IEICE TRANS. INF. & SYST., VOL.E92–D, NO.5 MAY 2009

PAPER Special Section on Information and Communication System Security

An Authenticated On-Demand Routing Protocol with Key Exchange for Secure MANET*

Youngho PARK[†a)], ***Nonmember and*** **Kyung-Hyune RHEE**[†b)], ***Member***

SUMMARY In the meantime, most secure ad hoc routing protocols based on cryptography just have assumed that pair-wise secret keys or public keys were distributed among nodes before running a routing protocol. In this paper, we raise a question about key management related to existing secure routing protocols, and then we propose an authenticated on-demand ad hoc routing protocol with key exchange by applying the ID-based keyed authenticator. In particular, we focus on providing an authentication mechanism to Dynamic Source Routing protocol combined with Diffie-Hellman key exchange protocol, and then we demonstrate simulated performance evaluations. The main contribution of our work is to provide a concurrent establishment of a route and a session key in a secure manner between source and destination nodes in ad hoc networks.

key words: *secure ad hoc routing, authentication, ID-based key, key exchange, DSR*

1. Introduction

In contrast with wired networks where routers are a part of the network infrastructure, there is no pre-defined network infrastructure to establish routes in mobile ad hoc networks (MANET). Instead, each mobile node serves as a router to forward messages sent by other nodes to any other nodes in MANET.

As most network protocols, ad hoc routing protocols are often designed for non-adversarial networks and thus forgo security features. In a friendly network environment, we expect a node to relay packets, to share information truthfully and to generate packets only when needed. However, all nodes are not always cooperative in such a network or some nodes are even malicious. What is worse, a malicious node may fabricate and modify routing packets that pass through it. Subsequently, networks can be fragmented by the wrong routing information advertised by the malicious nodes.

From an application-layer viewpoint, attacks to ad hoc routing protocol are regarded as instances of a Denial-of-Service (DoS) attack. For example, Dynamic Source Routing (DSR) protocol [17] utilizes source routing which the entire path is specified in packets. This routing protocol lacks any integrity check, and hence a simple DoS attack can be launched by just altering the route information in the packet, such that the packet cannot be delivered to the destination correctly. Therefore, a viable ad hoc routing mechanism must be able to withstand such attacks, and cryptography is a commonly used preventive measure to counter fabrication and modification attack.

1.1 Challenges

It is no doubt that cryptography is a fundamental countermeasure against various types of attacks. However, in cryptography, a secure communication channel is supported by cryptographic keys. Therefore, the two principals who wish to establish a secure channel between them should run an authentication protocol which has a sub-task of establishing an authenticated key between them. Such a protocol is called as an authenticated key establishment protocol [16].

Generally, most cryptographic approaches to secure ad hoc networks prefer to using symmetric key cryptographic algorithms due to its computational efficiency. However, it generally requires at most $n(n-1)/2$ key distributions complexity for n nodes. Hence, there is a trade-off between computation efficiency and complex key distribution for symmetric key based schemes. Moreover, when we consider the dynamic feature of ad hoc networks where the number of nodes may increase or decrease, pair-wise shared key distribution may be impractical.

If a shared key distribution is not available, an alternative solution is a key exchange scheme by which pairs of nodes agree on shared keys to be used for their pairwise secure communication. Key exchange is a peer-to-peer protocol which does not rely on a centralized key distribution, and it is more favorable for ad hoc network. However, this is an interactive scheme and an active route must be established in advance to exchange each other's public shares. Accordingly, the routing protocol also has to be protected in order to guarantee the viable key exchange protocol in ad hoc networks.

At this moment, we come to face with questions. Suppose that a new node participates in an ad hoc network in which there is no key distribution facilities and this node possesses no shared keys or no public keys of other nodes in advance. How can we secure the routing protocol without authenticated key establishment? And, how can we ensure the end-to-end key exchange with no assurance of a trusted

Manuscript received August 4, 2008.
Manuscript revised December 29, 2008.

[†]The authors are with the Division of Electronic, Computer and Telecommunication Engineering of Pukyong National University, Republic of Korea.

[*]This study was financially supported by Pukyong National University in the 2008 Post-Doc. program. This work was partially supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD, Basic Research Promotion Fund) (KRF-2008-521-D00454).

a) E-mail: pyhoya@pknu.ac.kr
b) E-mail: khrhee@pknu.ac.kr (Corresponding author)

DOI: 10.1587/transinf.E92.D.810

route in multi-hop wireless ad hoc network? One simple but elegant solution is to use an ID-based non-interactive key establishment scheme, and then we run both secure routing protocol and key exchange protocol simultaneously. In the meantime, most secure routing protocols assume that every node has pre-established keys, whereas most key establishment schemes for secure ad hoc networks assume that there are pre-existing routes among nodes. However, there is a contradiction between the assumption of the pre-existing route and the need of secure route discovery.

1.2 Related Work

In recent years, a various secure ad hoc routing protocols have been proposed [6], [7], [10], [14], [15], and those protocols assume a pre-distribution of shared secret keys or public keys among nodes before constituting an ad hoc network. In fact, SEAD [6], SRP [10] and ARAN [14] assume that all nodes share pair-wise secret keys or their public key certificates are known to their neighboring nodes in advance. Moreover, SRP is vulnerable to attacks that involves adding or deleting nodes to or from the route because it did not authenticate intermediate nodes. Ariadne [7] is the well-known secure ad hoc routing protocol, but it requires a TESLA key distribution scheme [11] which suffers from a loose time synchronization among nodes. Consequently, previously proposed schemes are not enough to satisfy our challenge because most previous schemes assume that every nodes has pre-distributed shared keys or certified public keys of other nodes.

The authors of SRDP [8] proposed some cryptographic schemes for HMAC-based secure DSR route discovery protocol incited by the Ariadne. One of their schemes is to use Diffie-Hellman key agreement scheme for accumulating message authentication tags. However, this scheme also assumed that every node already obtained authenticated public keys of other nodes, and it also additionally requires public key certificates.

One interesting work to us is the NIKAP [9]. The authors of NIKAP had the same idea of ours and proposed a non-interactive key establishment scheme using the notion of self-certified public key [12]. By using their non-interactive key establishment scheme, they made it possible for nodes to establish pair-wise keys in the course of routing protocol. However, the authors also assumed that self-certified public keys of all nodes were distributed to others within one-hop broadcast transmission range, so a hop-by-hop pair of nodes could derive a pair-wise key. It is not reasonable in multi-hop wireless ad hoc network because some end-nodes more than two hops away cannot establish a pair-wise key in a non-interactive manner.

1.3 Overview and Contribution

Our goal is to establish a viable route and a session key between end-to-end nodes on demand at the same time in authenticated manner even though they have no pre-distributed pair-wise secret keys or public keys of each other. In particular, we will integrate Diffie-Hellman (DH) key exchange protocol with DSR protocol which has received a lot of attention from the security community. Both protocols are on-demand mechanisms. That is, key exchange is performed when a secure communication is needed, and DSR is performed to discover a path from source to destination when a communication is needed. We name the proposed protocol ADSR-KE (*Authenticated DSR with Key Establishment*) and consider the followings as our design goals:

- *Route integrity*: It must prevent a malicious node from modifying routing information.
- *Authenticity*: Any unauthorized node should not be able to participate in routing protocol.
- *Key establishment*: Once completing route discovery protocol, the source and the destination node should establish a session key that can be subsequently used for secure end-to-end communication.

One interesting aspect of our work is a novel application of non-interactive key sharing scheme to secure ad hoc network. In order to guarantee the integrity and authenticity of DSR route discovery and key exchange protocol, we apply the notion of Boyd et al.'s ID-based keyed authenticator [3] based on Sakai et al.'s non-interactive key establishment scheme [13] assuming that each node has its ID-based private key issued by a trusted authority (TA), and the public key is the identity string of a node. The scheme is advantageous to ad hoc network with limited infrastructure that nodes can perform an authentication protocol just by using their publicly known identifiers even though there is no key distribution facilities in the network.

Furthermore, ID-based cryptography would be suitable for DSR because the identities of nodes on the route are stated in packet and these identities are regarded as public keys. Therefore, a node can verify the authenticity of the received packet by generating the ID-based keyed authenticator using the identifiers specified in the source route field of DSR and by checking the accumulated authentication tag appended in the packet.

Consequently, the proposed protocol requires neither a public key certificate management nor a complex shared key distribution within ad hoc network. ID-based private key issued by a TA is sufficient for each node to authenticate routing packets and establish a shared key at the same time in our work. Based on this observation, our scheme is suitable for temporary ad hoc network environments where it is difficult to connect to a TA within ad hoc network and to distribute pair-wise session keys due to a separated TA from the network. It is our main contribution.

The rest of this paper is organized as follows: In Sect. 2, we briefly describe the notion of ID-based keyed authenticator. We present our proposed protocol in Sect. 3. We discuss the security and the performance by simulation in Sect. 4, and then conclude in Sect. 5.

2. ID-Based Keyed Authenticator

We briefly describe the ID-based keyed authenticator [3] based on Sakai et al.'s non-interactive key distribution scheme [13] which serves as a basis of our work. Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two cyclic groups of prime order q. Pairing based cryptosystems make use of a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ which has the following properties: 1) *Bilinear*: for $P \in \mathbb{G}_1$ and $a, b \in Z_q^*$, $\hat{e}(aP, bP) = \hat{e}(P, P)^{ab}$; 2) *Non-degenerate*: $\hat{e}(P, P) \neq 1$. Typically, the map $\hat{e}$ will be derived from the Weil or Tate pairings on an elliptic curve over a finite field [2].

Let $s \in Z_q^*$ be the master secret of TA whose role is to issue an ID-based private key $S_i = sH_1(id_i) \in \mathbb{G}_1$ derived from the identifier id_i, where $H_1 : \{0,1\}^* \to \mathbb{G}_1$ is an admissible encoding function to map an arbitrary string to a point in $\mathbb{G}_1$. Suppose that id_i wants to send a message m to id_j in an authenticated manner, and id_i has no shared secret key with id_j but id_i knows that its desired communication partner's identity is id_j. Assuming that each entity keeps its ID-based private key issued by TA, the ID-based keyed authenticator can be computed as follows:

1. id_i computes $k_{ij} = F(\hat{e}(S_i, H_1(id_j)))$ by using its private key S_i and the identifier id_j, where F is a pseudo random function, and
2. id_i computes $\sigma_i = HMAC(k_{ij}, m)$ and sends $\langle m, \sigma_i \rangle$ to id_j. This σ_i is the authenticator of id_i to id_j.

Upon receiving the $\langle m, \sigma_i \rangle$, id_j can verify the authenticator as follows:

1. j computes $k_{ji} = F(\hat{e}(H_1(id_i), S_j))$ by using its private key S_j and the sender's identifier id_i , that is, $k_{ji} = k_{ij}$ and then,
2. checks if $\sigma_i \stackrel{?}{=} HMAC(k_{ji}, m)$.
   If the verification is hold, then id_j is assured that the received message m was really sent by id_i.

This is the basic function of ID-based keyed authenticator. As shown in the above scheme, id_i and id_j can derive a shared key $k_{ij} = k_{ji}$ from the other party's public identity without help of key distribution mechanism. The correctness of k_{ij} and k_{ji} can be shown by $\hat{e}(S_i, H_1(id_j)) = \hat{e}(H_1(id_i), H_1(id_j))^s = \hat{e}(H_i(id_i), S_j)$. Dupont and Enge proved that Sakai et al.'s scheme is secure under the assumption of the BDH problem and the random oracle model in [5].

3. Authenticated DSR with Key Exchange

In this section, we present our ADSR-KE protocol which allows the source node to discover an authenticated path to the destination node and establish a session key at the same time. Since, our main focus is to guarantee the authenticity and integrity of routing protocol, we do not address internal selfish nodes or misbehaving nodes which drop packets or do not reply routing protocol. A solution to these problems cannot be satisfied with only cryptographic mechanisms and it needs some additional operations, such as auditing or network watching schemes [4].

Table 1 Notations for ADSR-KE.

Notation	Description
rreq	route request type.
rrep	route reply type.
$\sigma_{i...j}$	accumulated authentication tag from i to j.
$\mathcal{H}()$	cryptographic hash function.
$\mathcal{H}_{k_{ij}}()$	HMAC with the key k_{ij}.
g^x, g^y	DH parameters for $g^{xy} (\mathrm{mod}\ p)$, respectively.
k_{ij}	non-interactively shared key between i and j.
$\mathcal{K}_{SD}$	session key by DH between node S and D.

There are two types of strategies for authenticated route discovery; *forward* authentication of Route Request packets and *backward* authentication of Route Reply packets. In forward authentication, each intermediate node computes and accumulates its authentication tag with respect to the route request packet which is propagated from the source to the destination before forwarding it to next hop. On the other hand, in backward authentication, each node accumulates its authentication tag with respect to the route reply packet which is propagated from the destination to the source. These strategies have both advantage and disadvantage from the viewpoints of performance and security. We refer to [8] for detailed features of both strategies.

We present both versions of Forward ADSR-KE and Backward ADSR-KE. Before presenting our protocol, bring to mind that the focus of this paper is an application of non-interactive cryptographic scheme to secure ad hoc network rather than routing processes or behaviors of mobile nodes.

3.1 Notations and Assumptions

Throughout this paper, we use the notations in Table 1 and our ADSR-KE protocol message is basically formed as the following format:

$$\{type, src, dst, \#seq, DH, \sigma, (routes)\}$$

$type$	: packet type.
src	: source node.
dst	: destination node.
$\#seq$	: sequence number.
DH	: Diffie-Hellman parameter.
$\{route\}$	: node list for source routes.
σ	: authentication tag.

Although we do not put any key distribution mechanism in the ad hoc network, we assume that every node with id_i keeps its ID-based private key $S_i = sH_1(id_i)$ issued by an off-line TA, as shown in Sect. 2, before participating in the network. This is a kind of key issuance similar to a certificate issuance in the traditional PKI, hence it differs from the key distribution in ad hoc network. Who can act as such a TA is application dependent. For example, the headquarters and the conference host can be act as TA in the case of military tactical and the conference ad hoc network, respectively.

Related to ID-based key issuance, an expiry date can be

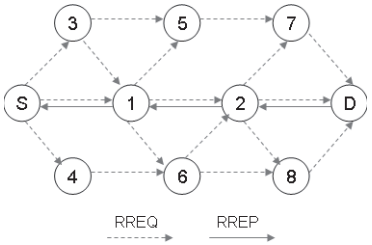

Fig. 1 Example topology for route discovery.

easily embedded in the key itself, e.g. by concatenating the current date to the identity string to limit the validity period of an ID-based key to one day [2]. In the case of temporary ad hoc network, the lifetime of the network may not be long. Therefore it is reasonable to issue a short-lived private key for one day or a few days, and it can minimize the effects of key compromise.

In the notations in Table 1, key k_{ij} for the authentication tag $\sigma_i = \mathcal{H}_{k_{ij}}(m)$ from node i to j is not necessarily established before running the protocol. This key can be shared by both nodes using ID-based keys without interaction on the protocol run, and can be used for computing the authentication tag for message m according to the procedure in Sect. 2. Therefore, the node i is able to generate the authenticator σ_i to the node j by using the on-the-fly key k_{ij} without any share of the node j. Of course, the destination node j can also verify the σ_i by using the key $k_{ji} = k_{ij}$.

However, due to the static feature of ID-based private key derived from the identifier of each node, it is not recommended to use the ID-based non-interactively shared keys for another cryptographic purpose such as confidentiality. Hence, we need an additional key establishment protocol to establish a session key between the end-to-end nodes for securing subsequent communication. This is the reason why we embed DH key exchange scheme into the route discovery protocol.

3.2 Route Discovery with Key Exchange

To clarify and understand our protocol, we assume the route in the topology, S - 1 - 2 - D, shown in the Figure 1 as an example. Detailed protocol is presented in the boxed description.

Suppose that the node S wants to find a path to the destination node D and establish a shared secret key. S first initiates a route request (RREQ) including its DH key exchange parameter g^x (where g is a generator of Z_p^* of a prime order p and x is S's session random value) and authentication tag σ_S to convince the target node of the legitimacy for the RREQ using the key k_{SD}. Then S broadcasts RREQ message to the network.

Each neighbor node i which received this RREQ appends its identifier to source route field and updates the authentication tag by computing the key k_{iD} toward D, and then forwards the RREQ to its neighbors. Through this process, the route request reaches the destination node D in the end. At this phase, not only the source node S but also each intermediate node i can set k_{SD} and k_{iD} for ID-based keyed authenticator, respectively, by using D's public identity.

Upon receiving the RREQ, D first computes k_{SD} and every key k_{iD}, where i is the identifier specified in the source route field, and verifies accumulated authentication tag. If the authentication tag is valid, then D computes session key $\mathcal{K}_{SD} = \mathcal{H}(g^{xy})$ by using its session random value y and g^x included in route request according to DH key establishment protocol, and then sends route reply (RREP) to S. This RREP will reversely pass through the nodes specified in the source route field of RREQ. At this moment, D also appends not only its DH parameter g^y corresponding to $\mathcal{K}_{SD}$ but also authentication tag for the RREP by $\mathcal{K}_{SD}$ and k_{SD}.

Note that the keys used for computing authentication tag in RREQ by the source node and RREP by the destination node are different shared keys. For the source node, it

Protocol: Forward ADSR-KE.
/* route request phase */

1. $S \rightarrow * :$ { rreq, S, D, #seq, g^x, σ_S, () };
   where $\sigma_S = \mathcal{H}_{k_{SD}}$(rrep, S, D, #seq, g^x).
2. $1 \rightarrow * :$ { rreq, S, D, #seq, g^x, σ_{S1}, (1) };
   where $\sigma_{S1} = \mathcal{H}_{k_{1D}}(\sigma_S, 1)$.
3. $2 \rightarrow * :$ {rreq, S, D, #seq, g^x, σ_{S12}, (1, 2)};
   where $\sigma_{S12} = \mathcal{H}_{k_{2D}}(\sigma_{S1}, 2)$.
4. D computes k_{SD} and each k_{iD}, and then verifies,
   $\sigma_{S12} \stackrel{?}{=} \mathcal{H}_{k_{2D}}(\mathcal{H}_{k_{1D}}(\mathcal{H}_{k_{SD}}$(rreq,$S$, D,#seq,g^x),1),2);
   if valid, D sets $\mathcal{K}_{SD} = \mathcal{H}(g^{xy})$ for its session random y.

/* route reply phase */

1. $D \rightarrow 2 :$ { rrep, D, S, #seq, g^y, σ_D, (1, 2) };
   where σ_D=$\mathcal{H}_{k_{SD}}(\mathcal{H}_{\mathcal{K}_{SD}}$(rrep, D, S, #seq, g^y, (1, 2))).
2. $2 \rightarrow 1 :$ { rrep, D, S, #seq, g^y, σ_D, (1, 2) };
3. $1 \rightarrow S :$ { rrep, D, S, #seq, g^y, σ_D, (1, 2) };
4. S sets $\mathcal{K}_{SD} = \mathcal{H}(g^{xy})$, and verifies authentication tag,
   $\sigma_D \stackrel{?}{=} \mathcal{H}_{k_{SD}}(\mathcal{H}_{\mathcal{K}_{SD}}$(rrep, D, S, #seq, g^y, (1, 2)));
   if valid, S accepts the route and the session key K_{SD}.

Protocol: Backward ADSR-KE.
/* route request phase */

1. $S \rightarrow * :$ { rrep, S, D, #seq, g^x, σ_S, () };
   where $\sigma_S = \mathcal{H}_{k_{SD}}$(rreq, S, D, #seq, g^x).
2. $1 \rightarrow * :$ { rreq, S, D, #seq, g^x, σ_S, (1) };
3. $2 \rightarrow * :$ { rreq, S, D, #seq,, g^x, σ_S (1, 2) };
4. D computes k_{SD}, and checks
   $\sigma_S \stackrel{?}{=} \mathcal{H}_{k_{SD}}$(rreq, S, D, #seq, g^x);
   if valid, D sets $\mathcal{K}_{SD} = \mathcal{H}(g^{xy})$ for its session random y.

/* route reply phase */

1. $D \rightarrow 2 :$ { rrep, D, S, #seq, g^y, σ_D, (1, 2) };
   where $\sigma_D = \mathcal{H}_{k_{SD}}(\mathcal{H}_{\mathcal{K}_{SD}}$(rrep, D, S, #seq, g^y, (1,2))).
2. $2 \rightarrow 1 :$ { rrep, D, S, #seq, g^y, σ_{D2}, (1, 2) };
   where $\sigma_{D2} = \mathcal{H}_{k_{2S}}(\sigma_D)$.
3. $1 \rightarrow S :$ { rrep, D, S, #seq, $g^y D$, σ_{D21}, (1, 2) };
   where $\sigma_{D21} = \mathcal{H}_{k_{1S}}(\sigma_{D2})$.
4. S computes $\mathcal{K}_{SD} = \mathcal{H}(g^{xy})$ and each k_{Si}, and verifies
   $\sigma_{D21} \stackrel{?}{=} \mathcal{H}_{k_{S1}}(\mathcal{H}_{k_{S2}}(\mathcal{H}_{\mathcal{K}_{SD}}$(rrep, D,S,#seq, g^y,(1,2))));
   if valid, S accepts the route and the session key $\mathcal{K}_{SD}$.

initially has no shared key with the destination, so it uses an ID-based non-interactively shared key k_{SD}. On the other hand, the destination D can derive a session key $\mathcal{K}_{SD}$ which will be shared with S, and compute σ_D by using the $\mathcal{K}_{SD}$. When D replies to S, D appends its DH parameter g^y into the RREP so that S can compute the session key $\mathcal{K}_{SD}$. Therefore, the σ_D in RREP acts as not only an authenticator of D to S but also a key confirmation message at the same time because if σ_D is valid, it means that S and D successfully agreed on session key $\mathcal{K}_{SD}$.

We just describe the process of Forward ADSR-KE in detail in this paper, but the process of Backward ADSR-KE are almost same as those of Forward ADSR-KE except that update of the authentication tag of each intermediate node on the route is performed during the route reply phase toward S.

4. Analysis and Discussion

4.1 Security

In this section, we intuitively analyze security of the proposed protocol in terms of authenticity and integrity under the assumption of secure ID-based keyed authenticator.

Suppose that there are n intermediate nodes on the route, denoted by $1, 2, \ldots, n$, between a source node S and a destination node D. Each non-interactively shared key k_{iD} toward D is computed by $k_{iD} = \hat{e}(S_i, H_1(Q_D))(i \in \{1, 2, \ldots, n\})$ respectively, and the accumulated authentication tag $\sigma_{1\ldots n}$ is computed by $\mathcal{H}_{k_{nD}}(\mathcal{H}_{k_{n-1D}}(\ldots(\mathcal{H}_{k_{1D}}(\sigma_S, 1), 2)\ldots), n)$, where the previous-hop HMAC result is taken as an input to HMAC update at each intermediate node. The destination node D can verify the authentication tag by computing each non-interactively shared key using the identifiers specified in RREQ packet. In order for an adversary to break this routing protocol, the adversary must be able to reconstruct the accumulated authentication tag without being detected by the D. However, it is infeasible to counterfeit the authentication tag without knowing any ID-based non-interactively shared keys.

When we assume the trustiness of a TA, where the TA would issue ID-based private keys to legally identified nodes, a non-interactively shared key k_{ij} can be computed only by the nodes i and j. Therefore, only legitimated nodes permitted by a TA can participate in routing protocol, but unauthorized nodes cannot take part in the protocol in our ADSR-KE.

By using accumulated HMAC during the route discovery phase, we could ensure not only integrity of routing message but also authenticity of the nodes on the route. Routing packet contains mutable fields in which each intermediate node on the route modifies the content. Therefore, in order to prevent unauthorized nodes from modifying route information illegally, it is required to authenticate intermediate nodes in addition to end-to-end authentication. In our route discovery phase, we make each intermediate node j updates the tag σ_i (where $i < j$) of its previous hop to $\sigma_{ij} = \mathcal{H}_{k_{jD}}(\sigma_i, j)$. Therefore, modification of our routing protocol means forgery of the HMAC chains, and it is infeasible if we assume the robustness of HMAC scheme.

Another aspect of ADSR-KE is authenticated key exchange for establishing a session key between the source and the destination node after performing route discovery protocol. When we detach the Diffie-Hellman parameter and HMAC from the RREQ and RREP, it can be viewed as an authenticated key exchange protocol using ID-based keyed authenticator. In fact, the authors of [3] proved that the key exchange using this authenticator can become an SK-secure key agreement protocol [1].

Table 2 Comparison of functions of secure routing protocols.

	pre-shared key	auth. node	crypto. schemes
SAODV [15]	required	end-to-end	signature, hash
Ariadne [7]	required	all	HMAC, TESLA
ARAN [14]	not required	hop-by-hop	signature, certificates
ADSR-KE	not required	all	ID-based keys, HMAC

4.2 Operational Features

The issue of secure ad hoc routing protocol has received significant attention for a few years, and many protocols have been proposed as the results. Each solution, however, has not only advantage but also disadvantage, so we cannot say which one is the best protocol. Therefore, we just briefly examine some operational prerequisites of some well-known secure routing protocols in Table 2 by comparing with our ADSR-KE.

In our protocol, the use of the non-interactive keyed authenticator has the benefit that it can get rid of the requisite for key distribution mechanism to establish a pair wise key among nodes in advance. It is an intermediate approach between secret key cryptography and public key cryptography because the authentication is based on the HMAC instead of digital signature although we use an ID-based public key variant. So, the initial computational overhead of ADSR-KE is greater than that of HMAC-based scheme using pre-shared secret keys but less than that of digital signature scheme.

Another advantage of ADSR-KE is that it can reduce communication costs between the source and the destination node by simultaneously running key exchange protocol with routing protocol. In the meantime, key exchange and secure routing are separately handled. Since key exchange is an end-to-end protocol between communicating parties, it cannot help relying on underlying routing protocol and the routing protocol must be secured as a matter of course. So, we think that simultaneous establishment of route and session key between a source and a destination node is more efficient than that of key establishment after route discovery. We show the communication efficiency of our protocol in the following section.

Table 3 Simulation parameters.

scenario parameters	
number of nodes	50 nodes
maximum velocity (v_{max})	20 m/sec
dimension of space	1500 m × 300 m
nominal radio range	250 m
source-destination pairs	20
packet rate	4 packets/sec
IFQ size	50 packets
application data payload size	512 byte/packet
radio link	802.11b
DSR parameters	
initial route request timeout	2 sec
maximum route request timeout	40 sec
cache size	32 routes
cache replacement policy	FIFO
crypto. parameters	
authentication tag size	160 bit
pairing time	0.033 sec
DH modulus size	1024 bit
SRDP's DH-based MAC time	0.024 sec
Ariande's TESLA time interval	1 sec
Ariadne's propagation time	0.2 sec
Ariadne's max. time sync. error	0.1 sec

4.3 Simulated Performance

To simulate and evaluate ADSR-KE as compared to Ariadne [6] and SRDP [8] which are the variants of DSR, we used NS-2 network simulator [19] which has been used extensively in evaluating the performance of ad hoc network routing protocols. Moreover, to simulate Ariadne and SRDP, we use the same parameters specified in [6] and [8]. We modified the original DSR source code in the NS-2, but we do not consider optimization of each protocol except the basic DSR; we increased the packet size to reflect the additional fields necessary for DH key exchange and authentication tag, and we adjusted the packet transmission time to compensate for the delay necessary for processing cryptographic operations. The pairing function of ID-based keyed authenticator was estimated by using the pairing-based cryptography toolkit [18] on Pentium III mobile 700 MHz processor with 512 MB memory. We considered 160 bits keyed hash function for HMAC and a subgroup of order q in an elliptic curve E over F_p, where p is a 512 bits prime and q is a 160 bits prime.

Table 3 shows some specific parameters used in our simulation. Similar to the scenario of Ariadne's simulation, each node moves in the rectangular space of size 1500 m × 300 m according to the random waypoint mobility model. Each node is initially placed at a random location and pauses for a period of time called the *pause time*; then it chooses a new location at random and moves there with a uniformly chosen velocity between 0 and the maximum value v_{max}. The traffic pattern being used is 20 random sessions among 50 nodes, and each of them is a constant bit rate (CBR) flow at a rate of 512 bytes and 4 packets per second. All simulations were run by using 7 movement scenarios with different pause time from 0 to 600 second. We evaluated the following metrics for each simulation run:

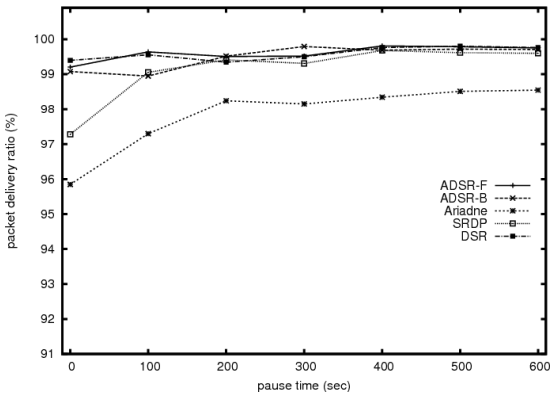

Fig. 2 Packet delivery ratio.

Packet delivery ratio (PDR). This is the fraction of total number of packets successfully received by the destination nodes to the number of packets sent by the source nodes. A higher value of PDR is a good indicator of the protocol performance.

$$PDR = \frac{(number\ of\ packets\ received)}{(number\ of\ packets\ sent)}$$

Figure 2 shows the packet delivery ratio. We can see that Forward and Backward ADSR-KE nearly overlapped with the basic DSR protocol and outperformed Ariadne and SRDP at a higher level of mobility. For pause times greater than 200 seconds, most security enhanced protocols have few difference from the basic DSR because the longer pause time makes the node rather stationary and hence the routes among nodes are almost static for these pause times once a route is discovered. On the other hand, Ariadne has lower PDR than others because Ariadne's route discovery operates more slowly due to the TESLA propagation time used for authentication.

Normalized routing load (NRL). This is calculated as the ratio of the number of transmitted routing packets to the number of actually received packets. The lower numerical result means the protocol more efficient.

$$NRL = \frac{(number\ of\ routing\ packets\ sent)}{(number\ of\ packets\ received)}$$

Figure 3 and Fig. 4 show the normalized routing load and packet overhead, respectively. As shown in these figures, we can find that Forward ADSR-KE burdens about average 8.4% more routing load than the DSR but shows the similar curve pattern compared with the basic DSR protocol. Nevertheless, ADSR-KE does not make a heavy burden on the whole rather than other protocols when the pause time is more than 200 seconds. In SRDP and Ariadne, since routing packets are processed slowly due to a longer public key certificate verification and TESLA propagation time respectively, they cause more redundant routing packets to be sent.

Average end-to-end delay (AED). This is defined as the average delay in transmission of a packet between two nodes.

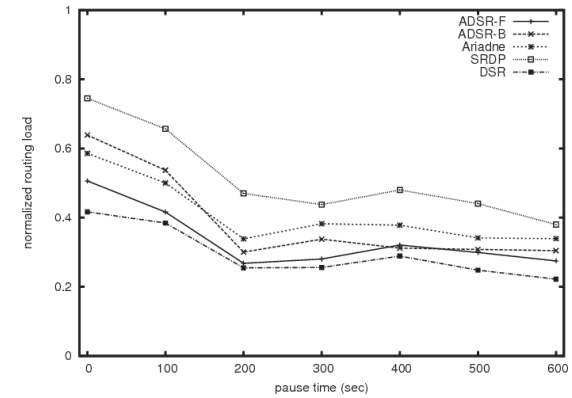

Fig. 3 Normalized routing load.

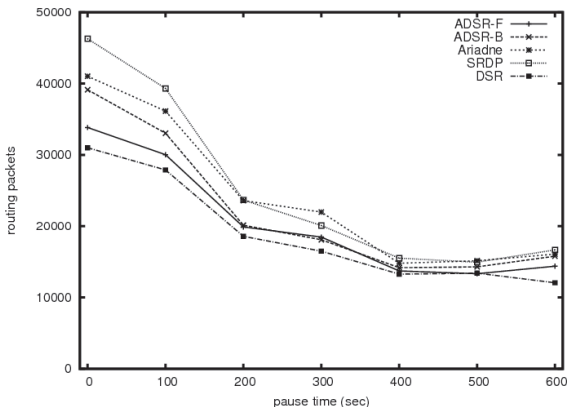

Fig. 4 Routing packet overhead.

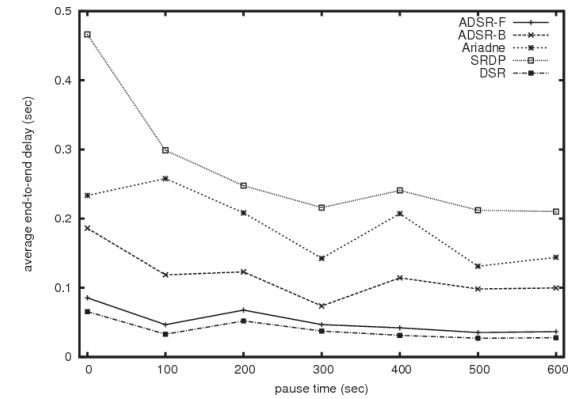

Fig. 5 Average end-to-end delay.

$$AED = \frac{\sum_{i=0}^{n}(received\ time_i - sent\ time_i)}{(total\ number\ of\ packets\ received)}$$

The remarkable result of ADSR-KE is the average end-to-end delay as shown in Fig. 5. Due to the computation of ID-based keyed authenticator of our ADSR-KE and DH-based MAC of SRDP, it may burden the processing overhead to each intermediate node. Nevertheless, Forward ADSR-KE just adds about 9 milliseconds (ms) on average to the basic DSR protocol and much faster than other protocols throughout the all scenarios. At this result, ADSR-KE's delay is measured by including a key establishment processing time during the route discovery while other protocols' are just pure route discovery processing time. Hence, additional run of end-to-end key exchange protocol is required when we use other mechanisms for secure ad hoc network, i.e., secure route discovery and then key exchange. SRDP has the longest delay which is induced by the public key certification and DH-based MAC computation, and Ariadne has longer delay than ADSR-KE due to the TESLA time interval for key retrieval.

5. Conclusion

In this paper, we proposed a secure on-demand ad hoc routing protocol with key exchange protocol, named ADSR-KE. The main contribution of ADSR-KE is ensuring authentication and integrity of routing protocol, and establishing a session key at the same time without any pre-shared security association between a source and a destination node. To achieve our goal, we considered the notion of ID-based keyed authenticator based on non-interactive key distribution scheme. This enables ad hoc nodes to securely set up a route for communication and to establish a session key between a source and a destination node at the same time. Our proposed scheme is an intermediate approach between secret key cryptography and public key cryptography because the authentication is not based on the digital signature but the HMAC although we used an ID-based public key variant. Hence, our ADSR-KE does not require not only a beforehand shared key distribution but also public key certificate management. We also confirmed the effectiveness of the proposed protocol throughout the simulation results in terms of number of packets and delay. Although we assumed DSR as our underlying on-demand routing protocol in this paper, we expect that our design concept can be applied to other on-demand protocols in the ad hoc network environments.

References

[1] M. Bellare, R. Canetti, and H. Krawczyk, "A modular approach to the design and analysis of authentication and key exchange protocols," Proc. 30th Annual ACM Symposium on Theory of Computing, pp.419–428, 1998.
[2] D. Boneh and M. Franklin, "Identity based encryption from the Weil pairing," Proc. Advances in Cryptology - CRYPTO '01, LNCS 2139, pp.213–229, Springer, 2001.
[3] C. Boyd, W. Mao, and K.G. Paterson, "Key agreement using statically keyed authenticators," Proc. Applied Cryptography and Network Security, LNCS 3089, pp.248–262, Springer, 2004.
[4] S. Buchegger and J.-Y. Le Boudec, "Performance analysis of the CONFIDANT protocol (Cooperation of nodes: Fairness in dynamic ad-hoc networks)," Proc. 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing, pp.226–236, 2002.
[5] R. Dupont and A. Enge, "Provably secure non-interactive key distribution based on pairings," Discrete Appl. Math., vol.154, no.2, pp.270–276, 2006.
[6] Y. Hu, D. Johnson, and A. Perrig, "SEAD: Secure efficient distance

vector routing for mobile wireless ad hoc networks," Proc. 4th IEEE Workshop on Mobile Computing Systems and Applications, pp.3–13, 2002.
[7] Y. Hu, A. Perrig, and D.B. Johnson, "Ariadne: A secure on-demand routing protocol for ad hoc networks," Proc. Eighth ACM International Conference on Mobile Computing and Networking (MobiCom 2002), pp.12–23, 2002
[8] J. Kim and G. Tsudik, "SRDP: Securing route discovery in DSR," Proc. Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, MobiQuitous 2005, pp.247–258, 2005.
[9] Z. Li and J.J. Garcia-Luna-Aceves, "Non-interactive key establishment in mobile ad hoc networks," Ad Hoc Networks, vol.5, no.7, pp.1194–1203, 2007.
[10] P. Papadimitratos and Z. Haas, "Secure routing for mobile ad hoc networks," Proc. SCS Communication Networks and Distributed Systems Modelling Simulation Conference (CNDS), pp.193–204, 2002.
[11] A. Perrig, R. Canetti, J.D. Tygar, and D. Song, "Efficient authentication and signing of multicast streams over lossy channels," Proc. IEEE Symposium on Security and Privacy, pp.56–73, 2000.
[12] H. Pertersen and P. Horster, "Self-certified keys - Concepts and applications," Proc. 3rd Conference of Communication and Multemedia Security, 1997.
[13] R. Sakai, K. Ohgishi, and M. Kasahara, "Cryptosystems based on pairing," Proc. Symposium on Cryptography and Information Security (SCIS 2000), 2000.
[14] K. Sanzgiri, D. LaFlamme, B. Kahill, and B.N. Levine, "Authenticated routing for ad hoc networks," Proc. 10th IEEE International Conference on Network Protocols, pp.78–87, 2002.
[15] M.G. Zapata and N. Asokan, "Securing ad hoc routing protocols," Proc. 3rd ACM Workshop on Wireless Security (WiSE '02), pp.1–10, 2002.
[16] W. Mao, Modern Cryptography, Prectice Hall, 2003.
[17] "The dynamic source routing protocol for mobile ad hoc networks (DSR)," IETF MANET Working Group Internet-Draft, July 2004.
[18] Pairing-based Cryptography Toolkit, http://crypto.stanford.edu /pbc/
[19] Network Simulator-NS2, http://www.isi.edu/nsnam/ns/

Youngho Park received his M.S. and Ph.D. degrees in Department of Computer Science and Information Security from Pukyong National University, Republic of Korea in 2002 and 2006, respectively. He is currently a postdoctor course researcher in Department of Information Engineering, Pukyong National University. His research interests are related with information security, applied cryptography and network security; authentication, key management, secure mobile ad hoc network including vehicular ad hoc network.

Kyung-Hyune Rhee received his M.S. and Ph.D. degrees from Korea Advanced Institute of Science and Technology (KAIST), Daejon Korea in 1985 and 1992, respectively. He worked as a senior researcher in Electronic and Telecommunications Research Institute (ETRI), Daejon Korea from 1985 to 1993. He also worked as a visiting scholar in University of Adelaide, University of Tokyo, and University of California, Irvine, respectively. He is currently a professor in the Division of Electronic, Computer and Telecommunication Engineering of Pukyong National University, Republic of Korea. His research interests are related to cryptography and its applications, wireless communication security and digital rights management.