An Energy-efficient and Real-time Anonymous Routing Protocol for Ad hoc Networks

Fangbin Liu*, Fengyu Liu, Hong Zhang
Institute of Computer Science, Nanjing University of Science and Technology
200 Xiao Ling Wei Street, Nanjing 210094, China
Corresponding author. Tel.: +86 13813871343.E-mail address:nj_lfb@163.com

***Abstract*—Anonymous routing in ad hoc networks has been one of the major concerns over the past few years. Most researches have only focused on the route anonymity without considering energy efficiency and real-time performance which are also major concerns in ad hoc networks. In this paper, we propose a novel efficient anonymous routing protocol which provides not only anonymity but also energy efficiency and real-time performance in ad hoc networks. Our analysis and simulation study verify that our protocol is much better than existing anonymous routing protocols on the aspects of energy efficiency and real-time performance.**

***Index Terms*— Energy-efficient; real-time; zero knowledge proofs; anonymous routing, Ad Hoc networks.**

I. INTRODUCTION

Mobile ad hoc networks (MANETs) established in a hostile environment is vulnerable to security threats. Due to the intrinsic characteristics of MANETs, such as node mobility and open wireless transmissions, the adversary can intercept routing information and obtain the source identity, destination identity and source-destination relationship, then trace, locate and physically destroy legitimate nodes. To prevent the adversary from getting the source identity, destination identity and source-destination relationship, many anonymous routing protocols have been proposed, such as SDAR [1], ANODR [2] and MASK [3]. Most of the proposed anonymous protocols satisfy the anonymity well, but each has at least one of the following problems: consuming a large amount of energy and providing poor real-time performance, which are unacceptable in MANETs where members move fast with limited energy. The reasons of the two problems are: 1. Causing a large amount of public key operations, such as in SDAR and ANODR. As we know, public key operations need a lot of computation time, and it consumes a large amount of energy and prolongs end-to-end delay. 2. Session keys should be established in advance through node detections, such as in SDAR, MASK and AnonDSR [4]. Because exchanging of session keys between two nodes (between two neighbors or between source and destination) incurs a lot of control packets and public key encryptions, it not only consumes a large amount of energy but also prolongs the route acquisition time. Therefore, in order to reduce energy consumption and guarantee real-time performance, we must avoid using public key operations and proactive node detections. In this paper, we propose an energy-efficient and real-time anonymous routing protocol for Ad hoc Networks (ERAP), which not only reduces public key encryptions but also avoids the establishment of session keys by proactive node detection.

The contributions of this paper are:

- We show that previous anonymous routing protocols, which consume a large amount of energy and decrease real-time performance, are unacceptable in MANETs. Nodes in MANETs are energy limited and fast-moving, but those anonymous routing protocols [1,2,3,4] incur a lot of public key encryptions, which require a large amount of energy and CPU time, or the use of proactive node detections to establish session keys between source and destination or between two neighbors, and this will incur a lot of control packets and public key encryptions. This calls for an efficient anonymous routing protocol which can reduce public key encryptions and guarantee real-time performance.
- We propose the Zero Knowledge Proofs of Trapdoor Protocol and use it to construct the trapdoor in the RREQ phase instead of using public key encryptions. Public key encryptions used in constructing trapdoor, which consume a large amount of energy and other resources of the mobile node, are unacceptable in the low-end mobile devices. This calls for an efficient way to construct the trapdoor. In this paper, we use Zero Knowledge Proofs of Trapdoor Protocol to construct trapdoor.
- We also propose an efficient anonymous routing protocol ERAP as the countermeasure. To reduce public key encryptions and guarantee real-time performance, ERAP adopts the following methods as the countermeasures. 1. Zero Knowledge Proofs of Trapdoor Protocol. 2. Anonymous Public Key (APK). 3. Establishment of shared session key between two neighbors only executes once. The three methods will be discussed in detail later.

The rest of the paper is organized in the following way: Section II elaborates related work, which shows that previous anonymous protocols consume a large amount of energy and reduce real-time performance. In section III, we describe the Zero Knowledge Proofs of Trapdoor Protocol. Section IV is about the underlying models. Section V is about our efficient anonymous routing protocol, which is referred to as ERAP. Section VI

doi:10.4304/jcp.7.4.900-912

presents the main features of our protocol and analyzes the anonymity and security of our protocols. In section VII, we evaluate ERAP's routing performance. Finally, sectionVIII summarizes this paper.

II. ANONYMOUS ROUTING REVISITED

Several anonymous ad hoc routing protocols have been proposed, for example, SDAR [1], ANODR [2], MASK [3], AnonDSR [4], EARP [5], RAODR [6] , CAR [7], MASR [8], but they all have the same problems: consuming a large amount of energy and providing poor real-time performance. Since SDAR [1], ANODR [2], MASK [3] and AnonDSR [4] are the typical protocols in those proposed anonymous protocols, we will analyze them in detail in the following.

A. SDAR [1]

Before the path discovery, first, SDAR establishes a Trust Management System [1], which helps SDAR to only use trust nodes to establish an anonymous path. To establish and maintain the Trust Management System, the network has to pay heavy overheads. Firstly, the central node keeps track of its neighbors by listening for a HELLO message which includes the sender's public key, and then it establishes session keys with its neighbors by using those public keys. When using public keys to establish session key, the nodes have to consume a large amount of energy, and because the HELLO message includes the sender's public key, the sender must pay more overhead than those nodes which use AODV[9] routing protocol to send the HELLO message. Secondly, in each community, the central node classifies its neighboring node into three categories based on their trust levels, and generates two different keys (MTLCK and HTLCK [1]) for the medium and high trust level. Take the descending of a node's trust level for example, when a node's trust level goes down from high to the medium trust level (probably because the node drops the packet silently without forwarding it or maliciously updates the packet before forwarding it [1]), the central node must update the HTLCK, which will incur a lot of public key encryptions and consume a large amount of energy.

When the Trust Management System has been established, a source node can establish a path based on the Trust Management System, but SDAR incurs a lot of public key encryptions during the RREQ phase. When a node X receives a RREQ packet, firstly it will try to decrypt the trapdoor, $E_{PK_R}(ID_R, K_S, PL_S)$ [1], to check if itself is the destined receiver, secondly it will sign the received packet, finally it will encrypt the ID of node X, a session key K_X and so on. All of the three steps are public key operations. More importantly, SDAR combines those public key operations with RREQ flood policy, which incurs many public key operations and consumes a large amount of energy. When the destination has received the RREQ message, it has to verify and decrypt the RREQ packet to get the session keys shared with the intermediate nodes, which will incur more public key operations. Let us compute the number of public key encryptions incurred by a source node by sending a RREQ. We assume there are N' nodes according to the trust level claimed by the source node in the whole network, L is the average path length and E denotes one public key encryption (we do not distinguish encryption from decryption).There are $(3*N'+2*L)*E$ public key encryptions per RREQ package, which consume a large amount of energy and decrease SDAR performance. This is unacceptable in the MANETs.

SDAR also exposes intermediate node's ID to the source and destination node, which makes the intermediate nodes unanonymous.

In the Trust Management System, when a node drops the packet silently without forwarding it, the central node would identify this as a malicious behavior and let the node's trust level go down. But SDAR let the node drop the same RREQ packet and other previously received data packets. Those two policies are inconsistent, and SDAR does not tell us how to solve it.

EARP [5] also introduces the Trust Management System, and it must consume a large amount of energy to establish and maintain the Trust Management System. EARP uses a temporary public key to collect intermediate nodes' IDs and session keys, which incurs a lot of public key encryptions and consume a large amount of energy. There is a problem in EARP that it uses a symmetric key to construct the trapdoor, but it does not give an approach to establish the symmetric key between source and destination nodes.

B. ANODR [2]

ANODR is an identity-free and on-demand anonymous routing protocol, which has many good features, but it uses public key encryption to construct the trapdoor, and this may cause some problems. Take node X for example, in the RREQ phase, when node X receives a RREQ packet, it will try to decrypt the trapdoor with its private key to check if itself is the destined node. Combined with RREQ flood policy, this will incur a great deal of public key operations and consumes a large amount of energy. Although ANODR uses the shared session key which is established in the first RREQ phase to construct the trapdoor in the following RREQ phases, the indeterminate nodes can not distinguish whether this RREQ packet is the first RREQ packet, so the indeterminate nodes must try both the session key and the private key to decrypt the trapdoor. As a result, the session key does not advance the performance of ANODR.

In the RREP phase, nodes on the path and their neighbors must try to decrypt the RREP packet with their one-time private keys to get the session keys, and verify whether they are the next nodes. Because there are many one-time private keys which are used in different path discovery phases in each node, they must try several times to decrypt one RREP packet. Let us compute the number of public key encryptions when a source node wants to establish an anonymous path. We assume there are N nodes in the MANETs, and every node has M neighbors and P one-time private keys on average. There are $(N + L*M*P/2)*E$ public key encryptions for each

RREQ and RREP phase, which normally consumes a large amount of energy and is unacceptable in the MANETs.

ANODR is a pure on-demand routing protocol and it does not need to detect nodes and establish session keys before rout discovery, so it has fewer control packets but better real-time performance than SDAR, MASK and AnonDSR.

C. MASK [3]

Before path discovery, MASK takes an anonymous neighborhood authentication [3] which is based on Pairing-based cryptography. Because Pairing-based cryptography is almost the same as public key cryptography in consuming CPU time and energy, the neighborhood authentication would incur a lot of computational overhead and consume a large amount of energy.

The big problem here is that MASK exposes the destination ID, which is unacceptable in anonymous routing protocol.

D. AnonDSR [4]

AnonDSR is consisted of three protocols. One of them is the Security Parameter Establishment Protocol whose RREQ packet is

$<RREQ,SecType,seqnum,ID_{src},ID_{dest},Rrec,SecPara>$ .When the source node wants to establish an anonymous route, SecType is anonymous. Therefore, when the adversary finds out that the SecType is anonymous, it knows the source wants to establish an anonymous route, and the source and destination are ID_{src} and ID_{dest} . In other words, the adversary can find the source and destination IDs from the Security Parameter Establishment Protocol in advance.

It also takes AnonDSR a long time to establish an anonymous route, because it must use the Security Parameter Establishment Protocol to establish the security parameters before anonymous route discovery. This will decrease the real-time performance, which is unacceptable in MANETs where nodes move fast. Execution of the Security Parameter Establishment Protocol also consumes a lot of energy.

The second protocol of AnonDSR is Anonymous Route Discovery Protocol, and it uses a one-time temporary public key created by the source node to collect the session key K_X created by the intermediate node X for the destination node in the RREQ phase. Because every RREQ packet is flooding over the whole network, every node in the network must use the one-time temporary public key to encrypt the session key for every RREQ packet, which incurs a lot of public key encryptions and consumes a large amount of energy.

E. Comparison

Based on the analyses above, we can see that the large amount of energy consumption and decrease of real-time performance occur in two places. One is during the decrypting of trapdoor and another is during the establishment of session-key between two neighbors or among source, destination and intermediate nodes.

Why do we have to use the trapdoor? When a source node sends a RREQ packet to a destined receiver, the destination can verify that it is the destined receiver from the RREQ packet, while other nodes can not verify the destination from the RREQ packet. MASK does not use a trapdoor in RREQ packet, so it exposes the destination ID. There are two ways to construct the trapdoor. One is to use session key shared by the source and destination. The problem of this method is that the shared key has to be established. AnonDSR uses the Security Parameter Establishment Protocol to establish the shared key, but it has the problem mentioned above. Another method is to use the public key of the destination node to construct the trapdoor, such as SDAR and ANODR, but it also has problem. So we must find an efficient way to construct the trapdoor, and that is the Zero Knowledge Proofs of Trapdoor Protocol (ZKPTP).

The routes established by SDAR and AnonDSR are MIX-net's onion-based schemes, so SDAR and AnonDSR must collect the intermediate nodes' session key for source and destination in the RREQ phase. As analyzed above, SDAR and AnonDSR incur too many public key encryptions and consume a large amount of energy when collecting those session keys, which is unacceptable in MANETs. Compared to SDAR and AnonDSR, routes established by ANODR and MASK are virtual circuit-based. However, MASK's proactive neighbor detection and authentication consume a large amount of energy as analyzed above, while in ANODR every two neighbors use a one-time public key (pk_1time_X) to establish a session key between them in RREP phase, which has a better performance than SDAR, AnonDSR and MASK, but ANODR also has a problem in RREP phase as analyzed above, and we solve this by introducing a Anonymous Public Key (APK).

III. ZERO KNOWLEDGE PROOFS OF TRAPDOOR PROTOCOL

A. zero-knowledge proofs

In 1985, Goldwasser, Micali, and Rackoff first proposed the basic concept of Zero-Knowledge Proof [10] . Zero-Knowledge Proof protocol is an instance of interactive proof protocol， by which a "prover" can prove possession of a certain piece of information to a "verifier" without revealing it. Take a password-based authentication for example, during the authentication the verifier authenticates the prover based on a password, so the verifier has certain knowledge of the password. Therefore, the verifier can impersonate the prover to a third party with whom the prover may share the same password. The Zero-Knowledge Proof protocol can prevent this kind of situation from happening, because the main objective of Zero-Knowledge Proof protocol is to make the prover be able to convince the verifier that it knows the secret without revealing any information of the secret itself.

Zero-Knowledge Proofs have the following features:

Completeness: The protocol is considered complete, if it works for an honest verifier and an honest prover with a

very high probability. The acceptable level of probability depends on the application.

Soundness: The probability that a prover who does not know the secret information can get away with it can be made arbitrarily small.

B. Zero Knowledge Proofs of Trapdoor Protocol

The purpose of trapdoor in anonymous routing protocols is to let the destination verify that it is the destined receiver, while other nodes can not learn anything about the identity of the destination from the trapdoor. It has the same purpose as that of Zero Knowledge Proofs Protocol. Therefore, we can borrow the Zero Knowledge Proofs Protocol to construct a trapdoor.

Our Zero Knowledge Proofs of Trapdoor Protocol (ZKPTP) is a variant of the Feige-Fiat-Shamir Identification Protocol[11], which is a perfect application of Zero Knowledge Proofs. Like in the Feige-Fiat-Shamir Identification Protocol, our scheme assumes the existence of a trusted center whose sole purpose is to publish a modulus n, which is the product of two large primes of the form 4r + 3. Such moduli (known as Blum integers) are used in a variety of cryptographic applications, and their most useful characteristic is that -1 is a quadratic non-residue whose Jacobi symbol is +1 (mod n). Everyone can use the universal n, but no one should know its factorization.

In the network, everyone, such as the destination node D, performs the following steps:

Choosing k random numbers $c_1,...,c_k$ in Z_n, which constitute the secret identity of D.

Choosing each d_i (randomly and independently) as $\pm 1/c_i^2 (\mathrm{mod}\, n)$, which constitutes the open identity of D.

That is

$$gcd(c_i,n)=1 \qquad (1)$$
$$d_i c_i^2 \equiv \pm 1 (\mathrm{mod}\, n)$$
$$d_i < n \quad 1 \le i \le n$$

Then we execute our ZKPTP as the following:

1. The source node S picks a subset $A=\{s_1,s_2,...,s_j\} \subseteq \{1,2,...,k\}$ and a number r randomly, computes $x= r^2 \ mod\ n$, and keeps r secret.

2. S uses the open identity of the destined node (node D) to compute the value y, and embed y^2, A, x into the RREQ packet.

$$W_d = \prod_{i=1}^{j} d_{s_i} \bmod n$$
$$y = rW_d \ mod\ n$$

3. When a node X receives the RREQ packet, it computes W_c with its secret identity.

$$W_c = \prod_{i=1}^{j} d_{c_i} \bmod n$$

And verifies

$$x \equiv y^2 W_c^4 (\mathrm{mod}\, n) \qquad (2)$$

4. If formula (2) is true, then node X is the destination D; otherwise, node X is not the destination.

C. Analysis of our Zero Knowledge Proofs of Trapdoor Protocol

In this section, we compare our ZKPTP with the Feige-Fiat-Shamir Identification Protocol and analyze its security and efficiency.

a. Comparison

In this section, we compare our ZKPTP with the Feige-Fiat-Shamir Identification Protocol.

1. Feige-Fiat-Shamir Identification Protocol is an interactive protocol which must be executed for more than once, so it is not applicable in the anonymous routing establishment process in which source and destination nodes just need to interact with each other for one time. While in the path discovery phase, our ZKPTP scheme is the same, and we can embed our ZKPTP into route discovery phase.

2. In our protocol, the prover (source node) uses the verifier's(destination node) identity to construct the "zero knowledge", while in Feige-Fiat-Shamir Identification Protocol the prover uses its own identity to construct the "zero knowledge".

3). The objective of our protocol is to verify whether the identity from which the "zero knowledge" originates belongs to the verifier, while the objective of Feige-Fiat-Shamir Identification Protocol is to verify whether the identity belongs to the prover.

b. Security

1. We assume that factoring is difficult and formula (1) is true, or the adversary can get n's factors p and q immediately.

2. Because factoring and square root extraction are computationally equivalent, the adversary could not obtain the secret identity of D from its open identity, neither do r from x.

3. Since $y = rW_d \ mod\ n$ and r is a random number, the adversary could not get the open identity of D from y, and r must be kept secret.

4. If X node is destination node, formula (2) should be true, because:

$$y^2 W_c^4 \equiv r^2 W_d^2 W_c^4 \equiv r^2 (W_d W_c^2)^2 \equiv r^2 \equiv x \ mod\ n$$

5. There might appear a collision that formula (2) is true while the open identity does not belong to D, but the probability is a negligible quantity. Let's compute the probability.

We assume the number of elements in subset A is J, and there are N nodes in the MANETs.

5.1. The probability of choosing the same subset A in two secret identities is:

$$P_1 \le \frac{1}{n^J}$$

5.2. The probability of the collision appearing once in the whole MANETs is:

$$P = 1-(1-P_1)^{m-1} \le 1-(1-\frac{1}{n^J})^{m-1}$$

In which, n is the product of two large primes, and P is a negligible quantity.

c. Efficiency

Compared with public key cryptograph, the Feige-Fiat-Shamir Identification Protocol needs much less computation, which is mentioned in [11].Since we can select a smaller subset A than that in Feige-Fiat-Shamir Identification Protocol(even though A only has one member, ZKPTP is still valid), and our ZKPTP is just executed once for every verification, our ZKPTP incurs less computation than Feige-Fiat-Shamir Identification Protocol. Therefore, our ZKPTP is more efficient than public key cryptograph.

IV. Underlying Models

A. Attacking model

Anonymity threats are from the attackers that are passive in nature. The attackers keep silent until physical attacks are launched. The passive threat model in [2] was characterized very well, which we will use in our protocol.

Eavesdropper and traffic analyst: Such an adversary can perform eavesdropping and collect information from intercepted traffic. The passive adversary can get the unprotected routing information, such as IP header, and, in regard to multihop routing, any unchanged packet characteristics like unique packet length and unchanged packet field (even if the field is encrypted in a semantically secure system). The adversary who is equipped with GPS can know nodes exact locations. We assume that there is a global adversary that can scan the entire network area in short delay round by round. The baseline traffic it can intercept is the routing traffic from the legitimate side. An eavesdropper with enough resource is capable of analyzing intercepted traffic on-the-scene. This ability gives the traffic analyst quick turnaround action time about the event it detects and reduces the chance of evasion for those victim nodes.

Node intruder: We must consider node compromise, and a passive node intruder keeps silent and participates in collaborative network operations (e.g., ad hoc routing) to boost its passive attack strength. This implies that a countermeasure must not be vulnerable to a single point of failure/compromise.

Colluding attackers: Adversaries having different levels of attacking ability can collaborate through a separated channel to combine their knowledge and to coordinate their attacking activities. A subset of network members may be compromised.

DoS attacks: An adversary can send fake route request or route response packets which exhaust the computation resources and energy of all nodes in the networks, and a countermeasure against the DoS attacks must be adopted.

There are also other attacks such as replay attack and timing analysis.

B. Network model

We assume wireless links are symmetric. If a node X is in transmission range of some node Y, then Y is in transmission range of X. A mobile node's physical interface is capable of using omni-directional radio to transmit packets. Within its transmission range, a network node can send a unicast packet to a specific node or a broadcast packet to all local nodes. A node may hide its identity pseudonym using an anonymous broadcast address. In 802.11, a distinguished predefined multicast address of all 1s can be used as source MAC address or destination MAC address to realize anonymity for local senders and receivers. In addition, by anonymous acknowledgment and retransmission, a local sender and a local receiver can implement locally reliable unicast. If the count of retransmission exceeds a predefined threshold, the sender considers the connection on the hop lost.

V. An Efficient Anonymous Routing Protocol for Ad hoc Networks

In this section, we introduce our efficient anonymous routing protocol for the establishment of anonymous paths in MANETs. The main objective of our protocol is to reduce energy consumption and guarantee real-time performance with the same level of anonymity and security. We divide our protocol into three phases: the path discovery phase, the path reverse phase and the data transfer phase. In the first phase, we just use the trapdoor to find the destination, neighbors establish session keys with each other in the second phase and the official data exchange is processed during the data transfer phase.

The main notations used in this paper are the following:

$seqnum$: The unique identifier for the RREQ message.

tr_{dest} : The trapdoor constructed by source node.

ID_{dest} : The identity of the destination node.

PK_{dest} : The public key of the destination node.

SK : The session key between source node and destination node.

K_f : A symmetric key generated by source node.

$E_{PK}(M)$: A message encrypted with public key PK.

$E_K(M)$: A message encrypted with symmetric key K.

$K_{X,X+1}$: A symmetric key generated by node X and used by node X and X+1.

$N_{X,X+1}$: A random pseudonym generated by node X and used by node X and X+1.

APK_X : The Anonymous Public Key used by node X. APK_X is a general public key, but other nodes can not find the mapping between APK_X and node X. Therefore, although some nodes may know the APK_X , they do not know whom it belongs to.

$H()$: A one-way hash function.

A. path discovery phase

The path discovery phase allows for a source node S which wants to communicate anonymously with destination node D to discover and establish a routing path through a number of intermediate nodes. The character of this phase is no public key operation. Before the path discovery phase, there is no proactive node detections to establish session keys.

The source node S triggers the path discovery phase by sending a route request(RREQ) message to all nodes within its wireless transmission range. The RREQ

message includes four parts. The first part is consisted of message type, RREQ, and a unique identifier for the message, seqnum. The second part contains $H(K_1),...,H(K_i),APK_X$ and tr_{dest} . $K_1,...,K_i$ are the session keys shared by node X and its neighbors. The node forwarding the RREQ message may have shared session keys established in other route establishment phases with some of its neighbors, so we do not have to establish them again. APK_X is generated by node X, which is used in a t time window for all RREQ messages which are received by node X during the t time window. APK_X will be used in the path reverse phase to establish a session key between node X and its neighbors if they have not shared a session key. tr_{dest} is a trapdoor which the source node constructs with our ZKPTP and $Tr_{dest} = (y^2, A, x)$. The third part is $E_{K_f}(seqnum)$, which is used by the intermediated nodes to check if the route reply message comes from the real destination node in the path reverse phase. The fourth part is consisted of $E_{PK_{dest}}(ID_{dest}, SK, K_f)$ and $E_{SK}(M_S)$. ID_{dest} , the destination node ID, SK , a shared session key between source node and destination node generated by source node, and K_f , a symmetric key generated by source node, all of them are encrypted with the destination node's publish key PK_{dest} . $M_S = (ID_{src}, Sign_{src})$ and $Sign_{src} = H(ID_{src}, ID_{dest}, SK, K_f, tr_{dest})$. The RREQ message is in the following format:

$$< \{RREQ, seqnum\}, \{H(K_1),...,H(K_i),APK_X, tr_{dest}\}, E_{K_f}(seqnum), \{E_{PK_{dest}}(ID_{dest}, SK, K_f), E_{SK}(M_S)\} >$$

We assume each node keeps a session key table which maps a session key HASH value to the session key. This is shown in Table 1.

TABLE 1. session key table

Index of KX-1,X	Session key shared by node X-1 and X	Pseudonym shared by node X-1 and X
H(KX-1,X)	KX-1,X	NX-1,X
…	…	…

TABLE 2. The routing tables of node

Session key shared with the ancestor node	Pseudonym shared with the ancestor node	Session key shared with the next node	Pseudonym shared with the next node	Anonymous public key of the ancestor node	*Seqnum* encrypted with Kf	The unique identifier for the RREQ message
KX-1,X	NX-1,X	KX,X+1	NX,X+1	APKX-1	$E_{K_f}(seqnum)$	*seqnum*

When a node X receives a RREQ message, it processes the message according to the following steps:

1. Check if the message has already been received from other nodes within its wireless transmission range by using the *seqnum* as the unique identifier of the message. If the message has been received already, drop it silently and stop; otherwise, continue.

2. Save the RREQ message, and replace APK_{X-1} and $H(K_1),...,H(K_i)$ with APK_X and $H(K_1)',...,H(K_j)'$ in node X, then broadcast the new RREQ message.

3. Check if it has shared a session key with node X-1 by comparing $H(K_1),...,H(K_i)$ with its session key table. If it finds a $H(K_{X-1,X})$ in its session key table, which means it has shared a session key with node X-1, it saves *seqnum* and $E_{K_f}(seqnum)$ to its routing table ,then execute 5; otherwise execute 4.

4. Save APK_{X-1} , *seqnum* and $E_{K_f}(seqnum)$ to its routing table, then execute 5.

5. Check if it is the destination node with the trapdoor. The node gets y^2 , A and x from tr_{dest} , and then execute steps 3 and 4 of ZKPTP. If it is the destination node, trigger the path reverse phase; otherwise drop the RREQ message and stop.

The path discovery phase is depicted in Figure 1.

$$RREQ_{S,A} = < \{RREQ, seqnum\}, \{H(K_{S1}),...H(K_{si}), APK_S, tr_{dest}\}, \{E_{K_f}(seqnum), E_{PK_{dest}}(ID_{dest}, SK, K_f), E_{SK}(M_S)\} >$$

$$RREQ_{A,B} = < \{RREQ, seqnum\}, \{H(K_{A1}),...H(K_{Aj}), APK_A, tr_{dest}\}, \{E_{K_f}(seqnum), E_{PK_{dest}}(ID_{dest}, SK, K_f), E_{SK}(M_S)\} >$$

$$RREQ_{B,C} = < \{RREQ, seqnum\}, \{H(K_{B1}),...H(K_{Bk}), APK_B, tr_{dest}\}, \{E_{K_f}(seqnum), E_{PK_{dest}}(ID_{dest}, SK, K_f), E_{SK}(M_S)\} >$$

$$RREQ_{C,D} = < \{RREQ, seqnum\}, \{H(K_{C1}),...H(K_{Cl}), APK_C, tr_{dest}\}, \{E_{K_f}(seqnum), E_{PK_{dest}}(ID_{dest}, SK, K_f), E_{SK}(M_S)\} >$$

Figure 1. Path Discovery Phase

B. Path reverse phase

When the destination node has received the RREQ message, it performs the following steps:

1. It gets ID_{dest} , SK and K_f from $E_{PK_{dest}}(ID_{dest}, SK, K_f)$, then uses SK to get ID_{src} and $Sign_{src}$ from $E_{SK}(M_S)$, and ensures none of ID_{src}, ID_{dest} , SK , K_f and tr_{dest} has been changed by comparing them with $Sign_{src}$.

2. If the destination node D has already shared a session key with its ancestor node, it constructs the RREP message as the following:

$<NRREP, N_{X,X+1}, E_{K_{X,X+1}}(seqnum, K_f')>$

NRREP means that the second part of the message is a pseudonym. $K_{X,X+1}$ is the shared session key between the destination node D and its ancestor, and $N_{X,X+1}$ is the corresponding pseudonym. K_f' is a commitment value of K_f , which comes from $E_{K_{X,X+1}}(seqnum, K_f')$ .In this case, node D and its ancestor do not have to establish a new session key, which will reduce public key operations and energy consumption.

Otherwise, there is no session key between them, and the constructed RREP message is:

$<ARREP, E_{APK_X}(K_{X,X+1}, N_{X,X+1}), E_{K_{X,X+1}}(seqnum, K_f')>$

ARREP means that the second part of the message is encrypted by an anonymous public key. APK_X is the anonymous public key of D's ancestor.

3. Then the destination node D broadcasts the RREP message.

When an intermediate node X receives the RREP message:

4. If the first part of the RREP message is NRREP, then

4.1. Node X finds $K_{X,X+1}$ in its routing table with $N_{X,X+1}$ which comes from the RREP message, then decrypts the third part of the RREP message and gets seqnum and K_f' .

4.2. It uses the seqnum to find the $E_{K_f}(seqnum)$ in its routing table and checks if $E_{K_f'}(seqnum) = E_{K_f}(seqnum)$ .If the RREP comes from the real destination node, the equation must be valid, and it goes to 4.3,otherwise node X drops the RREP message and stops.

4.3. It uses the seqnum to find the session key shared with its ancestor node X-1 in its routing table. If it succeeds, it constructs the following RREP message:

$<NRREP, N_{X-1,X}, E_{K_{X-1,X}}(seqnum, K_f')>$

Otherwise, it means there is no session key between them, and node X uses the seqnum to find the anonymous public key APK_{X-1} generated by its ancestor node X-1 in its routing table. It constructs the following RREP message:

$<ARREP, E_{APK_{X-1}}(K_{X-1,X}, N_{X-1,X}), E_{K_{X-1,X}}(seqnum, K_f')>$

In which, $K_{X-1,X}$ and $N_{X-1,X}$ are generated by node X, and node X saves them in its routing table.

4.4. Broadcast the RREP message.

5. If the first part of the RREP message is ARREP, then

5.1. Node X decrypts the second part of the RREP message by using the private key corresponding to the APK_X , and obtains $K_{X,X+1}$ .Then $K_{X,X+1}$ is used to decrypt the third part of the RREP message to obtain seqnum and K_f' .

5.2. Node X uses seqnum to find $E_{K_f}(seqnum)$ in its routing table. If there is no corresponding entry, node X drops the message and stops, otherwise it checks if $E_{K_f'}(seqnum) = E_{K_f}(seqnum)$ .If fails, node X drops the message and stops; otherwise it goes to 5.3.

5.3. The rest are as the same as steps 4.3 and 4.4 in the NRREP case.

The path reverse phase is depicted in Figure 2.

C. data transfer phase

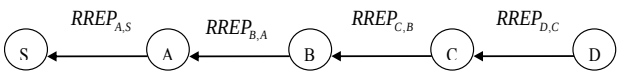
After the path reverse phase, the anonymous route has been established, which is shown in Figure 3.

The route table, take that of node B for example, is shown in Table 3.

When the source node S wants to send data to destination node D, it generates a data packet as the following and broadcasts it:

$<N_{S,A}, E_{K_{S,A}}(N'_{S,A}, E_{K_{S,D}}(data))>$

$N_{S,A}$ is the pseudonym shared with the next node, and $N'_{S,A}$ is a new pseudonym corresponding to $N_{S,A}$, which will be used in the next data transfer phase. $K_{S,A}$ and $K_{S,D}$ are the session keys shared with node A and the destination node.



$$RREP_{D,C} = < NRREP, N_{C,D}, E_{K_{C,D}}(seqnum, K_f') >$$
$$or < ARREP, E_{APK_C}(K_{C,D}, N_{C,D}), E_{K_{C,D}}(seqnum, K_f') >$$
$$RREP_{C,B} = < NRREP, N_{B,C}, E_{K_{B,C}}(seqnum, K_f') >$$
$$or < ARREP, E_{APK_B}(K_{B,C}, N_{B,C}), E_{K_{B,C}}(seqnum, K_f') >$$
$$RREP_{B,A} = < NRREP, N_{A,B}, E_{K_{A,B}}(seqnum, K_f') >$$
$$or < ARREP, E_{APK_A}(K_{A,B}, N_{A,B}), E_{K_{A,B}}(seqnum, K_f') >$$
$$RREP_{A,S} = < NRREP, N_{S,A}, E_{K_{S,A}}(seqnum, K_f') >$$
$$or < ARREP, E_{APK_S}(K_{S,A}, N_{S,A}), E_{K_{S,A}}(seqnum, K_f') >$$

Figure 2. path reverse phase

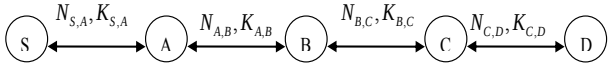
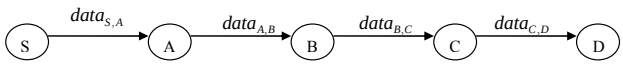


Figure 3. anonymous route

TABLE 3. Route table of node B

Session key shared with the ancestor node	Pseudonym shared with the ancestor node	Session key shared with the next node	Pseudonym shared with the next node
$K_{A,B}$	$N_{A,B}$	$K_{B,C}$	$N_{B,C}$



$$data_{S,A} =< N_{S,A}, E_{K_{S,A}}(N'_{S,A}, E_{K_{S,D}}(data)) >$$

$$data_{A,B} =< N_{A,B}, E_{K_{A,B}}(N'_{A,B}, E_{K_{S,D}}(data)) >$$

$$data_{B,C} =< N_{B,C}, E_{K_{B,C}}(N'_{B,C}, E_{K_{S,D}}(data)) >$$

$$data_{C,D} =< N_{C,D}, E_{K_{C,D}}(N'_{C,D}, E_{K_{S,D}}(data)) >$$

Figure 4. data packet transfers along the anonymous route

When node A receives the data packet, it checks if it has a pseudonym $N_{S,A}$ in its routing table. Obviously, $N_{S,A}$ is in A's routing table, then node A gets the session key $K_{S,A}$ corresponding to $N_{S,A}$ and uses the key to decrypt the second part of the data packet. A replaces $N_{S,A}$ with $N_{S,A}'$ in its routing table and generates a new pseudonym $N_{A,B}'$, then A encrypts $N_{A,B}'$ and $E_{K_{S,D}}(data)$ with $K_{A,B}$ and replaces $N_{S,A}$ with $N_{A,B}$ in the data packet. Finally, node A broadcasts the data packet, and the data packet will arrive at the destination along the anonymous path. The data packet transferred along the anonymous route is depicted in Figure 4.

VI. FEATURES OF THE ERAP PROTOCOL

The most important features of ERAP are energy efficiency and real-time performance at the same level of anonymity and security.

A. Energy efficiency

We reduce energy consumption in two ways. One is by reducing public key operations and another is by reducing control packets.

a. Reducing public key operations:

As we know, public key operation consumes a large amount of CPU time to encrypt a packet, which means it consumes a large amount of energy and prolongs end-to-end delay, but nodes in MANETs are energy limited and quickly moving, so we must try to avoid using public key operation.

In previous anonymous routing protocols, such as SDAR, ANODR, which normally use public key to construct the trapdoor, a large amount of public key operations are introduced. When we use the ZKPTP to construct the trapdoor, it should be energy efficient.

As we have compared in section II, ANODR has a better performance than other proposed protocols in establishing shared session key between neighbors, but it has a problem in the RREP phase as analyzed in 2.2, which incurs many public key operations. We use an Anonymous Public Key, not a one-time public key like in ANODR, to establish shared session key between two neighbors. Therefore, when a node receives a RREP packet, the node does not have to try several times to decrypt the packet; it only needs to use the anonymous private key. If there has been a session key between two neighbors, they do not have to establish a new session key again, and this characteristic as well as the Anonymous Public key will reduce public key operations.

b. Reducing control packets:

In MANETs, excessive control packets also consume a lot of energy, so a good protocol must introduce as few control packets as possible.

To maintain the Trust Management System, SDAR introduces a lot of control packets, as analyzed in section II. In MASK, before path discovery, an anonymous neighborhood authentication must be conducted, which incurs a lot of heavy control packets. Before anonymous route discovery, AnonDSR has to execute its Security Parameter Establishment Protocol to establish security parameters, which incurs additional control packets. Our protocol ERAP does not have to prepare anything before anonymous route discovery, so it does not incur additional control packets.

B. Real-time

Anonymous routing protocols, such as AnonDSR, must prepare some parameters before anonymous route discovery, and it prolongs the route establishment time. This is not applicable to MANETs in which nodes move fast. As we know, public key operations consume a lot of CPU time, so excessive public key operations in path establishment phase will prolong end-to-end delay. As our ERAP does not have to prepare anything before anonymous route discovery and has fewer public key operations than other anonymous protocols, it has a better real-time performance.

C. Not decreasing anonymity and security

a. Source and Destination nodes anonymity

Because the IDs of source and destination are encrypted, and the destination open identity used in trapdoor is protected by ZKPTP, the adversary can not analyze the IDs of source and destination in the path discovery phase. During the path reverse phase and data transfer phase, we only need to use node's pseudonym, so source and destination nodes are anonymous during path reverse phase and data transfer phase.

b. Intermediate nodes anonymity

The intermediate nodes only use their pseudonyms to participate into the route establishment, and those pseudonyms are updated regularly during the data transfer phase, so the adversary can not get their real IDs.

c. Untraceability

As RREQ messages are broadcasted, they can not be traced. During the path reverse phase and data transfer phase, every party of the message updates per hop, and the adversary can not find the relationship between two random messages, so it can not trace the message. We also use the Neighborhood traffic mixing mentioned in [12] to stop timing analysis.

d. DoS attack

To prevent a adversary from sending fake route request packets, we can introduce anonymous authentication in path discovery phase. There are many good anonymous

authentication protocols for Ad Hoc networks, such as [15] and [16], which can make nodes refuse broadcasting RREQ packets created by the adversary.

ANODR could not prevent DoS attack and it also can use a anonymous authentication protocol to prevent DoS attack.

If a adversary sends fake route reply packets, a legitimate node can refuse forwarding the RREP packets, as the legitimate node will find the equation $E_{K_f'}(seqnum) = E_{K_f}(seqnum)$ non-valid.

e. Node intruder and Colluding attackers

No matter an intruder on the path or not, it can not get anything about nodes on the path, because nodes on the path are anonymous. Even if an intruder is on the path, it can not know who the neighbor nodes are. Colluding attackers also can not get anything about nodes on the path.

f. replay attack

In every path discovery phase, we use different $seqnum$, SK and K_f , so the adversary can not replay RREQ packets. In the path reverse phase, a node discards those route reply packets with the same $E_{K_{X-1,X}}(seqnum, K_f')$, which can prevent replaying RREP packets.

D. Comparisons

In Table 4, we compare our ERAP with ANODR, SDAR, AnonDSR and MASK.

TABLE 4. The asymmetric encryption operations of the anonymous routing protocol

	ERAP	ANODR	SDAR	AnonDSR	MASK	EARP
Recipient anonymity	Protected by ZKPTP	Crypto-protected for destination	Crypto-protected for destination	Exposed by its Security Parameter Establishment Protocol	Exposed	Crypto-protected for destination
Neighbor exposure	No	No	Exposed	No	No	No
PKC in RREQ flood	0	0 or N*E	(3*N'+2*L)*E	N*E	0	(N'+L)*E
PKC in RREP	0-M*L*E	M*L*P*E/2	0	0	0	0
precondition	No	No	Establish the Trust Management System first	Establish security parameters first	take an anonymous neighborhood authentication first	Assume source and destination. have shared a symmetric key . Establish the Trust Management System first

N': the number of nodes in the Trust Management System

PKC: public key encryption

L: the average path length

E: denotes one public key encryption

N nodes in the MANETs

M: the number of one node' neighbors on average

P: the number of one node' one-time private keys on average

In ERAP, once there is a session key between two neighbors, they only use the session key in the path reverse phase, so the number of public key operation is less than M*L*E. With the time elapsing, the number will gradually reduce to 0, so the number of public key encryptions shown in the table is 0-M*L*E.

In ANODR, since it uses the destination node public key to construct trapdoor in the first path discovery phase, the number of public key is N*E. In the following path discovery phase, ANODR uses the session key to construct trapdoor, so the number of public key operations is 0.

VII. PERFORMANCE

In the simulation, the aforementioned ad-hoc anonymous routing protocols are presented for comparison with each other. Our simulation relates to the energy consumption especially incurred by the cryptosystems and the establishment of precondition, such as Trust Management System in SDAR and Establishment of security parameters in AnonDSR. We also focus on the real-time performance which is an important aspect of the mobile network.

We evaluate the performance of these protocols in terms of the following metrics. (i) Energy consumption: to establish an anonymous path, how much the average energy consumption is. (ii) Average route acquisition latency [13] : the average latency for discovering a route, i.e., the time elapsed between the first RREQ message and the first RREP message received by the source. (iii) Packet delivery fraction [13] : the ratio between the number of data packets received and those originated by the sources. (iv) Normalized control packet overhead [13] : the number of routing control packets transmitted by a node normalized by the number of delivered data packets, averaging over all the nodes. Each hop-wise transmission of a routing packet is counted as one transmission. Here, we do not consider MASK, which exposes the destination ID in the RREQ message.

A. Cryptographic Implementation

As we mentioned above, cryptographic processing is the most important factor which affects energy consumption and real-time performance in anonymous routing protocol. As energy consumption rises proportionally to the cryptographic processing time, we can use the data from Jiejun Kong [12]. The processing overhead of various cryptosystems is shown in Table 5.

TABLE 5. Processing overhead of various cryptosystems (on iPAQ3670 pocket PC with Intel Strong-ARM 206MHZ CPU) [12]

cryptosystem	decryption	encryption
RSA(1024-bit key)	900ms	30ms
ECAES(160-bit key)	42ms	160ms
El Gamal(1024-bit key)	80ms	100ms
AES/Rijndael(128-bit key & block)	29.2Mbps	29.1Mbps
RC6(128-bit key & block)	53.8Mbps	49.2Mbps
Mars(128-bit key &block)	36.8Mbps	36.8Mbps
Serpent(128-bit key &block)	15.2Mbps	17.2Mbps
TwoFish(128-bit key &block)	30.9Mbps	30.8Mbps

In our simulation, we use RSA (1024-bit key) as the public key cryptosystem, AES/Rijndael (128-bit) as the symmetric cryptosystem, and SHA-1(160-bit) as the hash function.

B. Simulation model

The simulation is based on ns-2 [14] network simulator which provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless networks.

We use three simulation models. The first one with different simulation time is:

The network field is 2000m x 2000m with 400 initially uniformly-distributed nodes. For the physical propagation model we used the two-way ground model. We applied the parameters of a 2.4GHZ Lucent Orinoco WaveLAN DSSS Radio Interface. The data rate was set to 11 Mb/s and the effective transmission distance of each node is 300m. We used 802.11 DCF as the link layer protocol. The node mobility model is a random-way point model. The pause time is fixed to be 60 seconds. The moving speed of the node is 5m/s. In other words, the minimum and maximum speeds are both 5m/s. CBR sources have been used to model data traffic. Every node generates a packet of 100 bytes per minute. The source-destination pairs are chosen randomly from all of the nodes. The simulation time is between 1200s and 9600s. All data is the average of fifty simulation results.

The second model with different moving speed is:

The nodes' mobility is controlled in such manner that minimum and maximum speeds are always the same, which are between 0 to 10m/s.The simulation time is 3600s, and other parameters are the same as those in the first model.

The third model with different number of nodes is:

The number of nodes is between 225 and 484.The simulation time is 3600s, and other parameters are the same as those in the first simulation model.

C. Routing performance measurement

Here, we present the results and divide these results into two groups. The first group is energy consumption impacted by mobility, simulation time and node number. The second group is consisted of Average route acquisition latency, Packet delivery fraction and Normalized control packet overhead under different node mobility.

a. Comparison of energy consumption

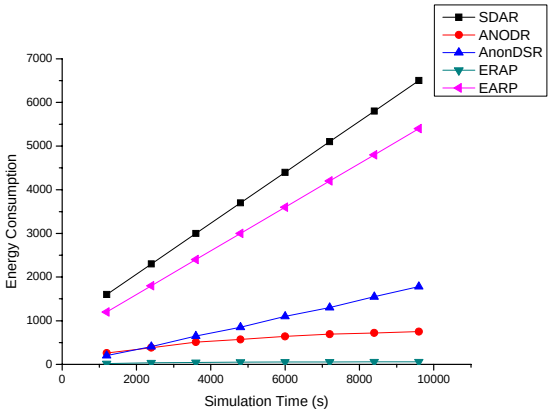


Figure 5.1. Energy consumption over simulation time

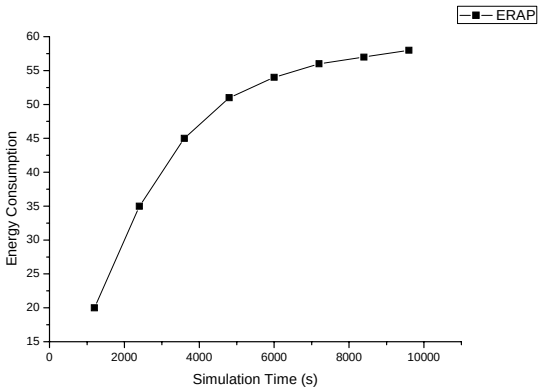


Figure 5.2. Energy consumption of ERAP over simulation time

Figure 5.1 and 5.2 illustrate change of energy consumption over the simulation time. Figure 5.1 shows the comparison between different protocols and Figure 5.2 shows specific energy consumption of ERAP. As shown in Figure 5.1, SDAR and EARP consume the most energy, because they use public key cryptography to establish and maintain the Trust Management System, thus incur lots of public key operations in path establishment. AnonDSR shows less energy consumption because it does not have to establish the Trust Management System, so it incurs fewer public key operations in path establishment. Since ANODR establishes session keys in the path reverse phase, it consumes less energy than AnonDSR. As shown in Figure 5.1 and 5.2, our ERAP consumes the least energy.

Because, firstly, it uses our ZKPTP to construct the trapdoor but not the public key cryptography; secondly, it establishes the session keys in the path reverse phase, and once a session key has been established between two neighbors, they do not need establish it again and can just use the session key to establish a path in the following path establishment. As shown in Figure 5.2, change of energy consumption slows down when the simulation time is over 5000s.The reason is that most of the neighbors have established session keys after 5000s, and they can use the session keys immediately.

Figure 6.1 and 6.2 illustrate change of energy consumption impacted by mobility. Figure 6.1 shows the comparison between different protocols and Figure 6.2 shows specific energy consumption of ERAP. SDAR also consumes the most energy, followed by EARP, AnonDSR, ANODR and ERAP. When the node speed becomes faster and faster, nodes in SDAR and EARP consume more and more energy to maintain the Trust Management System. As shown in Figure 6.2, change of energy consumption slows down when the speed reaches 6 m/s. When the node speed is not very fast, acceleration will incur more new neighbors, so it must establish more session keys with those new neighbors in the path reverse phase. But when the speed has reached a certain level, 6 m/s here, the number of new neighbors does not change any more, and there are no more session keys established, so change of energy consumption becomes slow.

Figure 7.1 and 7.2 illustrate change of energy consumption impacted by the number of nodes. Figure 7.1 shows the comparison between different protocols and Figure 7.2 shows specific energy consumption of ERAP. As shown in Figure 7.1, all of those protocols consume more and more energy as the number of nodes increases, and SDAR has the biggest increment while ERAP has the smallest.

b. Comparison of Protocol performance

Figure 8 shows the average delay of route acquisition under different node mobility. As shown in Figure 8, the latency for establishing a route by using SDAR is longer than others, while ERAP is the shortest one, because ERAP incurs the fewest public key operations in path establishment. AnonDSR takes a Security Parameter Establishment Protocol before anonymous route discovery, which also prolongs the route acquisition time.

Figure 9 shows the comparison of the packet delivery ratio. All the curves show a more of less steady decrement as mobility increases. The ERAP protocol presents the best performance while the SDAR protocol shows the quickest decreasing trend. Since SDAR ,EARP and AnonDSR incur public key operations both in the path discovery phase and at the destination nodes, all of them decrease more quickly than ERAP and ANODR. Because SDAR incurs more public key operations than EARP and AnonDSR in the path discovery phase, SDAR decreases more quickly than EARP and AnonDSR. As we know, ERAP only needs public key operation in the first path reverse phase, in other words, once two neighbors have a session key, they do not have to incur public key operation between them in later path reverse phases, so ERAP incurs less public key operations than ANODR in the path reverse phase. In the path discovery phase, ERAP uses ZKPTP to construct the trapdoor while ANODR uses the public key cryptograph, so ERAP decreases more slowly than ANODR. In a mobile environment, public key operation incurs excessive delay in the path establishment process, which makes it harder to establish and maintain routes, so the more public key operations a protocol has the faster it decreases.

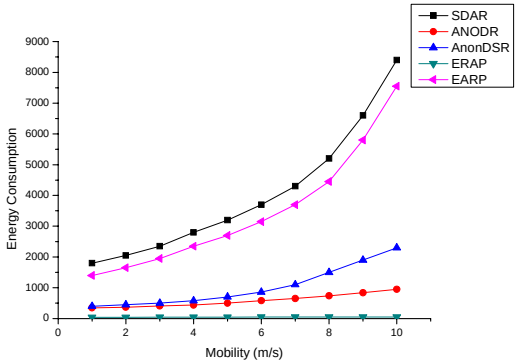


Figure 6.1. Energy consumption impacted by mobility

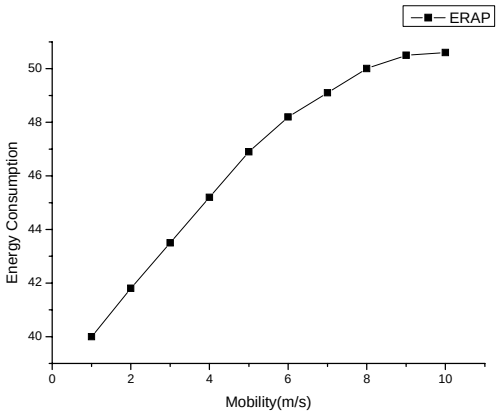


Figure 6.2. Energy consumption of ERAP impacted by mobility

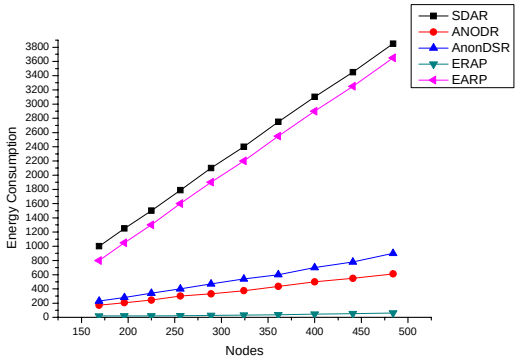


Figure 7.1. Energy consumption impacted by number of nodes

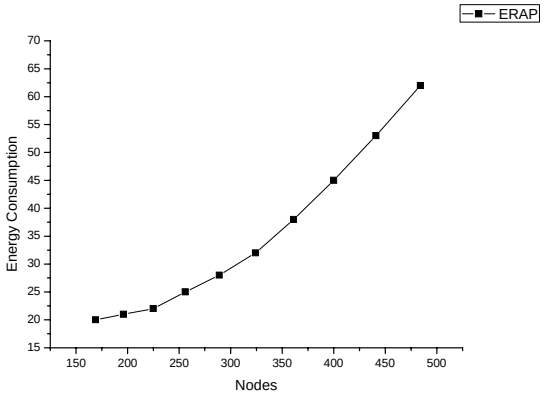


Figure 7.2. Energy consumption of ERAP impacted by number of nodes

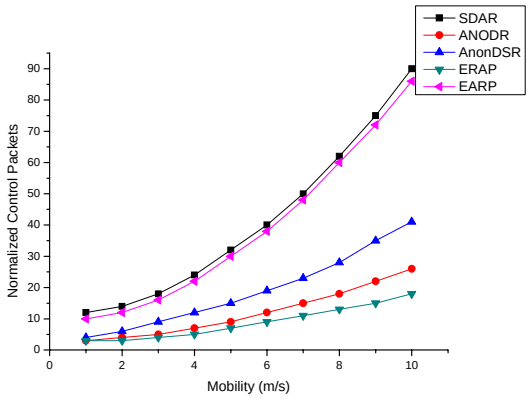


Figure 10. Normalized control packets

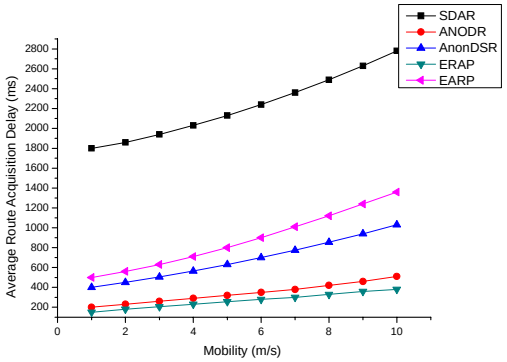


Figure 8. Route acquisition delay

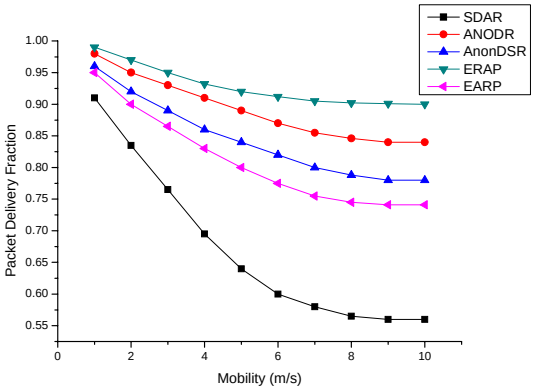


Figure 9. Packet delivery ratio

Figure 10 compares the number of normalized control packets of all protocols. SDAR has the highest number of normalized control packets, and the main reason is that SDAR must maintain the Trust Management System. So dose EARP. Since AnonDSR must take the Security Parameter Establishment Protocol before the anonymous path discovery phase, it has more normalized control packets than ANODR and ERAP.

VIII. CONCLUSION

Nodes in MANETs are energy limited and fast moving, but previous anonymous routing protocols have not considered energy efficiency and real-time performance. In this paper, we propose an efficient anonymous routing protocol for Ad Hoc networks, which provides not only anonymity but also energy efficiency and real-time performance. As public key cryptograph consumes a lot of CPU time, which means it will consume a large amount of energy and prolong end-to-end delay, we use the following methods to reduce public key operations. Firstly, we use our ZKPTP to construct the trapdoor; secondly, in the path reverse phase we use Anonymous Public Key to reduce public key operations; last but not least, once every two neighbors have established a session key, they do not have to establish the session key again in the following path establishment because they can use the old one. In our anonymous routing protocol, we also avoid using proactive node detections to establish session keys, and this enhances its performance.

REFERENCES

[1] Azzedine Boukerche, Khalil El-Khatib, Li Xu,Larry Korba. SDAR: A Secure Distributed Anonymous Routing Protocol for Wireless and Mobile Ad Hoc Networks. 29th Annual IEEE International Conference on Local Computer Networks (LCN'04), Tampa, Florida, USA, November, 2004

[2] Jiejun Kong, Xiaoyan Hong, Mario Gerla. An Identity-Free and On-Demand Routing Scheme against Anonymity Threats in Mobile Ad Hoc Networks. IEEE Transactions on Mobile Computing, Frequency,2007, Volume 6:888-902

[3] Yanchao Zhang, Wei Liu, Yuguang Fang. MASK: Anonymous On-Demand Routing in Mobile Ad Hoc Networks. IEEE Transactions on wireless communications, Vol.5. NO.9, September 2006

[4] Ronggong Song, Larry Korba, George Yee. AnonDSR:Efficient Anonymous Dynamic Source Routing for Mobile Ad-Hoc Networks. Proceedings of the 2005 ACM Workshop on Security of Ad Hoc and Sensor Networks, Alexandra, Virginia, USA , January， 2005

[5] Xiaoqing Li, Hui Li, Jianfeng Ma, Weidong Zhang. An Efficient Anonymous Routing Protocol for Mobile Ad Hoc

Networks, Information Assurance and Security, 2009. IAS '09. Fifth International Conference on , vol.2, no., pp.287-290, 18-20 Aug. 2009.

[6] Ronggong Song, Korba L. A robust anonymous ad hoc on-demand routing, Military Communications Conference, 2009. MILCOM 2009. IEEE , vol., no., pp.1-7, 18-21 Oct. 2009

[7] Shokri Reza, Yazdani Nasser, Khonsari Ahmad. Chain-Based Anonymous Routing for Wireless Ad Hoc Networks, Consumer Communications and Networking Conference, 2007. CCNC 2007. 4th IEEE , vol., no., pp.297-302, Jan. 2007

[8] Jun Pan, Jianhua Li. MASR: An Efficient Strong Anonymous Routing Protocol for Mobile Ad Hoc Networks, Management and Service Science, 2009. MASS '09. International Conference on, vol., no., pp.1-6

[9] Charles E.Perkins, Elizabeth M.Royer. Ad hoc On-Demand Distance Vector Routing Mobile Computing Systems and Applications. Proceedings. WMCSA'99. Second IEEE Workshop on, 1999

[10] Goldwasser,S.,S.Micali and C.Rackoff. Knowledge Complexity of Interactive Proof Systems, Proceedings of STOC 1985, PP. 291-304

[11] Uriel Feige,Amos Fiat and Adi Shamir. Zero Knowledge Proofs of Identity. Proceedings of the nineteenth annual ACM symposium on Theory of computing, New York, USA,1987

[12] Jiejun Kong, Xiaoyan Hong. ANODR: anonymous on Demand Routing with Untraceable Routes for Mobile Ad-hoc Networks. Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing table of contents, Annapolis, Maryland, USA, 2003

[13] J. Kong, J. Liu. On Performance Cost of On-demand Anonymous Routing Protocols in Mobile Ad Hoc Networks. Spring US, 2007.

[14] S.McCanne and S.Floyd,.Advances in Network Simulation[EB/OL]. http://www.isi.edu/nsnam/., July 2010

[15] Ciszkowski Tomasz, Kotulski Zbigniew. ANAP: Anonymous Authentication Protocol in Mobile Ad hoc Networks. eprint arXiv:cs, 2006.

[16] Jung Ha Paik, Bum Han Kim, Dong Hoon Lee. A3RP : Anonymous and Authenticated Ad Hoc Routing Protocol. 2008 International Conference on Information Security and Assurance, 2008.