The Hierarchical Ad Hoc Networks Anonymous Routing Scheme

Yuan Xiaopeng, Wu Jianjun, Guo Xian, Feng Tao, Li Xinghua, and Wang Jing

***Abstract*—Anonymous routing protocol is designed for avoiding from being leaked by nodes during communication in mobile ad hoc networks and insuring the communication route can't be discovered. Whereas most proposed anonymous secure routing protocols known in Ad Hoc networks can't meet anonymity sufficient or apply to hierarchical Ad Hoc networks. In addition, the protocols almost didn't have formal proof. Our mechanism based on ring signature, looking for honest neighbors. The source *node* set "trap-door" by using the public key of destination. Adopt the on-demand routing protocol to complete this program. This new protocols meet the hierarchical Ad Hoc networks by joining the network identifier. Finally, we adopt the formal proof tool UC(universally composable model) to prove the safety and anonymity of the new protocols.**

***Index Terms*—Ad hoc, ring signature, UC, anonymous routing.**

I. Introduction

Anonymous communication was first proposed by Chaum [1]. Depending on the object of protection, anonymous form of protection can be divided into three types: sender anonymity, recipient anonymity and relationship anonymity. Researchers made improvements in the classic on-demand routing protocols AODV and DSR in order to meet anonymous communication. Such as ANODR ASR, MASK [2]. However most of these protocols only considered relationship anonymity. Intermediate nodes communication will bring malicious nods, which may lead to information leaking that can't meet anonymity sufficient. Lin *et al.* [3] proposed an anonymous authentication routing protocol which is based on ring signature. it integrates a suit of interoperative authenticated key exchange mechanisms into the routing algorithm design, which can insure the intermidate nods are honst nodes. Whereas the proposed ring signature can't unforgeable against chosen-subring attacks [4]. In this paper will introduce a new anonymous routing scheme which has three advantages: (1) it can provides not only anonymity to the route from source to destination, but sender anonymity. (2) it is suitable for hierarchical ad hoc networks. (3) it achieves universally composable security.

II. Preliminaries

Ring Signature. Bender *et al.* [4] gave a strict definition of the ring signature. Kazuki *et al.* [5] gave an ideal function of UC security according to the anonymous program to achieve anonymity and unforgeability. They also proved the Bender's protocols meet the UC security.

Gen: Generate signing key pair $(pk_s, sk_s) \leftarrow Gen'(1^k)$, Generate encryption key pair $(pk_E, sk_E) \leftarrow EGen'(1^k)$ and erase sk_E, Choose an initial ZAP (2-round, public-coin, witness indistinguishable proof system) message $r \leftarrow \{0,1\}^{l(k)}$, Output the public key $PK := (pk_S, pk_E, r)$, and the secret key $SK := sk_S$.

Sign: Parse each PK_i as $(pk_{S.i}, pk_{E,i}, r_i)$, and parse SK_{i^*} as SK_{S,i^*}. Set $R_E := \{pk_{E,1}, 1, \cdots pk_{E,n}\}$ and $R_S = \{pk_{S,1}, 1, \cdots pk_{S,n}\}$. Set $M^* := M \mid PK_1 \mid \cdots \mid PK_n$, where "|" denotes concatenation. Compute the signature $\sigma'_{i*} \leftarrow Sign'_{sk_{S,i*}}(M^*)$. Choose a random coins ω_0, ω_1 and compute $C_0^* := Enc^*_{R_E}(\sigma'_{i*}; \omega_0)$ and $C_1^* : Enc^*_{R_E}(0^k; \omega_0)$. For $j \in \{0,1\}$, let x_j denote the statement: "$(R_S, M^*, R_E, C_j^*) \in L$", and let $x := x_0 \vee x_1$. compute the proof $\pi \leftarrow P_{r1}(x, (pk_{s,i*}, \sigma'_{i*}, \omega_0))$. The signature is $\sigma := (C_0^*, C_1^*, \pi)$.

Verify: Parse σ as (C_0^*, C_1^*, π). Parse each PK_i as $(pk_{S.i}, pk_{E,i}, r_i)$. Set $M^* := M \mid PK_1 \mid \cdots \mid PK_n$; set $R_E := \{pk_{E,1}, 1, \cdots pk_{E,n}\}$; and set $R_S = \{pk_{S,1}, 1, \cdots pk_{S,n}\}$. For $j \in \{0,1\}$, let x_j denote the statement "$(R_S, M^*, R_E, C_j^*) \in L$", and let $x := x_0 \vee x_1$. Output $v_{r1}(x, \pi)$.

UC Model. Canetti first proposed a universally composable(UC) framework [6], UC model can be combined to ensure the protocol security. Under UC model, the protocol running in the case with other different protocols, or as a system component of a protocol, still security guarantees agreement.

UC Emulation. A protocol *π* UC-realizes an ideal functionality *F* if for any real-life adversary *A*, there exists an ideal adversary *S* such that for any environment *Z*, the

Manuscript received June 13, 2014; revised September 2, 2014. This work was supported in by The National Natural Science Foundation of China under Grant 61462060 and 61461027. This work also supported in by Science and Technology Program of Gansu Province under Grant 1308RJZA277 and 145RJZA078.

Yuan xiaopeng is with High-level School Admissions Office of Gansu Province, Lanzhou, China.

Wu Jianjun, Guo Xian, and Wang jing are with School of Computer and Communication Lanzhou University of Technology, Lanzhou, China (e-mail: wangjing@lut.cn).

Feng Tao is with School of Computer and Communication Lanzhou University of Technology, Lanzhou, he is also with the Ministry of Education Key Laboratory of Computer Networks and Information Security Xidian University, Xi'an, China.

Li Xinghua is with the Ministry of Education Key Laboratory of Computer Networks and Information Security Xidian University, Xi'an, China.

DOI: 10.7763/IJCCE.2015.V4.374

probability that *Z* is able to distinguish between an interaction with *A* and real parties running protocol *π* and an interaction with *S* and dummy parties accessing *F* in the ideal process is at most a negligible probability.

Composition Theorem. Let *ρ* be a protocol that securely realizes the ideal functionality *F*, and let *π* be a protocol in the F-hybrid model. We say that $\pi^{\rho/F}$, with the ideal functionality *F* which is replaced by *ρ*, UC-realizes *π*. In particular, if *π* securely realizes the ideal functionality *G* in the F-hybrid model, then $\pi^{\rho/F}$ securely realizes *G* from scratch.

III. Anonymous Routing Scheme for Hierarchical Ad Hoc Networks

A. Networks of Background

Here, the background of what we study is based on the hierarchical ad hoc networks [7], Network nodes are divided according multi-frequency approach. Participating in the network nodes can be divided into the cluster head nodes and ordinary nodes. We shall make the following assumptions:

1) A bidirectional link between two mobile nodes within the transmission range can be established
2) Nodes in the networks have enough computing power and decryption operation perform
3) Each node has a unique identity (ID). the source node can get the destination node's ID(if the destination node in the other subgroup, including the network identity(*NF*))
4) Each node maintains two tables: one is a local neighborhood table (Fig. 1), another is local route table(Fig. 2) whose format is as follows:

Neighbor Address	Session Key	TTL

Fig. 1. Local neighborhood table.

Seq#	Dest_ID	Ancestor	Successor	TTL

Fig. 2. Local route table.

Neighbor Address: neighbor node's address; Session Key: records its session key between itself and the corresponding neighbor, TTL: time to live, if it hits 0, the table removes. Seq#: a number represents a unique route, it maintains the route freshness, prevent replay attacks. Dest_ID: the identity of the destination for the source or the identity of the source for the destination. Ancestor: records its upstream node's address. Successor: records its downstream node's.

B. Adversary Model

Adversary capabilities can generally be divided into passive or active. Attack's goal is to identify the sender and recipient information. Passive attacks monitoring the network traffic data and analysis information. We consider the adversary which controls several nodes of network but does not have full control over the entire network.

Neigborhood Anonymous Authentication Key Exchange Protocol. Cluster head nodes and ordinary nodes do anonymous authentication key exchage protocol based on introduced ring signature. They want to anonymously authenticate each other, where both Alice and Bob know that they are talking to an authentic peer in the ring without knowing the real identity of their peer. They both choose a random number $x \in Z_q^*$ and $y \in Z_q^*$ and compute xP, yP (where *P* is the system parameter), when the authentication succeeds, they record address in their trust neighborhood table and possess a new session key at the end of protocol (See Fig. 3).

$$\begin{aligned} k_{ab} &= x(yP)\cdots Alice \\ &= y(xP)\cdots Bob \end{aligned}$$

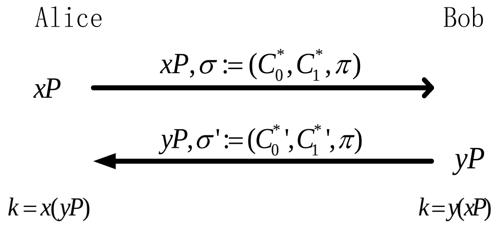
Alice
Bob
$xP, \sigma := (C_0^*, C_1^*, \pi)$
xP
$yP, \sigma' := (C_0^*{}', C_1^*{}', \pi)$
yP
$k = x(yP)$
$k = y(xP)$

Fig. 3. Key exchange based on the ring signature.

C. Key Pre-distribution Phase

We assume an offline security manger (SM) exists for identity check and pre-distribution:

1) G_1 is an additive group of prime order *q*, G_2 is a multiplicative group with the same order as G_1 and $\hat{e}$: $G_1 \times G_1$, G_2 be the bilinear pairing. Define a secure hash functions H_1: $\{0,1\}^* \to G_1$;
2) SM choose random number $s \in Z_q^*$ as its master key and computes $P_{pub} = sG$ as its public key;
3) SM calculates for each node an identity-based public/private key pair (PK_N, SK_N), where $PK_N = H_1(ID_N), SK_N = sPK_N$;
4) Each node is preloaded with the public parameters $< G_1, G_2, \hat{e}, q, G, P_{pub}, H_1 >$ and its private key;
5) SM sends cluster head node its network identifier (*NF*)

D. Route Discovery Phase

1) Intra-group anonymous routing request

Step 1: *S* generates a unique number seq#, selects a random number $a \in [1, p-1]$ to compute g^a and $H(g^a \parallel K_{SD} \parallel 0)$, where $K_{SD} = \hat{e}(H(ID_D), SK_{ID_S})$. then *S* makes $M_S = [H_1(ID_S), ID_D, g^a, H(g^a \parallel K_{SD} \parallel 0)]$, after that ,we use the public key of destination *D* to encrypt M_S as $C_S = E_{PK_D}(M_S)$, *S* also sets the number of hops from *S* to *D* as HopCount, $ARREQ =< RREQ, seq\#, HopCount, Cs >$ Forward the packet, change its route table as Fig. 4.

Step 2: Upon receiving the ARREQ, intermediate nodes go through the following procedure: Check if it is from one of its trusted neighbor nodes based on the sender's address, if so, it continues. Otherwise it drops. Check if the ARREQ has already been received based on the seq#, if so, it drops. Otherwise it continues. Check if the node is the destination by decrypting C_S with its private key. If the node can decrypt it successfully, the node is receiver; otherwise, the node is not the receiver.

If the node is not the intended receiver and $(HopCount--)\geq 0$, then it forwards the ARREQ to all its neighbors via broadcasting. After execution, intermediate nodes (ID_I) change its route table (Fig. 5).

sequence	Dest_ID	Ancestor	Successor	TTL
Seq#	ID_D	N/A	?	T_S

Fig. 4. Source node changes its route table.

sequence	Dest_ID	Ancestor	Successor	TTL
Seq#	N/A	I_{I-1}_addr		T_S

Fig. 5. Intermediate node changes its route table.

If the node is the intended receive D, it can correctly recover M_S and parse it as $H_1(ID_S), ID_D, g^a$ and $H(g^a \| K_{SD} \| 0)$ then, it can verify: $H(g^a \| K_{SD} \| 0) = H(g^a \| \hat{e}(H(ID_S), SK_{ID_D} \| 0)$ only *S* and *D* can compute: $K_{SD} = \hat{e}(H(ID_D), SK_{ID_S}) = \hat{e}(H(ID_S), SK_{ID_D}$ in this way, it can prove *D* is the destination node. The destination D maintains the route table (Fig. 6).

sequence	Dest_ID	Ancestor	Successor	TTL
Seq#	$H_1(ID_S)$	In_addr	N/A	T_S

Fig. 6. Destination node changes its route table.

2) *Inter-group anonymous route request*

When a node want to communicate with the node in the different groups, the source node *S* will also generate the relevant parameters and add the network identity *NF* into the ARREQ. The following form as follows: $ARREQ =< RREQ, seq\#, NF, HopCount, Cs >$

Intermediate nodes will discard the ARREQ packet as long as find *NF*, which means the destination node is in the other group. After that, change their route tables like intra communication.

When the cluster head node receives the ARREQ, it will know the source node want to communicate with the other node of the other subgroups. Because the system uses a multi-band classification, cluster head node use another frequency to forwards the ARREQ to its neighbor cluster head nodes. After that, it changes its route table.

When the neighbor cluster head node receives the RREQ, it checks *NF*, if not, forward, otherwise, discard the *NF* of ARREQ. $ARREQ =< RREQ, seq\#, HopCount, Cs >$.

In this way, it can find the destination node by intra routing discovery. Intermediate nodes complete their route tables. Finally, it can achieve the inter-group routing node discovery.

E. Route Reverse Phase

Inter-group route and intra-group route reverse both executed as follows. The nodes receive the packet from the neighbor honest nodes and can find its forward nodes by seq#.

Step 1. D randomly select $b \in [1, p-1]$, computers g^b and $H(g^b \| K_{SD} \| 1)$, where $M_D = [H_1(ID_S), ID_D, g^b, H(g^b \| K_{SD} \| 1)]$ *D* use $PK_S = H_1(ID_N)$ to encrypt M_D, as $C_D = E_{PK_S}(M_D)$. According to seq# look up its upstream, and appends authentication tag encrypt message C_D together with seq# using secret key K_{DI_n} (key exchange based on the ring signature) shared with the upstream. *D* sends ARREP to I_n, which is formatted as follows: $ARREP =< RREP, seq\#, C_D, MAC_{K_{DI_n}}(rt\_seqno, C_D) >$ computes shared session key $SK_{SD} = (g^a)^b$.

Step 2. *D* uses its shared secret key K_{DI_n} to verify $MAC_{K_{DI_n}}(rt\_seqno, C_D)$, If seq# is found in its local table, it continues, otherwise, it drops. If authentication fails, it drops the ARREP, otherwise, it continues.

Step 3. When the source node *S* receives ARREP, it can authenticate, S also can compute session key $SK_{SD} = g^{ab}$, the data packet transmission start the source to destination node is established.

F. Data Forwarding Phase

Step 1. *S* uses session key to encrypt *M* as $C = E_{SK_{SD}}(M)$, it can find its downstream node I_1 from its local route table, and uses their session key to encrypt C, seq# as $MAC_{K_{SI_1}}(C)$, $R_{I_1} = eK_{SI_1}(seq\#)$. At last, it sends $(R_{I_1}, C, MAC_{K_{SI_1}}(C))$ to node I_1.

Step 2. When $\begin{aligned} k_{ab} &= x(yP) \cdots Alice \\ &= y(xP) \cdots Bob \end{aligned}$ receiving the packet, node use its public key to authentication. If success, decrypt R_{I_1}, and then lookup downstream by seq# from local route table. At last, it changes the information of packet by their session key.

Step 3. When the destination receives $(R_D, C, MAC_{KI_ND}(C))$, it uses its shared secret key to verify $MAC_{KI_ND}(C)$. If it success, decrypting R_D. It finds the corresponding session key $SK_{SD} = g^{ab}$, which is taken to recover *M*. Similarly *D* also can send confidential data to *S* in the same way.

IV. Protocol Analysis Based on UC Model

The *SI* unit for magnetic field strength *H* is A/m. However, if you wish to use units of *T*, either refer to magnetic flux density *B* or magnetic field strength symbolized as $\mu_0 H$. Use the center dot to separate compound units, e.g., "A·m^2."

The proposed protocol π can be divided into three sub-agreement π_1 , π_2 , π_3. π_1 is the authentication key exchange phase; π_2 is anonymous routing stage; π_3 is the data transfer phase. According to common characteristics, the security of π is equivalent π_1 , π_2 , π_3 combination of security.

Lemma 1: The protocol π_1 meet the UC security is based on the ring signature, which can against adaptively chosen message attack.

Proof: [5] reference.

Lemma 2: The protocol π_3 meet the UC security.

Proof: On the basis of the protocol π_1, neighborhood nodes get their MAC session key. Source node and destination node establish session key by select random number *a*, *b* and sharing parameters *g*, which meet the bilinear difficulties in problem solving. So the data transfer is the UC security [8].

Definition 1: the ideal route request process F_{ARREQ} [9], $\mathrm{F}_{\mathrm{ARREQ}}$ execute instruction under the background of the security parameter *k*, participates $P_1,...,P_n$, and adversary S. $\mathrm{F}_{\mathrm{ARREQ}}$ internal structure: the path of P_i is B_i . The record entry form $(sid, P_s, P_{j1}, P_{j2},..., P_i)$, where P_s is the source node, P_{j1}, P_{j2} are intermediate nodes. Bad represent those compromised nodes. Ead means those eavesdropping nodes. Text symbol "→" indicates that the message flow from left to right; "$\wedge$" is the logical and; "$\vee$" is logical or; $NB_A(P_i)$ is the P_i 's neighbor set; $NB_H(P_i)$ is the P_i 's honest set of neighbors. The communication channel is C_I .

Lemma 3: protocol π_2 is UC anonymous in the F_{SC} [6] hybrid model.

Proof: Set the attack *A* running in the routing request of the F_{SC} model. We construct an attack *S* of $F_{ARREQ.}$ *S* runs a simulated *A*. The purpose of simulation is to prove that the environment *Z* can't distinguish it interacts with *S* of F_{ARREQ} or interacts with *A* in the F_{SC} hybrid model. *S* simulates F_{SC}.

S maintains two tables: parameter table list, which is used to send *ARREQ* to *A*; message list $List_{ARREQ}$ which is used to send message to F_{ARREQ}. When *S* receives $NB_A(P_i)$ ARREQ for the first time, it randomly generated seq#, one-time key pair (E_{pk}, E_{sk}), and use the destination public key to encryption M_s , saving $(sid, seq\#, E_{pk_D}(M_s))$ to the list. *S* eavesdrops P_x. when it received ($NB_A(P_i)$, *ARREQ*) from C_I , looking up List $(sid, seq\#, E_{pk_D}(M_s))$ by sid. It will pack the message into *m* of ARREQ. *S*:(N_i, N_x, *m*) →*A*, *S* simulate N_X (corresponding to P_x of F_{ARREQ}), receive *ARREQ* from N_i forwarded. That means S simulates N_X to receive and forward the packet of *ARREQ*.

If $P_x \in Bad$ received $(NB_A(P_i), ARREQ)$, then $S:(N_i, N_x, m) \to A$, where the message *m* is the same with the contents of eavesdropped. Then, it uses private key of P_X to decrypt $E_{pk_D}(M_s)$ of message m and send the result to *A*. Save (N_i, N_x, m) to $list_{ARREQ.}$ Assume the nodes in the path from N_x to N_y were captured, N_y forward the rout request m to N_j ($N_j \notin Bad$) from *A* to *S*. *S* get (sid, N_i, N_x, m) $S:(P_i, P'_{y1}, P'_{y2},..., P'_y\ P_j, ok) \to F_{ARREQ}$ by looking up the recorded $List_{ARREQ}$, if *ARREQ* and $list_{ARREQ}$ have no corresponding date, discarded the packet.

We define the sequence of mixing machine environment proves that process simulation. The probability that *Z* is able to distinguish between an interaction with *A* and real parties running protocol π_2 and an interaction with *S* and dummy parties accessing F_{ARREQ} is at most a negligible probability.

Hybrid H_{y0}. It generates key pairs for all honest nodes and execute the protocol π_2 for each node. According the protocol π_2 it can simulate the host nodes interact with *A* and *Z*. The interaction output of *Z* and H_{y0} is the same with the output of *Z* in the real-life world. Hybrid H_{y1} and H_{y0} is basically the same, the differences is that: if $P_x \in Ead$, N_x forwards the packet *ARREQ* to N_i, H_{y1} gets the route path by decryption *ARREQ*. If the route was not recorded before, then this path is marked as false path Hybrid H_{y2} and H_{y1} is basically the same, the differences is that: upon $P_x \in NB_A(P_i)$ $\wedge$ $(P_x \in Bad$ $\vee$ $P_x \in Ead)$ $\wedge$ $P_i \notin Bad$, when N_i forwards the routing request to N_x, H_{y2} uses private key to decrypt the message of $E_{pk_D}(M_s)$, and sends the result to attack *A*. Hybrid H_{y3} and H_{y2} is basically the same, the differences is that: if $P_i \notin Bad$ $\wedge$ $P_x \in NB_A(P_i)$ $\wedge$($P_x \in Bad$ $\vee$ $P_x \in Ead)$ when N_i forward the routing request to N_x, H_{y3} sends the route request to attack *A*. Hybrid H_{y4} and H_{y3} is basically the same, the differences is that: H_{y4} do not execute protocol π_2 for honest nodes, but get the path through the ideal functionality, that means it don't execute any encryption or decryption. The output of *Z* and H_{y4} interaction is the same with the output of *Z* in F_{ARREQ}. If H_{yi} and H_{yi+1} (*i* = 0,1,2 ,3) is interacting , the output of *Z* is the same, the protocol π_2 is anonymous. The signature of against Chosen-message attacks is to guarantee the the success probability of tampering honest node path can be ignored.

The output of *Z* and H_{y1}, H_{y2} interaction can't computationally indistinguishable. If it can distinguish, consider another sequence of mixing machine $H_{yl,}$ before the condition of H_{y2} is first time satisfied, H_{yl} did what H_{yl} should do. Then execute according to H_{y2}. In the polynomial sector, H_{yl} and H_{yl} will be undistinguishable. Otherwise, the adversary can declassification the IND-CPA security of public key encryption. Assume challenge nodes *N*, an optional node N_j let (N_j , SK_j) = (*N*, SK_N); challenger generates key-pairs for the other nodes. When execute the mix machine for *i*+1 times, let challenger public key is E_{PK}, if it can distinguish that means to crack the security of IND-CPA.

The output of *Z* and H_{y2}, H_{y3} interaction is computationally indistinguishable. UC secure channel proves the indistinguishable of the ciphertext, which is generated by different plaintext.

The output of *Z* and H_{y3}, H_{y4} interaction is computationally indistinguishable. The reason is the ideal functionality executes the function of honest nodes in the Hy_3.

Theorem 1 anonymous routing protocols π can achieve UC security Proof. It is proved by Lemma 1-Lemma 3.

V. Comparison with Related Work

TABLE I: Comparison of the Proposed Scheme and Related Work

	ANODR [10]	MASK [11]	ASRPAKE[3]	OURS
Sender anonymity	×	×	×	√
anonymous authentication	×	×	√	√
Anonymous routing	×	×	√	√
hierarchical routing	×	×	×	√
UC security	×	×	×	√

Our protocols can provide end-to-end anonymity of a route, sender anonymity, the security of authenticated session key shared. In addition, the protocols suitable for hierarchical ad hoc networks and achieve UC security (Table I).

VI. Conclusion

In this paper, we proposed a new anonymous routing based on the anonymous ring signature authentication routing protocol, the protocol is more suitable for large-scale ad hoc network and achieve the sender anonymity, routing anonymity. At last, we use the UC model to prove the security of this protocol. As the future research, we plan to improve route efficiency.

References

[1] L. chaumd, "Untraceable electronic mail, return dresses, and digital pseudonyms," *Communications of the ACM*, vol. 24, no. 2, pp. 84–88, 1981.

[2] S. S. Varghese and J. I. J. Raja, "A survey on anonymous routing protocols in MANE," in *Proc. the 12th International Conference on Networking*, 2010.

[3] ASRPAKE, "An anonymous secure routing protocol with authenticated key exchange for wireless Ad hoc networks," in *Proc. the ICC 2007*.

[4] A. Bender, J. Katz, and R. Morselli, "Ring signatures: Stronger definitions and constructions without random oracles," in *TCC 2006*, pp. 60–79.

[5] K. Yoneyama and K. Ohta, "Ring signatures," *IPSJ Digital Courier*, vol. 3, pp. 571–584, 2007.

[6] R. Canetti. Universally composable security. [Online]. Available: http://eprint.iacr.org/2000/067.

[7] W. D. Yang, "Weight-based clustering algorithm for mobile ad hoc network," in *Proc. Cross Strait Quad-Regional Radio Science and Wireless Technology Conference*, 2011, pp. 787–791.

[8] J. Camenisch and A. Lysyanskaya, *A Formal Treatment of Onion Routing*, Berlin: Springer, 2005.

[9] Z. Yang, "Formal treatment of an anonymous on demand routing protocol in MANET," *Journal of Computer Research and Development*, 2008.

[10] K. Kong and X. Hong, "ANODR: an anonymous on demand routing with untraceable routes for mobile ad hoc networks," in *Proc. MobiHoc'03*, 2003, pp. 291-302.

[11] Y. Zhang, W. Liu, and W. Lou, "Anonymous communications in mobile ad hoc networks," in *Proc. the 24th Annual Joint Conference*, IEEE, 2005, pp. 1940–1951.

Yuan Xiaopeng was born in 1981. He recevied a bachelor degree. He is now working as an engineer in High-Level School Admissions Office of Gansu Province, Lanzhou, China. His research areas are in software design and information security.

Feng Tao was born in 1970, he recevied a PhD degree and has been a supervisor of School of Computer and Communication Lanzhou University of Technology, Lanzhou, China. His research areas are in networking and information security.