南　華　大　學

資訊管理學系

碩士論文

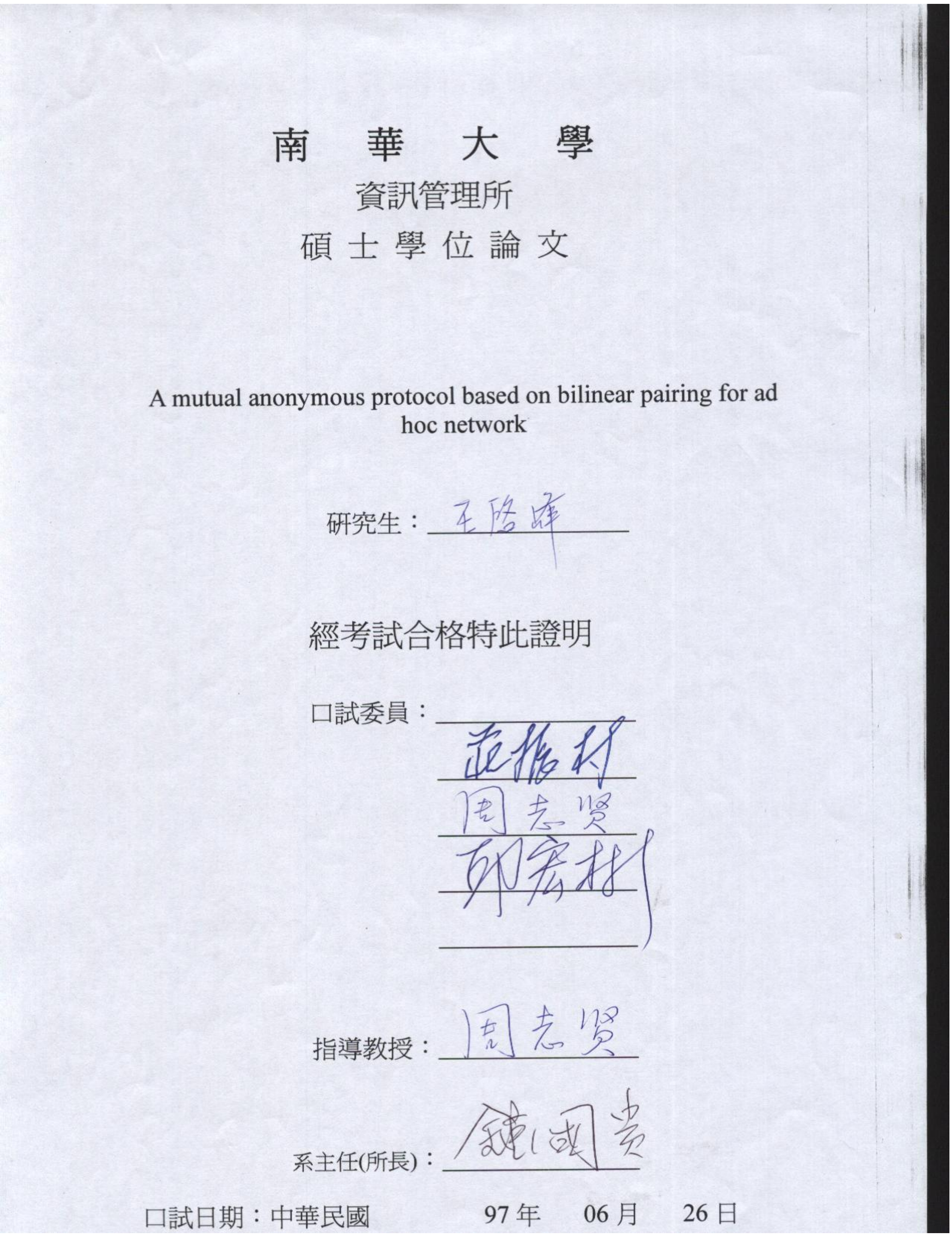
安全、有效率且自由路徑的雙向匿名網路架構運用於隨意型無線網路

A secure efficient and path-free mutual anonymous routing protocol for ad hoc network

研 究 生：王啟峰

指導教授：周志賢

中華民國　97 年　6 月　30　日



南華大學

資訊管理所

碩士學位論文

A mutual anonymous protocol based on bilinear pairing for ad hoc network

研究生：王路峰

經考試合格特此證明

口試委員：莊振村

周志賢

邱宏彬

指導教授：周志賢

系主任(所長)：鍾國貴

口試日期：中華民國 97年 06月 26日

誌謝

本篇論文之所以能得以完成主要仰賴指導教授周志賢老師及陳雅玲學姐不辭辛勞地細心指導，在每次的討論中給予我寫作上正確的觀念及方向，在論文修改期間更是細心地給予指導以及其過去寶貴的寫作經驗。還有在百忙之中抽空為我和宗亨擔任口試委員的莊振村教授及邱宏彬教授所提出的修改意見，有了這些師長的細心教導才能使學生我可以有更縝密的思考來完成此篇論文。

在南華大學資訊管理研究所的這兩年，首先必須要感謝我的家人，在我最低潮的時候能給予我莫大的鼓勵與肯定，使我感受到家人的溫暖，另外還要感謝台北大學的簡明哲老師，稻江管理學院的盧宓承、郭文益老師，感謝他們在這短短的兩年時光中給了我許多的磨練與成長的機會，而在學校中，首先感謝我的直屬學長 晉城及仲儒帶領我奠定紮實的密碼學觀念，再來是我的搭檔宗亨，感謝他幫了我許多的忙，假如沒有他的幫忙，我想，這篇論文也不會順利的完成吧，最後，更要謝謝一路陪伴著我走來的你們，小捲、鈺琪、洪欣、猴子等以及其他所有協進會辦公室的成員，因為你們，讓我在這兩年的研究所生活中，更加的有意義及豐富的色彩。

最後，在這兩年的生活中雖然辛苦，但也因為有了家人、師長、同學及朋友間的相互鼓勵及扶持，我才能完成這兩年來的學業以及體驗到不同的學生生活，謝謝你們。

安全、有效率且自由路徑的雙向匿名網路架構運用於隨意型無線網路

學生：王啟峰　　　　　　　　　　　　　　　　指導教授：周志賢

南　華　大　學　資訊管理學系碩士班

摘　　要

近年來，許多應用於隨意型無線網路的匿名安全網路架構漸漸受到人們的重視。然而，我們發現許多的網路架構研究卻只使用了 PKI 或是 Onion 的加密技術，以期待能達到更完善的匿名性與安全性，但在我們研究的過程中卻發現，許多的研究並不能達到完全的匿名效果或是更加安全的網路架構，因此我們提出了一個安全、有效率並且採自由路徑方式設計了一個雙向匿名的網路安全架構以應用於無線網路。

關鍵字：匿名的路由網路架構，bilinear pairing 運算技術，隨意型無線網路。

A Secure Efficient and path-free mutual anonymous routing protocol for Ad Hoc network

Student: Chi-Feng Wang Advisor: Dr. Chou Jue-Sam

Department of Information Management
The M.I.M Program
Nan-Hua University

ABSTRACT

There are many researchers work on anonymous secure routing in mobile ad-hoc networks. However, most of them using onion encryption which makes the system very complicated in concept. Moreover, they usually need the requirement that the source and destination nodes pre-share a session key. This makes the system very inefficient in key management. Therefore, in this paper, we propose a mutual anonymous ad hoc routing protocol based on bilinear pairing which not only can provide anonymity property for both of the sender and the receiver, but is very simple in concept. After our analysis, we can conclude that our scheme can resist against various attacks.

Keyword: Anonymous routing protocol, bilinear pairing, ID-based routing protocol, Onion.

目 錄

表　目　錄

圖 目 錄

1. Introduction

Recently, wireless ad-hoc networks have got a great deal of attentions for they needn't (or require less) any fixed infrastructure in an open environment. In such a network, when a node wants to communicate with the other party, it must inject it's identify into the packets to identify himself; thereby, an attacker can obtain the identity of a source or destination node. This makes the transactions traceable between the two communicating parties. Hence, for the purpose of making the transactions untraceable, anonymity becomes an important issue in mobile ad-hoc networks. It expects that the identities of the two communicating parties can be hidden from all possible adversaries.

There are many researchers work in this area [1-15, 26]. In 2004, Zhu et al. [1] proposed a scheme to provide anonymous property for the source/destination nodes, and the security of discovered routes against various attacks. However, their scheme requires each communicating pair of nodes must pre-share a session key. This induces the key management problem. In addition, the secrecy shared between the source and destination, KT is unchanged. This makes their scheme lack of backward and forward secrecy. Also in 2004, Boukerche et al. [2] proposed a scheme which can allow trustworthy intermediate nodes to participate in the path construction without jeopardizing the anonymity of the communicating nodes. They define the trust level of a node to be based on its past behavior. However, they do not define the details about a node's trust level. This would cause the serious problem of inconsistent trust level view in the system. For example, a node is viewed as level A but may be viewed as level B for another node. In addition, when the path discovery message delivered from source to destination, all intermediary nodes would produce the temporary session keys. This makes the system suffer from key management problem, too. Besides, the destination node knows all intermediate nodes' identities which make the communication. In 2006, Yanchao Zhang et al. [7] proposed the MASK protocol to achieve the anonymous property needed in an ad hoc network. However, in 2006, Li et al. [13] points out that their scheme lacks the anonymous property since the identity of the destination node is encompassed in the RREP packet. Also, in 2006, Seys et al.[9] proposed the ARM protocol, but their protocol not only needs pre-share session keys but also needs pre-share pseudonym name tables.

In 2007, Lu et al. [3] proposed a scheme to provide route anonymity from the source to the destination and integrate the authenticated key exchange into the routing algorithm. However, their scheme also incurs key management problem. Moreover, each node's identity (Ni) can be record by ancestor and successor node when it broadcasts a packet. This makes the message delivery traceable. Also, in 2007, Han, J et al. [26] consider that anonymity property comprises three issues: (1) initiator anonymity, (2) responder anonymity, and (3) mutual anonymity (providing both the initiator and responder anonymity). They, [26] mention that anonymous protocols in most wireless scenarios were extremely difficult in practice to guarantee the reliability of message delivery via path-based approaches, since a path in a wireless environment is highly dynamic. In addition, the paths should be periodically updated for security concerns. Hence, they propose a scalable secret-sharing-based mutual anonymity protocol, termed PUZZLE, which enables both the anonymity of query issuance and file delivery for MOPNETs. Their scheme critical that most of the existing protocols are path-based which is bad explained above. However, their scheme still employs the path-based method in the reply phase.

In this paper, we propose a mutual anonymous and path-free protocol which is secure and efficient for implementation in an ad hoc network. Our design based on bilinear pairing using the broadcasting feature of mobile network to achieve mutual anonymity of the two communicating parties. In our scheme, each node has a private key issued by a trusted authority in the key predistribution phase and uses two pseudonym tables when communicating with other members. The analysis shows that our scheme can achieve all of the security requirements, e.g., route anonymity, resistance of man in the middle attack, the backward and forward secrecy, and so on.

The structure of this article is as follows. The introduction is presented in Section 1 and the preliminaries are shown in Section 2. In section 3, we show the proposed scheme. Its security analysis is done in Section 4. In Section 5, we compare its performance including computational cost and bandwidth consumption, and several security attributes with other protocols. In Section 6, we describe how to modify our scheme to achieve the security requirement of preventing KCI attack. Finally, a conclusion is given in Section 7.

2. Preliminaries

2.1. ID-based bilinear pairing

In 1984, Shamir [22] proposed an ID-based encryption and signature scheme. In the scheme, each user can use his identity as his public key to make the key distribution easier than the conventional ones. In 1993 and 1994, some related works on elliptic curve were proposed which are the foundations of bilinear pairing [17, 18]. After that, in 2001, an ID-based bilinear pairings defined on elliptic curves were proved and applied to cryptography [19]. Since then, many protocols have been designed based on the Weil pairing [19, 20, 21, 23, 24, 25]. Now, we briefly introduce bilinear pairings as follows.

Let P be a generator of G1 that is a cyclic group whose order is a prime q and G2 be a cyclic multiplicative group of the same order. We assume that solving the discrete logarithm problem (DLP) in both G1 and G2 is difficult in polynomial time. Let e: G1×G1→G2 be a bilinear pairing satisfying the following conditions.

(1) Bilinear : $e(aP, bQ) = e(P, Q)^{ab}$, for any $a, b \in Z_q$ and $P, Q, R \in G_1$.
(2) Computability : There is an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in G_1$.
(3) Non - degenerate : there exists $P \in G_1$ and $Q \in G_1$ such that $e(P, Q) \neq 1$.

Then, we make a description of BDHP (Bilinear Deffie-Hellman problem) as follows: Using (P,aP,bP,cP) to compute $e(P,Q)^{abc}$, for any $a,b,c \in Z_q$ and $P,Q \in G_1$.

Generally speaking, it is hard to solve the BDHP in polynomial time. The security level of bilinear pairings is equal to the discrete logarithm problem but with more efficiency.

2.2. Diffie-Hellman Problems

With the group G1 described in section 2.1, the following hard cryptographic problems have been defined which are applied to our proposed scheme.

(1). Discrete Logarithm (DL) Problem: Given P, Q Є G1, find an integer n such that P=nQ whenever such integer exists.

(2). Computational Diffie-Hellman (CDH) Problem: Given a triple (P, aP, bP) Є G1 for a, b Є Zq*, find the element abP.

(3). Decision Diffie-Hellman (DDH) problem: Given a quadruple (P, aP, bP, cP) Є G1 for a, b, c Є Zq*, decide whether c=ab mod q or not.

(4). Gap Diffie-Hellman (GDH) Problem: A class of problems where the CDH problem is hard but DDH problem is easy.

(5). Bilinear Diffie-Hellman (BDH) Problem: Given a quadruple (P, aP, bP, cP) Є G1 for some a, b, c Є Zq*, compute ẽ(P,P)abc.

Groups in which the CDH problem is hard but DDH problem is easy are called GAP Diffie-Hellman (GDH) groups.

2.3. Characteristics of Wireless System

The major challenges in designing protocol for ad hoc networks are the lack of a fixed infrastructure and the highly dynamic nature since nodes can join and leave the network at any time. Ad hoc networks generally have the following characteristics:

(1). Dynamic network topology:

The topology of the network may change frequently, since the nodes are mobile.

(2). Limited bandwidth:

The bandwidth of wireless systems is lower than traditional network systems. This may limit the number and size of messages sent during protocol execution.

(3). Energy constrained nodes:

Nodes in ad hoc networks usually use batteries as their power source.

(4). Limited security:

The wireless network usually has limited security support.

3. Proposed scheme

For traditional schemes in securing ad hoc routing are too complex in concept. And the key distribution is a main problem in the system. In this paper, we propose a novel scheme based on bilinear pairing to resolve these problems. We first make the assumptions of our protocol in Section 3.1. Then, show our protocol in Section 3.2.

3.1. Assumptions

In our protocol, we make the following assumptions.

(1) Links between wireless nodes are symmetric. That is, if node A is in the transmission range of node B, then node B is in the transmission range of node A as well.

(2) Each wireless node has a unique identifier which can make the node uniquely recognizable in the network.

(3) Adversaries have unbounded eavesdropping capability but only with bounded computing and node intrusion capabilities.

3.2. Our proposed protocol

After introducing our system's assumptions, we now present our protocol. It consists of four phases: (1) Setup phase, (2) Request phase, (3) Reply phase, and (4) Data transfer phase. In the following, we will first list the definitions of used notations in table 1, and then describe the four phases.

Table 1. definitions of used notations in our protocol

TA	the trusted authority
PT	a pseudonym table
s	the private key of TA
SP	the public key of trusted party (=sP) and P is a generator
Q_i	the public key of node i
S_i	the private key (=sQ_i) of node i, computed by TA
Sq	an unique number used in the *Request/Reply* phase
r'	a random number chosen by the source node
r''	a random number chosen by the source node
r_k	a random number chosen by system administrator in time interval k
T	the timestamp of a node
G_1	an additive cyclic group of prime order q
G_2	a multiplicative cyclic group of prime order q
H (.)	a collision resistant hash function, mapping a point in G_2 to a bit string
e	a bilinear pairing map, e: $G_1 \times G_1 \rightarrow G_2$
$En_{SK}(M)$	a message M encrypted with key SK
N_{ij}	a point in G_2 computed as e(S_i , Q_j) = e(Q_i , S_j), for j = 1 to N and $i \neq j$

(1) Setup phase

In our protocol, there exists a trusted third party. TA. He produces a private key s and a generator P. Then, he calculates his public key as SP. When node i wants to join the network, TA will distribute a public/private key pair, Q_i/S_i for him and a random number table which contain many random number by used in each time to him. Node i can then use his private key and random number table to calculate the pseudonym shared with other nodes and build up two pseudonym tables. He computes the pseudonym shared with node j as $H(r_kN_{ij})$= $H(r_ke(S_i, Q_j))$, (for j=1 to N and $i \neq j$ under the assumption that there are *N* nodes in the network). Each two pseudonym tables has two fields, Node ID and Pseudonym. One is for sending and the other for receiving. We name the one for sending as sending Pseudonym table and the other for receiving as receiving Pseudonym table as shown in figure l.

Pseudonym tables generated by Node i (Assume that $i = A$)

sending pseudonym table	
Sorted ID	Pseudonym
B	$H(r_1N_{iB})$
	$H(r_2N_{iB})$
	⋮
	$H(r_lN_{iB})$
⋮	
j	$H(r_1N_{ij})$
	$H(r_2N_{ij})$
	⋮
	$H(r_lN_{ij})$

receiving pseudonym table	
Sorted Pseudonym	ID
$H(r_1N_{iC})$	C
$H(r_1N_{iB})$	B
⋮	
$H(r_1N_{iD})$	D
⋮	
$H(r_lN_{iD})$	D
⋮	⋮
$H(r_lN_{iB})$	B

* $H(r_kN_{ij})$ is the pseudonym shared betwreen node i and j in time internal k for $k = 1$ to l

Figure 1. the two sorted Pseudonym Tables of node *i*

In addition, each node also calculate $e(P, Sj)$ employ the generator P and private key of themselves and builds up the private table in advance. The private table has two fields, Node ID and the value calculated by bilinear pair which contains *SP* and public key of each node. It is shown in figure 2.

The private table generated by Node i

Node ID	Value
B	$H(e(SP , Q_B))$
C	$H(e(SP , Q_C))$
⋮	⋮
i	$H(e(SP , Q_i))$
⋮	⋮

Figure 2. the Private Table of node *i*

(2) Request phase

Whenever node *i* wants to send confidential data to node *j* in the time internal *k*, he would perform the following steps.

Step1. Chooses two random numbers *r*’, *r*” and searches the sending *PT* using j as index to find $H(r_kN_{ij})$.

Step2. Generates a unique Sq_1 for this route, computes $r'Q_i$, $H(N_{ij})\oplus r''$, $H(r_kN_{ij})\oplus H(e(SP,Q_j))$ and pre-computes session key shared with node *j* as SK=H(e(r’Si, r”Qj)).

Step3. Makes a Request packet, $[Sq_1, H(r_kN_{ij})\oplus H(e(SP,Q_j)), H(N_{ij})\oplus r''$, $r'Q_i$, $T]$, and broadcasts this packet to all other nodes within its wireless transmission range.

Step4. When all nodes receiving the packet, each node, say node *B*, would check to see if the packet has already been received using the unique Sq_1. If it has been, he rejects. Otherwise, node *B* stores the value of $r'Q_i$ which is to be used as an indicator for deciding whether to drop the packet in the reply phase and computes $H(r_kN_{ij})\oplus H(e(SP,Q_j))\oplus H(e(P,S_j))$, getting $H(r_kN_{ij})'$. Then he uses $H(r_kN_{ij})'$ as index to search his receiving pseudonym table. If he can find such an item, then he is the destination node. He then knows node *i* wants to communicate with him by extracting the corresponding Node ID in his receiving pseudonym table. (Here, we assume that the field value in the corresponding Node ID of his receiving pseudonym table is *i*). Then, node *B* broadcasts the packet to all other nodes within its wireless transmission range.

Step5. Go to Step4.

(3) Reply phase

In this phase, whenever destination node j responds to source node i, he will perform the following steps.

Step1. Generates a unique Sq2 for reply phase, computes $H(N_{ij})\oplus(H(N_{ij})\oplus r'')$, getting *r*" and computes the session key shared with node *i* as $H(e(r''S_j,r'Q_i))$.

Step2. Broadcasts the packet = [Sq_2, $r'Q_i$, $r''Q_j$, *T*+1], to all other nodes within its wireless transmission range.

Step3. When all nodes receiving the packet, each node, say node *B*, would check to see if the packet has already been received using the unique Sq_2. If it has been, he rejects. Otherwise, he checks $r'Q_i$ to see if this is the value stored in the request phase. If so, either he is the source node or the intermediate node. If he is the source node, he can use the session key $H(e(r'S_i,r''Q_j))$ to communicate with node *j*. If he is the intermediate node, he broadcasts the message to all other nodes within its wireless transmission range. Else, he is not both of the two cases, he rejects.

(4) Data transfer phase

After completing the request and reply phases, the two communicating parties has established a common session. For node *i*, he computes SK $(=H(e(r'S_i,r''Q_j)))$ and for node *j*, he computes $SK=H(e(r''S_j,r'Q_i))$. Then they use *SK* to communicate with each other. Source node *i* sends the confidential *M* to the destination node *j* by broadcasting the data packet, $[[M]_{SK}$, $r'Q_i$, T'], to all other nodes within its wireless transmission range. After receiving the data packet, only node *j* can decrypt the message $[M]_{SK}$. Since node *j* is the only node who can compute the session key $SK(= H(e(r''S_j, r'Q_i)))$.

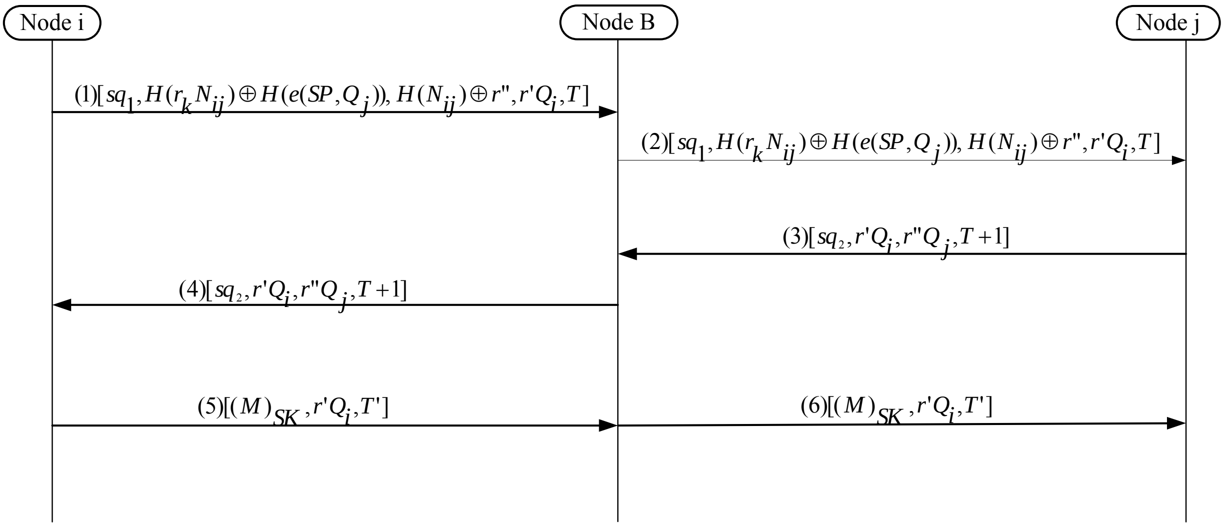

Node i
Node B
Node j
$(1)[sq_1, H(r_k N_{ij}) \oplus H(e(SP, Q_j)), H(N_{ij}) \oplus r'', r'Q_i, T]$
$(2)[sq_1, H(r_k N_{ij}) \oplus H(e(SP, Q_j)), H(N_{ij}) \oplus r'', r'Q_i, T]$
$(3)[sq_2, r'Q_i, r''Q_j, T+1]$
$(4)[sq_2, r'Q_i, r''Q_j, T+1]$
$(5)[(M)_{SK}, r'Q_i, T']$
$(6)[(M)_{SK}, r'Q_i, T']$

Figure 3. Our Protocol

4. Security analysis

In this section, we analyze our protocol on the route anonymity property and the route security properties including: (1) against man in the middle attack, (2) backward and forward secrecy, and (3) untracebility. We describe them as follows.

4.1 Route anonymity

Only the destination node *j* can know where the packet comes from computing $H(e(P,S_j))\oplus H(r_kN_{ij})\oplus H(e(SP,Q_j))$, obtaining $H(r_kN_{ij})$. Although an attacker can acquire the value of $H(r_kN_{ij})$, he can't find who is the destination node and who is the source node by searching his receiving *PT*. Since there doesn't exist a table entry consisting such a value. Therefore, our protocol possesses the anonymity property of the route for both of the source and the destination nodes.

4.2 Route security

(1) Against man in the middle attack (MIMA)

Man in the middle attack means that there is an adversary *E* sniffing transmitted information on a communication line between the sender and receiver. He wants to impersonate the source node to destination node without being detected and vice versa. In our protocol, after the setup phase, each node has his two sorted *PT*s. Although *E* can obtain $H(r_kN_{ij})$, *E* can't know who the two communicating parties are. Not to mention, he can compute their session key. Moreover, due to the session key computation of a communicating pair is either $H(e(r'S_i, r''Q_j)$ or $H(e(r''S_j, r'Q_i)$, an attacker, without knowing S_i and S_j, can not impersonate *i* to communicate with *j* and vice versa. Therefore, the MIMA fails in our protocol.

(2) **The backward and forward secrecy**

Backward secrecy means that when the current session key of two nodes is compromised, their session keys used before are still secure. Similarly, the forward secrecy is defined in the other direction. In our protocol, the session key between a pair of nodes is computed by $H(e(r'S_i, r''Q_j)$ or $H(e(r''S_j, r'Q_i)$. Other than S_i, S_j, Q_i and Q_j, the session key computation also depends on the two random numbers, *r*' and *r*" which can assure the independence of each session key. Hence, our protocol possesses the forward and backward secrecy.

(3) **Untraceable**

On our scheme, we use $H(r_k, N_{ij})$ as an index to search the receiving *PT*. $H(r_k N_{ij}) \oplus H(e(SP, Q_j))$ is a random number. An attacker can not know which pair of nodes to communicate with each other. Although he has $\frac{1}{|R|}$ probability (there have *R* nodes between source and destination node in the ad hoc network) to guess out which node the source wants to communicate with and computes $(H(r_k N_{ij}) \oplus H(e(SP, Q_j))) \oplus H(e(SP, Q_j))$, obtaining $H(r_k N_{ij})$, he can not knows who is the source node.

5. Performance and Security comparisons

In this section, we compare the performance of our protocol in the aspect of: (1) computational cost and bandwidth consumption, and (2) security attributes, with some other studies in Section 5.1 and Section 5.2, respectively. The definitions of used notations are listed in table 2. The computational cost comparisons and bandwidth consumption comparisons are shown in table 3 and table 4, corresponding. Then, the comparisons of security attributes and necessity of pre-sharing secret keys are given in table 5. For clarity, in our comparisons, we assume that there are n intermediate nodes between the source node and the destination node.

Table 2. Notations used in the comparisons

S	searching the *PT*
$\oplus$	an exclusive or operation
Bp	a bilinear pairing operation
Exp	a modular exponential operation for computing temporary public key
Asym	an operation using of asymmetric approach to en/decrypt messages
Sym	an operation using of symmetric approach to en/decrypt messages
H	an one-way hash function operation
n	the number of all intermediate nodes
cmp	the number of comparisons
crte	the number of creating table entries
mul	a point multiplication in G_1

5.1 Comparisons of computational cost and bandwidth consumption

(1) Computational Cost Comparisons

Table 3. Computational cost comparisons

Protocols	Our protocol	ASR [1]	SDAR [2]	SARPAKE [3]	MASK [7]	ARM [9]
Request	⊕: n+4 S: n+1 mul: 1	Asym:(n+2), Exp:2, H:2,	Asym:5n+4, Sym: 2, H: 2n+2, crte: n+2	Asym: (n+2), Exp: 1, H: 2, cmp: 1	crte: n+2	ASym:2(n+1), Sym: (n+4), Exp: 1
Reply	mul : 1 S: n+1	Asym:(n+1), Sym:3n+5, cmp:(n+2),	Sym: 5n-4, H: 8n-9,	Asym: (2n+3), Exp: 3, H: 2, cmp: 1	Sym: n+2 crte: n+2	Asym: (n+1), Sym: (n+6), cmp: (n+1)
Total	⊕: n+4 S: 2n+2 mul: 2	Asym:2n+3, Sym:3n+5, cmp:(n+2), Exp:2, H:2,	Asym:5n+4, Sym: 5n-2, H: 10n-7, crte: n+2	Asym: 3n+5, Exp: 4, H: 4, cmp: 2	Sym: n+2 crte: 2n+4	ASym:3n+3, Sym: 2n+10, Exp: 1 cmp: (n+1)

In our protocol, each bilinear pair operation can be pre-computed before the protocol run. Hence, we omit these bilinear pair operations in computational cost comparisons. When a source node wants to transmit a request packet to a destination node, he needs 2⊕ + 1mul ($r'Q_i$), the other *n* intermediate nodes (we assume that there are n intermediate nodes) each need (1⊕+1S) operations and the destination node needs (2⊕+1S) operations. Totally, it needs (n+4)⊕(n+ 1)S, and 1mul. In reply phase, the computation cast of destination node needs 1mul ($r''Q_j$) operation. The source node needs 1S operation and the n neighbor nodes each need 1S operation. They only forward, drop or accept the packet. Totally, it sums up to (n+1)S+1mul. In [1], to transmit a RREQ packet, the source and destination node both need 1H + 1Asym + 1Exp, the other n intermediate nodes each needs 1Asym. In RREP phase, it totally needs (n+2) cmp, (n+1) Asym for computing $\{T_D\}_{PKi}$, (n+1) Sym for computing $T_D(seq, K_s)$ and 2(n+2) Sym for computing $K_s(seq, END)$. To sum up, in this phase, it requires (n+2) cmp, (n+1) Asym and 3n+5 Sym. In the RREQ phase of [2], the source node needs 2Asym+1Sym+1H operations. Each intermediate node needs 3Asym+1H. The destination node needs (2n+2)Asym+1Sym+(n+1)H. To sum up, in this phase, it requires (5n+4) Asym, 2 Sym, (2n+2) H, (n+2) crte. In RREP phase, source node needs 2nSyn+3nH. Each intermediate node needs 3n-5Sym+5n-9H and the destination node needs (2n+2)Sym+(2n+2)H. To sum up, in this phase, it requires (5n-4) Sym, (8n-9) H. In the RREQ phase of [3], the source node needs 1Asym+1H+1Exp operations. Each intermediate node needs 1Asym. The destination node needs 1Asym+1H+1cmp. In the RREP phase, the destination node needs 1Asym+1H+2Exp. Each intermediate node needs 2Asym and the source node needs 2Asym+1H+1Exp+1cmp. In the RREQ phase of [7], the source node, all intermediate nodes and the destination node totally need (n+2) crte operations. In the RREP phase, the source node, all intermediate nodes and the destination node totally needs (n+2) Sym and (n+2) crte operations. In [9], the source node needs 1Asym+2Sym+1Exp (1Exp for computing its public key), intermediate nodes need n(Sym+Asym) and destination node needs 2Sym+(n+1)Asym. It totally needs 2(n+1)ASym+(n+4)Sym+1Exp in RREQ phase. In RREP phase, the source node needs 2Sym+1cmp, intermediate nodes need n (Sym+cmp) and destination node needs (n+1) Asym+4Sym, so that all need (n+1)Asym +(n+6)Sym +(n+1)cmp.

From the table (1), we can see that our scheme is the most effective protocol.

(2) Bandwidth Consumption Comparisons

Bandwidth consumption is an important issue in ad hoc network, for example, a low bandwidth consumption protocol can make the system's power consumption lower and the transmission speed quicker. In the following, we only compare the maximal bits transferred between each pair of intermediate nodes in the path and assume that the symmetric encryptions used by protocols [1, 2, 3, 7, 9] is AES-192 and the public key used is RSA-1024. M presents the message the scheme encrypts.

For the computation in RSA and Elgamal cryptographic system is typically 1024 bits long. Elliptic curves (ECC) has an computational advantage than RSA and Elgamal, it uses only a 160 bit key to provide the same level of security. The bits length of the parameters transferred in our scheme, $H(r_k N_{ij}) \oplus H(e(SP, Q_j))$, $H(N_{ij}) \oplus r''$, $r'Q_i$, T, is 160*4 in request phase. And the bits of transferred parameters, $r'Q_i$, $r''Q_j$, T+1, are 160*3 bits in reply phase.

In [1], the scheme needs 192*2 ($K_T(M)$, $K_S(M)$) + 1024 (PK_O) + 128 (random number) bits in Route Request phase. And 1024 (PK_i) + 192 ($T_D(M)$) bits in Route Response message with PK_i and T_D. In [2], the size of message transmitted is 160*(n + 1) + 192 + 1024 bits in path discovery phase and 192 bits in path reverse phase. In [3], it costs 1024 bits in path discovery phase and 1024*2 bits in reverse phase. In [7], it costs 128*2 bits in anonymous route discovery phase and 192 + 128 bits in route reply phase. In [9], it costs 128*2 + 160*2 + 192 + 1024 bits in discovery phase and 192*2 bits in reply phase. We show our comparisons in table 4.

Table 4. Comparisons of bandwidth consumption of our protocol with others

Protocols	Our protocol	ASR [1]	SDAR [2]	SARPAKE [3]	MASK [7]	ARM [9]
Request	4*160	192*2 + 1024 + 128	160*(n + 1) + 192 + 1024	1024	128*2	128*2 + 160*2 + 192 + 1024
Reply	3*160	192	192*(n + 1)	1024*2	192 + 128	192*2
Total	1120	1536	352*n + 1568	3072	576	2176

5.2 Security attributes Comparisons

Table 5. Comparisons of security and necessity of pre-sharing secret keys

Protocols	Our protocol	ASR [1]	SDAR [2]	SARPAKE [3]	MASK [7]	ARM [9]
MIMA attack resistance	yes	yes	yes	yes	yes	yes
Anonymity	yes	yes	yes	yes	no	yes
backward and forward secrecy	yes	no	yes	yes	yes	yes
Needn't pre-share secret keys	yes	no	no	no	no	no
untraceable	yes	no	no	no	no	no

In this section, we explore the security attributes of the other protocols [1, 2, 3, 7, 9], are path-based methods and hence easier to be traced if there exists a system monitor. Besides, they all need preshare secret keys. Moreover, in

[1], doesn't possess forward and backward secrecy since they assume a secrecy shared between the source and destination node but doesn't update it in each session. In [7] put the identity of destination node in RREQ message, intermediate node would know who destination node is. This violates the anonymity property [13]. Azzedline Boukerche et al.'s scheme [2] would have risk to be traced, since every intermediate node would produce a symmetric key and forward with RREQ message to destination node, then destination node would employ the symmetric key to encryption the RREP message layer by layer. This scheme would have the risk for adversary to trace the source node. Rongxing Lu et al.'s research [3] also have the same risk, since they employ the public key of ancestor node to encryption the message in path reverse phase that would account the adversary to trace RREP message and find source node. Yanchao Zhang et al.'s research [7] assumes each pair of nodes has the session key to encryption the reply message and that would also have risk to be traced to find the source node. After the comparisons as shown in table 5. We can conclude that our protocol not only can satisfy the security requirements but also is more efficient in bandwidth consumption than other schemes.

From table 4 and table 5, we can see that our scheme outperforms all of the proposed schemes in route anonymity and security except for [7]. It seems that our scheme is less efficient than [7]. However, [7] needs to pre-share secret keys among all nodes in the network in advance and can't achieve the anonymous property as pointed by [13].

6. Discussion

In this session, we describe how our scheme can achieve the requirement of preventing KCI attack. KCI attack means that when the private key of a node has been compromised, the adversary can impersonate the other node to communicate with the compromised node. We can enhance our protocol and describe the needed changes in the corresponding phase as follows. The enhanced protocol is also depicted in figure 4.

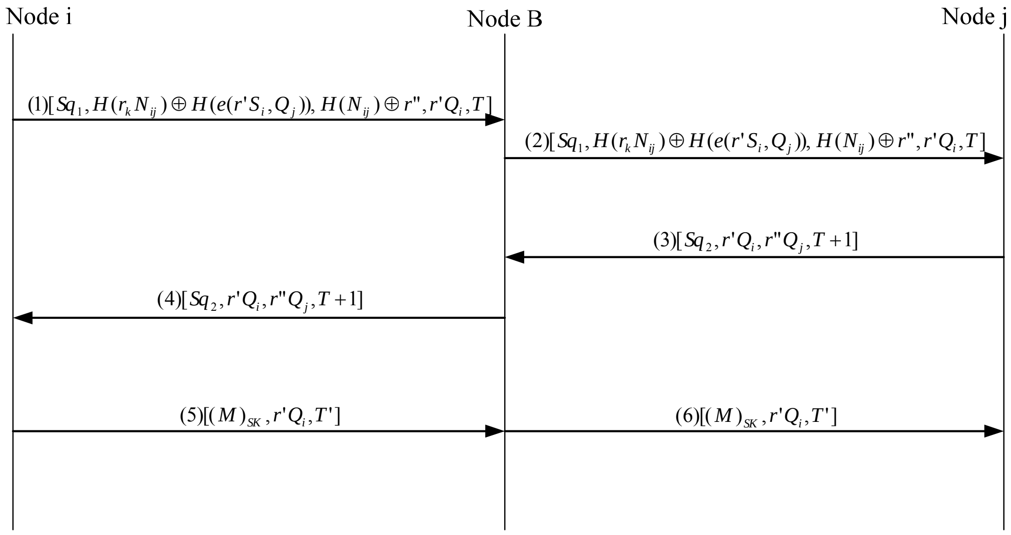

Node i
Node B
Node j
(1)$[Sq_1, H(r_k N_{ij}) \oplus H(e(r'S_i, Q_j)), H(N_{ij}) \oplus r'', r'Q_i, T]$
(2)$[Sq_1, H(r_k N_{ij}) \oplus H(e(r'S_i, Q_j)), H(N_{ij}) \oplus r'', r'Q_i, T]$
(3)$[Sq_2, r'Q_i, r''Q_j, T+1]$
(4)$[Sq_2, r'Q_i, r''Q_j, T+1]$
(5)$[(M)_{SK}, r'Q_i, T']$
(6)$[(M)_{SK}, r'Q_i, T']$

Figure 4. enhanced Protocol

(a) Request phase

Whenever node *i* wants to send confidential data to node *j*, he performs the following steps.

Step1. Chooses two random numbers *r*’, *r*” and searches the sending *PT* using *j* as index to find $H(r_k N_{ij})$.

Step2. Generates a unique Sq_1 for this route, computes $r'Q_i$, $H(N_{ij})\oplus r''$ and $H(r_k N_{ij})\oplus H(e(r'S_i,Q_j))$.

Step3. Makes a Request packet, [Sq_1, $H(r_k N_{ij})\oplus H(e(r'S_i,Q_j))$, $H(N_{ij})\oplus r''$, $r'Q_i$, T], and broadcasts this packet to all other nodes within its wireless transmission range.

Step4. When all nodes receiving the packet, each node, say node *B*, would check to see if the packet has already been received using the unique Sq_1. If it has been, he rejects. Otherwise, node *B* stores the value of $r'Q_i$ and computes: $H(r_k N_{ij})\oplus H(e(r'S_i,Q_j))\oplus H(e(r'Q_i,S_B))$, getting $H(r_kN_{ij})$'. Then he uses $H(r_kN_{ij})$' as index to search his receiving pseudonym table. If he can find such an item, then he is the destination node. He knows node *i* wants to communicate with him by extracting out the corresponding Node ID in the receiving pseudonym table. Here, we assume that the field value in the corresponding Node ID of his receiving pseudonym table is *i*. Otherwise, node *B* broadcasts the packet to all other nodes within its wireless transmission range.

Step5. Goes to Step4.

(b) Reply phase

In this phase, whenever destination node j receives the request packet and responds to source node i, he will perform the following steps.

Step1. Computes $H(e(r'Q_i, S_j)) \oplus H(r_k N_{ij}) \oplus H(e(r'S_i, Q_j))$, obtaining $H(r_k N_{ij})$. Then, uses $H(r_k N_{ij})$ as the searching key to search his receiving *PT*. Here, we assume the corresponding field value of node ID is *i*.

Step2. Generates a unique *Sq*2, computes $H(N_{ij}) \oplus (H(N_{ij}) \oplus r'')$, getting *r*" and computes the session key shared with node i as $H(e(r''S_j, r'Q_i))$.

Step3. Broadcasts the packet = [*Sq*2, $r'Q_i$, $r''Q_j$, *T*+1], to all other nodes within its wireless transmission range.

Step4. When all nodes receiving the packet, each node, say node *B*, would check to see if the packet has already been received using the unique *Sq*2. If so, he rejects. Otherwise, he checks $r'Q_i$ to see if this is the exact value stored in the request phase. If it is, then either he is the source node or the intermediate node. If he is the source node, then he can use the session key $H(e(r'S_i, r''Q_j))$ to communicate with *j*; otherwise, if he is the intermediate node, he broadcasts the message to all other nodes within its wireless transmission range. Else, he is not both the two cases, he rejects.

Step5. Goes to Step4.

(c) Data transfer phase

After completing the request and reply phases, the two communicating parties, node *i* and *j*, has established a common session key *SK*

$(=H(e(r'S_i, r''Q_j)))$. Then they use *SK* to encrypt the exchanged message *M*. The transmitted data packet would consist of $En_{SK}(M)$, $r'Q_i$ and *T*'. That is, source node *i* sends the confidential *M* to the destination node *j* by broadcasting the data packet, [$[M]_{SK}$, $r'Q_i$, *T*'], to all other nodes within its wireless transmission range. After receiving the data packet, only node *j* can decrypt the message $[M]_{SK}$. Since he is the only node who can compute the session key $SK(= H(e(r''S_j, r'Q_i)))$.

As an illustration of perverting KCI attack, we assume that an adversary *E* acquires the secret key S_i (= sQ_i) of node *i* and wants to impersonate node *j* to communicate with node *i*. Although, he can compute $H(r_kN_{ij})(= H(r_ke(S_i, Q_j)))$ and get *r*" from $H(N_{ij}) \oplus r''$. He can't calculate the session key by computing $SK=H(e(r'S_i, r''Q_j)$ for he doesn't know the random number *r*' (committed in $H(r_k N_{ij}) \oplus H(e(r'S_i, Q_j))$) chosen by node *i* due to ECDLP. In addition, *E* does not know the private key S_j of node *j*, from the above mentioned, we can see that *E* can't impersonate node *j* to communicate with node *i*. Similarly, it can be easily seen that the other direction holds as well. Therefore, our enhanced protocol can prevent KCI attack.

7. Conclusions

Traditional protocols mainly use the path-based onion-like approaches to ensure the route anonymity in a mobile ad-hoc network. However, this is extremely difficult in practice to guarantee the reliability of message delivery since path can be traced by a malicious network monitor. Hence, in this paper we proposed a mutual anonymous protocol based on broadcasting feature and bilinear pairings to achieve mutual anonymity. In the protocol, each node employs the pseudonym tables to find out who will be the destined node. Up to now, our protocol is the first scheme which not only can provide anonymity property but is very efficient since the bilinear pairings of our proposed scheme can be pre-computed in the setup phase and thus outperforms the other protocols.

References

1. Bo Zhu, Zhigup Wan, Mohan S. Kankanhalli, Feng Bao, Robert H. Deng. "*Anonymous Secure Routing in Mobile Ad-Hoc Networks*, " Local Computer Networks, 2004. 29th Annual IEEE International Conference on 16-18 Nov. 2004 Page(s):102 - 108
2. Azzedline Boukerche, Khalil El-Khatib, Li Xu, Larry Korba "*SDAR: A Secure Distributed Anonymous Routing Protocol for Wireless and Mobile Ad Hoc Networks*, " Local Computer Networks, 2004. 29th Annual IEEE International Conference on 16-18 Nov. 2004 Page(s):618 – 624
3. Rongxing Lu, Zhenfu Cao, Licheng Wang, Congkai Sun " *A secure anonymous routing protocol with authenticated key exchange for ad hoc networks*, " Computer Standards & Interfaces, Volume 29, Issue 5, July 2007, Pages 521-527
4. Reza Shokri, Nasser Yazdani, Ahmad Khonsari "*Chain-based Anonymous Routing for Wireless Ad Hoc Networks*, " Consumer Communications and Networking Conference, 2007. CCNC 2007. 2007 4th IEEE Jan. 2007 Page(s):297 - 302
5. Sk. Md. Mizanur Rahman, Masahiro MAMBO, Atsuo INOMATA, Eiji OKAMOTO "*An Anonymous On-Demand Position-based Routing in Mobile Ad Hoc Networks*, " Applications and the Internet, 2006. SAINT 2006. International Symposium on 23-27 Jan. 2006 Page(s):7 pp.
6. Zhou Zhi, Yow Kin Choong "*Anonymizing Geographic Ad Hoc Routing for Preserving Location Privacy*, " Distributed Computing Systems

Workshops, 2005. 25th IEEE International Conference on 6-10 June 2005 Page(s):646 - 651

7. Yanchao Zhang, Student Member, IEEE, Wei Liu, Wenjing Lou, Member, IEEE, and Yuguang Fang, Senior Member, IEEE "*MASK: Anonymous On-Demand Routing in Mobile Ad Hoc Networks*, " Wireless Communications, IEEE Transactions on Volume 5, Issue 9, September 2006 Page(s):2376 - 2385
8. Azzedline Boukerche, Khalil El-Khatib, Li Xu, Larry Korba "*Anonymity Enabling Scheme for Wireless Ad hoc Networks*, " Global Telecommunications Conference Workshops, 2004. GlobeCom Workshops 2004. IEEE 29 Nov.-3 Dec. 2004 Page(s):136 - 140
9. Stefaan Seys and Bart Preneel ", " Advanced Information Networking and Applications, 2006. AINA 2006. 20th International Conference on Volume 2, 18-20 April 2006 Page(s):133 - 137
10. Chao-Chin Chou, Student Member, IEEE, Davis S. L. Wei, Member, IEEE, C.-C. Jay Kuo, Fellow, IEEE, and Kshirasagar Naik, Member, IEEE "*An Efficient Anonymous Communication Protocol for Peer-to-Peer Applications over Mobile Ad-hoc Networks*, " Selected Areas in Communications, IEEE Journal on Volume 25, Issue 1, Jan. 2007 Page(s):192 - 203
11. Jiejun Kong, Xiaoyan Hong, and Mario Gerla, Fellow, IEEE "*An identity-Free and On-Demand Routing Scheme against Anonymity Threats in Mobile Ad Hoc Networks*, " Mobile Computing, IEEE Transactions on Volume 6, Issue 8, Aug. 2007 Page(s):888 - 902
12. Reza Shokri, Maysam Yabandeh, Nasser Yazdani "*Anonymous routing in*

MANET using Random Identifiers, ” Networking, 2007. ICN '07. Sixth International Conference on 22-28 April 2007

13. Song Li, Anthony Ephremides “*Anonymous Routing: A Cross-Layer Coupling between Application and Network Layer*, ” Information Sciences and Systems, 2006 40th Annual Conference on Publication Date: 22-24 March 2006 Page(s):783 - 788

14. Lijun Qian and Ning Song, Xiangfang Li “*Secure anonymous routing in clustered multihop wireless ad hoc networks*, ” Information Sciences and Systems, 2006 40th Annual Conference on 22-24 March 2006 Page(s):1629 – 1634

15. D.Balfanz, G. Durfee, N. Shanlar, D. Smetters, J. Stasson, and H. –C. Wong, “*Secret handshakes from pairing-based ley agreements*, ” Proceedings of the 2003 IEEE Symposium on Security and Privacy, Oakland, California, 2003, p. 180

16. A. Joux. “*A one round protocol for tripartite Diffie–Hellman*, ”in: Proceedings of Algorithmic Number Theory Symposium Lecture Notes in Computer Science, vol. 1838, Springer-Verlag, Berlin, 2000, pp. 385–394.

17. G.Frey, H. Ruck. “*A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves*, ”Mathematics of Computation 62 (1994) 865–874.

18. A. Menezes, T. Okamoto, S. Vanston. “*Reducing elliptic curve logarithms to logarithms in a finite field*, ”IEEE Transaction on Information Theory 39 (1993) 1639–1646.

19. D. Boneh, M. Franklin. “*Identity-based encryption from the Weil*

pairing, "in: Advances in Cryptology—Crypto_01, LNCS 2139, Springer-Verlag, 2001, pp. 213–229.

20. N.P. Smart. "*An identity based authentication key agreement protocol based on pairing*, "Electron. Lett. 38 (2002) 630–632.
21. K.G. Paterson. "*ID-based signature from pairings on elliptic curves*, "Electron. Lett. 38 (18) (2002), 1025–1026.
22. Shamir. "*Identity based cryptosystems & signature schemes*, "Advances in Cryptology, CRYPTO'84, Lecture Notes-Computer Science, 1984, pp. 47–53
23. Fangguo Zhang and Xiaofeng Chen. "*Attack on an ID-based authenticated group key agreement scheme from PKC 2004*, "Information Processing Letters,Volume 91, Issue 4,31 August 2004,Pages 191-193
24. Kyungah Shim and Sungsik Woo. "*Weakness in ID-based one round authenticated tripartite multiple-key agreement protocol with pairings*, "Applied Mathematics and Computation, Volume 166, Issue 3, 26 July 2005, Pages 523-530
25. Popescu, C. "*A secure authenticated key agreement protocol*, "Electrotechnical Conference, MELECON 2004. Proceedings of the 12th IEEE Mediterranean Volume 2, 12-15 May 2004 Page(s):783 - 786 Vol.2
26. Han, J.; Liu, Y. "*Mutual Anonymity for P2P Systems*, " Parallel and Distributed Systems, Transactions on: Accepted for future publication Volume PP. Forthcoming, 2003 Pages(s): 1-1 Digital Object Identifier 10.1109/TPDS.2007.70805