node. However, this is very unlikely to happen. Tor network itself does not require

the Tor entry node to be a trusted node. Also, if the circuit ID has not changed and the IP-packets between the mobile node at different locations and the Tor entry node are not encrypted, then the observer can trace the movement of the mobile node from the unchanged circuit ID and hence can obtain its location. Therefore, a further extension is required to satisfy the requirement of location privacy.

### 4.3 Architecture MA3. Enhancing Mobility-Equipped Tor with Location Privacy

The main problem with the architecture MA2 is the exposure of the mobile node's location to the Tor entry node, so that the movement of the mobile node will be traceable. In this section, we present a further enhancement to this design, by forcing the mobile node (i.e. the Tor client) to change its circuit every time it moves to a new network. By this enforcement, it will ensure that the Tor entry node will always be different. The restriction is that all other circuits must have the same exit node in order to ensure that the TCP connection between the exit node and the server can continue functioning. Fortunately, the circuits are established *a priori*. This mechanism will allow the mobile node to establish the circuits prior to its movement and hence, the swapping between one circuit to another will not cost too much delay. An additional data must be inserted into the cell's component to allow the exit node to concatenate the connection to the server between the old and the new circuit. The detail of this design and implementation is as follows.

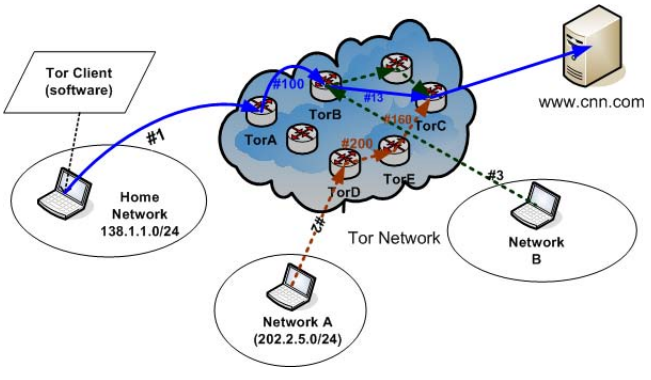


**Fig. 3.** Illustration of Mobile Node's Movement to achieve Location Privacy

*Initialization*. Prior to the network activity (and network movement), the mobile node (i.e. the Tor client) must establish several circuits that use the *same* exit node and store them in its circuit pool. These circuits are inactive when they are not in use.

*Mobile Node Movements*. When the mobile node moves to a new network, it will firstly acquire a new IP address. Then, it selects a new circuit from the available circuit pool. As the circuit has been established *a priori* with the mobile node's initial IP address, it also needs to be updated with the new IP address that has just been acquired. Then, we employ the same mechanism as used in Architecture MA2. That is, the mobile node sends a *Resume* command to the Tor entry node. However, this time it is the Tor entry node of the new circuit.

Then, the mobile node sends a relay cell to the exit node through the new circuit's connection aiming at switching the circuit. The relay cell consists of the following components: a command to notify the exit node to switch the circuit (*ResumeCon*) and a connection identifier that the mobile node uses to notify the exit node of the same destination (*CID*). Once the exit node receives the relay cell, it decrypts the packet (*aka* onion layer). Then, it executes the command by searching its database for the circuit that is currently used with the connection to the server using *CID*, i.e. the old circuit. Finally, it deactivates the old circuit (by removing *CID* from the old circuit's record) and activates the new circuit with the connection to the application server. Note that we name the relay cell's circuit as the new circuit. Due to page limitation, we refer the readers to [23] for more details and diagrams.

*Analysis*. It is clear that the Tor entry node cannot trace the location of the mobile node. This is due to the fact that the circuit ID and the Tor entry node are always changed when the mobile node moves to a new network. Hence, there is no need to encrypt the circuit ID between the mobile node and the Tor entry node to provide location privacy against the observer. Moreover, even though the exit node can obtain a list of its previous nodes of all circuits belonging to the connection from the mobile node to the application server, it does not have enough information to trace the movement of the mobile node, since there is more than a hop that connects the exit node to the mobile node. We note that by allowing the number of hops in a circuit to vary, we can achieve a better and efficient location privacy protection as it is harder for the adversary to predict even the size of the circuit.

The assumption put in place in Tor networks includes the following. On one extreme, we note that the collusion of all nodes is not permitted, or else the anonymity properties, i.e. sender anonymity, receiver anonymity and unlinkability cannot be provided. On the other extreme, we also note that Tor does not require that all Tor nodes must be trusted either. We note that these two assumptions are indeed valid in practice.

**Theorem 1.** *Our design* MA3 *provides mobility, anonymity and location privacy according to our definition in Section 2.*

*Justification*. The mobility of our design MA3 is provided by the inherent Tor networks. For the anonymity, we should consider the three properties, namely sender anonymity, receiver anonymity and unlinkability. In the following, we briefly show that the security of our design can be reduced to the security of Tor.

*Sender Anonymity.* Consider the following game between $\mathcal{A}$ and $\mathcal{C}$. Assume that $\mathcal{A}$ is an attacker that can break the sender anonymity interaction in our design. In this setting, we set $\mathcal{C}$ as an observer to a Tor network in the real world. Firstly, $\mathcal{C}$ provides all the required Tor parameters to $\mathcal{A}$ and a set of senders $\{\mathcal{S}_1, \mathcal{S}_2, \cdots, \mathcal{S}_n\} \in \mathcal{S}$. The attacking phase can be done by $\mathcal{A}$ by querying $\mathcal{C}$ for any particular sender $\mathcal{S}_j \in \mathcal{S}$ for a particular message $m_j \in \mathcal{M}$. To answer this query, $\mathcal{C}$ can invoke the real world that contains the Tor networks and obtain the real transcript from the Tor networks. The transcript will be provided to $\mathcal{A}$ and hence, the simulation runs completely. The view of the simulated environment is identical to the real world, and hence, the simulation is perfect. Finally, $\mathcal{A}$ outputs two senders $\mathcal{S}_0, \mathcal{S}_1$ of his choice and a target message $m^*$ that has not been queried before and $\mathcal{C}$ provides a transcript $\Omega_i$ for a coin toss $i \in \{0, 1\}$. Then, $\mathcal{A}$ can output the choice of $i$ that $\mathcal{C}$ selected. Note that this output means that $\mathcal{A}$ has successfully break the underlying sender anonymity of the Tor network in the real world, and hence we obtain contradiction.

*Receiver Anonymity and Unlinkability.* Receiver anonymity and unlinkability can be done in similar fashion as above. The underlying idea is to show if there exists an adversary $\mathcal{A}$ who can break the interaction, then this adversary will also break the underlying Tor networks. Therefore, the contradiction is obtained.

*Location Privacy.* When we consider the mobile node as the receiver of the communication, location privacy interaction is similar to the receiver anonymity, except the location of the receivers can vary. The attacker will not be able to break the location privacy interaction since the circuit for each different location will also be different. If the attacker can break the location privacy interaction in our design, it means that the attacker is capable to observe the whole structure of the Tor networks, and hence, the adversary is in fact a global adversary. The fact that a global adversary does not exist means that our design is secure against location privacy.

## 5 Conclusion and Further Works

In this paper, we presented a mechanism to achieve mobility and anonymity in IP-based networks concurrently. We started the paper by firstly defining the required properties, that include mobility, anonymity and location privacy. We noted that adding mobility to an IP-based network will imply losing location privacy. We presented a concrete design and implementation based on the existing IP-based network to achieve both mobility and anonymity at the same time. We note that our work in this paper can be considered as the first step towards formalizing mobility, anonymity and location privacy. In our future work, we will consider the location attribute in our design and therefore we can achieve a more robust model. Therefore, our future work will be able to capture a more powerful adversary and a broader scenario.

## References


1. Freedom Network, http://www.freedom.net/
2. Onion Router History, http://www.onion-router.net/history.html
3. Chaum, D.: Untraceable Electronic Mail, Return Address, and Digital Pseudonyms. Comm. of ACM 24(2), 84–88 (1981)
4. Choi, S., Kim, K., Kim, B.: Practical Solution for Location Privacy in Mobile IPv6. In: Chae, K.-J., Yung, M. (eds.) Information Security Applications. LNCS, vol. 2908, pp. 69–83. Springer, Heidelberg (2004)
5. Diaz, C., Seys, S., Claessens, J., Preneel, B.: Towards Measuring Anonymity. In: Dingledine, R., Syverson, P.F. (eds.) PET 2002. LNCS, vol. 2482, Springer, Heidelberg (2003)
6. Droms, R.: Dynamic Host Configuration Protocol, RFC 2131 (1997)
7. Escudero-Pascual, A., Hdenfalk, M., Heselius, P.: Flying Freedom: Location Privacy in Mobile Internetworking. In: INET2001, CD-proceedings. (2001)
8. Fasbender, A., Kesdogan, D., Kubitz, O.: Analysis of Security and Privacy in Mobile IP. In: 4th International Conference on Telecommunication Systems Modeling and Analysis (1996)
9. Freedman, M.J., Morris, R.: Tarzan: A Peer-to-Peer Anonymizing Network Layer. In: CCS 2002, USA (2002)
10. Goldberg, I.: A Pseudonymous Communications Infrastructure for the Internet. PhD thesis, UC Berkeley (2000)
11. Johnson, D., Perkins, C., Arkko, J.: IP Mobility Support for IPv6, RFC 3775 (2004)
12. Jones, A.: Anonymous Communication on the Internet (September 2004), http://www10.cs.rose-hulman.edu/Papers/Jones.pdf
13. Koodi, R.: IP Address Location Privacy and Mobile IPv6: Problem Statement RFC 4882 (May 2007)
14. Koponen, T., Gurtov, A., Nikander, P.: Application Mobility with HIP. In: Proc. ICT 2005 (2005)
15. Marc Rennhard, B.P.: Practical Anonymity for the Masses with Morphmix. In: Juels, A. (ed.) FC 2004. LNCS, vol. 3110, Springer, Heidelberg (2004)
16. Montenegro, G.: Reverse Tunneling for Mobile IP, RFC 2344 (May 1998)
17. Perkins, C.: IP Mobility Support for IPv4, RFC 3344 (2002)
18. Reed, M., Syverson, P., Goldschlag, D.: Anonymous Connections and Onion Routing. IEEE Journal on Selected Areas in Communications 16(4), 482–494 (1998)
19. Reiter, M., Rubin, A.: Crowds: Anonymity for Web Transactions. ACM Trans. Inf. Syst. Secur. 1(1), 66–92 (1998)
20. Roger Dingledine, P.S., Mathewson, N.: Tor: The Second-Generation Onion Router. In: Proc. of the 13th USENIX Security Symposium (2004)
21. Salkintzis, A.K. (ed.): Mobile Internet: enabling technologies and services. CRC Press, Boca Raton (2004)
22. Valko, A.G.: Cellular IP: a new approach to Internet host mobility. SIGCOMM Comput. Commun. Rev. 29(1), 50–65 (1999)
23. Wiangsripanawan, R., Susilo, W., Safavi-Naini, R.: Achieving Mobility and Anonymity in IP Based Networks (full version). Available upon request from the first author