Chameleon and the Identity-Anonymity Paradox: Anonymity in Mobile Ad Hoc Networks

Leonardo A. Martucci, Christer Andersson, and Simone Fischer-Hübner

Karlstads University, Department of Computer Science
Universitetsgatan 2, 651-88 Karlstad, Sweden
{leonardo.martucci, christer.andersson, simone.fischer-huebner}@kau.se

Abstract. In this paper we first present the identity-anonymity paradox, which explains why identities are needed to achieve reliable anonymity. Then, we introduce Chameleon, a novel anonymous overlay network for mobile ad hoc environments, and describe it in details with the support of state transition diagrams. To the best of our knowledge, this is the first low-latency anonymous communication mechanism designed for a mobile ad hoc network setting.

1 Introduction

Mobile ad hoc networks are constituted of mobile platforms that establish on-the-fly wireless connections among themselves, and ephemera networks without central entities to control it. The quest for privacy in mobile ad hoc networks is currently focused on introducing anonymity in the network layer, with several anonymous routing protocols being recently proposed [12, 22, 5]. However, such solutions prevent the usage of standardized ad hoc routing protocols, meaning, in practice, that all network nodes must run a non-standard routing protocol. Our proposal, Chameleon, is an anonymous overlay network tailored for mobile ad hoc environments, aiming, with reasonable performance costs, to provide sender anonymity against recipients and relationship anonymity against local observers. In addition, Chameleon provides conditional anonymity against malicious Chameleon users, as well as protection against single attackers trying to compromise large portions of a network by assuming multiple identities. Chameleon builds on a flexible design that provides isolation and independence from both the application and transport layers, allowing the usage of standardized mobile ad hoc routing protocols. To the best of our knowledge, Chameleon is the first low-latency anonymous overlay network being applied in a mobile ad hoc setting. Another overlay anonymous communication mechanism was recently presented by Jiang *et al.* [10], who propose a number of adaptations to make Chaum's classical mix concept [7] suitable for ad hoc networks. However, their solution is not low-latency, since it uses stop-and-go mixes and suggests the usage of bandwidth-consuming dummy-traffic.

Chameleon was specially designed with the characteristics of mobile ad hoc environments in mind. Therefore, when designing Chameleon, key characteristics of those environments, such as limited battery lifetime, user mobility and vanishing nodes, for instance, were taken into account. The core functionalities of Chameleon are inspired by the traditional Crowds system [16] for anonymizing HTTP traffic. This decision was made according to a previous evaluation of Peer-to-Peer (P2P) based anonymous overlay networks in the context of ad hoc networks [3]. Although none of the studied techniques were fully compliant with the characteristics of mobile ad hoc networks, the Crowds system [16] was deemed as an appropriate choice for a foundation upon which Chameleon could be developed. A number of adaptations to Crowds were made. For example, Chameleon enables end-to-end encryption between a sender and a recipient, employs certificates to hinder attackers from assuming multiple identifies, and acts as a general overlay network accepting all messages from the application layer.

The rest of this paper is organized as follows. Section 2 presents a discussion regarding identification and anonymity in mobile ad hoc networks, which we called the identity-anonymity paradox. In Section 3 we introduce Chameleon by describing its architecture and assumptions. In Section 4 we present a detailed description of Chameleon with the support of state-transition diagrams. Section 5 presents the theoretical analysis of the Chameleon protocol. Finally, Section 6 presents concluding remarks and future research plans.

2 The Identity-Anonymity Paradox

In order to implement identities in Chameleon, each Chameleon node owns a set of certificates used to authenticate against other Chameleon nodes. We assume that certificates are obtained either by a side-channel, or when the nodes are in contact with the certificate authority, possibly located in a fixed network. This section discusses why digital certificates were selected as identifiers in Chameleon, and also why we consider that the most reasonable option for all anonymous communication mechanisms and also security models for mobile ad hoc networks to be proposed from now on.

By definition [8], mobile ad hoc networks *may* operate in isolation – that is, in the absence of any fixed infrastructure. Therefore, the concept of autonomous systems is not applicable in mobile ad hoc environments, as there is no entity controlling the network and providing services such as routing, security or addressing[1]. The lack of standardized addressing schemes allows network nodes to change their IP addresses (and MAC addresses as well), or even to have multiple network interfaces (either real or virtual) with multiple identifiers. Thus, obtaining unique, persistent and trustworthy identifiers from layers below application (regarding the TCP/IP model) is not realistic. The consequence of such fact is that traditional identification systems that rely on the usage of network or data link information are basically useless in such environments.

The lack of reliable network and data link identification might give the impression that nodes in mobile ad hoc networks are naturally anonymous, especially if we consider using the Sybil attack[2] [9] as an enabler for achieving anonymity. The Sybil attack would allow the usage of multiple identifiers simultaneously with a lifetime equivalent to the lifetime of one session or TCP connection, for instance. Therefore, both IP and

[1] There are currently no standards for IP assignment in mobile ad hoc networks. Recently, the Autoconf Internet Engineering Task Force (IETF) Working Group [2] was assigned to study, among other questions, the problem of addressing in mobile ad hoc networks.

[2] In a Sybil attack, malicious users assume multiple identities, preventing the usage of security mechanisms based on filters or trust assumptions.

MAC addresses would constantly change and, in principle, it would not be possible to associate or track those identifiers.

Although the concepts of anonymity and identities can be understood as opposites, without identities, reliable anonymity is not achievable in mobile ad hoc environments. First, because such scheme would be vulnerable to traffic analysis and positioning techniques. Furthermore senders and recipients could be easily pinpointed and their relationships exposed since both senders and receivers establish direct connections, thereby, having their anonymity properties compromised. In addition, the lack of persistent identities is harmful for the network sanity, since all security mechanism for mobile ad hoc networks would hold without some form of trustworthy identifiers. We named this need of identifiers to achieve anonymity as the *identity-anonymity paradox*.

The consequences of this paradox and its relation with the Sybil attack lead to a clear interpretation of the definition of mobile ad hoc networks in the RFC 2501 regarding the operation in isolation and a better understanding of the foundations behind the issue of identifiers in proposed security mechanisms for mobile ad hoc environments. A taxonomy of such mechanisms is presented below, where security models are classified into three families regarding the way that identifiers are generated and obtained:

i. *intermittently connected to an established infrastructure* – security models belonging to this group assume that mobile ad hoc networks connect periodically (or at least occasionally) to an established infrastructure, such as the Internet. Therefore, it is possible to rely on the established security infrastructure that already exists in the Internet, such as a PKI (Public Key Infrastructure), and therefore, distribute digital certificates among the participants of an ad hoc network. Security schemes in this group include proposals that rely on Internet access [11] and proposals combining crypto-based techniques [4] with digital certificates;

ii. *setting a Certificate Authority in the mobile ad hoc network* – the assumption is that one or more devices have a special role in the network, such as personal Certificates Authorities (CA) and repositories. These CA are responsible for issuing certificates or credentials to devices in the mobile ad hoc networks. There are two basic approaches to set one or more CA in a mobile ad hoc network:

   (a) one or more devices have a special role in the network, such as issuing certificates and publishing revocation lists, for instance. Solutions such as the Resurrecting Duckling model [18] are based on a central device that controls the network. In Martucci *et al.* [14], a security architecture is presented using multiple CA-like devices that control and secure a service-oriented ad hoc network. These solutions can operate isolated from an established infrastructure, although one or more nodes play a special role regarding security;

   (b) a set of ad hoc network devices has parts of a private key that is used to issue certificates usually based on threshold cryptography. As long as a sufficient part of these nodes is the network range, digital certificates can be issued. Threshold cryptography was first proposed in the context of ad hoc networks in Zhou and Haas [23]. How many nodes and which nodes are needed to issue a certificate is usually implementation dependent;

iii. *PGP-like (Pretty Good Privacy) security models* – the assumption is that every device has one or more public/private key pairs and that every device can issue its own certificates and distribute them as well. Security often relies on the concept of web of trust. Such solutions are distributed enough to operate in complete isolation from any deployed infrastructure, however there are absolute no guarantees regarding protection against Sybil attacks, what is a major drawback of security models belonging to this family, such as the proposal of Capkun *et al.* [6] for instance.

Several conclusions can be drawn when putting the aforementioned taxonomy, the RFC 2501 definition and identity-anonymity paradox into the same picture. First, security schemes for ad hoc networks need to guarantee the uniqueness of the network identifiers, usually by the means of digital certificates. Second, the provisioning of reliable anonymous communication for nodes in a mobile ad hoc network, persistent identifiers are also needed. Third, to achieve reliable certificate distribution in ad hoc networks to prevent Sybil attacks, some sort of trusted third party (either centralized or distributed) is needed, which includes solutions from families *i* and *ii*, but not from family *iii*. Finally, regarding the RFC 2501 definition, to our understanding, a mobile ad hoc network may either depend intermittently on some deployed infrastructure (and therefore may operate in isolation for a given time frame) or it could operate in complete isolation from the deployed infrastructure, given that some support systems (a third trusted party) is deployed in the mobile ad hoc network.

Given all the aforementioned reasons, identities in Chameleon are implemented as digital certificates. The strategy for issuing and distributing identifiers depends on the security model chosen. From the point of view of the security model, Chameleon is an add-on for providing anonymous communication.

3 Chameleon: an Anonymous Overlay Network

The idea of Chameleon is that one user's action is hidden within the actions of many other users. By sending messages through virtual paths, a user can participate in a communication session while at the same time hiding his identity among the identities of the other users in the mobile ad hoc network.

A virtual path functions by routing encrypted messages through chains of nodes. To protect against traffic analysis, the appearance of the messages is changed at each node in the path through encryption. Generally, there are two main strategies for constructing virtual paths for anonymous overlay networks. One approach, applied in layered encryption approaches, is to let the first node decide the whole path by wrapping a message in several layers of encryption – one for each intermediary node along the path. These layers are thereafter peeled off (by decryption), one by one, at each subsequent node on the path. In the second strategy, the first node decides its successor, and then the intermediate nodes decide their respective successors, until some node decides to end the path, based on some criteria, and then forwards the message to the destination.

To deal with high mobility and to enable efficient path repairing in case of disappearing nodes, Chameleon employs the latter strategy for establishing virtual paths. Therefore, during path establishment, the decision of extending the path or not depends on the result of the toss of a biased coin, which bias is determined by a "probability of forwarding" p_f, where p_f is bounded by the interval $[0.5, 1)$. With the probability

$(1 - p_f)$, the path is ended and a connection is established with the destination; otherwise the path is extended to another randomly chosen node, at which the same process is repeated. The path length L is thus probabilistic and denotes the sum of the appearances for each node on the path (excluding the destination node), and $\min(L) = 2$. The expected path length, L_{exp}, is given in equation (1) [16]:

$$L_{exp} = (p_f)/(1 - p_f) + 2 \tag{1}$$

Virtual paths are bidirectional, meaning that messages can travel forward (towards the destination) or backward (towards the source). As in Crowds, the destination's IP address is known only to the nodes belonging to the path, and path rebuilding is performed in the forward direction only (to enable path rebuilding also in the backward direction, intermediary nodes would require greater knowledge about the path and, eventually, the identity of the sender). To provide better protection against local observers, link encryption is employed between the nodes in the virtual path. Unlike Crowds, conditionally on the destination type, end-to-end encryption may also be applied between the sender and destination (see Section 4). Finally, Chameleon relies on the following assumptions:

i. It is expected that certificates are obtained a priori from a third trusted party, which is, most likely, located in a fixed network. Whether this assumption collides or not with the definition of mobile ad hoc networks in RFC 2501 [8] is polemic among authors in the field. In our opinion, it is expected for a node in a mobile ad hoc network to have occasional contact with a fixed network and, therefore, to a set of trusted devices. This assumption is also present in other papers dealing with the problem of anonymity in ad hoc networks, such as [12, 22, 5];
ii. Chameleon assumes that it is possible to establish secure sessions in the transport layer, with mutual authentication using digital certificates and symmetric key establishment. Secure sessions can be achieved using standard protocols, such as TLS.
iii. Since the IP and hardware addresses are not necessarily unique identifiers that can be linked, with a long-term one-to-one relationship, to a corresponding user, we assume that the mobile ad hoc environment is a service-based network, such as Jini [15], SLP (Service Location Protocol) [20] or UPnP [19] networks. Therefore, all network services, including potential anonymity services, are announced through a localization (or directory) service, such as Jini's Lookup Server.

4 Chameleon Protocol Description

In the remainder of this paper, we use the following notation for describing the networks nodes in a Chameleon scenario:

i. Ψ denotes the set of nodes $\{\psi_1, \psi_2, ..., \psi_n\}$ situated in the mobile ad hoc network;
ii. Γ denotes the set of Chameleon users $\{\gamma_1, \gamma_2, ..., \gamma_n\}$, where $\Gamma \subset \Psi$. A virtual path is defined as a path connecting the sender, γ_s, with the last node before the destination, γ_{last}, where γ_s and γ_{last} are interconnected by zero or more nodes from Γ. When we describe the protocol, γ_i denotes the current node. The cardinality of Γ is denoted $|\Gamma|$ (where $|\Gamma| \in \mathbb{N}$), and $min(|\Gamma| = 3)$, since this is the minimum amount of members needed to provide some level of anonymity;

iii. D denotes the destination, which can be classified in three disjoint sets: $D_{\overline{sec}}$ accepts only unencrypted requests; D_{sec} accepts secure requests using a standard secure transport protocol between γ_{last} and D, and; D_{Γ} understands Chameleon protocol messages, enabling end-to-end encryption between γ_s and D;
iv. $\Phi \subset \Gamma$ denotes a set of decentralized directory servers $\{\phi_1, \phi_2, ..., \phi_n\}$ announcing the set of network addresses of the nodes in Γ, IP_{Γ}, along with their digital certificates, to other nodes in Γ. To reveal as little as possible information to Φ, each node in Γ requests IP_{Γ} at regular time intervals. The restriction $\Phi \subset \Gamma$ decreases the likelihood of corrupted directory servers announcing false information, since they can be detected as malicious nodes and filtered out by other Chameleon users. The announcement of IP_{Γ} follows one of the main principles of zero configuration networking [21], which assumes the existence of a service discovery system in network environments such as mobile ad hoc networks. The nodes in Φ act as a distributed version of the blender in Crowds.

The following notation is used for the messages types in Chameleon:

i. θ denote application data passed to Chameleon from the application layer;
ii. m_{γ_i,γ_j} denote messages passed between Chameleon nodes γ_i and γ_j via the lower layers. The messages m_{γ_i,γ_j} are link encrypted between γ_i and γ_j using the symmetric key $E_{k_{\gamma_i,\gamma_j}}$ (established using a secure transport layer protocol). For the cases where $D \in D_{sec}$ or $D \in D_{\overline{sec}}$, the payload of m_{γ_i,γ_j} includes: IP_D – the IP address of D; $p\#_{\gamma_i,\gamma_j}$ – a path identifier (a randomly generated integer for identifying packet streams between nodes γ_i and γ_j); and the data payload θ – see equation (2), where $\cdot$ denotes concatenation. For the case where $D \in D_{\Gamma}$, m_{γ_i,γ_j} has two optional fields to achieve end-to-end encryption and data integrity – see equation (3). The first field contains a symmetric key $k_{\gamma_s,D}$, which is encrypted with the D's public key, Pu_D. The symmetric key $k_{\gamma_s,D}$ is used to set an end-to-end secure channel between γ_s and D. The second field is used to send the output of a keyed-hash function for message integrity, with input data θ and key $k_{\gamma_s,D}$;

$$m_{\gamma_i,\gamma_j} = E_{k_{\gamma_i,\gamma_j}}[p_{\#\gamma_i,\gamma_j} \cdot IP_D \cdot \theta] \tag{2}$$

$$m_{\gamma_i,\gamma_j} = E_{k_{\gamma_i,\gamma_j}}[p_{\#\gamma_i,\gamma_j} \cdot IP_D \cdot E_{k_{\gamma_s D}}[\theta] \cdot E_{Pu_D}[k_{\gamma_s D}] \cdot hash_{k_{\gamma_s,D}}(\theta)] \tag{3}$$

iii. An acknowledgment message is generated in γ_{last} and sent towards γ_s to inform that a message has reached its destination. Equation (4) describes the $ack\gamma_{i+1}, \gamma_i$ acknowledgement message sent from γ_{i+1} to γ_i.

$$ack_{\gamma_{i+1},\gamma_i} = E_{k_{\gamma_{i+1},\gamma_i}}[p_{\#\gamma_{i+1},\gamma_i}] \tag{4}$$

Each node in Chameleon maintains a routing table with the following entries: the destination's IP address (IP_D); the backward and forward path identifiers ($p_{\#\gamma_{i-1},\gamma_i}$ and $p_{\#\gamma_i,\gamma_{i+1}}$); the address of the preceding and succeeding nodes in the virtual path ($IP_{\gamma_{i-1}}$ and $IP_{\gamma_{i+1}}$) and; the time-to-live (TTL) counter, a decremental counter indicating the remaining lifetime of a given entry in the table. The path identifiers are managed in the same way as the *path_id* in Crowds [16]. In Chameleon, the tuple $[IP_{\gamma_i}, IP_{\gamma_{i+1}}, p_{\#\gamma_i,\gamma_{i+1}}]$ identifies a path connection between two nodes γ_i and γ_{i+1}. A Chameleon node can

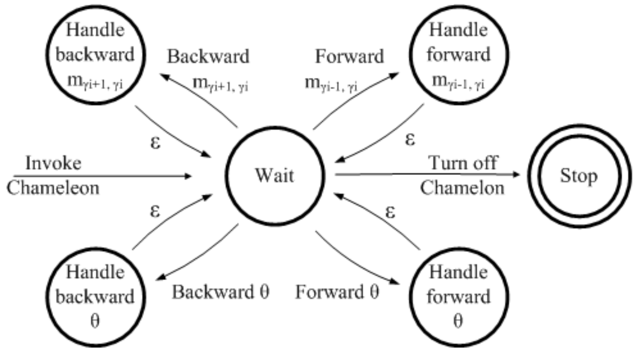
Handle backward $m_{\gamma i+1, \gamma i}$
Backward $m_{\gamma i+1, \gamma i}$
Forward $m_{\gamma i-1, \gamma i}$
Handle forward $m_{\gamma i-1, \gamma i}$
ε
ε
Invoke Chameleon
Wait
Turn off Chamelon
Stop
ε
ε
Handle backward θ
Backward θ
Forward θ
Handle forward θ

Fig. 1. The Chameleon main state transition diagram for each node in Chameleon. A node can play the roles of γ_s, γ_i, or γ_{last}, depending on the type the incoming message.

be described as a local proxy server following the state transition diagram in Figure 1. Its role is threefold; first, it may serve as the user's local proxy to which the user's applications forward their data, θ. In this case the node constitute the first node on the virtual path, γ_S. This situation is represented by the "Handle forward θ" state in Figure 1, which in turn can be expanded to the diagram in Figure 2. In the second case, a node can be an intermediary peer in one or more virtual paths. This situation is represented by the "Handle forward $m_{\gamma_{i-1},\gamma_i}$" (which can be expanded to the diagram in Figure 3) and "Handle backward $m_{\gamma_{i+1},\gamma_i}$" state in Figure 1 depending on the message direction. Finally, a node can act as the last peer in a virtual path, γ_{last}. In this case, it acts as a proxy server towards D.

In the remainder of this section, we key out the protocol details by (1) describing virtual path establishment, (2) describing how data is sent from γ_s to D, and, (3) describing how virtual paths are repaired in the event of a path break.

A. *Building virtual paths.* In Chameleon, virtual paths are constructed as follows, assuming that there is no entry in the routing table for the designated IP_D:

  (i) Path establishment is initiated when a node γ_s receives θ from the application layer. Then, γ_s randomly selects[3] a node γ_1 from Γ, as visualized in the "Select γ_1" state in Figure 2. Then, γ_s and γ_1 establish a secure session in the transport layer, exchanging a symmetric key k_{γ_s,γ_1} for link encryption. The sender γ_s then assembles and encrypts m_{γ_s,γ_1} (in which θ is piggy-backed) and forwards m_{γ_s,γ_1} to γ_1 ("Send m_{γ_s,γ_1} to γ_1" state in Figure 2). In cases when γ_s cannot send m_{γ_s,γ_1} to γ_1, it selects another new random node γ_1 from Γ and repeats the process;

  (ii) Now, γ_i (i. e., $i = 1$), triggers the state transition diagram in Figure 3, and starts by decrypting $m_{\gamma_{i-1},\gamma_i}$. Assuming there is no corresponding entry for $m_{\gamma_{i-1},\gamma_i}$ in the Chameleon routing table of γ_i, a biased coin is tossed ("Toss biased coin" state in Figure 3). If the decision of the coin toss is to end the path,

[3] If γ_s posses no recent information about Γ, it contacts a directory server ϕ_i and requests this information. The nodes γ_s and ϕ_i mutually authenticate using their certificates.

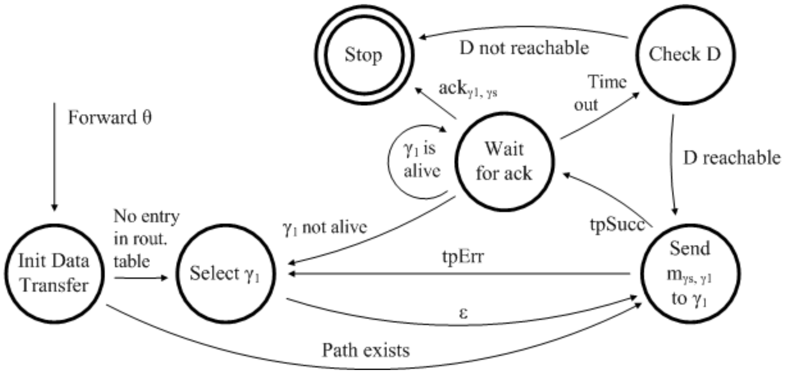

Fig. 2. State transition diagram for a node γ_s receiving data from the application layer. The acronyms *tpSucc* and *tpErr*, used in this section, denote transitions indicating whether the sending of a message was accomplished successfully (*tpSucc*) or not (*tpErr*) in the transport layer.

θ (encapsulated in $m_{\gamma_{i-1},\gamma_i}$) is forwarded to D. In this case, γ_i becomes the last node in the virtual path, γ_{last}. Otherwise, the path is extended one hop and a new node γ_{i+1} is selected randomly from Γ. The message $m_{\gamma_i,\gamma_{i+1}}$ is then encrypted and forwarded to γ_{i+1}, where this process is repeated. Eventually, a path will be established between γ_s and γ_{last}, where γ_s and γ_{last} are interconnected by zero or more intermediary Chameleon nodes.

B. *Sending and forwarding data*. In Chameleon, data is passed from γ_s to D in the following way, assuming that a virtual path is already established:

(i) When γ_s receives θ from an application, γ_s assembles and encrypts m_{γ_s,γ_1}, and sends it to γ_1, as depicted in the "Send Message m_{γ_s,γ_1} to γ_1" state in Figure 2;

(ii) Regarding the intermediary nodes, an incoming $m_{\gamma_{i-1},\gamma_i}$ is treated according to the state transition diagram depicted in Figure 3. At each node, $m_{\gamma_{i-1},\gamma_i}$ is decrypted, and $m_{\gamma_1,\gamma_{i+1}}$ is generated and encrypted before being forwarded. Eventually, the last node on the path, γ_{last}, will receive $m_{\gamma_{last-1},\gamma_{last}}$. Then, γ_{last} sends θ to D (either encrypted or unencrypted, depending on the destination type, see Section 4). Provided that the connection with D was successful, $ack_{\gamma_{last},\gamma_{last-1}}$ is sent backwards along the path to acknowledge γ_s that D did receive θ;

(iii) The sending of data in the backward direction is initiated when γ_{last} receives θ from D. Then, γ_{last} encapsulates θ in $m_{\gamma_{last},\gamma_{last-1}}$ and sends it to γ_{last-1} on the virtual path. Since messages travelling in the backward direction are not acknowledged, the state transition diagram in Figure 1 always returns to the "Stop" state, independent of whether or not it was possible to send the message to γ_{last-1}. This process is repeated at each intermediary node until the message eventually reaches γ_s. If a timeout threshold is exceeded, the "Check D" state is invoked (Figure 2), where γ_s checks the status of D (this is possible since the ad hoc network is a service-based network). The timeout should be large enough to allow intermediary nodes to conduct path repairing, but, on the other hand, not too large, since this would risk to compromise the protocol performance.

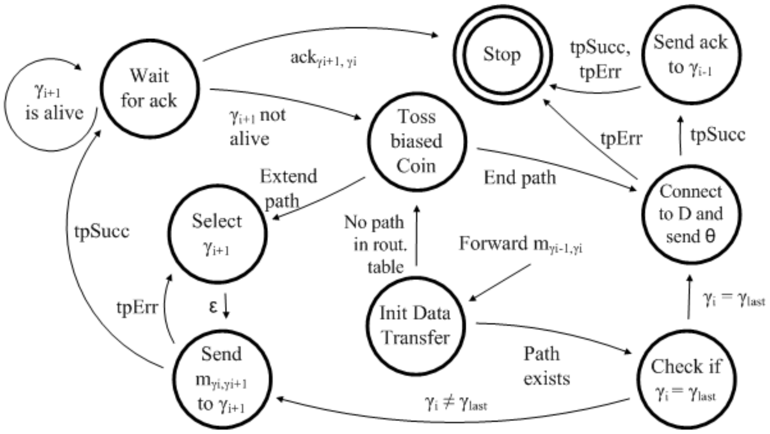

Fig. 3. State transition diagram for a node γ_i receiving a message $m_{\gamma_{i-1},\gamma_i}$, including path repairing.

C. *Repairing virtual paths.* Path repairing is initiated in two situations: first, when γ_i fails to send $m_{\gamma_i,\gamma_{i+1}}$ to γ_{i+1}, and, second, when γ_i waits for $ack_{\gamma_{i+1},\gamma_i}$ and notices that γ_{i+1} is not alive (γ_i polls γ_{i+1} at regular intervals during the "Wait for $ack_{\gamma_{i+1},\gamma_i}$" state to assert that γ_{i+1} is still alive, as illustrated in Figures 2 and 3). The node γ_i tosses a biased coin and either forwards θ directly to D or selects a new node γ_{i+1} as its successor in the path. In this way, the path is restored from the point where it was broken, and not from the beginning. No explicit path destruction is conducted after the communication session via the virtual paths has ended. Instead, the TTL field in the routing table ensures that inactive path entries are deleted.

5 Theoretical Analysis

Six different requirements were defined in [3] which an anonymous overlay network should adhere to (at least to an acceptable degree, since the requirements are not orthogonal) in order to be suitable in mobile ad hoc network environments. Next, we list these requirements, and discuss to what extent Chameleon meets them:

1. *Scalability*: the workload on each participant in Chameleon remains virtually constant as the number of participants grows, as in Crowds [16]. It is proved in [16] that for each node in the network, the expected number of virtual paths a node will be appearing on at a particular time is given by: $\frac{1}{(1-p_f)^2} * (1 + \frac{1}{|\Gamma|})$;
2. *Strong anonymity properties*: an anonymous overlay network should provide adequate protection against, for instance, malicious users and different types of eavesdroppers. The Chameleon attacker model is more complete and suitable for mobile ad hoc networks than the one used in Crowds. It assumes that all nodes (attackers included) have the same radio range. The following types of attackers are included:
   (a) *Local observer* ($\psi_{obs} \in \Psi$): a passive observer whose radio range covers γ_s;

Table 1. Degrees of anonymity in Chameleon.

	Sender Anonymity	*Receiver Anonymity*	*Relationship Anonymity*
Local observer (ψ_{obs})	`possible innocence`	`beyond suspicion` (for large networks)	`beyond suspicion` (for large networks)

Malicious insiders (Γ')	`probable innocence`
	if $\lvert\Gamma\rvert \geq \frac{p_f}{(p_f - \frac{1}{2})} * (\lvert\Gamma'\rvert + 1)$
		P(`absolute privacy`)
		$= \left(\frac{\lvert\Gamma\rvert - \lvert\Gamma'\rvert}{\lvert\Gamma\rvert}\right)^{L_{exp}-1}$
			`probable innocence`

Malicious outsider (ψ')	`probable innocence`
	if $L_{exp} \geq 4$
		`probable innocence`
		if $L_{exp} \geq 4$
			`beyond suspicion`

Destination	`beyond suspicion` `for` $\lvert\Gamma\rvert \geq 3$	$-$	`beyond suspicion`

(b) *Malicious insiders* ($\Gamma' \subset \Gamma$): this attacker is represented by $|\Gamma'|$ (collaborating) malicious members of Γ, aiming to occupy all positions on the virtual path;

(c) *Malicious outsider* ($\psi' \in \Psi$): this is a malicious node aiming to control an intermediary node linking a pair of Chameleon nodes in a given virtual path;

(d) *Destination* (D): this attacker attempts to disclose the identity of γ_s;

(e) *Malicious directory servers* ($\phi' \subset \Phi$): these constitute attackers hosting directory services for the purposes of misusing information about Γ, by the means of announcing different subsets of Γ in different instances of ϕ' and then mount a partition attack. Or, alternatively, announce a reduced set of Γ in order to increase the percentile of Γ' nodes in the announced set.

The metric used is the same metric used for evaluating the anonymity properties of Crowds [16]. In this metric, each user is considered separately, and the resulting value spectra is a function of (among other parameters) the size of the anonymity set, the probability of forwarding and the amount of malicious insiders. The degree of anonymity for a subject γ_i can be expressed on a continuous scale ranging from `absolute privacy` to `provably exposed` via `beyond suspicion`, `probable innocence`, `possible innocence` and `provably exposed`. Chameleon offers sender and relationship anonymity against local observers. Unlike Crowds, Chameleon enables both link-to-link and end-to-end encryption for certain destination types. However, due to performance reasons Chameleon does not protect against a global observer[4]. In Table 1, the offered degrees of anonymity in Chameleon are summarized. The proof for these values can be found in [13]. Malicious directory servers are not included in the table since their goal is to compromise the anonymity level by supporting other malicious users. Possible countermeasures against ϕ' include the usage of redundant servers or cycling through Φ. In the extreme case, every node could run an instance of ϕ, but the performance trade-off would be high;

3. *Fair distribution of work*: an anonymous overlay network should be fair regarding the distribution of workload among the participants. A possible source for unfair-

[4] Protection against a global observer can only be achieved if all nodes transmit in a constant rate independently of the real data traffic (i. e., demands the usage of dummy traffic).

ness in Chameleon is the workload implied for the operators of the directory servers Φ. However, this is dependent of the service-based network technology selected.;

4. *Performance-wise lightweight solution*: in order to reduce computational overhead and increase battery lifetime, an anonymous overlay network should generate few messages and perform few public key operations. Chameleon uses public key encryption sparsely and avoids layered encryption. The protocol overhead is low; assuming knowledge about Γ, $2L$ public key operations and $2L - 1$ Chameleon messages are needed to establish a path, where L denotes the path length. In comparison, MorphMix [17] generates $6L + (L - 2)(L + 1)$ messages and needs at least $13L$ public key operations when establishing a path. Additionally, in contrast to Chameleon, the earlier mentioned mix-based proposal by Jiang *et al.* [10] uses nested public key encryption for both path establishment and message transfer. Lastly, no performance consuming dummy traffic is used;
5. *Adherence to the P2P-model*: mobile ad hoc networks are most often assumed to function without the aid of central services [8]. Unlike e. g., Crowds, Chameleon is a P2P-compliant protocol, although all nodes in Γ need to agree on the value of p_f;
6. *Manage a dynamic topology*: in most proposed mobile ad hoc network scenarios, it is assumed that nodes frequently enter and leave the network. Chameleon addresses dynamic topologies by, among other things, an optimized path repairing process in the forward direction. A virtual path is repaired only from the point of breach, in contrast to other approaches that rebuild a broken path entirely from scratch.

6 Conclusions

This paper introduced Chameleon, a low-latency anonymous overlay network tailored for mobile ad hoc networks that provides, for instance, efficient path repairing, and a reduced amount of control messages in comparison to other anonymous overlay networks. Chameleon does not rely on dummy traffic or layered encryption and it was inspired by the Crowds system, although it differs from Crowds in a number of ways, including: end-to-end encryption between the sender and recipient, certificate-based protection against Sybil attacks, and a distributed service discovery mechanism (and also an attacker model consistent with mobile ad hoc networks). We also presented the identity-anonymity paradox, which states the need of persistent identifiers to achieve reliable anonymity in mobile ad hoc networks. Current research plans include analyzing protocol performance by the means of simulation.

References

1. *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC'03)*, New York, NY, USA, 1–3 Jun 2003. ACM Press.
2. IETF Ad Hoc Network Autoconfiguration Working Group. Ad Hoc Network Autoconfiguration (autoconf), 2006. See http://www3.ietf.org/html.charters/autoconf-charter.html.
3. Christer Andersson, Leonardo Martucci, and Simone Fischer-Hübner. Requirements for Privacy-Enhancements for Mobile Ad Hoc Networks. In 3rd *German Workshop on Ad Hoc Networks (WMAN 2005), Proceedings of INFORMATIK 2005 - Informatik LIVE! Band 2*, volume 68 of *LNI*, pages 344–348. GI, 19–22 Sep 2005.

4. Tuomas Aura. Cryptographically Generated Addresses (cga). RFC-3972, Mar 2005. See http://www.ietf.org/rfc/rfc3972.txt.
5. Azzedine Boukerche, Khalil El-Khatib, Li Xu, and Larry Korba. SDAR: A Secure Distributed Anonymous Routing Protocol for Wireless and Mobile Ad Hoc Networks. In *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN'04)*, pages 618–624, 2004.
6. Srdjan Capkun, Jean-Pierre Hubaux, and Levente Buttyàn. Mobility Helps Security in Ad Hoc Networks. In *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC'03)* [1], pages 46–56.
7. David Chaum. Untraceable Electronic Mail, Return Addresses and Digital Pseudonyms. *Communication of the ACM*, 24(2):84–88, Feb 1981.
8. M. Scott Corson and Joseph Macker. Mobile Ad Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations. RFC-2501, Jan 1999. See http://www.ietf.org/rfc/rfc2501.txt.
9. John R. Douceur. The Sybil Attack. In P. Druschel, F. Kaashoek, and A. Rowstron, editors, *Peer-to-Peer Systems: Proceedings of the 1st International Peer-to-Peer Systems Workshop (IPTPS)*, volume 2429, pages 251–260. Springer-Verlag, 7–8 Mar 2002.
10. Shu Jiang, Nitin H. Vaidya, and Wei Zhao. A Mix Route Algorithm for Mix-net in Wireless Mobile Ad Hoc Networks. In *Proceedings of the 1st IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS2004)*, 24–27 Oct 2004.
11. Frank Kargl, Stefan Schlott, and Michael Weber. Identification in Ad Hoc Networks. In *Proceedings of the 39th Hawaiian International Conference on System Sciences (HICSS-39)*. IEEE Computer Society, 4–7 Jan 2006.
12. Jeijun Kong and Xiaoyan Hong. ANODR: ANonymous On Demand Routing with Untraceable Routes for Mobile Ad Hoc Networks. In *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC'03)* [1].
13. Leonardo Martucci, Christer Andersson, and Simone Fischer-Hübner. Chameleon: Anonymous communications for mobile ad hoc networks. Technical report, Karlstad University, Karlstad, Sweden, To be published, 2006.
14. Leonardo A. Martucci, Christiane M. Schweitzer, Yeda R. Venturini, Tereza C. M. B. Carvalho, and Wilson V. Ruggiero. A Trust-Based Security Architecture for Small and Medium-Sized Mobile Ad Hoc Networks. In *Proceedings of the 3rd Annual Mediterranean Ad Hoc Networking Workshop, Med-Hoc-Net*, pages 278–290, Jun 2004.
15. SUN Microsystems. The Jini Architecture Specification – Version 1.2, 2001.
16. Michael Reiter and Avi Rubin. Crowds: Anonymity for Web Transactions. In *DIMACS Technical report*, pages 97–115, 1997.
17. Marc Rennhard and Bernhard Plattner. Introducing Morphmix: Peer-to-Peer based Anonymous Internet Usage with Collusion Detection. In *Proceedings of the Workshop on Privacy in Electronic Society (WPES02)*, 21 Nov 2002.
18. Frank Stajano and Ross Anderson. The Resurrecting Duckling: Security Issues for Ad Hoc Wireless Networks. In *Proceedings of the 3rd AT&T Software Symposium*, Oct 1999.
19. UPnP Forum. UPnP Device Architecture, Version 1.0, Jun 2000.
20. John Veizades, Erik Guttman, Charles E. Perkins, and Scott Kaplan. Service Location Protocol. RFC-2165, Jun 1997. See http://www.ietf.org/rfc/rfc2165.txt.
21. IETF Zero Configuration Networking Working Group. Zero Configuration Networking (zeroconf), 2003. See http://www.zeroconf.org/zeroconf-charter.html.
22. Yanchao Zhang, Wei Liu, and Wenjing Lou. Anonymous Communication in Mobile Ad Hoc Networks. In *Proceedings of the* 24*th Annual Joint Conference of the IEEE Communication Society (INFOCOM 2005)*, Miami, FL, USA, 13–17 Mar 2005.
23. Lidong Zhou and Zygmunt J. Haas. Securing Ad Hoc Networks. *IEEE Network*, 13(6):24–30, 1999.