Chain-based Anonymous Routing for Wireless Ad Hoc Networks

Reza Shokri, Nasser Yazdani, Ahmad Khonsari
ECE Department, University of Tehran, Tehran, Iran
r.shokri@ece.ut.ac.ir, yazdani@ut.ac.ir, ak@ipm.ir

Abstract— **Wireless ad hoc networks are so vulnerable to passive attacks and eavesdropping adversaries due to their shared medium which makes network traffic easy to capture and analyze. Therefore, security and privacy protections are of extreme importance for protocols and applications in such networks. In this paper, we introduce a new framework for anonymous routing, named *chain-based routing*, to improve the privacy. In our framework, nodes on a path are virtually bound to each other like a chain. Each node is only aware of its associated links in a flow and does not require any other information about source, destination, or other parts of the chain. Based on this framework, we propose an on-demand routing protocol, called Chain-based Anonymous Routing (CAR), which uses unicast-based broadcast data transfer to fulfill anonymous communication in wireless ad hoc networks. Through hiding identifiers of nodes inside the chain, CAR realizes sender, receiver, and relationship *anonymity* in addition to *untraceability* in the network. Moreover, it is resistant to a wide range of passive attacks while adapting to implement other security mechanisms in the presence of active attacks.**

Keywords-component; Wireless ad hoc networks, security, anonymity, chain-based routing, unicast-based broadcast.

I. INTRODUCTION

In a wireless ad hoc network, mobile nodes cooperate within an infrastructureless network. Due to the shared wireless medium and open peer-to-peer network architecture, ad hoc networks intrinsically are vulnerable to a wide range of security threats. Desire for a protected communication in a potentially hostile environment has made security and privacy a major challenging issue in such networks.

In this paper, we consider wireless ad hoc networks deployed in hostile environments. The shared and open medium of such networks introduces abundant opportunities for passive eavesdropping on data communications. This means that internal or external adversaries can easily overhear all the messages flying in the air even without physically compromising any node. While end-to-end security mechanisms can provide confidentiality for transferred data between communication nodes, some valuable information such as location and relationships of communicating entities might be determined from traffic analysis easily. This is extremely dangerous since adversaries can identify critical nodes and launch directed attacks to them. Here, we seek efficient solutions to dangerous passive attacks through anonymous communication in the network. This means that the communication partners are secret and nodes could communicate, while their identifier is not detectable from the exchanged messages.

We introduce a new framework for anonymous routing, named chain-based routing. In our framework, nodes on a path are virtually bounded to each other like a chain. Each node is only aware of its immediate neighbors in the chain, and does not require any other information about source, destination or other parts of the chain. As a result, sender, receiver and relationship anonymity are held in the framework and attackers are not capable of finding out the relationships among nodes by observing the system (unlinkability) [1]. Moreover, attackers have not the ability of tracing transferred packets to find the end points of a connection (untraceability), and also they cannot recognize packets belonging to the same ongoing communication flow in multiple sessions. Then, based on the proposed framework, we design a novel on-demand routing protocol, called CAR, which provides untraceability of routing, unlinkability among nodes, and anonymity in communications between wireless ad hoc devices. CAR ensures unlocatability and untrackability, meaning that although adversaries might know the real network identifiers, they are unable to decide whom and where the corresponding nodes are in the network and who is communicating with whom. Using a chain of fresh and secure identifiers and replacing the complex cryptography algorithms with simple and robust operations, the algorithm enables us to achieve a secure and flexible anonymity protocol with low overhead on nodes. As a main objective of our method the discovered routes are verified. This prevents the attackers from launching some DoS and replay attacks. CAR is an efficient routing protocol confirmed by detailed simulation results. It is also resistant to several threats and can be adapted to the other security methods to counter active attacks.

The rest of the paper is organized as follows. Section II explains the related works. In Section III, we illustrate our framework for chain-based anonymous routing. Section IV demonstrates the system model. CAR is explained in Section V. Evaluation and performance analysis are illustrated in Sections VI and VII. Finally, Section VIII concludes the paper.

II. RELATED WORK

In this section, we briefly revisit several anonymous routing schemes recently proposed for wireless ad hoc networks. Almost all of the methods are inspired from fixed wired network techniques [2][3][4][5]. Unfortunately special characteristics of ad hoc networks such as absence of fixed infrastructure, mobility of nodes, energy and computation power constraints make previous approaches inapplicable to mobile ad hoc networks. Some works address the anonymity problem in terms of routing schemes. [6] explores the use of mixes [3] in MANETs. An anonymous on-demand routing protocol, named ANODR, has been proposed in [7] to conceal

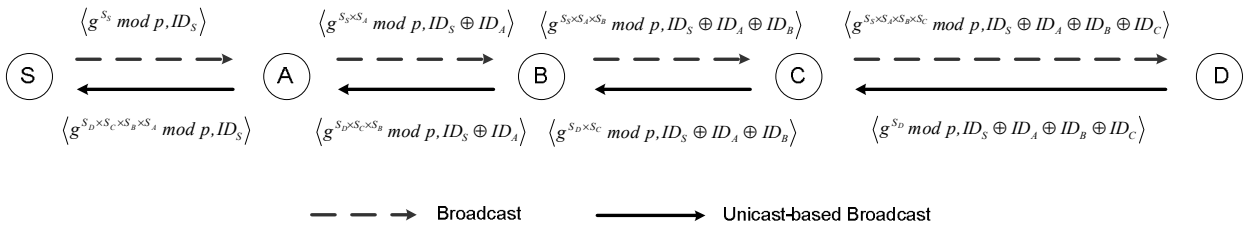


Figure 1. the sequence of <*PathChain*, *PathID*> pairs exchange in a chain between S and D.

network identifiers of communicating nodes using onion routing. In ASR [8] and MASK [9], nodes forwarding a route request message keep the state of the messages to reforward them later. SDAR [10] is another protocol which uses encrypted route messages to achieve anonymity. ARM [11], as another method, uses random data padding in addition to an onion like technique [5] to harden the protocol in front of passive adversaries. Moreover, the effect of mobility on an anonymous routing protocol is shown in [12] illustrating the impact of anonymous algorithms on routing performance.

Low efficiency and large overhead on relay nodes in route discovery and forwarding phases are the main drawbacks of the mentioned protocols. In addition, in most of these methods the sender should be aware of the entire path. Moreover, some of the protocols put the overhead of several encoding/decoding operations on either relays or end nodes that makes them inapplicable to ad hoc networks.

III. CHAIN-BASED ROUTING FRAMEWORK

First, we explain the idea behind the chain-based routing as an on-demand routing framework. The main objective of our framework is to allow end nodes authenticate each other and verify the path while discover routes anonymously. Every node, either an end points or a relay node, assigns a fresh secret key to each session which is used in the construction of the chain between the source and destination nodes. During the route discovery phase, in which the source node broadcasts its request to the network, each node alters the message and rebroadcasts it to its neighbors. So, several messages reach to the destination having traversed multiple paths (chains). Every node in a chain has two links corresponding to its upstream and downstream nodes labeled with different unique identifiers fastened together using secret keys of the nodes. More precisely, we extend the Diffie-Hellman key exchange method [13] to create the chain. Definition 1 explains the procedure in details.

Definition 1. Consider k sequential nodes in a particular path $\langle n_1,...,n_i,...,n_k \rangle$, a certain prime number p, and a generator g where $g \in Z_p^*$. Each node i calculates f_i as follows and passes it to the next node along the path:

$$f_i = \begin{cases} g^{S_i} \bmod p & i = 1 \\ f_{i-1}^{S_i} \bmod p & i \neq 1 \end{cases}$$

where S_i is the secret key of node i.

We define $PathChain_{1,k}$ as the value of f_k, which is the result of subsequent calculation of f_i, in the path sequentially from the first node to the last one (k^{th} node). Consequently, value of $PathChain_{1,k}$ equals f_k. We have:

$$\begin{aligned} f_k &= (((g^{S_1} \bmod p)^{S_2} \bmod p)...)^{S_k} \bmod p \\ &= g^{\prod_{j=\{1,...,k\}} S_j} \bmod p \end{aligned}$$

Obviously, $PathChain_{1,k}$ strictly depends on all nodes along the path. Furthermore, it is not possible for any node to recognize its position among the path. Hence, no *PathChain* value should reveal any information about its traveled path.

Proposition 1. For a given path $\langle n_1,...,n_i,...,n_k \rangle$, the value of $PathChain_{1,k}$ equals to $PathChain_{k,1}$, where p, g, and S_i are the same for both of them.

Proof. The straight implementation of Definition 1 results in $PathChain_{1,k}$. While, backward implementation of it ($PathChain_{k,1}$) obviously results in the following value:

$$\begin{aligned} PathChain_{k,1} &= (((g^{S_k} \bmod p)^{S_{k-1}} \bmod p)...)^{S_1} \bmod p \\ &= g^{\prod_{j=\{k,...,1\}} S_j} \bmod p = PathChain_{1,k} \end{aligned}$$

This is the same as $PathChain_{1,k}$. ■

Thanks to the difficulty of discrete logarithmic in Diffie-Hellman problem, owner of a secrete key only could relates the *PathChain* values of its immediate nodes in the chain while no other one can do this. To construct the chain a route discovery performs by the sender to provide various possible paths. Response from the other end of connection stabilizes the chain. Based on the Proposition 1, for a given path, constructed chain between the source and destination is equals to the reverse chain value. Then, both source and destination nodes could authenticate each other and verify the path using the final *PathChain* value. One consideration is that several *PathChain* values from multiple paths might reach to the destination whether it must reply to one of them (or many of them in multipath routing). Since the *PathChain* is not reversible, to direct the reply message through the same path that the request has traversed, we use *PathID* as demonstrated in Definition 2. The reversible manner of *PathID* is shown in Proposition 2.

Definition 2. Consider k sequential nodes in a particular path $\langle n_1,...,n_i,...,n_k \rangle$, $PathID_{1,i}$ (calculated from the first node to the i^{th} node) and $PathID_{k,i}$ (calculated in reverse order from the last node to the i^{th} node) are defined as following:

$$PathID_{1,i} = \begin{cases} ID_1 & i = 1 \\ PathID_{1,i-1} \oplus ID_i & i \neq 1 \end{cases}, \quad PathID_{k,i} = \begin{cases} PathID_{1,k} \oplus ID_k & i = k \\ PathID_{k,i+1} \oplus ID_i & i \neq k \end{cases}$$

where ID_i is the current transient identifier of node i.

Proposition 2. For a given path $\langle n_1,...,n_i,...,n_k \rangle$, the value of $PathID_{1,i}$ is equal to $PathID_{k,i+1}$ for an intermediate node i.

Proof. It is straightforward. ■

Based on Proposition 2, using *PathID* in addition to *PathChain* enables us to route the Route Replay message corresponding to a *PathChain* through the same nodes it traversed in the Route Request phase. Figure 1 shows the sequence of <*PathChain*, *PathID*> pairs are exchanged in a chain during route discovery process. After the route discovery, data packets are routed in the network by means of *PathChain*.

To conclude, our framework has a number of characteristics including non-source-based routing, having flexible and reliable route selection, resilience against any spoofing and hijacking in the path discovery phase, and imposing very low overhead on nodes during data forwarding. Moreover, while all of communications are established anonymously, no global eavesdropper can link two nodes as suspect for having contact. More details about the behavior of chain-based routing in front of multiple adversaries are discussed in Section V.

IV. System Model

Before presenting our protocol, we state our assumptions, data model, and model of adversary. Moreover, Table I summarizes the main notations used in the paper.

A. Assumptions

In order to represent the problem formally and to make it practical and conceivable for implementation and analysis, we have used the following assumptions, which have been presented in other previous studies [7][8][9][11]. Links between wireless nodes are bi-directional. So, routes between nodes could be symmetric. Each wireless node has a unique identifier (e.g. IP address) which makes the node uniquely recognizable. Each node has a certificate and private key by means of a trusted certificate authority in key predistribution phase. Besides, every node knows the certificates of its peer. Each node has a pool of pre-calculated large prime numbers (p) and their corresponding primitive roots in Z_p^*. There are some adversaries in the network such that they intentionally do not follow routing or forwarding protocols and also may want to expose the traffic information. Any active attacker which has compromised some nodes in the network owns all the cryptographic key information of them and distributes such information among other attackers.

B. Data Model

Our method employs some data structures which will be covered in detail along the paper. The major data structures used in our protocols are as following:

Flow Table – Is kept in the source and destination, contains the state information of a session after route setup. Attributes of the table are: *PeerID*, *SessionKey*, *Identifier*, *Key*, *Nonce*, *PrimeNumber*, *Generator*, *RoundNumber*, and *FlowActiveness*.

Forwarding Table – Contains information necessary to forward packets in relay nodes. Attributes of forwarding table are: *Down-PathID*, *Down-PathChain*, *Up-PathChain*, *Identifier*, *Key*, *RoundNumber*, *NACK*, and *RouteActiveness*.

RREQ message – Is the exchanged message during the Route Request phase. It contains the following fields: *Trapdoor*, *PathID*, *PathChain*, *PrimeNumber*, and *RoundNumber*. The "*Trapdoor*" is compound by:

$$Trapdoor_j = E_{PK_D}\left(ID_s \,||\, ID_D \,||\, N_S r_j \,||\, G_S r_j \,||\, j\right) \quad (1)$$

RREP message – Is the message exchanged during the Route Reply phase. It includes following fields: *Response*, *PathID*, and *PathChain*. The "*Response*" is compound by:

$$Response_j = E_{SK_j}\left(ID_S \,||\, ID_D \,||\, N_S r_j + 1 \,||\, j\right) \quad (2)$$

RERR message – Is used in route maintenance phase when a route failure detected. It only consists of two fields: *PathChain*, and *RoundNumber*.

Data Packet – Our method modifies the header of the IP packet such that the source and destination fields are replaced by two new fields for the sake of anonymity. Also, the address fields of MAC header are replaced with the broadcast address. The header of data packets consists of two fields: *PathChain* and *SequenceNumber*.

C. Adversary Model

Our attacker model consists of two main classes of attackers: passive and active. A passive attacker does not send any message; it just eavesdrops on the network and tries to find the relationships between nodes by observing the network traffic. An active attacker injects packets into the network while it eavesdrops as well. Such attackers attempt to corrupt the network protocols. In this paper we also take into account some cooperating attackers inside the network.

V. CAR Protocol

To realize anonymous communication in wireless ad hoc networks we present an on-demand routing protocol, called CAR, to establish chains between source-destination pairs. In this protocol, each node maintains the flow table and forwarding table, as mentioned in the previous section. Like all on-demand routing protocols, CAR has two major stages: route discovery and route maintenance. After the route discovery process, nodes can communicate with each other using *PathChains* field.

TABLE I. NOTATIONS USED IN THE PAPER.

S:	Source node of a communication.	$K_i r_j$:	The DH private key of node i in j^{th} round.
D:	Destination node of a communication.	$ID_i r_j$:	The transient forwarding identity of node i in j^{th} round.
PV_i:	The private key of node i.	$Enc_k(M)$:	Encryption of message M with the key k.
PK_i:	The public key of node i corresponding to PV_i.	$Dec_k(M)$:	Decryption of message M with the key k.
SK_j:	The session key of j^{th} round.	$N_i r_j$:	A nonce generated by node i for using in j^{th} round.
$P_i r_j$:	A large prime number, used in j^{th} round by node i.	$G_i r_j$:	A primitive root of $P_i r_j$ in node i. ($G_i r_j \in Z_p^*$).

A. Route Request

Route Discovery is initiated by sending RREQ (Route Request) message whenever a source node needs to communicate with another node for which it has no routing information in its flow table. The format of RREQ message has been illustrated in Section IV.B. The source node first creates *Trapdoor* by using the destination public key as illustrated in (1). Later, the *Trapdoor* is used by other nodes receiving this message to check whether the packet destined for them. The *Nonce* and *RoundNumber* fields of *Trapdoor* are used to prevent replay and modification attacks in the path. In addition, the *Trapdoor* and *RoundNumber* fields are used for loop-free routes construction. After construction of the RREQ message, the source node broadcast it in the network.

Every node, *i*, receiving RREQ checks whether it is the destination of the message. To do this, it tries to decrypt Trapdoor of the message by its private key, PV_i, and then compares the ID_D part of the result with its real identifier and *RoundNumber* of the message with the *RoundNumber* field of decrypted *Trapdoor*. If the node is not the intended destination, then it only forwards the packet. It regenerates *PathID* and *PathChain* parameters by means of message header contents and its local keys and then broadcasts the updated RREQ to its neighbors, besides inserting new corresponding entry to its forwarding table. It is worth noting that *PathIDs* are regenerated for every new route discovery. As a result, *PathIDs* are gained from a compromised node give no information to the attackers from the other connections of that node. On the other hand, duplicated messages must be dropped. Each node recognizes the RREQ that had been previously forwarded, by checking the *Trapdoor* and *RoundNumber* fields in its forwarding table. If there is an entry corresponds to them, the packet is duplicated.

In CAR, it is possible that several RREQ messages from multiple paths reach to the destination from a unique source. In such situations, destination node selects the first one, because it almost sure is received from the shortest path. Consequently, it creates a new entry related to the message in its flow table and replies to RREQ by starting the Route Reply phase.

B. Route Reply

In the first step, destination node calculates the final *PathChain* and sets it to the session key (SK_j) of the connection. As mentioned before, since the final *PathChain* is created in the source node is equal to SK_j, this is used by the source node to authenticate the destination. Moreover, since the session key is depends on all of the nodes in the path, it verifies that the RREQ and RREP (Route Reply) messages pass through the same sequence of nodes. Thus, the destination node creates a Response field in the RREP message, encrypted with SK_j as illustrated in (2). While the only node that can open the Trapdoor field of RREQ is the intended destination, no other node possibly will be aware about the real identifiers encrypted in it. Putting these identifiers encrypted in the *Response* field helps to authenticate the destination node. Adding the *RoundNumber* and incremented *Nonce* in the *Response* field, makes replay attacks impossible.

Using the generator field of *Trapdoor* ($G_s r_j$) and the PrimeNumber of the RREQ message ($P_s r_j$), the destination node first generates *PathChain* value and puts it in the RREP message. Besides, *PathID* of the RREP message is equal to the same *PathID* field of the received RREQ message. After constructing the RREP message, the destination node broadcasts it to its neighbors. Immediate sending of this message may lead to increasing potential interference around the node, because of the reception of several RREQ message from multiple paths. Thus, we delay the packet based on a uniform distribution from 0 to 5ms. In contrast to the Route Request phase, where the message is broadcasted throughout the network, in Route Reply phase, control message is sent back to the source node through the unicast-based broadcast. In this phase, identification of duplicated packets is done by checking the Response filed. A node receiving an RREP propagates the first RREP towards the source. If it receives further RREPs, it updates its forwarding information and propagates the RREP only if the RREP contains a new *Response* value. As the RREP travels back to the source, each node which has a forwarding entry related to the message updates and enables its forwarding entry and relays RREP to the next node by broadcasting it. When the packet reaches to the source node, it tries to decrypt the *Response* filed. To insure that the packet is received from the desirable destination and path, the source node compares the decrypted attributes of *Response* with its Flow Table. If there is an entry corresponding to them, it updates its Flow Table. Afterward, data transmission will be done in the reserved path through the network.

C. Route Maintenance

Movement of nodes which are not laid along an active path does not affect the corresponding routing information. If any node belonging to a path moves and consequently becomes unreachable from its neighbors, path breakage occurs. Furthermore, if the source node still requires a route to the destination, as an alternative it can restart the discovery process. To determine whether a route is still needed, a node may check whether the route has been used recently, as well as inspect upper-layer protocol's control blocks to see if the connection remains open. In addition, it can only remove the corresponding entry for such a destination from its flow table. Consequently when the source wants to send a packet destined for that destination it gets informed about path breakage.

To inform the source of a path from link failure along the path, intermediate nodes must be aware of unreachablity of upstream neighbors along the path. Every node in the network maintains a simple variable about reachability of next hop nodes in its routing table. In CAR, when a node forwards a packet, it can not determine whether the next hop in the path has received the packet correctly, since the receiver never sends any acknowledgment. Due to the wireless medium of the network, when a node transmits a packet, the predecessor node along the path hears it and can treat it as the implicit acknowledgement. For each transmitted packet corresponding to a route entry, the node increases the *NACK* (not acknowledged yet) by one. In addition, for each implicit acknowledgement from the next hop, the node decreases the variable by one. When *NACK* reaches to a threshold (here three), the corresponding forwarding entry is erased and a route error (RERR) message is sent towards the source of

corresponding route to inform it from a link failure within the path. A RERR message consists of the *Down-PathChain* of erased forwarding entry. Every node acts with RERR messages as same as data packets. Furthermore, if it has a route for the message, the corresponding forwarding entry will be erased.

D. Data Forwarding

Data dissemination is very simple in CAR. The source node of a connection first puts the *PathChain* value of the entry in the Flow Table which is correspondent to the desired destination node and also new sequence number of packet for current session, on data packet. Then, it broadcast the packet to its neighbors. Only one node (next node in the chain) has a *Down-PathChain* which is equal to the packet's *PathChain*. Other nodes drop the packet and prevent from broadcasting of the message to the network. Furthermore, the packet traverses a strip area around the chain through the network to reach the destination. In fact, destination is the node which has a flow table entry with *PathChain* value equal to the packet's one.

VI. Security Analysis

We analyze our protocol in the presence of several active and passive attacks and show that CAR ensures anonymity, unlinkability and untraceability in the network.

Modification attacks compromise the integrity of routing computations. In our proposed algorithm, an attacker can modify control and data packets. If the attacker modifies RREQ messages, it will be dropped by receiving neighbors. Since every node broadcasts the RREQ messages to its neighbors, the probability that an attacker can prevent establishment of a route is extremely low. Modifying RREP, RERR, and data messages can alter the route and hence the flow of communication between source and destination. But, as the modified message will be dropped by all the receiving nodes, the altered flow is stopped as near as possible and will not affect the whole network. This will not allow an attacker to redirect the messages toward a different destination or increase the delay of communication. Another problem arises when a selfish node wants to save battery life for its own communication and endangers the correct network operation simply by not participating in the routing protocol or by not executing the packet forwarding. This problem, called *Lack of Cooperation*, is solved in CAR by means of multi-path routing. In other words a selfish node cannot alter the procedure because other nodes cooperate to introduce all other possible routes to the destination. Here, we relax the problem of *Impersonation* and *Fabrication*, since they can be addressed by an effective authentication protocol and are out of the scope of this paper.

In case of passive attacks, we analyze the power of an eavesdropper to degrade the anonymity of our protocol. Since real identifier of mobile nodes is kept confidential during running of the protocol, we have successfully realized anonymous communication. In other words, sender and receiver anonymity and their relationship anonymity have been achieved. Based on the inherent characteristics of the defined chain, in route discovery and also forwarding phase it is not clear for a node that a message comes directly from the source or just a relay node. On the other hand, after set up of the route a forwarder cannot specify which neighbor will capture the message. Also, it is not apparent even for the last relay nodes which neighbor is the destination of a connection. In other words, every node is unaware of its location in the route. Moreover, an adversarial eavesdropper learns nothing more than some seemingly random numbers from the transient messages. Subsequently, no node can trace any route or data message to discover end points of the connection and hence, untraceability is realized in CAR. It is worth noting that when all of the nodes along a path are compromised, the path is traceable and linkable.

VII. Performance Analysis

Here, we evaluate the performance of CAR and compares to other routing protocols through simulation experiments.

A. Simulation Model

We have carried out several experiences using Network Simulator, ns-2 [14]. Distributed Coordination Function (DCF) of IEEE 802.11 is used as MAC layer and an unslotted Carrier Sense Multiple Access (CSMA) technique with collision avoidance (CSMA/CA) is used to transmit data packets. We have simulated an ad hoc network with 150 nodes uniformly deployed in a 2400×600 m^2 square field. The transmission range was 250m. Random Waypoint mobility model was used to simulate node's motion behavior. According to the model, a node travels to a random chosen location in a certain speed and stays for a whole before going to another random location. In our simulation, mobility speed varied from 0 to 10 m/sec, and the pause time was set to 30 seconds. Continuous bit rate (CBR) traffic sessions were used to generate network data traffic. For each session, data packets of 512 bytes were generated in a rate of 4 packets per second. Source-Destination pairs were random over the network. During running of our simulation for 10 minutes, 5 pairs of nodes were maintained to communicate. Simulations were conducted in identical network scenarios and routing configuration across all the schemes. Results were averaged over multiple runs with different seeds for the random number generators.

The processing overhead used in our simulation is based on actual measurement on a pocket PC. Table II shows the measurement presented by [15] on the performance of different cryptosystems. For public key cryptosystems, the table shows processing latency per operation, and for symmetric key cryptosystems, it shows encryption/decryption bit-rate. The overhead of modular arithmetic used in CAR for generating *PathChain* values in the route discovery phase is about half of the Tate paring overhead have been used in MASK. Besides, due to offline generation of large prime numbers and corresponding generators used in chain manipulation, the related overhead is ignored. For MASK and ANODR, taking consideration of the crypto systems proposed by the original authors, we choose the cryptosystem specification for simulation like [16].

TABLE II. PROCESSING OVERHEAD OF VARIOUS CRYPTOSYSTEMS (ON INTEL STRONGARM 200MHZ CPU BASED POCKET PC, LINUX)

Cryptosystem	Decryption	Encryption
ECC (163-bit key)	24.5 ms	46.5 ms
AES/Rijndael (128-bit key & block)	29.2 Mbps	29.1 Mbps

We evaluate the performance of our protocol in relation to these metrics: *packet delivery fraction* (the ratio of the data packets delivered to the destinations to those generated by the sources), *normalized routing load* (the number of routing packets transmitted per data packet delivered at the destination), and the *average data packets end-to-end delay* (the time from when the source generates the data packet to when the destination receives it. This includes all possible delays caused by buffering during route discovery latency, queuing at the interface queue, retransmission delays at the MAC, and propagation and transfer times).

B. Simulation Results

Figure 2 shows comparison of mentioned metrics among CAR, AODV, ANODR, and MASK. Diagram (a) shows the effect of mobility on packet delivery fraction. The original AODV protocol indicates the best performance possible on this metric. CAR in comparison to other anonymous protocols uses faster operation and this leads to better results in low mobility speeds. On the other hand, in high speed mobility, due to the more potential interference on established routes in unicast-based broadcast model, more packets will be dropped and the delivery fraction is the same as the other ones. Diagram (b) illustrates the data packet latency. Thanks to the very low overhead on relay nodes in CAR, the end-to-end latency of data packets is almost near the AODV. Also, MASK has lower latency in contrast with ANODR, as a result of more efficient symmetric encryption algorithms used. Diagram (c) compares the normalized control overhead in terms of bytes. ANODR generates the most normalized control bytes. CAR and MASK put less control packet overhead on the network in comparison to other anonymous routing protocols. In low speed mobility, CAR generates lesser control packets than the others. However, interference in the high mobility which leads to route error messages and restarting of route discovery causes little more normalized control bytes.

VIII. CONCLUSION

Security and privacy are one of the most challenging issues in wireless ad hoc networks because of their open nature of communication. In this paper, we have proposed a novel and general framework for anonymous routing, called chain-based routing, to achieve anonymity in routing and data dissemination. Based on this framework, a new anonymous routing protocol, named CAR, have been designed for wireless ad hoc networks. By a careful and simple design, CAR provides anonymity of sender and receiver in any communication, as well as node unlocatability. Also, it realizes relationship anonymity and connection untraceability in spite of any internal or external eavesdroppers. Moreover, we have shown that CAR is resilient to different types of active attacks as well as passive ones. Finally, due to low computational overhead of CAR and based on the experimental results our protocol could be implemented as a versatile and cost effective anonymous routing protocol in wireless ad hoc networks. As the drawback of CAR we can point to the high potential of interference in the network in high mobility while broadcast packets conflict with unicast-based broadcast flows.

REFERENCES

[1] A. Pfitzmann, M. Köhntopp. "Anonymity, Unobservability, and Pseudonymity – A Proposal for Terminology", 2005.

[2] M. Reiter, A.D. Rubin. "Crowds: Anonymity for Web Transactions". ACM Transactions on Information and System Security, 1998.

[3] D. Chaum. "Untraceable Electronic Mail, Return Address, and Digital Pseudonyms", Communication of the ACM, 1981.

[4] D. Chaum. "The Dining Cryptographers Problem: Unconditional Sender and Recipient Untraceability", Journal of Cryptography, 1988.

[5] M. Reed, et. al. "Proxies for Anonymous Routing". In 12th Annual Computer Security Application Conference, 1995.

[6] S. Jiang, N. Vaidya, W. Zhao, "Dynamic mix method in wireless ad hoc networks", In IEEE Milcom'01, 2001.

[7] J Kong, X Hong. "ANODR: ANonymous On Demand Routing with Untraceable Routes for Mobile Ad-hoc Networks", In MOBIHOC 2003.

[8] B. Zhu, et. al. "Anonymous secure routing in mobile ad hoc networks", In IEEE LCN, 2004.

[9] Y. Zhang, W. Liu, W. Lou. "Anonymous Communications in Mobile Ad Hoc Networks", In IEEE Infocom, 2005.

[10] A. Boukerche, et.al. "A Novel Solution for Achieving Anonymity in Wireless Ad Hoc Networks", In the ACM Workshop on Performance Evaluation of Wireless Ad hoc, Sensor, and Ubiquitous Networks, 2004.

[11] S. Seys, B. Preneel. "ARM: Anonymous Routing Protocol for Mobile Ad Hoc Networks", In International Conference on Advanced Information Networking and Applications, 2006.

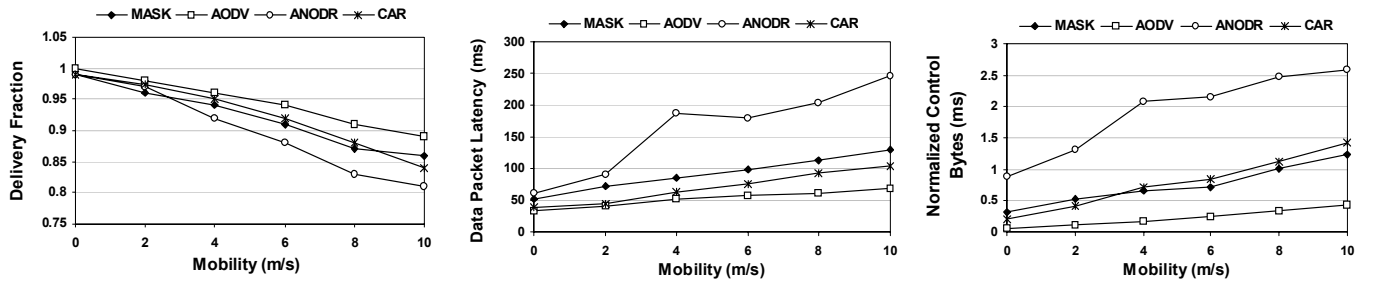
[12] J. Kong, et. al. "Mobility Changes Anonymity: Mobile Ad Hoc Networks Need Efficient Anonymous Routing", In IEEE Symposium on Computers and Communications (ISCC), 2005.

[13] W. Diffie and M. E. Hellman. "New directions in cryptography", In IEEE Trans. Inform. Theory, 1976.

[14] NS-2, available at http://www.isi.edu/nsnam/ns/.

[15] V. Gupta, et. al. "Performance Analysis of Elliptic Curve Cryptography for SSL", In ACM Workshop on Wireless Security, 2002.

[16] Jun Liu, Jiejun Kong, Xiaoyan Hong, Mario Gerla ,"Performance Evaluation of Anonymous Routing Protocols in MANETs", IEEE Wireless Communications and Networking Conference 2006, April 2006.



(a) Packet Delivery vs. Mobility Rate (b) Data Packet Latency vs. Mobility Rate (c) Normalized Control Bytes vs. Mobility Rate

Figure 2. Comparison of simulation results between CAR and other routing protocols (AODV, ANODR, and MASK).