AnonDSR: Efficient Anonymous Dynamic Source Routing for Mobile Ad-Hoc Networks

Song, Ronggong; Korba, Larry; Yee, George

National Research Council Canada
Conseil national de recherches Canada

Institute for Information Technology
Institut de technologie de l'information

NRC · CNRC

AnonDSR: Efficient Anonymous Dynamic Source Routing for Mobile Ad-Hoc Networks *

Song, R., Korba, L., and Yee, G.
November 2005

* published in the Proceedings of the 2005 ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN 2005) in conjunction with the 12th ACM Conference on Computer & Communications Security (CCS 2005). Alexandria, Virginia, USA. pp. 32-42. November 7-11, 2005. NRC 48304.

Canada

AnonDSR: Efficient Anonymous Dynamic Source Routing for Mobile Ad-Hoc Networks

Ronggong Song
Institute for Information Technology
National Research Council Canada
Ottawa, Ontario K1A 0R6, Canada
+1-613-990-6869
Ronggong.Song@nrc.ca

Larry Korba
Institute for Information Technology
National Research Council Canada
Ottawa, Ontario K1A 0R6, Canada
+1-613-998-6967
Larry.Korba@nrc.ca

George Yee
Institute for Information Technology
National Research Council Canada
Ottawa, Ontario K1A 0R6, Canada
+1-613-990-4284
George.Yee@nrc.ca

ABSTRACT
Security, anonymity, and scalability are still important issues for mobile ad hoc network routing protocols. We first expose the limitations of several existing mobile ad hoc network routing protocols with security and anonymity constraints and analyze their scalabilities. Based on the analysis, we propose a new anonymous dynamic source routing protocol (AnonDSR) to provide three levels of security protection. We compare their scalabilities with security constraints, and analyze the new protocol to show it has strong security and anonymity protection, and very good scalability.

Categories and Subject Descriptors
C.2.2 [**Computer-Communication Networks**]: Network Protocols – *routing protocols*

General Terms
Security

Keywords
Mobile Ad Hoc Network, Dynamic Source Routing, Anonymity, Security.

1. INTRODUCTION
The dynamic source routing protocol (DSR) [1, 2, 3] has many good features for mobile ad hoc networks. These features include efficiency, simplicity, self-organization and self-configuration without the requirement of network infrastructure or a particular network topology. However, the original DSR does not include any security and anonymity protection making it vulnerable to a variety of security attacks.

In order with the objective of secure communications for ad hoc networks, Vetriselvi and Parthasarathi [4] proposed a secure dynamic source routing protocol (SDSR) in 2003. Kargl et al. [5] proposed another secure dynamic source routing protocol (SDSR) in 2005. They use the same cryptographic mechanism – Diffie Hellmann key agreement protocol [6] to create a shared session key for a security communication between the source node and destination node. But these security routing protocols including Ariadne [16] and SRP [17] cannot protect the routing information from traffic analysis attacks such as message coding attack, communication pattern attack, and others [7]. Thus adversaries can trace network routes and find the source and destination nodes of any communication, making serious threats to instigate covert missions against user privacy. To prevent adversaries from tracing a packet flow to its source or destination and other traffic analysis attacks, Kong and Hong [8] presented an anonymous routing protocol for mobile ad-hoc networks (ANODR) in 2003. We describe in detail two limitations in Kong-Hong's ANODR protocol through a cryptographic analysis in the next section. One limitation is that the source and destination cannot make a cryptographic onion for their communication data after the anonymous route discovery protocol since each node encrypts the routing information with their own secret key during the route discovery so that the source and destination don't know the whole route. Obviously, they cannot create the shared session keys with each intermediate node on the routes to make an anonymous cryptographic onion for communication data. Another limitation is that the trapdoor used in the protocol is not practical since a destination node really does not know which shared session key should be used for the trapdoor if the destination node has many shared session keys with different nodes in an ad-hoc network. El-Khatib et al. [9, 13, 14] presented another secure distributed anonymous routing protocol (SDAR) for mobile ad-hoc networks in 2003. The protocol uses a public key algorithm as the trapdoor, making the system very unscalable since each intermediate forwarding node needs to attempt to decrypt the trapdoor during a route discovery. Asymmetric decryption usually has a very high computational complexity for a public key algorithm according to the demonstration in [11].

In order to provide a strong security and anonymity protection and better scalability for mobile ad hoc networks, we propose a new efficient anonymous dynamic source routing (AnonDSR). The new routing consists of three protocols. The first protocol is used to create a shared secret key and random nonce between the source and destination for secure and anonymous communications. The second protocol uses the shared secret key and nonce to create a trapdoor and employ an anonymous onion routing between the source and destination. To make the system more scalable, we only use encryption with public key in the intermediate forwarding nodes during the route discovery since

SASN'05, November 7, 2005, Alexandria, Virginia, USA.

ACM 1-59593-227-5/05/0011.

many public key cryptosystems like RSA [10] choose a short key as public key. As we demonstrated in [11], the RSA encryption with standard public key has very good scalability for short messages (e.g. size lesser than 2048 bits for RSA-2048). Furthermore, each intermediate node in an anonymous route owns a shared session key with the source and destination when the protocol is completed. In the last protocol, the source and destination use their session keys shared with the intermediate nodes to encrypt all communications with the cryptographic onion method [12]. In addition, we demonstrate, using cryptanalysis, that AnonDSR can provide strong security and anonymity protection for mobile ad hoc networks, and has very good scalability by comparing it with existing anonymous ad hoc network routings. We also show the first protocol can provide secure or non-secure communications directly for secure or non-secure applications, respectively.

The rest of the paper is organized as follows. Two anonymous ad hoc network routing protocols are briefly reviewed and their limitations are analyzed in the next section. In Section 3, a new efficient anonymous dynamic source routing protocol, AnonDSR, is presented. The security, anonymity, and scalability of AnonDSR are discussed, analyzed, and compared with other existing secure and anonymous routing protocols. The performance of AnonDSR is simulated in Section 5, and compared with other protocols. Finally, concluding remarks are given in Section 6.

2. EXISTING ANONYMOUS AD HOC NETWORK ROUTING PROTOCOLS

We review two anonymous ad hoc network routing protocols in this section: Kong-Hong's ANODR [8] and El-Khatib et al.'s SDAR [9, 13, 14], analyze their limitations on security, anonymity, scalability, and practice, and discuss several challenges for the design of a practical anonymous ad hoc routing protocol.

2.1 Terminology and Notations

Terminology and notations used in the paper are defined as follows:

- ID_A: identity for node A
- K_X: a random symmetric key
- $H()$: a one-way hash function
- PK_A: public key for node A
- SK_A: private key for node A
- N: a random nonce
- P: padding
- PL: padding length
- $Sign_A$: the node A's signature
- $E_K(M)$: a message M encrypted with a session key K
- $E_{PK}(M)$: a message M encrypted with a public key PK

2.2 Review and Analysis of Kong-Hong's ANODR

Kong-Hong's anonymous on demand routing consists of three phases: route request phase (*RREQ*), route reply phase (*RREP*), and data transfer phase as follows.

- ***RREQ* phase:** This phase is initiated by a communication source. The source node first creates the following RREQ packet and broadcasts the packet to its neighbor nodes that are covered within the radio communication range of the source node.

  $$<RREQ, seqnum, tr_{dest}, onion>$$

  In the RREQ packet, *seqnum* is a global unique sequence number, tr_{dest} is a cryptographic trapdoor, i.e. $tr_{dest}= E_{K_T}(ID_{dest}, N_A)$ where K_T is a shared secret key between the source and destination and ID_{dest} is the destination tag, and *onion* is a cryptographic routing message that is encrypted by each intermediate node along with the path discovery, for instance, $E_{K_D}(N_D, E_{K_C}(N_C, E_{K_B}(N_B, E_{K_A}(N_A, src))))$ made by nodes *A, B, C,* and *D*.

  The local neighbor nodes try to open the trapdoor tr_{dest} when they receive an RREQ packet the first time. A node is the destination if the node can successfully open the trapdoor. Otherwise, the node works as an intermediate RREQ packet forwarding node to embed a random nonce N_X to the boomerang onion, encrypt the result with a random symmetric key K_X, and broadcast the RREQ packet to its neighbors, where the trapdoor information N_X and K_X are only known to the node *X*. In addition, each intermediate forwarding node discards any RREQ packet when it receives them the second time.

- ***RREP* phase:** The destination node creates the following RREP packet and broadcasts the packet to its neighbor nodes.

  $$<RREP, N_{dest}, pr_{dest}, onion>$$

  In the RREP packet, N_{dest} is a local unique random route pseudonym created by the destination node, pr_{dest} is the proof of global trapdoor opening, and *onion* is the same cryptographic message that is bounced back by the destination node.

  The local neighbor nodes try to open the *onion* when they receive an RREP packet. The node will discard the RREP packet if it cannot successfully open the *onion* with its trapdoor information K_X and N_X. Otherwise, it means that the node is on the anonymous route. The node then uses its nonce N_X instead of the nonce N_{dest} and the decrypted inside layer *onion* instead of the received *onion* in the RREP packet, and broadcasts the new RREP packet to its neighbors. Each intermediate forwarding node has the same performance as above until the RREP packet is bounced back to the source node.

- **Data transfer phase:** The source node wraps its data packet with the outgoing route pseudonym and broadcasts the packet locally. The local nodes check the route pseudonym in their forwarding tables after receiving the packet. Only the matched node changes the route pseudonym to the matched outgoing pseudonym and broadcasts the changed packet locally. All other local nodes discard the packet.

There exist two important limitations in ANODR as we analyzed as follows.

- **Trapdoor issue:** ANODR recommends using a symmetric key cryptography as the trapdoor solution in the *RREQ* phase for scalability. However, it does not describe how the source and

destination establish a shared secret key. Furthermore, even if we assume that they already have a shared secret key for the trapdoor, the issue is how the destination and intermediate forwarding nodes know which key should be used to open the trapdoor in an RREQ packet during an anonymous route discovery since the RREQ packet hides the identity information of the source and destination for anonymity, and every node may have many shared secret keys with other different nodes. Therefore, each node needs to try all its shared secret keys for the trapdoor. This requirement makes the system impractical and unscalable despite the use of symmetric key cryptography. In addition, the same issue exists in the *RREP* phase since each RREP forwarding node does not know which key should be used for the trapdoor boomerang onion. In a security application, key management usually plays a very important role. A practical trapdoor requires careful design.

- **Anonymity issue:** ANODR only uses the route pseudonym to wrap the communication data so that a global attacker can easily find the path only by comparing the communication data part since each intermediate forwarding node on the anonymous route only changes the route pseudonym and does not change the communication data part. The current popular solution is to create a cryptographic onion for the communication data. To do this, the source or destination node must have a shared session key with each intermediate forwarding node for an anonymous route. However, ANODR does not provide a mechanism to establish these shared session keys during the *RREQ* phase and *RREP* phase.

2.3 Review and Analysis of El-Khatib et al.'s SDAR

El-Khatib et al.'s secure distributed anonymous routing algorithm can be summarized as three phases similar with ANODR above but SDAR uses a public key cryptography for routing protection.

- ***RREQ* phase:** In SDAR, the source node creates the following RREQ packet and broadcasts it locally

  $<RREQ, PK_{temp}, tr_{dest}, path>$,

  where PK_{temp} is a one-time temporary public key created by the source node and another function of the temporary public key is to work as a unique sequence number, tr_{dest} is a public key cryptographic trapdoor, i.e. $tr_{dest} = E_{PK_{dest}} (ID_{dest}, K_A)$ where PK_{dest} is the destination's public key and K_A is a shared session key between the source and destination for the secure and anonymous route protection, *path* is a cryptographic routing message that is encrypted and attached by each intermediate node along with the path discovery, for instance, $E_{K_A}(ID_A, PK_A, PK_{temp}, SK_{temp}, N_A, PL, P, Sign_A) \| E_{PK_{temp}} (ID_B, K_B, N_B, Sign_B) \| E_{PK_{temp}} (ID_C, K_C, N_C, Sign_C) \| E_{PK_{temp}} (ID_D, K_D, N_D, Sign_D)$ made by nodes *A, B, C,* and *D*. Other processing on each intermediate node is similar with ANODR.

- ***RREP* phase:** The destination node creates the following RREP packet and broadcasts it locally after receiving the RREQ packet

  $<RREP, N_{next}, onion>$,

  where *onion* is a cryptographic path reverse message created by the destination node, for instance, $E_{K_D}(N_C, E_{K_C}(N_B, E_{K_B}(N_A, E_{K_A}(N_B, K_B, N_C, K_C, N_D, K_D, PL, P))))$ made by the destination node for nodes *D, C, B,* and *A*. Each intermediate forwarding node on the reverse path can get its next node's nonce by decrypting the onion and change it in the RREP packet as N_{next}.

- **Data transfer phase:** The source and destination nodes create a cryptographic onion for their communication data using the session keys they share with the intermediate forwarding nodes.

There exist three important limitations in SDAR as we analyzed as follows:

- **Trapdoor issue:** SDAR uses a public key cryptography as the trapdoor solution so that each intermediate forwarding node must try to open the trapdoor with its private key in order to check whether or not it is the destination. This makes the system very unscalable when there are many RREQ packets since a public key cryptosystem usually uses a long private key as the decryption key resulting in a decryption with very high computational complexity.

- **Scalability issue:** In addition to the scalability issue associated with the trapdoor, SDAR has another scalability issue evident during the path discovery phase. In the *RREQ* phase, each intermediate forwarding node requires the creation of an encrypted routing message containing a signature made by the node. Since making a signature uses the private key of the node and has a high computational complexity, this will delay the anonymous routing message and again make the system very unscalable.

- **Security issue:** SDAR has at least one security issue. For instance, in the *RREQ* phase, an intermediate forwarding node or hacker can delete the last part of the routing message that was attached by its previous nodes. It then makes a new routing message and forwards it to its next node. The destination node cannot find this attack although each node embeds a signature in the protected path message.

Based on the cryptanalysis of the above two anonymous ad hoc routing protocols, we find that there are several important challenges for designing a practical anonymous ad hoc routing protocol. The first challenge is to design a practical trapdoor for anonymous routing. Obviously, a public key trapdoor is not scalable. A symmetric key trapdoor may be a good choice but it must be designed carefully with the routing protocol to ensure practicality. The second challenge is to provide anonymity for all routing and data messages since any unchanged part during communication can easily expose the anonymous path. The last challenge is scalability. Symmetric key cryptographic operations usually have very good scalability but a public key cryptosystem can provide an efficient way to establish secret session keys for the use of symmetric key cryptosystems. In addition, not all public key cryptographic operations have poor scalability as we demonstrated in [11], for example, encryption and verification with public key usually scale very well since a public key is a short key in many public key cryptosystems like RSA. The public key cryptographic operations that scale poorly are the operations using the private key such as decryption and signature. In

summary, we must design the protocol carefully and avoid using the unscalable cryptographic operations in the intermediate forwarding nodes if the public key system is required.

3. EFFICIENT ANONYMOUS DYNAMIC SOURCE ROUTING

Based on the analysis and challenges above, we describe the design of a new anonymous dynamic source routing (AnonDSR) in this section. The new routing consists of three protocols: security parameter establishment, anonymous route discovery, and anonymous data transfer.

3.1 Security Parameter Establishment Protocol

The security parameter establishment protocol is used to establish the security parameters for secure and anonymous communications according to the secure type in the packet. It also can build a route for non-secure communications directly. The protocol has two phases: *RREQ* phase and *RREP* phase.

- ***RREQ* phase:** The source node first creates the following RREQ packet and broadcasts the packet locally

  $< RREQ, SecType, seqnum, ID_{src}, ID_{dest}, RRec, SecPara >$,

  where *SecType* represents a security type of the RREQ packet; *seqnum* is a global unique sequence number; ID_{src} and ID_{dest} are the identities of the source and destination nodes; *RRec* is the source route record; and *SecPara* contains the security parameters that the source node provides. *seqnum*, ID_{src}, ID_{dest}, and *RRec* are the same as the elements in the original DSR. *SecType* can be non-secure, secure, or anonymous depending to the security requirements of a practical application. If *SecType* is non-secure, the value of *SecPara* is empty. If *SecType* is secure or anonymous, $SecPara = \{ E_{PK_{dest}} (N_K, K, Para), Sign_{src}\}$ where N_K is a secret index of the shared secret key *K*; *Para* is the cryptographic parameters such as encryption algorithm and version used in the secure data transfer protocol or the anonymous route discovery protocol; and $Sign_{src}$ is a signature signed by the source node for verification, i.e. $Sign_{src} = E_{SK_{src}} (H(seqnum, ID_{src}, ID_{dest}, N_K, K, Para))$.

  Each intermediate forwarding node puts its address into *RRec* and broadcasts the new packet locally. Every node discards the same packet when it receives the second time by comparing the sequence number and packet type. If a node finds it is the destination node and the RREQ packet is a non-secure packet, the protocol jumps into the following *RREP* phase. If it finds the RREQ packet is a secure or anonymous packet, it needs to decrypt the *SecPara* part first, verifies whether the packet is correct, record the information $< N_K, ID_{src}, K, Para >$ into its shared secret key ring, and jumps into the *RREP* phase.

- ***RREP* phase:** The destination node broadcasts the following RREP packet locally to respond the RREQ packet

  $< RREP, SecType, seqnum, ID_{src}, ID_{dest}, RRec, SecPara >$

  where *SecType*, *seqnum*, ID_{src}, and ID_{dest} are the same as the elements in the RREQ packet above; *RRec* is the record of whole path; *SecPara* is the security parameters that the destination node requests to change.

  Each intermediate forwarding node first checks whether it is on the route when it receives a RREP packet the first time. If it is on the route, it broadcasts the packet locally without any change. It discards the packet if it is not on the route or it receives the packet the second time. If the security type of the packet is non-secure or secure, the node adds a route into its routing table for data transfer use. Figure 1 depicts a non-secure route. For secure communications, the only difference is that the source and destination has a shared secret session key.

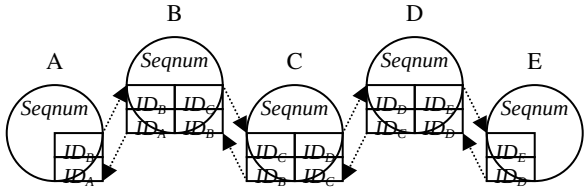

Figure 1. A secure or non-secure route

After the above protocol has completed execution, the source node chooses a corresponding data transfer protocol for its communication with the destination. If the security type in the above protocol is secure or non-secure, the source and destination nodes use the above route for their communications. If the security type is anonymous, the source node uses the following anonymous route discovery protocol to build an anonymous route for their communications.

3.2 Anonymous Route Discovery Protocol

The anonymous route discovery protocol establishes an anonymous route between a pair of source and destination nodes that is resistant against traffic analysis attacks launched by any adversaries including the intermediate forwarding nodes. The protocol is used when the source and destination want to create an anonymous path for their communications and they already have a shared secret key and secret key index in their key ring (established by the security parameter establishment protocol described above). The protocol consists of two phases: *RREQ* phase and *RREP* phase.

- ***RREQ* phase:** The source node first creates the following ANON-RREQ packet and broadcasts the packet locally

  $< ANON\text{-}RREQ, PK_{temp}, tr_{dest}, onion >$

  where *ANON-RREQ* represents a ANON-RREQ packet type, PK_{temp} is a one-time temporary public key created by the source node and also works as a unique sequence number in the *RREQ* phase, and tr_{dest} is a symmetric key cryptographic trapdoor that only the destination can open with a shared secret key. We use the following cryptographic mechanism for the trapdoor $tr_{dest} = \{N_K, E_K(ID_{dest}, SK_{temp})\}$ where N_K and *K* are the key index and shared secret key established in the security parameter establishment protocol and stored in the key ring of the source node and destination node, and SK_{temp} is the corresponding private key of the one-time public key PK_{temp}. *onion* is a cryptographic onion message that records the anonymous path

with security protection. Figure 2 depicts a protected path discovery onion (PDO) in each intermediate forwarding node during a path discovery and a protected path reverse onion (PRO) in the next *RREP* phase. In Figure 2, K_X is a session key that each intermediate forwarding node shares with the source and destination for building an anonymous communication channel, N_X is a local unique route pseudonym, $N_{K'}$ and K' are the new key index and shared secret key that are used to update the old key index N_K and secret key K for the next communication use if this protocol is successful, and $Sign_A = E_{SK_A}(H(PK_{temp}, SK_{temp}, K_A, ID_A, ID_E, PK_A, N_K, K, N_{K'}, K', PL, P))$.

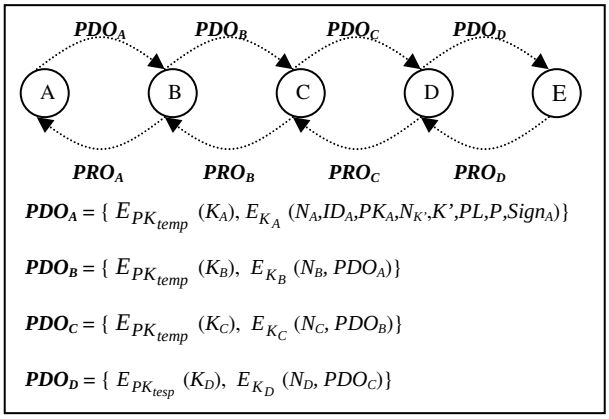

Figure 2. AnonDSR protected path discovery onion

Each intermediate forwarding node creates a session key K_X shared with the source and destination, and encrypts the session key with the one-time public key PK_{temp} when it receives an ANON-RREQ packet the first time. It then creates a local unique route pseudonym N_X, encrypts the pseudonym and the received path discovery onion together with the session key, and broadcasts the new packet locally. After that, the node then tries to open the trapdoor tr_{dest}. If it finds it is the destination, the protocol will jump into the *RREP* phase. Every node discards a same ANON-RREQ packet when it receives it the second time by comparing the packet type and the one-time public key.

- ***RREP* phase:** The destination node decrypts the protected cryptographic onion in the ANON-RREQ packet above using the private key it gets from the trapdoor and verifies if all data are correct. As in Figure 2, with the anonymous route pseudonyms (N_A, N_B, N_C, N_D) and corresponding session keys (K_A, K_B, K_C, K_D) it gets from the path discovery onion, the destination node creates a path reverse onion as follows (see Figure 2)

$$\boldsymbol{PRO_D} = E_{K_D}(N_C, E_{K_C}(N_B, E_{K_B}(N_A, E_{K_A}(N_B, K_B, N_C, K_C, N_D, K_D, PL, P, Sign_E)))),$$

where $Sign_E = E_{SK_E}(H(N_A, K_A, N_B, K_B, N_C, K_C, N_D, K_D, PL, P)$. It then adds a route $< N_D, K_D, N_A, K_A, N_B, K_B, N_C, K_C,>$ into its routing table in which it uses the first nonce N_D as its anonymous route pseudonym for this communication. Finally, the destination node creates the following ANON-RREP packet and broadcasts it locally

$$<ANON\text{-}RREP, N_D, PRO_D>.$$

All local nodes check if N_D is their pseudonym after they receive the ANON-RREP packet above. They discard the packet if N_D is not their pseudonym. Obviously, node *D* can find it is on the route. Node *D* decrypts one layer of the onion PRO_D using its session key K_D corresponding to the pseudonym N_D and gets N_C and PRO_C where

$$\boldsymbol{PRO_C} = E_{K_C}(N_B, E_{K_B}(N_A, E_{K_A}(N_B, K_B, N_C, K_C, N_D, K_D, PL, P, Sign_E))).$$

It then adds a route $< N_D, N_C, K_D>$ into its routing table in which it uses the nonces N_D and N_C as its anonymous route pseudonyms with the node *E* and *C* respectively, and creates a new ANON-RREP packet $<ANON\text{-}RREP, N_C, PRO_C>$ using N_C and PRO_C instead of N_D and PRO_D, and broadcasts the new packet locally. Since N_X is a random number, the conflict opportunity of N_X locally is very low. Even if another node has the same pseudonym, the node just discards the packet when it finds it cannot decrypt the onion. One by one, node *C* creates a new ANON-RREP packet $<ANON\text{-}RREP, N_B, PRO_B>$ and broadcasts it locally, where

$$\boldsymbol{PRO_B} = E_{K_B}(N_A, E_{K_A}(N_B, K_B, N_C, K_C, N_D, K_D, PL, P, Sign_E)),$$

and adds a route $< N_C, N_B, K_C>$ into its routing table. Finally, the ANON-RREP packet $<ANON\text{-}RREP, N_A, PRO_A>$ arrives at the source node *A* where

$$\boldsymbol{PRO_A} = E_{K_A}(N_B, K_B, N_C, K_C, N_D, K_D, PL, P, Sign_E).$$

The source node decrypts the last layer of the onion PRO_A and gets all anonymous route pseudonyms and their corresponding session keys. It then adds a route $< N_A, K_A, N_D, K_D, N_C, K_C, N_B, K_B>$ into its routing table in which it uses N_A as its anonymous route pseudonym. Figure 3 depicts the anonymous route with corresponding pseudonyms and session keys.

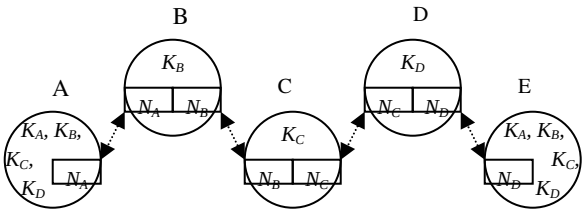

Figure 3. AnonDSR anonymous route

After the protocol has completed execution, the source and destination have updated their shared secret key and key index for the next anonymous communication use. They also have all anonymous route pseudonyms and session keys for the current anonymous communication use. In the protocol, we use a

cryptographic onion to protect the anonymous route record in the ANON-RREQ packet and ANON-RREP packet. In addition, each intermediate forwarding node has one public key encryption and one symmetric key encryption for building PDO, one symmetric key decryption for opening the trapdoor in the *RREQ* phase, and one symmetric key decryption for decrypting one layer of the onion in the *RREP* phase. As we mentioned [11], cryptographic operations such as public key encryption, symmetric key encryption and decryption, have very good scalability.

3.3 Anonymous Data Transfer Protocol

The anonymous data transfer protocol builds a cryptographic onion for anonymous communication data protection. The protocol is only used when an anonymous route discovery protocol is completed. The protocol is described as follows.

The source node creates a cryptographic onion for the communication data that the source wants to send to the destination, creates the following ANON-DATA packet, and broadcasts the packet locally

$$< ANON\text{-}DATA, N_{src}, onion >.$$

Figure 4 depicts the anonymous communication data onion (ADO) from the source to the destination in each intermediate forwarding node and the reverse anonymous communication data onion (RDO) from the destination to the source.

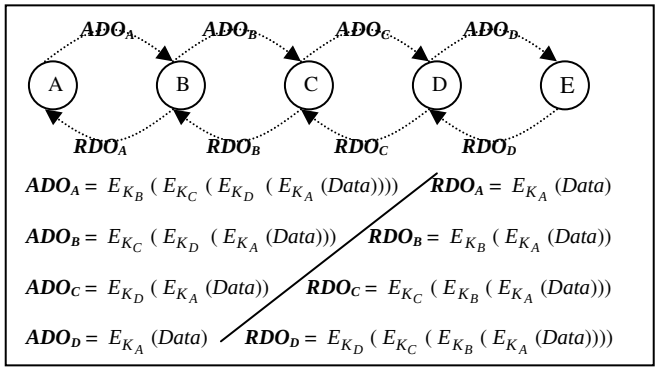

Figure 4. AnonDSR anonymous communication data onion

Each intermediate forwarding node checks whether the pseudonym of the data packet belongs to it and decrypts one layer of the data onion using its session key if it is on the anonymous route. It then changes the route pseudonym by its forwarding routing table, uses the decrypted onion instead of the received onion, and broadcasts the new packet locally. It discards the packet if it is not on the anonymous route. The procedure is repeated until the data packet arrives at the destination. A reverse anonymous communication data transfer from the destination to the source uses the reverse data onion (RDO) that is described in Figure 4.

In this protocol, the communication data is changed through each intermediate forwarding node and each intermediate forwarding node only has one cryptographic operation, i.e. a symmetric key decryption.

4. SECURITY, ANONYMITY, AND SCALABILITY ANALYSIS

We analyze the security and anonymity of the new AnonDSR routing protocol in this section, and give a comparison of security, anonymity, and scalability with other security and anonymous ad hoc routing protocols.

4.1 Security Analysis

The main security attacks for a communication message include active attacks such as modification and replay, and passive attacks such as eavesdropping. First, in the security parameter establishment protocol, an adversary cannot change the messages such as the sequence number, the identities of the source and destination, and the security parameters in an RREQ packet, which the security type is secure or anonymous, since they are signed by the source node. It is the same for all messages in a RREP packet since the messages are signed by the destination. If an adversary changes the route record message in a RREQ packet, the corresponding RREP packet cannot go back to the source so that the source node knows the protocol failed. A replay attack can be easily found by the source by checking whether the session key and key index are the same as the one sent in the RREQ packet.

In the anonymous route discovery protocol, first, an adversary cannot successfully change the one-time public key, trapdoor, and onion sent by the source in an ANON-RREQ packet since they are signed by the source. Second, if the onion is changed or deleted partially by some adversaries during the route discovery, the destination cannot open it. An ANON-RREP packet uses the same protection mechanisms, i.e. the cryptographic onion and signature. A replay attack on the trapdoor can be easily found by checking whether or not the session key index N_K is updated since after each successful anonymous communication, the source and destination will update their session key and key index. A replay on the onion just causes the protocol to fail since a new anonymous path uses different route pseudonyms and intermediate forwarding nodes from all old anonymous communications.

In the anonymous data transfer protocol, all communication data are protected by the cryptographic onion. Any change and/or partial deletion made by an adversary would result in the packet not arriving at the receiver or the receiver not being able to open the packet, in which case the packet would be discarded. A replay on the data onion causes the same problems as above since each session creates different session keys and route pseudonyms.

4.2 Anonymity Analysis

Anonymity for communication data means preventing adversaries from linking the communication message with the source or the destination, and making an anonymous route so that each intermediate forwarding node only knows its local route pseudonym and does not know who else is on the same anonymous route.

In the security parameter establishment protocol, any adversary can easily find the source and destination nodes for the message just from the packet but they don't have any knowledge of the established session key and key index since these messages are encrypted. When the source uses the session key and key index to

create a trapdoor in the anonymous route discovery protocol, other nodes (except the source and destination) cannot link the trapdoor with the source and destination. A global adversary may find the source and destination by monitoring an ANON-RREQ packet, since the packet contains a same one-time public key during the path discovery. However, the adversary does not know the route pseudonym and session key created by each intermediate node. Therefore, when the destination node chooses an anonymous route and sends the corresponding ANON-RREP packet back, the adversary does not know who are the sender and receiver of the packet if all nodes on the network use dummy messages for routing protection. This is because all messages in the ANON-RREP packet are totally changed through each intermediate forwarding node by the cryptographic onion method.

In the anonymous data transfer protocol, there exists the same situation as above, i.e. any ANON-DATA packet is totally changed through each intermediate forwarding node by the cryptographic onion method. An adversary cannot link an incoming ANON-DATA packet with an outgoing ANON-DATA packet in a node if the node uses dummy messages to protect traffic analysis attacks. Thus the new ad hoc routing provides very good anonymity protection.

One possible traffic analysis attack on AnonDSR is a collusion attack launched by the intermediate forwarding nodes on the route but they only know they are on the same anonymous route and still do not know who are the source and destination.

4.3 Comparison with Other Security Ad Hoc Routings

We briefly compare our AnonDSR with several existing security and anonymous ad hoc routings such as Kargl et al.'s SDSR, El-Khatib et al.'s SDAR, and Kong-Hong's ANODR on their security, anonymity, and scalability.

- **Security**: AnonDSR, SDSR, and SDAR provide an end-to-end encryption protection but SDAR has some security problems in the *RREQ* phase as we analyzed it in Section 2 above. ANODR does not discuss how to protect the communication data.
- **Anonymity**: AnonDSR and SDAR provide very strong anonymity protection for both the anonymous communication route and the communication data. ANODR only provides anonymity protection for an anonymous route discovery but the communication data packet can very easily expose this anonymous route as we analyzed in Section 2 above. SDSR does not provide anonymity protection.
- **Scalability**: The major factors affecting scalability are cryptographic operations required for the security and anonymity in ad hoc routing. In order to give a good understanding of their scalability in Table 1, we categorize the cryptographic operations as three types: symmetric key operations as in encryption and decryption, efficient public key operations as in encryption with public key and verification of a digital signature, high computational complexity public key operations like decryption with the private key and signature; network nodes as two types: the intermediate forwarding nodes, as well as the source and destination nodes since the cryptographic operations on the intermediate forwarding nodes have a big effect on the scalability of the whole network; and protocol procedures as two types: *RREQ* phase and *RREP* phase since the RREQ packet usually is broadcasts to the whole network but the RREP packet only follows one path returned to the source node. The different categories may be used to give an overall picture of network scalability of the protocols.

Table 1 depicts a comparison of the different routing protocols on security, anonymity, and scalability. Note that we don't consider the data transfer phase since they all use efficient symmetric key mechanisms.

Based on the scalability analysis above, we know that the most important factor affecting the scalability of the routing is the high computational complexity public key operations in the intermediate forwarding nodes of the *RREQ* phase. The second most important factor affecting the scalability is the high computational complexity public key operations in the intermediate forwarding nodes of the *RREP* phase. From Table 1, we see that SDAR and SDSR scale poorly since the delay caused by the high complexity public key operations is very high, i.e. nL and $2nL$ times respectively. Obviously, AnonDSR and ANODR have very good scalability according to these two factors since they don't have these kinds of operations in the intermediate nodes. The third most important factor is the high computational complexity public key operations in the source and destination nodes but these operations only affect their own anonymous route establishment and do not have a big effect on the whole network. From Table 1, we know ANODR has better scalability than our AnonDSR but ANODR has a big issue on anonymity protection as we analyzed in Section 2. In addition, AnonDSR only has a total of $L+2$ time high computational complexity public key operations in the source and destination nodes for an anonymous route establishment based on Table 1, which means its main effect on scalability is caused by the hops between the source and destination not the number of the RREQ or RREP packets on an ad hoc network. It is very practical if the hops between the source and destination are not too many. We have simulated their performance in the next Section.

5. PERFORMANCE

5.1 Cryptographic Implementation

As we mentioned above, cryptographic processing is the most important factor affecting the scalability of security and anonymous ad hoc networks. To provide strong security protection and good scalability, AnonDSR combines symmetric key cryptographic algorithm, public key cryptographic algorithm, and hash function together. In our simulation, we use RSA-2048 (2048 bit modulus) as the public key cryptosystem, AES/Rijndael (128 bit key) as the symmetric key cryptosystem, and SHA-1 (160 bit) as the hash function. They are very popular and strong cryptosystems currently applied in commercial environment like Web bank. Table 2 depicts our testing results of their processing overhead based on actual measurement under Intel Pentium 4 computer: 3.00GHz (CPU) and 1.00GMB (RAM), Windows XP operation system, and IAIK JCE (3.0) [18] crypto package.

Table 1. Comparison of SDSR, SDAR, ANODR, and AnonDSR

Categories \ Protocols			SDSR	SDAR	ANODR	AnonDSR
Security			Good	Some Issues	N/A	Good
Anonymity			No	Good	Some Issues	Good
Scalability						
RREQ Phase	Each Inter Node	Symmetric Key Operations	0	*n*	2*n*	2*n*
		Efficient Public Key Operations	0	*n*	0	*n*
		Complexity Public Key Operations	0	*n*	0	0
	Src. Dest. Node	Symmetric Key Operations	0	1	3	3
		Efficient Public Key Operations	1	*L*	0	*2*
		Complexity Public Key Operations	2	*L*	0	*L*+1
RREP Phase	Each Inter Node	Symmetric Key Operations	0	*n*	*n*	*n*
		Efficient Public Key Operations	0	0	0	0
		Complexity Public Key Operations	2*n*	0	0	0
	Src. Dest. Node	Symmetric Key Operations	1	*L*+1	1	*L*+1
		Efficient Public Key Operations	*L*	0	0	1
		Complexity Public Key Operations	2	0	0	1

Note:

1. *n* is the number of different RREQ or RREP packets on the ad hoc network;
2. *L* is the number of hops of a RREQ or RREP packet from the source node to the destination node.

Table 2. Processing overhead of various cryptosystems (on Intel Pentium 4 computer 3.00GHz CPU)

Cryptosystem	Encryption	Decryption
AES/Rjindael (128 bit key & block)	128Mbps	128Mbps
RSA (2048 bit Modulus)	3ms	86ms
	Hashing	
SHA-1	161Mbps	

5.2 Simulation Platform and Metrics

The simulation platform is NS-2 (Network Simulator 2) [15] running on Intel Pentium 4 computer. The node processing and delay are based on the protocols and above actual measurement of the cryptographic systems. The network used during the tests has 100 nodes. Each node has number of neighbors from 2 to 4. The parameters used for simulation are:

(i) Route Establishing Time (ms): The time that a protocol spends in order to build a route between a source node and a destination node;

(ii) Hops: the number of hops that a route passes from a source node to a destination node, where the route is built by a protocol and the hops vary from 1 to 18 in this simulation relying on the distance between the source and destination nodes.

5.3 Simulation Results

5.3.1 Simulation on AnonDSR

As we described above, AnonDSR can provide three levels of security protection for different applications. In this simulation, we test the rout establishing time based on the hops between a source node and a destination node. During testing, there are about 100 routing request messages multicast in the network, i.e. each node sends one routing request message in the simulation network. Figure 5 depicts the simulation results.

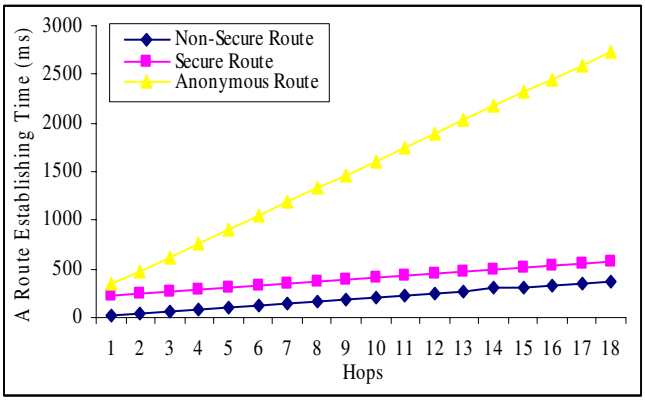

Figure 5. AnonDSR simulation results

From the simulation results it is clear that AnonDSR has very good scalability to establish a non-secure or secure route. For an anonymous route establishment, it cannot compete with non-secure and secure route establishment on scalability but the testing result shows that the scalability is very practical for real applications. According to the simulation, an anonymous route establishing time is less than 3 seconds under the conditions: 100 nodes network, 100 anonymous route request messages multicast on the network, 18 hops from a source node to a destination node, and RSA-2048 public key cryptosystem.

5.3.2 Comparison of DSR, SDSR, SDAR, and AnonDSR

In order to study the AnonDSR performance comparing with other protocols, we simulate four protocols in this simulation. The protocols include DSR [1, 2, 3], SDSR [5], SDAR [9, 13, 14], and AnonDSR. We simulate their route establishing time based on the number of hops from a source node to a destination node. For AnonDSR, we only simulate the anonymous route establishing time. The hops of a route vary from 1 to 18. Each node in the simulation network sends a request message and runs the protocols so that there are a total of 100 route request multicast messages during testing in the network. Figure 6 and 7 depicts the simulation results.

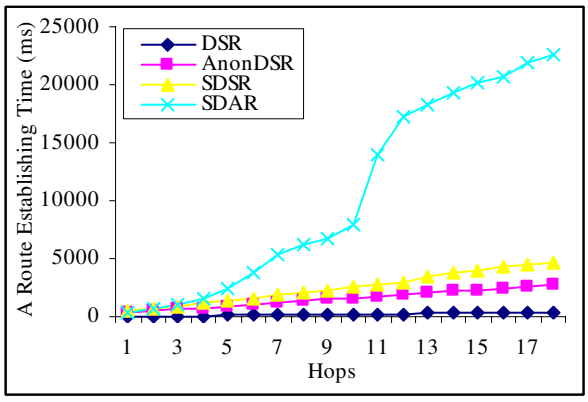

Figure 6. Performance of DSR, AnonDSR, SDSR, and SDAR

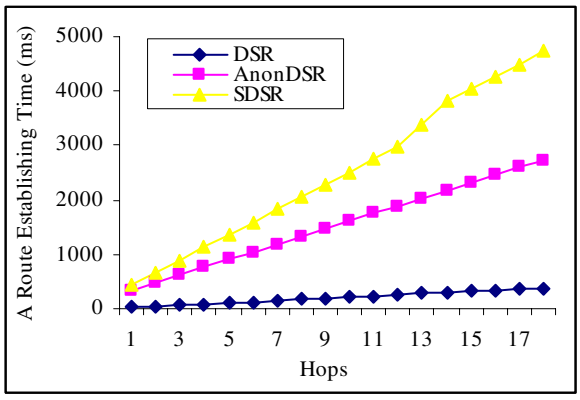

Figure 7. Performance of DSR, AnonDSR, and SDSR

From the testing results it is clear that AnonDSR has much better scalability than SDSR and SDAR when the hops between a source node and a destination node increase, especially comparing with SDAR for anonymous route establishment.

6. CONCLUSION

In order to provide an efficient, secure, and anonymous routing protocol for mobile ad hoc networks, we propose AnonDSR. We have analyzed and compared AnonDSR with other secure and anonymous ad hoc routing approaches such as SDSR, SDAR, and ANODR with respect to security, anonymity, and scalability, and demonstrated that AnonDSR has strong protection for user security and anonymity and very good scalability. Another advantage of AnonDSR is that it provides different security protection in order to satisfy various requirements of applications.

7. REFERENCES

[1] D. Johnson and D. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. In T. Imielinski and H. Korth (Eds.), Mobile Computing, chapter 5, pp. 152-181. Kluwer Academic Publishers, 1996.

[2] J. Broch, D. Johnson, and D. Maltz. The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks. Internet-Draft, draft-ietf-manet-dsr-03.txt, Oct. 1999.

[3] D. Johnson, D. Maltz, and J. Broch. DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks. In C. Perkins, editor, Ad Hoc Networking, chapter 5, pp. 139-172. Addison-Wesley, 2001.

[4] V. Vetriselvi and R. Parthasarathi. Secure Communication for Multi-Hop Ad-Hoc Network. IEEE TENCON 2003. Available at http://www.ewh.ieee.org/ecc/r10/Tencon2003/Articles/737.pdf.

[5] F. Kargl, A. Geiβ, S. Schlott, and M. Weber. Secure Dynamic Source Routing. In Proceedings of the 38th Hawaii International Conference on System Sciences, 2005.

[6] A. Menezes, P. Oorschot, and S. Vanstone. Handbook of Applied Cryptography. CRC Press Inc., 1997.

[7] R.Song and L.Korba. Review of Network-Based Approaches for Privacy. In Proceeding of the 14th Annual Canadian Information Technology Security Symporium, Ottawa, Canada. May. 2002. NRC 44905.

[8] J. Kong and X. 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), pp. 291--302, 2003.

[9] K. El-Khatib, L. Korba, R. Song, and G. Yee. Secure Dynamic Distributed Routing Algorithm for Ad Hoc Wireless Networks. In Proceeding of the ICPP 2003 First International Workshop on Wireless Security and Privacy (WiSPr2003), Kaohsiung, Taiwan, Oct.6-9, 2003. NRC 46517.

[10] R. L. Rivest, A. Shamir, and A. Adleman. A Method for Obtaining Signatures and Public Key Cryptosystems.

Communications of ACM, Vol. 21, No. 2, pp. 120-126, 1978.

[11] L. Korba and R. Song. Scalability, Security Technologies and Mobile Applications. In Proceeding of the first International Workshop on Mobility Aware Technologies and Applications (MATA'04), 2004. NRC 47169.

[12] D. Goldschlag, M. Reed, and P.Syverson. Onion Routing for Anonymous and Private Internet Connections. Communication of the ACM, Vol. 42, No. 2, pp. 39-41, 1999.

[13] A. Boukerche, K. El-Khatib, L. Xu, L. Korba. A Novel Solution for Achieving Anonymity in Wireless Ad Hoc Routing Protocol. Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor and Ubiquitous Networks (ACM PE-WASUN'2004), Venice, Italy, Oct. 4-6, 2004. NRC 47402.

[14] A. Boukerche, K. El-Khatib, L. Korba, L. Xu. A Secure Distributed Anonymous Routing Protocol for Ad Hoc Wireless Networks. Journal of Computer Communications, 2004. NRC 47393.

[15] The Network Simulator – ns-2. Available at http://www.isi.edu/nsnam/ns/. May, 2005.

[16] Y. C. Hu, A. Perrig, and D. B. Johnson. Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks. In Proceedings of MobiCom'02, Atlanta, Georgia, USA, Sep. 2002.

[17] P. Papadimitratos and Z. J. Haas. Secure Routing for Mobile Ad Hoc Networks. In Proceedings of CNDS'02, San Antonio, TX, USA, Jan. 2002.

[18] IAIK. Institute for Applied Information Processing and Communications. Available at http://jcewww.iaik.tu-graz.ac.at/.