IN3 Working Paper Series
INTERNET INTERDISCIPLINARY INSTITUTE
ISSN 2013-8644

# Anonymous Routing in Ad-hoc Networks

**Mercedes Rodríguez-García** (mrodriguezgarci@uoc.edu)
**Helena Rifà-Pous** (hrifa@uoc.edu)

*Universitat Oberta de Catalunya*

Working Paper

Working Paper Series WP00-000

Research group: K-ryptography and Information Security for Open Networks: KISON
Research group coordinator: David Megías Jiménez (UOC)

Submitted in: month yyyy
Accepted in: month yyyy
Published in: month yyyy

Universitat Oberta de Catalunya
IN3 INTERNET INTERDISCIPLINARY INSTITUTE
http://in3.uoc.edu


**Internet Interdisciplinary Institute (IN3)**
http://www.in3.uoc.edu
Edifici MediaTIC
c/ Roc Boronat, 117
08018 Barcelona
Espanya
Tel. 93 4505200

**Universitat Oberta de Catalunya (UOC)**
http://www.uoc.edu/
Av. Tibidabo, 39-43
08035 Barcelona
Espanya
Tel. 93 253 23 00

# Table of contents

# Anonymous Routing in Ad-hoc Networks

**Mercedes Rodríguez-García** (mrodriguezgarci@uoc.edu)

**Helena Rifà-Pous** (hrifa@uoc.edu)

**Abstract**
The concurrence of anonymity and scalability in the routing of mobile ad-hoc networks (MANETs) is a challenging issue of great interest. In this paper, we propose an efficient routing protocol that guarantees the anonymity of the sender and the recipient in the most scenarios, as well as the unlinkability between them. Another highlight is that the sender‘s identity remains anonymous before the recipient. The scheme is based on symmetric cryptography but, unlike previous proposals, the exchange of the shared key is performed without revealing the identities of the end nodes. The protocol provides a good level of scalability because no intermediate node has to perform cryptographic operations during the route discovery to verify whether it is the intended recipient.

**Keywords**
Anonymity, unlinkability, security, ad-hoc networks, MANET.

# Introduction

The mobile ad-hoc networks (MANETs) have achieved great prominence among the research community for their important commercial and military applications. Unlike traditional wireless networks, the MANETs do not require the existence of previous infrastructure and allow the immediate establishment of a communication scenario. Their nodes are mobile devices that function as hosts and routers at the same time. In this way, if a node wishes to communicate with other node that is outside its transmission range, a multihop connection is established. The intermediate nodes will help the sender reach the destination thanks to their routing functions.

However, the MANETs present critical and challenging security issues. One of these problems is the users' privacy. To discover a route in the MANETs, the routing protocols flood the network with route request messages. These messages contain the identities of the sender and recipient nodes. In this environment, if an attacker observes the network traffic, it can easily know which nodes are communicating with each other and their identities, a serious threat in private applications.

For the above reasons, our research focuses on the study of anonymous communications in mobile ad hoc networks. In particular, we investigate the following properties (Pfitzmann & Hansen, 2010) (Edman & Yener, 2009):

- *Sender anonymity* (or unlinkability between message and sender node): no node of the network can know the sender's identity, except the recipient.
- *Recipient anonymity* (or unlinkability between message and recipient node): no node of the network can know the recipient's identity, except the sender.
- *Relationship anonymity* (or unlinkability between sender node and recipient node): no node of the network can know which pair of nodes is having a communication, i.e. it is unable to identify either the sender or the recipient or both.
- *Sender anonymity in recipient*: the recipient cannot know the sender's identity.

In a network that preserves the above properties, whether an attacker intercepts a packet[1], it will be unable to identify the sender and the recipient.

Nowadays there are anonymous routing protocols for this type of networks, but very few present a good level of anonymity and scalability at the same time.

In this paper, we analyze and compare the anonymous routing proposals for mobile ad hoc networks. In section 2, we propose a novel efficient routing protocol that meets the above anonymity properties in the most scenarios. In section 3, we perform an

[1] In this paper, we use the terms message or packet interchangeably to represent the information sent from the source node to the destination node.

exhaustive evaluation of the protocol based on the identification of traceable items and the analysis of incoming and outgoing messages. Finally, section 4 concludes and identifies issues that require a further investigation.

# 1. Related work

Several routing protocols have been designed to preserve the anonymity of the nodes that wish to communicate in an ad-hoc network. These schemes use trapdoor functions[2] to hide the identities under pseudonyms.

The solutions proposed in ANODR (Kong & Hong, 2003), AnonDSR (Song, Korba, & Yee, 2005) and EARP (Li, Li, Ma, & Zhang, 2009) use symmetric key cryptography to build the recipient's pseudonym. The key is generated by the sender and must be provided the recipient before routing process. Only AnonDSR defines a method to exchange the key in a security parameter establishment phase (SPE protocol). However, the anonymity is broken because the end nodes' identities go in clear text during the SPE transmission. Any adversary can see them and know that among those nodes there will be a communication.

Unlike previous proposals, SDAR (Boukerche, El-Khatib, Xu, & Korba, 2005) uses public key cryptography to hide the recipient's identity. The asymmetric systems avoid the key exchange but require more computational load to open the trapdoor (Rifà-Pous & Herrera-Joancomartí, 2009) (Rifà-Pous & Herrera-Joancomartí, 2011).

When the source node has to find out a route towards the destination, floods the network with route request messages[3]. In the above protocols, all nodes receiving a route request message have to check the trapdoor of the pseudonym to know if they are the intended recipient. This action overloads the network nodes with unnecessary decryption operations. To minimize the problem, MASR (Pan & Li, 2009) associates an index to the shared symmetric key. Just the node that owns the index in its list will check the trapdoor. However, the index is reused in new communications toward the same destination and may facilitate its traceability.

[2] A trapdoor is a one-way function, it is straightforward to calculate in one direction and is very difficult to compute in reverse direction without the secret.

[3] The on-demand routing protocols, also called reactive protocols, begin to operate when a node wishes to communicate with another node. In that moment the source node sends a route request message (RREQ message) to all neighbors of its transmission range. When a node receives the RREQ message continues broadcasting it until reaching the destination. Each path node appends a layer with route information in an onion structure of the RREQ message. So, each layer of the onion defines a hop in the route.

In this direction TARo (Chen, Boreli, & Sivaraman, 2012) proposes a solution based on keyed one-way hash chains[4]. In a given communication, the source node uses as shared symmetric key and index any pair of consecutive elements of the chain. To avoid the drawback of MASR, each session uses a different pair. But TARo requires sharing a secret with the destination to build the chain and does not specify how.

For the same purpose, (Rifà-Pous, Panaousis, & Politis, 2012) presents a new approach based on asymmetric cryptography and hash functions. The recipient's identity is concealed with two pseudonyms. The first pseudonym is a truncated salted hash of the recipient's public key (high-level trapdoor -HLT-). The second one is a random sequence encrypted with recipient's public key (low-level trapdoor -LLT-). The recipient is the only node who can open the two trapdoors. When an intermediate node receives a RREQ message first checks HLT, a simple computation. Only the nodes able to open HLT, i.e. the hash collisions, will check LLT. In this protocol, the size of the truncated hash determines the anonymity level and the efficiency: a smaller hash provides a system more anonymous but less efficient and vice versa.

Regarding the privacy of the sender, only ANODR guarantees full anonymity during routing process. In this protocol, the sender keeps secret the key of its pseudonym, not even shares it with the receiver.

| PROTOCOL FEATURES | ANODR | ANONDSR | SDAR | TARO | RIFÀ |
|---|---|---|---|---|---|
| SPE PROTOCOL | No | Yes | No | No | No |
| NUMBER OF PSEUDONYMS | 1 | 1 | 1 | 1 | 2 (P1, P2) |
| CRYPTOGRAPHIC SYSTEM | Symmetric | Symmetric | Asymmetric | Symmetric | P1: Hash function<br>P2: Asymmetric |
| KEY INDEX | No | Yes | No | Yes | No |
| NODES THAT CAN OPEN THE TRAPDOOR | Recipient | Recipient | Recipient | Recipient | P1: Set of nodes<br>P2: Recipient |
| NODES THAT HAVE TO CHECK THE TRAPDOOR | All nodes | All nodes | All nodes | Recipient | P1: All nodes<br>P2: Set of nodes |
| SECRET INFORMATION IN CLEAR TEXT | Shared key | Sender's identity<br>Recipient's identity | No | Secret to build the hash chain | No |

**Table 1. Summary of methods to hide the recipient's identity**

[4] A one-way hash chain is a sequence of hash values, where two consecutive values in the chain ($K_{i-1}$, $K_i$) are one-way linkable, i.e., if a node knows $K_{i-1}$ then can calculate $K_i$, but it is unable to perform the calculation in the other direction. The first element is $K_0 = H(ID_s \,||\, ID_d)$, where $ID_s$ and $ID_d$ are the source and destination identities. To generate the other elements of the chain is calculated a hash function of the previous element concatenated to a secret, $K_i = H(K_{i-1} \,||\, \text{secret})$. This secret is a random sequence shared between sender and recipient. In TARo, $K_{i-1}$ is the index and $K_i$ is the shared key.

# 2. Proposed routing protocol

In this section, we describe in detail our anonymous routing protocol. The scheme uses symmetric cryptography to generate the sender's and recipient's pseudonyms. The sender keeps in secret the key of its pseudonym to achieve full anonymity. Unlike previous protocols, sender and recipient share the key of recipient's pseudonym without revealing their identities. The protocol is divided into four phases:

- Security Parameter Establishment phase (SPE phase).
- Route Request phase (RREQ phase).
- Route Reply phase (RREP phase).
- Data transmission phase (DATA phase).

The SPE phase is previous to the routing process and necessary in order to disclose the pair {key of the recipient's pseudonym, index} to the recipient. The scheme uses asymmetric cryptography to hide these parameters and a salted hash function to conceal the recipient's identity during the SPE transmission. The salt guarantees a different hash value in each session for the same identity. Its short lifetime hinders the dictionary attacks. If an adversary discovers the recipient's identity, he cannot link it with later messages of the same communication because:

- The transmission in the SPE phase follows a different path from the taken one in the phases RREQ, RREP and DATA.
- The recipient's pseudonym in the SPE phase is different from used one in the phases RREQ, RREP and DATA.

We assume the inclusion of a Trusted Third Party (TTP node) in the network. The function of this node is twofold: intermediate waypoint in the SPE phase and public key directory. When a node becomes part of the network generates its certificate and provides its public key to the TTP.

The RREQ and RREP phases constitute the routing process. The first finds out the route towards the receiver and it provides to this node the session keys of the intermediate nodes. In the second, the session keys are communicated to sender.

During the routing process, the end nodes' identities are sent under their pseudonyms. The index is used to avoid that the intermediate nodes perform cryptographic operations when checking if they are the destination. The scheme uses local route pseudonyms to identify the link between two adjacent nodes. In the RREQ phase, each intermediate node registers the pseudonym of its link towards the recipient. In the RREP phase, it registers the pseudonym of its link towards the sender. At the end the routing process, the intermediate nodes will only know the pseudonyms of the local links.

In the DATA phase, sender and recipient are ready to start the payload transmission. Both have all session keys and can build cryptographic onion structures to transmit data in a secure way.

## 2.1. Terminology and notations

The terminology and notations used in this scheme are defined below:

- TTP: TTP node (Trusted Third Party).
- S: Sender node.
- R: Receiver node.
- X: Any node.
- $ID_X$: Identity of X.
- $P_X$: Pseudonym of X.
- $PK_X$: Public key of X. This key is available in the public directory of TTP.
- $SK_X$: Private key corresponding to $PK_X$.
- $PK_S$': One-time public key generated by S in each communication and used to build the onion in RREQ phase.
- $SK_S$': One-time private key corresponding to $PK_S$' generated by S in each communication and used to decrypt the onion in RREQ phase.
- K*': One-time secret symmetric key generated by S in each communication and used to build $P_S$. S is the only node that knows this key.
- K': One-time symmetric key generated by S in each communication and used to build $P_R$.
- I: Index of the key K'.
- $E_{KEY}(.)$: Encryption using the key specified in the field KEY.
- $D_{KEY}(.)$: Decryption using the key specified in the field KEY.
- H(.): Hash function.
- Onion: Cryptographic multilayer structure to record the anonymous path towards R. Each node of the path appends a layer to the onion. Thus each layer depicts a hop in the path.
- $PDO_X$: Path discovery onion generated by X in the RREQ phase.
- $PRO_X$: Path response onion received by X in the RREP phase.
- $DTO_X$: Data transmission onion generated by X in the DATA phase.
- $N_X$: Local route pseudonym generated by X. It is a one-time random sequence.
- $K_X$': One-time symmetric key generated by X and used to build the $PRO_X$. Also known as session key.
- Type-xx: Type of message. xx identifies the type and its value can be SPE, RREQ, RREP or DATA. The type depends on the phase in which is generated the message.
- ,: The comma denotes the separation between fields.
- ': The apostrophe denotes the characteristic "one-time" of a key.
- *: The asterisk denotes that the symmetric key is not shared.


IN3 Working Paper Series is a monograph series promoted by the Internet Interdisciplinary Institute (IN3) of the UOC
IN3 Working Paper Series (2010) | ISSN 2013-8644 | http://in3wps.uoc.edu

## 2.2. Security parameter establishment phase (SPE phase)

In this phase the sender provides to the recipient the necessary parameters so that it can verify its pseudonym $P_R$ in later phases. These security parameters are hidden with asymmetric cryptography and sent in a SPE message.

In order to transmit the SPE message by a different path from the taken one in the phases RREQ, RREP and DATA, the route S → R is developed in two stages: S → TTP and TTP →R (figure 1). The TTP node acts as intermediate waypoint. In the following, we describe the process in detail.

**Stage 1**: S → TTP.- S sends to TTP the security parameters that wants to share with the recipient.

1. S generates the security parameters: the one-time symmetric key *K'* and the index *I* for this key.
2. S encrypts the security parameters and the recipient's identity with the public key of TTP: *Parameters* = $\{E_{PK_{TTP}} (ID_R, K', I)\}$
3. S composes the SPE message: *SPE message* = $\{Type\text{-}SPE, ID_{TTP}, Parameters\}$
4. S sends the SPE message to TTP via flooding.

**Stage 2:** TTP → R.- TTP forwards to R the security parameters in a new SPE message.

5. TTP decrypts the field *Parameters* with its private key to obtain $ID_R$, *K'* and *I*: $D_{SK_{TTP}}(Parameters) = ID_R, K', I$.
6. TTP encrypts *K'* and *I* with the public key of R to build the new field *Parameters*: *Parameters* = $\{E_{PK_R} (K', I)\}$.
7. TTP generates a salt and hides the public key of R with a salted hash function: $H(salt, PK_R)$
8. TTP composes the SPE message: *SPE message* = $\{Type\text{-}SPE, salt, H(salt, PK_R), Parameters\}$.
9. TTP sends the SPE message to R via flooding.
10. When a node X receives the *SPE message*, it checks whether $H(salt, PK_X) = H(salt, PK_R)$.
    - If the values match, X may be the recipient. X continues the step 11.
    - If the values do not match, X broadcasts the SPE message.
11. X tries to decrypt the field *Parameters* with its private key: $D_{SK_X}(Parameters)$.
    - If it achieves to decrypt it, X is the recipient and obtains K' and I. X continues the step 12.
    - If it does not achieve to decrypt it, X broadcasts the SPE message.
12. R records in its routing table the entry < *I, K'*>. In this point ends the SPE phase.

At the end of the SPE phase, the recipient will have obtained the parameters in a secure way and without intermediate nodes perform costly cryptographic operations.


IN3 Working Paper Series is a monograph series promoted by the Internet Interdisciplinary Institute (IN3) of the UOC
IN3 Working Paper Series (2010) | ISSN 2013-8644 | http://in3wps.uoc.edu

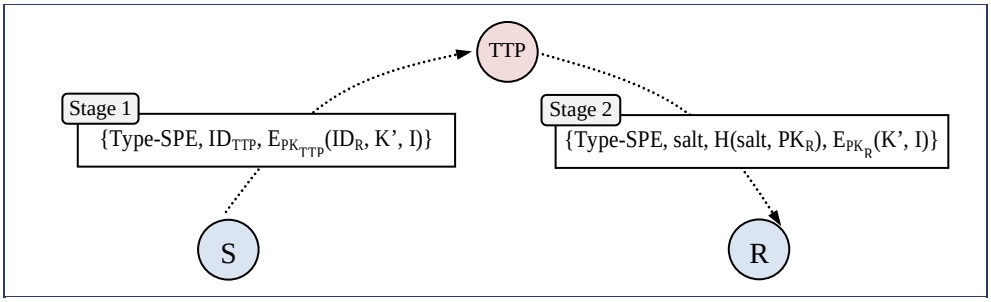


**Figure 1. Stages of the SPE phase**

## 2.3. Route request phase (RREQ phase)

This phase establishes a route between the sender and the recipient. For this, the sender triggers a route request message (RREQ message) to all neighbors of its transmission range. When a node receives the message includes its local route pseudonym and its session key in an onion structure[5] *PDO*. The message goes jumping from node to node until reaching the destination. Once in the recipient, this obtains the full path and all session keys. The figure 2 depicts the process with three intermediate nodes.

To reach the destination, the sender inserts into the RREQ message the recipient's pseudonym $P_R$ and the key index *I*. The sender's pseudonym $P_S$ is also included so that the recipient can send messages back in the next phases of the protocol.

To build the *PDO,* the sender generates the pair $(PK_S', SK_S')$. The public key $PK_S'$ is provided to all nodes of the path in order that they can encrypt their onion layer. The private key $SK_S'$ is provided the recipient through $P_R$ so that only the recipient can decrypt the *PDO* and obtain the routing information and the session keys.

The RREQ phase is initiated by the sender node as follows:

1. S generates the one-time key pair $(PK_S', SK_S')$. As $PK_S'$ is a unique sequence number also is used to identify the communication.
2. S generates the one-time secret symmetric key $K^{*\prime}$. This key is used to build $P_S$.
3. S generates the recipient's pseudonym: $P_R = E_{K'}(ID_R, SK_S')$. As $SK_S'$ and $K'$ change in each RREQ phase, the scheme guarantees a different $P_R$ in each session.
4. S generates its own pseudonym: $P_S = E_{K^{*\prime}}(ID_S)$. As $K^{*\prime}$ changes in each RREQ phase, the scheme guarantees a different $P_S$ in each session.
5. S generates its local route pseudonym: $N_S$.
6. S adds $PK_S'$, $K^{*\prime}$ and $N_S$ to its routing table: $< ID_R, I, K', PK_S', K^{*\prime}, N_S >$

[5] During RREQ and RREP phases, the scheme also uses onion structures to transmit the routing information in an anonymous way.

7. S generates the first layer of the path discovery onion: $PDO_S = \{E_{PK_S'}(N_S, P_S)\}$
8. S composes the RREQ message: $RREQ_S = \{Type\text{-}RREQ, I, P_R, PK_S', PDO_S\}$
9. S sends the RREQ message to all the nodes of its transmission range (neighbor nodes).

When a node X receives the RREQ message carries out the actions below. The generic format of the received message is $RREQ_{X-1} = \{Type\text{-}RREQ, I, P_R, PK_S', PDO_{X-1}\}$ where X-1 depicts the previous node.

10. X verifies if this RREQ message is received for second time, i.e., it verifies if the communication identifier $PK_S'$ is in its routing table.
   – If $PK_S'$ is present in the routing table, X receives the message for the second time. In this case, X discards the RREQ message.
   – If $PK_S'$ is not present in the routing table, X receives the message for the first time and continues the step 11.
11. X verifies if the index $I$ is in its routing table.
   – If $I$ is present in the routing table, X is the recipient and continues the step 12.
   – If $I$ is not present in the routing table, X is an intermediate node and continues the step 13.
12. X tries to open the trapdoor $P_R$ with $K'$ to obtain $ID_R$ and $SK_S'$: $D_{K'}(P_R) = ID_R, SK_S'$. If X can open it and $ID_X = ID_R$ then is confirmed that X is the recipient. In this case, X decrypts the $PDO$ with $SK_S'$ to get the $(K_X', N_X)$ of each intermediate node and the $(N_S, P_S)$ of the sender. Note that the recipient could receive several route requests since the RREQ messages are broadcasted. In any case, the recipient chooses the first request, i.e. the route that is firstly formed.
13. X generates its local route pseudonym $N_X$.
14. X generates a one-time symmetric key $K_X'$.
15. X records in its routing table the following entry: $< PK_S', K_X', N_X >$
16. X modifies the onion to include $N_X$ and $K_X'$, i.e., X appends its layer: $PDO_X = \{E_{PK_S'}(N_X, K_X', PDO_{X-1})\}$.
17. X replaces $PDO_{X-1}$ with $PDO_X$ in the RREQ message. The new message is $RREQ_X = \{Type\text{-}RREQ, I, P_R, PK_S', PDO_X\}$. Table 2 shows the onions and the RREQ messages modified by each path node.
18. X sends the new RREQ message to all the nodes of its transmission range.

The steps 10 to 18 are carried out by each intermediate node until reaching the recipient (step 12), point at which the phase ends.

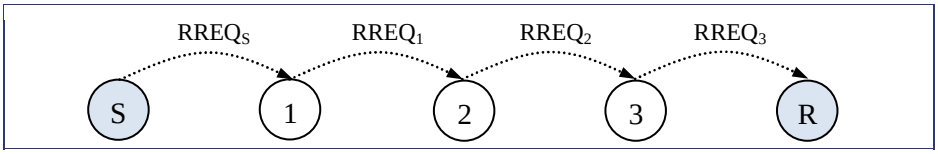


**Figure 2. RREQ phase**

| SENDER NODE | RREQ MESSAGE | PDO |
|---|---|---|
| S | $RREQ_S$ = {Type-RREQ, I, $P_R$, $PK_S$', $PDO_S$ } | $PDO_S = \{E_{PK_S}'(N_S, P_S)\}$ |
| 1 | $RREQ_1$ = {Type-RREQ, I, $P_R$, $PK_S$', $PDO_1$ } | $PDO_1 = \{E_{PK_S}'(N_1, K_1', PDO_S)\}$ |
| 2 | $RREQ_2$ = {Type-RREQ, I, $P_R$, $PK_S$', $PDO_2$ } | $PDO_2 = \{E_{PK_S}'(N_2, K_2', PDO_1)\}$ |
| 3 | $RREQ_3$ = {Type-RREQ, I, $P_R$, $PK_S$', $PDO_3$ } | $PDO_3 = \{E_{PK_S}'(N_3, K_3', PDO_2)\}$ |

**Table 2. RREQ message that sends each node of the path**

## 2.4. Route reply phase (RREP phase)

In this third phase, the recipient sends a route reply message (RREP message) back to the sender. The message contains all local route pseudonyms $N_X$ and all session keys $K_X'$ in an onion structure *PRO*. Each intermediate node *X* removes one layer of the onion, obtains the route pseudonym $N_{X-1}$ that it lacks to complete the registration of its path part and forwards the message to the next node *X-1*. When the message reaches the sender, this obtains all session keys.

For clarity, the process description is based on the example of the figure 3. The RREP phase is initiated by the recipient node as follows.

1. R generates the first layer of the path response onion: $E_K'(P_S, K_3', K_2', K_1')$.
2. R appends the other layers. $PRO_3 = \{E_{K_3}'(N_2, E_{K_2}'(N_1, E_{K_1}'(N_S, E_K'(P_S, K_3', K_2', K_1')))))\}$.
3. R appends to its routing table the adjacent local route pseudonym, the sender's pseudonym and the keys $K_X'$ of the intermediate nodes: $N_3$, $P_S$, $K_3'$, $K_2'$, $K_1'$.
4. R composes the RREP message: $RREP_3 = \{Type\text{-}RREP, N_3, PRO_3\}$.
5. R sends the RREP message to all the nodes of its transmission range.

When a node X receives the RREP message carries out the actions below. The generic format of the received message is $RREP_X = \{Type\text{-}RREP, N_X, PRO_X\}$.

6. X verifies if $N_X$ is its local route pseudonym, i.e., it verifies if $N_X$ is in its routing table.
   a. If $N_X$ is present in the routing table means that X is a route hop. X continues the step 7.
   b. If $N_X$ is not present in the routing table means that X is not a route hop. In this case, X discards the RREP message.
7. X decrypts $PRO_X$ with $K_X'$.
   a. If X is an intermediate node obtains two fields: the next local route pseudonym of the inverse path $N_{X-1}$ and the remaining onion $PRO_{X-1}$. With this


IN3 Working Paper Series is a monograph series promoted by the Internet Interdisciplinary Institute (IN3) of the UOC
IN3 Working Paper Series (2010) | ISSN 2013-8644 | http://in3wps.uoc.edu

action, X has extracted a layer to $PRO_X$, now the resulting onion is $PRO_{X-1}$. X continues the step 8.

b. If X is the sender node gets various fields: the sender's pseudonym $P_S$ and all the keys $K_X'$ of the intermediate nodes. In this case, X must try to open the trapdoor $P_S$ with its secret key $K^{*'}$ to obtain $ID_S$. If it can open it and $ID_X = ID_S$ then X is the sender node. The sender appends the keys $K_X'$ to the routing table and ends the RREP phase.

8. X appends $N_{X-1}$ to its routing table: < $PK_S'$, $K_X'$, $N_X$, $N_{X-1}$ >, e.g. the node 2 of the figure 3 has two adjacent route pseudonyms: $N_1$ and $N_2$. Before the RREP phase, the node has only $N_2$ in its table, after, it has $N_2$ and $N_1$.
9. X replaces $N_X$ with $N_{X-1}$ and $PRO_X$ with $PRO_{X-1}$. The new message is $RREP_{X-1}$ = $\{Type\text{-}RREP, N_{X-1}, PRO_{X-1}\}$. The table 3 shows the onions and the RREP messages modified by each node of the path.
10. X sends the new RREP message to all the nodes of its transmission range.

The steps 6 to 10 are carried out by each intermediate node until reaching the sender (step 7.b), point at which the phase ends.

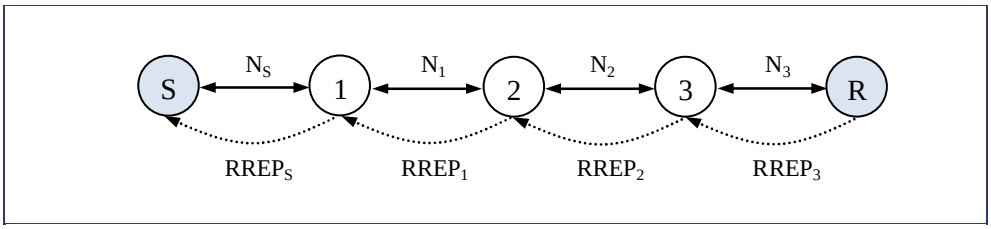


**Figure 3. RREP phase.**

| RECEIVER NODE | RREP MESSAGE | PRO |
|---|---|---|
| 3 | $RREP_3 = \{Type\text{-}RREP, N_3, PRO_3\}$ | $PRO_3 = \{E_{K_3'}(N_2, E_{K_2'}(N_1, E_{K_1'}(N_S, \mathbf{E_{K'}(P_S, K_3', K_2', K_1')}))\}$ |
| 2 | $RREP_2 = \{Type\text{-}RREP, N_2, PRO_2\}$ | $PRO_2 = \{E_{K_2'}(N_1, E_{K_1'}(N_S, \mathbf{E_{K'}(P_S, K_3', K_2', K_1')}))\}$ |
| 1 | $RREP_1 = \{Type\text{-}RREP, N_1, PRO_1\}$ | $PRO_1 = \{E_{K_1'}(N_S, \mathbf{E_{K'}(P_S, K_3', K_2', K_1')})\}$ |
| S | $RREP_S = \{Type\text{-}RREP, N_S, PRO_S\}$ | $PRO_S = \{\mathbf{E_{K'}(P_S, K_3', K_2', K_1')}\}$ |

**Table 3. RREP message that receives each node of the path**

| PHASE | S | X | R |
|---|---|---|---|
| SPE | <$ID_R$, I, K'> | --- | <I,K'> |
| RREQ | <$ID_R$, I, K', $PK_S$', K*', $N_S$> | < $PK_S$',$K_X$',$N_X$> | <$PK_S$',I,K'> |
| RREP | <$ID_R$, I, K', $PK_S$', K*', $N_S$, $K_3$', $K_2$', $K_1$'> | < $PK_S$',$K_X$',$N_X$,$N_{X-1}$> | <$PK_S$',I,K',$N_3$,$P_S$,$K_1$',$K_2$',$K_3$'> |

**Table 4. Routing table of the nodes S, X and R after executing each protocol phase**

## 2.5. Data transmission phase (DATA phase)

Once a route between S and R has been established, the sender initiates the data transmission. The communication can be developed in both directions: S → R and R → S. In either case, the data are hidden in a cryptographic onion *DTO* built with the keys of all nodes of the route (tables 5 and 6). The figure 4 depicts the communication in the two directions.

In a communication S → R, each intermediate node X checks whether the local route pseudonym $N_{X-1}$ of the received DATA message is its. In that case, X decrypts one layer of the *DTO* with its key $K_X$', replaces $N_{X-1}$ with $N_X$ in the DATA message and broadcasts the new DATA message. This process is repeated by each intermediate node until reaching the recipient.

In a communication R → S, each intermediate node X checks whether the local route pseudonym $N_X$ of the received DATA message is its. In that case, X decrypts one layer of the *DTO* with its key $K_X$', replaces $N_X$ with $N_{X-1}$ in the DATA message and broadcasts the new DATA message. This process is repeated by each intermediate node until reaching the sender.

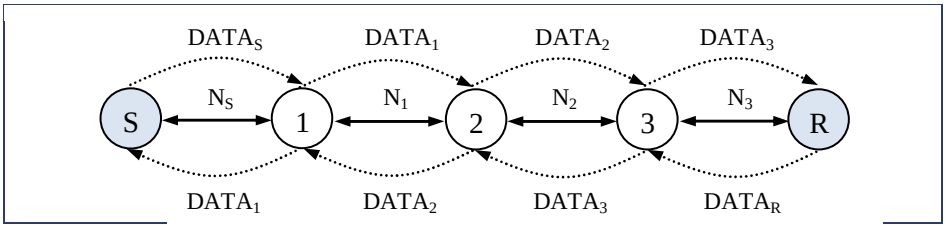


**Figure 4. DATA phase**

| SENDER NODE | DATA MESSAGE | DTO |
|---|---|---|
| S | $DATA_S$ = {Type-DATA, $N_S$, $DTO_S$ } | $DTO_S = \{E_{K_1}'(E_{K_2}'(E_{K_3}'(E_K'(data))))\}$ |
| 1 | $DATA_1$ = {Type-DATA, $N_1$, $DTO_1$ } | $DTO_1 = \{E_{K_2}'(E_{K_3}'(E_K'(data)))\}$ |
| 2 | $DATA_2$ = {Type-DATA, $N_2$, $DTO_2$ } | $DTO_2 = \{E_{K_3}'(E_K'(data))\}$ |
| 3 | $DATA_3$ = {Type-DATA, $N_3$, $DTO_3$ } | $DTO_3 = \{E_K'(data)\}$ |

**Table 5. DATA message that sends each node of the path S→R**

| SENDER NODE | DATA MESSAGE | DTO |
|---|---|---|
| R | $DATA_R$ = {Type-DATA, $N_3$, $DTO_R$} | $DTO_R = \{E_{K_3}'(E_{K_2}'(E_{K_1}'(E_K'(data))))\}$ |
| 3 | $DATA_3$ = {Type-DATA, $N_2$, $DTO_3$ } | $DTO_3 = \{E_{K_2}'(E_{K_1}'(E_K'(data)))\}$ |
| 2 | $DATA_2$ = {Type-DATA, $N_1$, $DTO_2$ } | $DTO_2 = \{E_{K_1}'(E_K'(data))\}$ |
| 1 | $DATA_1$ = {Type-DATA, $N_S$, $DTO_1$ } | $DTO_1 = \{E_K'(data)\}$ |

**Table 6. DATA message that sends each node of the path R→S**

# 3. Evaluation

In this section, we analyze the privacy, security and efficiency of the proposed protocol, and provide a comparison of scalability with others existing protocols.

## 3.1. Privacy analysis

In the analysis we evaluate the privacy properties which are protocol goal: sender anonymity, recipient anonymity, relationship anonymity and sender anonymity in recipient. Besides, we investigate whether the protocol preserves the unlinkability and indistinguishability, two families of properties proposed in (Chrétien & Delaune, 2013). Finally we discuss the anonymity of the phases SPE and DATA and the route anonymity.

To perform the analysis, we verify if the properties may be achieved in presence of a passive attacker, i.e. an eavesdropper who analyzes the incoming and outgoing messages from network nodes. We propose to identify the traceable items of the protocol to find out in which cases is possible to link the incoming messages in a node with the outgoing messages. As the input-output of a node depends of the role played

in the communication, if the protocol presents traceable items the adversary may discover such role.

### 3.1.1. Traceable items of the proposed protocol

As a preliminary step to the privacy analysis, it is necessary to identify the items that may use the adversary to track messages during the execution of the protocol.

- *Traceable items in the RREQ phase*. The messages of this phase contain a session identifier, the field $PK_S$. This item allows tracking the message along the route. Other fields as the key index I or even the recipient's pseudonym $P_R$ can also be used as identifiers, their values are unique in the network and constant in the session.
- *Traceable items in the RREP phase*. The RREP phase does not provide clues to the adversary. It lacks session identifier and, the message fields, e.g. route pseudonym and onion, change their values at each route hop. Besides, from an attacker's perspective, these changes are produced without any kind of correlation.
- *Traceable items in the DATA phase*. The DATA phase has the same behavior as the RREP phase, therefore it lacks traceable items.

The issue of traceable items in the RREQ phase is common among the efficient routing protocols. Even the protocols without session identifier have other fields that may be traceable, as the recipient's pseudonym.

### 3.1.2. Input-output analysis

The session identifier of the RREQ phase may be used to find out the role of a node in the communication. The adversary can link the inputs and outputs that belong to the same session and deduce the node role.

The table 7 shows the input-output given in each role. The notation {RREQ, *sid*} represents the RREQ message and *sid* its session identifier. According to (Chrétien & Delaune, 2013) the roles of a node may be: sender (role Src), receiver (role Dest), intermediate forwarder of a RREQ message (role Req) and intermediate receiver of a RREP message (role Rep). For clarity, Src is the role able to spontaneously start a RREQ phase and Dest is the role able to open the trapdoor of the recipient's pseudonym as response to a RREQ message.

Only the RREQ messages are taken into account because, as discussed in 3.1.1., it is impossible to apply this same tracking in the RREP phase. Nevertheless, it is necessary to clarify the roles Src and Dest in this phase. Src is the role able to open the trapdoor of the sender's pseudonym as response to a RREP message and Dest is the role able to start a RREP phase.

| | CASE | INPUT | OUTPUT | ROLE |
|---|---|---|---|---|
| RREQ message received for first time | 1 | × | {RREQ, sid} | Src |
| | 2 | {RREQ, sid} | {RREQ, sid} | Req |
| | 3 | {RREQ, sid} | × | Dest |
| RREQ message received for second time | 4 | {RREQ, sid} | × | Src |
| | 5 | {RREQ, sid} | × | Req |
| | 6 | {RREQ, sid} | × | Dest |

**Table 7. Inputs-outputs according to the node's role**

### 3.1.3. Sender anonymity

**Proposition 1**. The proposed protocol preserves the sender anonymity, as long as the observed nodes are neither the sender (scenario 1) nor the set of all its neighbors (scenario 2).

**Proof.** (scenario 1) Let *A* be a node with the role Src, i.e. *A* sends a broadcast of the message {RREQ, *sid*} as a result of the session start. Due to the *sid*, the adversary can detect the following case in the input-output analysis: *A* has sent a broadcast of {RREQ, *sid*} without receiving a previous message with the same *sid*. The case 1 of the table 7 is the only one that describes this situation and is associated to the role Src. Therefore, the adversary can know that *A* is a sender node.

(scenario 2) Let *N* be the set of all neighbors of the sender *A*, *N* is under the scrutiny of the adversary. If all the nodes of *N* receive a message {RREQ, *sid*} and none of them have been its transmitter, the adversary can detect the following case in the input-output analysis: all nodes of *N* have received a message {RREQ, *sid*} but none of them have transmitted it before. This case can only be the result of the following action: the node *A* has started a RREQ session. Therefore, the adversary can know that *A* is a sender node.

In any other scenario, the adversary cannot find out the sender's identity through an input-output analysis. Nor can he discover the identity by looking at the message because it is encrypted with a key that only knows the sender.

### 3.1.4. Recipient anonymity

**Proposition 2**. The proposed protocol preserves the recipient anonymity, as long as the observed nodes are neither the recipient (scenario 1) nor the set of all its neighbors (scenario 2).

**Proof.** (scenario 1) Let *A* be a node with the role Dest, i.e. *A* receives a message {RREQ, *sid*} and is able to open the trapdoor of the recipient's pseudonym. Due to the *sid*, the adversary can detect the following case in the input-output analysis: *A* has received a message {RREQ, *sid*} and has not sent a later broadcast of {RREQ, *sid*}. In the table 7 there are several cases that represent this situation, cases 3 to 6. If the adversary observes the previous inputs-outputs can discard that the message has been received for second time, cases 4 to 6. Since the case 3 is associated to the role Dest, the adversary can determine that *A* is the recipient node.

(scenario 2) Let *N* be the set of all neighbors of the recipient *A*, *N* is under the scrutiny of the adversary. If a node $B \in N$ sends a broadcast of the message {RREQ, *sid*}, all neighbors of *B* will broadcast the message except *A*. Due to the *sid*, the adversary can detect the following case in the input-output analysis: no node of *N* has received a message {RREQ, *sid*} from *A* after the broadcast sent by *B*. Therefore *A* has not broadcasted the message {RREQ, *sid*}. There is only one possibility: the node *A* is the recipient of {RREQ, *sid*}. So the adversary can determine that *A* is the recipient node.

In any other scenario, the adversary cannot find out the recipient's identity through an input-output analysis. Nor can he discover the identity by looking at the message because it is encrypted with a key that only knows the sender and recipient.

### 3.1.5. Relationship anonymity

According to (Pfitzmann & Hansen, 2010) the relationship anonymity is weaker than the sender anonymity and recipient anonymity, it can be guaranteed as long as either or both anonymity types are met. As consequence of the propositions 1 and 2, we have the following result.

**Corollary 1**. The proposed protocol preserves the relationship anonymity, as long as one of these scenarios is met:

- The observed nodes are neither the sender nor the set of all its neighbors.
- The observed nodes are neither the recipient nor the set of all its neighbors.

### 3.1.6. Sender anonymity in recipient

**Proposition 3**. The proposed protocol preserves the sender anonymity in recipient.

**Proof.** When the recipient receives the message RREQ and decrypts all onion layers, it obtains the sender's pseudonym. As the trapdoor of the pseudonym can just be opened with a key that only knows the sender, the recipient cannot discover its identity.

### 3.1.7. Indistinguishability

A protocol satisfies this property whether the adversary is unable to distinguish the actions that undertake a node. (Chrétien & Delaune, 2013) points out two types of indistinguishability:

- Indistinguishability w.r.t. Src: the roles Req and Rep are indistinguishable from the role Src.
- Indistinguishability w.r.t. Dest: the roles Req and Rep are indistinguishable from the role Dest.

**Proposition 4**. The proposed protocol does not preserve the indistinguishability w.r.t. Src.

As shown below, the role Rep is indistinguishable from Src. However, the role Req is not indistinguishable from Src.

**Proof.** (Req w.r.t. Src) Let *A* is a node that carries out two actions at the same time:

- *A* acts as Src: *A* starts a RREQ session, thus it sends a broadcast of the message {RREQ, $sid_1$}.
- *A* acts as Req: *A* receives and forwards a message {RREQ, $sid_2$} of other session.

As the RREQ messages contain a session identifier the adversary can detect the following cases in the input-output analysis:

- The node *A* has sent a broadcast of {RREQ, $sid_1$} without receiving a previous message with the same sid. The case 1 of the table 7 is the only one that describes this situation and is associated to the role Src. Therefore, the adversary knows that *A* has acted as Src for the message {RREQ, $sid_1$}.
- The node *A* has received and forwarded the message {RREQ, $sid_2$}. The case 2 of the table 7 is the only one that describes this situation and is associated to the role Req. Therefore, the adversary knows that *A* has acted as Req for the message {RREQ, $sid_2$}.

The above proves that the role Req is not indistinguishable from the role Src. (Rep w.r.t. Src) As the phase RREP has not traceable items, the adversary is unable to relate inputs and outputs. Therefore, the adversary cannot know the role taken by

the node in each message. It is shown that the role Rep is indistinguishable from the role Src.

**Proposition 5**. The proposed protocol does not preserve the indistinguishability w.r.t. Dest.

As shown below, the role Rep is indistinguishable from Dest. However, the role Req is not indistinguishable from Dest.

**Proof.** (Req w.r.t. Dest) Let *A* is a node that carries out two actions at the same time:

- *A* acts as Dest: *A* receives the message {RREQ, $sid_1$} and opens the trapdoor of the recipient's pseudonym.
- *A* acts as Req: *A* receives and forwards a message {RREQ, $sid_2$} of other session.

As the RREQ messages contain a session identifier the adversary can detect the following cases in the input-output analysis:

- The node *A* has received a message {RREQ, $sid_1$} and has not sent a later broadcast of {RREQ, $sid_1$}. In the table 7 there are several cases that represent this situation, cases 3 to 6. If the adversary observes the previous inputs-outputs can discard that the message has been received for second time, cases 4 to 6. As the case 3 is associated to the role Dest, the adversary can know that *A* has acted as Dest for the message {RREQ, $sid_1$}.
- The node *A* has received a message {RREQ, $sid_2$} and has sent a later broadcast of {RREQ, $sid_2$}. The case 2 of the table 7 is the only one that describes this situation and is associated to the role Req. Therefore, the adversary can know that *A* has acted as Req for the message {RREQ, $sid_2$}.

The above proves that the role Req is not indistinguishable from the role Dest.

(Rep w.r.t. Dest) As the phase RREP has not traceable items, the adversary is unable to relate inputs and outputs. Therefore, the adversary cannot know the role taken by the node in each message. It is shown that the role Rep is indistinguishable from the role Dest.

### 3.1.8. Unlinkability

The unlinkability, according to (Chrétien & Delaune, 2013), is the impossibility to determine whether two messages belong to the same session. They point out three types of the unlinkability:

- Unlinkability w.r.t. {Src,Req}/{Src,Req}: it is impossible to determine whether two RREQ messages belong to the same session.
- Unlinkability w.r.t. {Dest,Rep}/{Dest,Rep}: it is impossible to determine whether two RREP messages belong to the same session. Note that if an attacker is able to link two RREP messages will get valuable information about the established route.

- Unlinkability w.r.t. {Src,Req}/{Dest,Rep}: it is impossible to determine whether a RREQ message and a RREP message belong to the same session.

**Proposition 6**. The proposed protocol does not preserve the unlinkability w.r.t. {Src,Req }/{ Src,Req}.

**Proof.** As the RREQ messages contain a session identifier, an attacker can immediately determine whether two messages belong to the same session.

**Proposition 7**. The proposed protocol preserves the unlinkability w.r.t. {Dest,Rep}/{Dest,Rep}.

**Proof.** As the RREP messages lack traceable items, the attacker cannot determine whether two messages belong to the same session.

**Proposition 8**. The proposed protocol preserves the unlinkability w.r.t. {Src,Req}/{Dest,Rep}.

**Proof.** A RREQ message cannot link with a RREP message of the same session because there is no a common field with the same value that acts as linker.

### 3.1.9. Route anonymity

When the recipient chooses an anonymous route sends a RREP message back to sender to communicate it the keys of all path nodes. This message contains all route pseudonyms encrypted in an onion structure. Therefore, if an adversary wants to discover the route needs to resort to the input-output analysis. As, according to 3.1.1, the message has not traceable items, it concludes that the protocol preserves the route anonymity.

### 3.1.10. SPE phase anonymity

In the SPE messages, the recipient's identity is hidden with a salted hash function. As the salt changes in each SPE phase, the scheme guarantees a different hash value in each communication. Therefore, the dictionary attacks are hampered.

If an adversary discovers the recipient's identity in the SPE phase, it will be unable to link the recipient's identity with later messages of the same communication. As mentioned in section 2, this is due to two main points: in the RREQ, RREP and DATA phases the recipient's identity is encrypted with symmetric key and the route is different from the SPE route.

However, this phase has a weak point: if the TTP cheats, it can impersonate the recipient. As the TTP has knowledge of the key, the index and the receiver's identity, it can use that information to open the pseudonym trapdoor.

### 3.1.11. DATA phase anonymity

The DATA phase is similar in terms of privacy to the RREP phase. According to 3.1.1, a DATA message lacks traceable items because the values of its fields change in each hop of route without any correlation. As consequence, an adversary is unable to link an incoming DATA message with an outgoing DATA message. The eavesdropper has not elements to discover the identities of the nodes involved in the communication, so the phase preserves the anonymity.

On the other hand, as the data are encrypted in an onion structure, the adversary needs the keys of all path nodes to obtain the content of the message.

## 3.2. Security analysis

In this analysis, we evaluate if the protocol is secure in presence of an active attacker, i.e. an adversary who replays, alters and injects packets in the network, e.g. to perform replay, modification or denial of service (DoS) attacks.

In order to prevent replay attacks, both sender and recipient use the one-time items of the protocol. Since these fields change in each session, the nodes can detect if an adversary replays previous messages. The recipient can check whether the one-time public key, i.e. the session identifier, is already registered in its routing table to discover RREP messages received for second time. The sender can check whether the one-time symmetric keys generated by intermediate nodes are already registered in its routing table to discover RREP messages received for second time.

If an adversary changes the onion in a RREQ message, in the worst case, the RREP message will reach the attacker instead of the sender. But the adversary will be unable to open the inner onion layer because is encrypted with a key that only knows the sender and the recipient.

The protocol does not provide a mechanism against DoS attacks. An adversary may initiate multiple sessions and flood the network with RREQ messages or, e.g. to target the attack to the TTP node.

## 3.3. Scalability analysis

In this analysis, we compare the scalability of our protocol with others based on symmetric cryptographic, as Anodr and AnonDSR. The main factor that affects the scalability is the number of cryptographic operations that have to perform the

intermediate nodes during routing process. This is because the messages are flooded on the network.

Based on (Song, Korba, & Yee, 2005), we categorize the cryptographic operations into three types:

- Symmetric key operations.
- Efficient public key operations: encryption with public key and verification of a signature.
- Complexity public key operations: decryption with private key and signature.

Table 9 compares the number of cryptographic operations that has to perform each intermediate node in a given time. Our protocol provides better scalability than Anodr and AnonDSR because the intermediate nodes do not have to try to open the trapdoor of the recipient's pseudonym during RREQ phase.

| NODE | PHASE | ANODR | ANONDSR | OUR PROTOCOL |
|---|---|---|---|---|
| S | RREQ | S.E. to build the trapdoor.<br>S.E. to build its onion layer. | S.E. to build the trapdoor.<br>A.E., S.E. and S. to build its onion layer. | S.E. to build the trapdoor.<br>A.E. to build its onion layer. |
| | RREP | S.D. to obtain its onion layer. | S.D. to obtain its onion layer.<br>S.V. to check the message integrity. | S.D. to obtain its onion layer. |
| X | RREQ | S.D. to try to open the trapdoor<br>**Issue**: X does not know what key to use of its list. It will have to try it one to one.<br>S.E. to build its onion layer. | S.D. to try to open the trapdoor<br>A.E. and S.E. to build its onion layer. | **Not require cryptographic operations** to try to open the trapdoor.<br>A.E. to build its onion layer. |
| | RREP | S.D. to obtain its onion layer | S.D. to obtain its onion layer | S.D. to obtain its onion layer |
| R | RREQ | S.D. to open the trapdoor | S.D. to open the trapdoor.<br>A.D. and S.D. h times to obtain all onion layers.<br>S.V. to check the onion layer of the sender. | S.D. to open the trapdoor.<br>A.D. h times to obtain the all onion layers. |
| | RREP | Not require cryptographic operations | S. a time and S.E. h times to build the path reverse onion. | S.E. h times to build the path reverse onion. |

**Table 8. Comparative of cryptographic operations in each node.**
Note: S.E. is Symmetric Encryption**,** A.E. is Asymmetric Encryption, S.D. is Symmetric Decryption, A.D. is Asymmetric Decryption, S. is Signature generation, S.V. is Signature Verification and h is the number of route hops.


IN3 Working Paper Series is a monograph series promoted by the Internet Interdisciplinary Institute (IN3) of the UOC
IN3 Working Paper Series (2010) | ISSN 2013-8644 | http://in3wps.uoc.edu

| | | Nº OPERATIONS **IN EACH INTERMEDIATE NODE** | | |
|---|---|---|---|---|
| PHASE | TYPES OF OPERATIONS | ANODR | ANONDSR | OUR PROTOCOL |
| RREQ | Symmetric key operations | mn + n | 2n | 0 |
| | Efficient P.K. operations | 0 | n | n |
| | Complexity P.K. operations | 0 | 0 | 0 |
| RREP | Symmetric key operations | n | n | n |
| | Efficient P.K. operations | 0 | 0 | 0 |
| | Complexity P.K. operations | 0 | 0 | 0 |

**Table 9. Comparative of scalability (only intermediate nodes)**

Note:
1. n is the number of sessions opened at a given time, i.e. the number of different RREQ or RREP messages in circulation.
2. m is the number of keys registered in the intermediate node.

# 4. Conclusions

In this paper, we have proposed a novel anonymous routing protocol for mobile ad-hoc networks (MANETs). An exhaustive analysis shows that the protocol guarantees the sender and recipient anonymity in the most scenarios, the unlinkability between end nodes and the sender anonymity before the recipient. Compared with other methods based on symmetric cryptography, our protocol is more efficient because the intermediate nodes do not have to perform cryptographic operations during the route establishment to verify whether they are the intended recipient. The scheme is resistant against replay and modification attacks, but vulnerable to DoS attacks.

Unlike previous protocols, our work presents a method to share the key of the recipient's pseudonym before routing process in an anonymous and efficient way.

As future work, we suggest to research techniques that eliminate the traceable items in the route discovery phase of the protocol without sacrificing the efficiency. It is also interesting to investigate ways to prevent DoS attacks.

# Bibliographic references

Boukerche, A., El-Khatib, K., Xu, L., & Korba, L. (2005). An efficient secure distributed anonymous routing protocol for mobile and wireless ad hoc networks. *Computer Communications , 28* (10), 1193-1203.

Chen, J., Boreli, R., & Sivaraman, V. (2012). Improving the efficiency of anonymous routing for MANETs. *Computer Communications , 35* (5), 619-627.

Chrétien, R., & Delaune, S. (2013). Formal analysis of privacy for routing protocols in mobile ad hoc networks. *2nd international conference on Principles of Security and Trust (POST'13)*, (págs. 1-20).

Edman, M., & Yener, B. (2009). On anonymity in an electronic society: a survey of anonymous communication systems. *ACM Computing Surveys , 42* (1), 1-35.

Kong, J., & Hong, X. (2003). Anodr: anonymous on demand routing with untraceable routes for mobile ad-hoc networks. *4th ACM international symposium on mobile ad hoc networking & computing (MobiHoc '03)*, (págs. 291-302).

Li, X., Li, H., Ma, J., & Zhang, W. (2009). An Efficient Anonymous Routing Protocol for Mobile Ad Hoc Networks. *5th International Conference on Information Assurance and Security (IAS '09)*, *2*, págs. 287-290.

Pan, J., & Li, J. (2009). MASR: An Efficient Strong Anonymous Routing Protocol for Mobile Ad Hoc Networks. *International Conference on Management and Service Science (MASS '09)*, (págs. 1-6).

Pfitzmann, A., & Hansen, M. (2010). *A terminology for talking about privacy by data minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management v0.34.* Retrieved 10 01, 2013, from http://dud.inf.tu-dresden.de/Anon_Terminology.shtml

Rifà-Pous, H., & Herrera-Joancomartí, J. (2011). Computational and energy costs of cryptographic algorithms on handheld devices. *Future Internet , 3* (1), 31-48.

Rifà-Pous, H., & Herrera-Joancomartí, J. (2009). Cryptographic Energy Costs are Assumable in Ad Hoc Networks. *IEICE Transactions on Information and Systems , 92* (5), 1194-1196.

Rifà-Pous, H., Panaousis, E., & Politis, C. (2012). Recipients' Anonymity in Multihop Ad-hoc Networks. *EICE Transactions on Information and Systems , E95.D* (1), 181-184.

Song, R., Korba, L., & Yee, G. (2005). AnonDSR: Efficient anonymous dynamic source routing for mobile ad-hoc networks. *3rd ACM workshop on Security of ad hoc and sensor networks (SASN '05)*, (págs. 33-42).

***Resumen***

*La concurrencia de anonimato y escalabilidad en el enrutamiento de redes móviles ad-hoc (MANETs) es una cuestión difícil de gran interés. En este artículo, proponemos un protocolo de enrutamiento eficiente que garantiza el anonimato del emisor y del receptor en la mayoría de escenarios, así como la imposibilidad de vincularlos durante la comunicación. Otro aspecto a señalar es que la identidad del emisor permanece anónima frente al receptor. El esquema está basado en criptografía simétrica pero, a diferencia de propuestas anteriores, el intercambio de la clave compartida es realizado sin revelar las identidades de los nodos finales. El protocolo proporciona un buen nivel de escalabilidad porque ningún nodo intermedio tiene que realizar operaciones criptográficas durante el descubrimiento de ruta para verificar si es el destino.*

***Palabras clave***

*Anonimato, desvinculación, seguridad, redes ad-hoc, MANET.*

***Resum***

*La concurrència d'anonimat i escalabilitat en l'enrutament de xarxes mòbils ad-hoc (MANETs) és una qüestió difícil de gran interès. En aquest article, proposem un protocol d'enrutament eficient que garanteix l'anonimat de l'emissor i del receptor en la majoria d'escenaris, així com la impossibilitat de que es vinculin durant la comunicació. Un altre aspecte a destacar és que la identitat de l'emissor roman anònima per el receptor. L'esquema està fonamentat en criptografia simètrica però, a diferència de propostes anteriors, l'intercanvi de la clau compartida és realitza sense revelar les identitats dels nodes finals. El protocol proporciona un bon nivell d'escalabilitat perquè cap node intermedi ha de realitzar operacions criptogràfiques durant el descubriment de ruta per verificar si és el destí.*

***Paraules clau***

*Anonimat, desvinculació, seguretat, xarxes ad-hoc, MANET.*

***Renato Teixeira Bressan***
*renato.bressan@yahoo.com.br*
*Programa de postgrau de Comunicació i Societat*
*Universitat Federal de Juiz de Fora (Brasil)*

*Renato Teixeira Bressan és estudiant de postgrau i té una beca d'investigació al Programa de postgrau de Comunicació i Societat de la Universitat Federal de Juiz de Fora (Brasil). És membre de l'equip de Comunicació i Tecnologies Contemporànies, un grup de recerca del Consell Nacional de Recerca. Fa recerca sobre tecnologia i els nous mitjans des de 2006, quan encara era estudiant de grau de Comunicació. Entre 2006 i 2008, va ser membre del PET/MEC SESu (Programa d'Ensenyament Tutorial) i va conduir recerca sobre semiòtica, el web 2.0, la cibercultura i els jocs. Els seus temes principals de recerca són actualment la tecnocultura, els jocs, la representació diagramàtica, l'epistemologia, el cinema, les plataformes digitals (MMORPG, MMO, MUD, MUVE, etc.) i el web. Actualment, Renato també és membre de TV Motoradio (<http://tvmotoradio.com>), on treballa com a agent web, productor cultural, presentador i director de fotografia.*
*Més informació sobre l'autor a: <http://migre.me/m3PV>.*

***Renato Teixeira Bressan***
*renato.bressan@yahoo.com.br*
*Programa de postgrau de Comunicació i Societat*
*Universitat Federal de Juiz de Fora (Brasil)*

*Renato Teixeira Bressan és estudiant de postgrau i té una beca d'investigació al Programa de postgrau de Comunicació i Societat de la Universitat Federal de Juiz de Fora (Brasil). És membre de l'equip de Comunicació i Tecnologies Contemporànies, un grup de recerca del Consell Nacional de Recerca. Fa recerca sobre tecnologia i els nous mitjans des de 2006, quan encara era estudiant de grau de Comunicació. Entre 2006 i 2008, va ser membre del PET/MEC SESu (Programa d'Ensenyament Tutorial) i va conduir recerca sobre semiòtica, el web 2.0, la cibercultura i els jocs. Els seus temes principals de recerca són actualment la tecnocultura, els jocs, la representació diagramàtica, l'epistemologia, el cinema, les plataformes digitals (MMORPG, MMO, MUD, MUVE, etc.) i el web. Actualment, Renato també és membre de TV Motoradio (<http://tvmotoradio.com>), on treballa com a agent web, productor cultural, presentador i director de fotografia.*
*Més informació sobre l'autor a: <http://migre.me/m3PV>.*



IN3 Working Paper Series is a monograph series promoted by the Internet Interdisciplinary Institute (IN3) of the UOC
IN3 Working Paper Series (2010) | ISSN 2013-8644 | http://in3wps.uoc.edu