Secure Bootstrapping and Routing in an IPv6-Based Ad Hoc Network

Yu-Chee Tseng, Jehn-Ruey Jiang[*]*, Jih-Hsin Lee*
Department of Computer Science and Information Engineering
National Chiao-Tung University
[*]*Department of Information Management*
Hsuan-Chuang University
Hsin-Chu, Taiwan
yctseng@csie.nctu.edu.tw, jrjiang@hcu.edu.tw, shin.janice@msa.hinet.net

Abstract

The *mobile ad hoc network* (MANET), which is characterized by an infrastructureless architecture and multi-hop communication, has attracted a lot of attention recently. In the evolution of IP networks to version 6, adopting the same protocol would guarantee the success and portability of MANETs. In this paper, we propose a secure bootstrapping and routing protocol for MANETs. Mobile hosts can autoconfigure and even change their IP addresses based on the concept of CGA (cryptographically generated address), but they can not hide their identities easily. The protocol is modified from DSR (dynamic source routing) to support secure routing. The neighbor discovery and domain name registration in IPv6 are incorporated and enhanced with security functions. The protocol is characterized by the following features: (i) it is designed based on IPv6, (ii) relying on a DNS server, it allows bootstrapping a MANET with little pre-configuration overhead, so network formation is light-weight, and (iii) it is able to resist a variety of security attacks.

Keywords: Internet Protocol version 6 (IPv6), mobile ad hoc network (MANET), mobile computing, network initialization, secure routing, wireless communication.

1 Introduction

A *mobile ad hoc network* (MANET) is an infrastructureless network consisting of a set of mobile nodes that are able to communicate with each other in a multi-hop manner without the support of any base station or access point. A node in a MANET is not only a node but also a router that is responsible of relaying packets for other nodes. A MANET has the merit that it is quickly deployable. Applications of MANETs include communications in battlefields, disaster rescue operations, and outdoor activities.

Routing is essential for a MANET to operate correctly, and a lot of routing protocols have been proposed in the literature, including proactive (table-driven), reactive (demand-driven), and hybrid solutions [3, 6, 14]. Most of the existing protocols have assumed a MANET as a nonhostile, trusted environment. Unfortunately, in the presence of malicious nodes, a MANET is highly vulnerable to attacks due to its open environment, dynamically changing topology, and lack of centralized security infrastructure. To address this concern, several secure routing protocols have been proposed recently, such as SAODV [19], SRP [11], SAR [18], CSER [8], BSAR [2], Ariadne [4], and SEAD [5].

This article intends to present a secure bootstrapping and routing protocol for an IPv6-based MANET. We envision that IPv6 would be more widely deployed and accepted in the next stage. Adopting IPv6 in MANETs would warrant the success and portability of MANETs. In particular, the important address autoconfiguration feature in IPv6 should be adopted so that mobile nodes do not need predefined IP addresses before entering a MANET. This would greatly facilitate the formation of a MANET in an open environment. However, hosts should not be able to hide their identities (i.e., IP addresses) when doing something bad; otherwise, a lot of routing misbehaviors may happen. Further, while securing the network is essential, mobile nodes should maintain very limited pre-knowledge for this purpose.

In our design, we rely on the existence of an IPv6 DNS server in the MANET for the security purpose. For those hosts who intend to prevent the impersonation attack, they have to establish their IP-domain name mappings in the DNS server prior to network formation. Alternatively, the mapping can be established on-line, but in this case the domain names/IP addresses are taken in a first-come-first-serve manner. Even so, our approach still guarantees that a host can not arbitrarily claim the ownership of an IP address. For hosts with a stronger security demand, they can check with the DNS the IP address of a domain name before conducting communications. For hosts with a weaker security demand, they do not need to contact DNS, but the secure address autoconfiguration can still ensure, to a certain degree, the identities of their communication counterparts. In our design, a host only needs to know the public key of the DNS server to achieve the above goals. Via such a mechanism, we further propose our secure routing protocol such that the identities of all hosts alone a routing path can be verified. Thus, misbehaving hosts can be easily tracked, and thus routed around if necessary. We also propose how to assign credits to hosts depending on how reliable they relay packets in the past. Thus, trusted routes can be established after the network is run for a while.

The proposed secure routing protocol is derived based on the DSR protocol [6]. The protocol incorporates the concept of CGA (cryptographically generated address) [1], address autoconfiguration [12], and DNS autoregistration [12] and discovery [7] of IPv6. It allows the network to be bootstrapped without manual administration and can resist a variety of attacks, including the black hole, impersonation, replay, and message forging attacks. In comparison, most existing works are not directly targeted at IPv6 networks, and they usually assume stronger security associations among hosts prior to the network formation. Our work only relies on the existence of a DNS server in the MANET, and a host only needs to know the public key of the DNS server prior to entering the MANET. Thus, the network formation and bootstrapping is quite light-weight.

The rest of this paper is organized as follows. Some backgrounds are given in Section 2. Our proposal is presented in Section 3. Section 4 analyzes how our protocol prevents some attacks. Conclusions are drawn in Section 5.

2 Preliminaries

2.1 Review of Secure Routing Protocols

In a MANET, two security issues need to be addressed: one is to protect transmitted data and the other is to make the routing protocol secure. The former can be done through end-to-end protection and has been well addressed in wired networks. The latter is particularly challenging for MANETs with dynamically changing topologies. If we have a MANET whose members are a “team” and know a priori a “team-key”, this is not a big problem. However, if we want to create a MANET where everybody can participate, secure routing is necessary because there is no way to enforce everybody to be honest.

The SAODV (Secure Ad hoc On-demand Distance Vector Routing) protocol [19] is an extension of AODV [16]. Adversary nodes may forge AODV packets, listen to others, reply packets in their own interests, and report errors where there are none. To defend these attacks, it is assumed that each node has a certified public key. Hop-by-hop authentication is used to protect routing messages, and all intermediate nodes need to cryptographically validate the digital signatures appended with a routing message.

Assuming the existence of a *security association* between each pair of source and destination nodes, the SRP (Secure Routing Protocol) [11] guarantees that fabricated, compromised, or relayed route replies would either be rejected or never reach back the querying source. Compared to SAODV, the verification is not needed for intermediate nodes, thus removing the overheads. The security association can be obtained via the knowledge of the communication counterpart’s public key. SRP is robust in the presence of misbehaving nodes, and provides accurate routing information in a timely manner.

The SAR (Security-Aware Routing) protocol [18] incorporates security attributes as parameters in route discovery. SAR ensures that a route only consists of nodes at the same trusted level. However, such routes may not always exist. The CSER (Cooperative Security-Enforcement Routing) protocol [8] allows a path consisting of multiple segments, each starting and ending by nodes from the same security domain as the source node. The middle of each segment can contain untrusted nodes. The trust relationship among nodes is established at configuration time and all nodes in the same security domain must abide by a formal security policy and can assure a certain level of security. By such cooperative enforcement, CSER can effectively locate misbehaving nodes in a segment, and route around hostile areas.

SEAD [5] assumes a shared key among all nodes in the network and uses hash chain to authenticate relayed messages. The protocol builds on top of a proactive routing protocol, which is believed to be more costly than a reactive protocol. Ariadne [4] tries to make DSR secure. It requires one of the following key setups to authenticate the sender of a message: (1) a pairwise shared key among all nodes, (2) a system-wide distributed public key for each node, and (3) a public TESLA (Timed Efficient Stream Loss-tolerant Authentication) key for each node. Key setups (1) and (2) are expensive. While setup (3) is not so expensive, time synchronization among nodes is a prerequisite for this case.

The aforementioned protocols all assume the existence of some security associations among hosts, which must be pre-established or established on-line. This poses difficulty in a MANET. The BSAR protocol [2] is developed on top of SUCV (statistically unique and cryptographically verifiable) identifiers [9], which ensure a secure binding between IP addresses and keys without assuming any trusted certification authority (CA) or key distributed center (KDC). The concept of SUCV identifiers is similar to that of CGAs, which will be reviewed in Section 2.3. BSAR adopts DSR to discover new routes and allows a source node to verify the identity of the host who initiated a route reply or route error message. As compared to our work, we enhance BSAR by allowing a host to verify the identity of every host in a route, and thus a variety of attacks can be avoided.

2.2 IPv6 Address Autoconfiguration

In IPv6, there are two ways for a node to configure its address: *stateful* and *stateless*. Stateless configuration is more suitable for MANETs with a dynamic, infrastructureless architecture. In stateless auto-configuration, to obtain an IP address, a node has to generate a *link-local* address and then run the *duplicate address detection (DAD)* procedure of the *Neighbor Discovery Protocol (NDP)* [10]. In DAD, a node verifies the uniqueness of its link-local address by broadcasting a *NS (neighbor solicitation)* message to neighboring nodes. Any node with the same address as the announced link-local address should reply with a *NA (neighbor advertisement)* message to enforce the former to choose a new address and retry DAD.

For multi-hop MANETs, DAD verification of link-local addresses is insufficient to guarantee address uniqueness because the same addresses may be used by hosts that are several hops away. Reference [12] proposes to extend DAD by using routable *site-local* addresses. An extended DAD scheme is proposed in [15] for MANETs by requiring a node to flood an address request (AREQ) message and then wait for a potential address reply (AREP) message. Thus, AREQ and AREP are extensions of NS and NA, respectively. If an AREQ initiator does not receive an AREP after a specific period of time, it assumes that its address is unique and can be used for communication afterwards.

2.3 IPv6 Secure Neighbor Discovery via CGAs

IPv6 allows a host to autoconfigure its own address. In an open environment like MANET, a host may easily impersonate another host's address. A few works have addressed how to secure the neighbor discovery protocol. In [1], the *cryptographically generated addresses (CGAs)* are defined to make NS and NA massages verifiable in the absence of a centralized security infrastructure. A CGA is also known as a SUCV (statistically unique and crytographically verifiable) address [9]. The basic idea of CGAs is to associate a host's address with its public key in order for other hosts to verify the ownership of the address by the host. It is assumed that a node owns a public-private key pair (PK, SK) and there is a publicly known one-way, collision-resistant hashing function H. While the upper part of a host's IP address should follow some subnet masking rules, the lower part must consist of the hashing result H(PK, rn), where rn is a random number to avoid possible collisions. Afterwards, the host can send messages, such as NS and NA, with PK and rn attached. A receiving host can then verify the originality (i.e., IP address) of the sending host. Therefore, a host can not impersonate another host by taking the latter's IP address unless it compromises SK.

2.4 IPv6 DNS Auto-registration and Discovery

IP addresses are usually too long to remember; logical *domain names* are sometimes more preferable, especially for human. For a node to resolve names of others nodes, DNS servers are used. Three well-known site local IPv6 addresses are reserved for auto-discovery of DNS servers [17]. They are fec0:0:0:ffff::1, fec0:0:0:ffff::2, and fec0:0:0:ffff::3.

To verify the uniqueness of domain names, the *6DNAR (IPv6 Domain Name Auto-Registration)* protocol [13] proposes to incorporate domain name registration into the DAD procedure of NDP. A new "domain name" option is added in NS messages, through which a node can announce its domain name together with its IP address. As such, the uniqueness of domain names and IP's can be verified altogether. NA messages are also modified so as to announce duplicate domain names as well as IP addresses.

3 Secure Bootstrapping and Routing in a MANET

In this section, we present our secure bootstrapping and routing protocol in a MANET. The design basically follows the philosophy of IPv6. The following assumptions are made.

- There is a publicly known one-way, collision-resistant hashing function H, and there exists an IPv6 DNS server in the MANET. The DNS server has a public-private key pair, and the public key has been securely distributed to all mobile nodes prior to network formation.
- For a mobile host which intends to own a permanent domain name, an entry (domain name, IP address) should have been placed at the DNS server before the network is formed. In this case, impersonating such hosts would be impossible.
- For a mobile node which dose not intend to own a permanent domain name, its (domain name, IP address) entry can be registered with the DNS server online after the network is formed. We adopt the first-come-first-serve policy for registration of new domain names. (However, for a mobile host which only wants to be a

Table 1 Control messages used in our protocol

Type	Function	Parameters
AREQ	Address REQuest	(S_{IP}, seq, DN, ch, RR)
AREP	Address REPly	(S_{IP}, RR, $[S_{IP}, ch]R_{SK}$, R_{PK}, R_{rn})
DREP	DNS server REPly	(S_{IP}, RR, $[DN, ch]N_{SK}$)
RREQ	Route REQuest	(S_{IP}, D_{IP}, seq, SRR, $[S_{IP}, seq]S_{SK}$, S_{PK}, S_{rn})
RREP	Route REPly	(S_{IP}, D_{IP}, $[S_{IP}, seq, RR]D_{SK}$, D_{PK}, D_{rn})
CREP	Cached route REPly	(S'_{IP} , S_{IP}, D_{IP}, $RR_{S'\rightarrow S}$, $[S'_{IP}, seq', RR_{S'\rightarrow S}]S_{SK}$, S_{PK}, S_{rn},$[S_{IP}, seq, RR_{S\rightarrow D}]D_{SK}$, D_{PK}, D_{rn})
RERR	Route ERRor	(I_{IP} , I'_{IP} , $[I_{IP} , I'_{IP}]I_{SK}$, I_{PK}, I_{rn})

client, establishing a domain name is not always necessary.)

Our protocol uses several control messages, whose formats and parameters are summarized in Table 1 and Table 2, respectively.

3.1 Secure Address Auto-configuration

In this section, we introduce how a mobile host securely configures an IPv6 address and verifies its uniqueness in a MANET. The proposed solution is an integration and modification of the ideas in CGA [1], extended DAD [15], and 6DNAR [13].

To join a MANET, a host must obtain an IPv6 site-local address. This address is composed of four fields: a 10-bit site-local prefix fec0::/10, a 38-bit all-zero field, a 16-bit subnet ID, and a 64-bit hash value, as illustrated in Figure 1. In particular, the last 64-bit hash value H(PK, rn) is generated based on the concept of CGA, where rn is a random number to avoid possible collisions. The subnet ID makes no sense for a MANET and can be replaced by the gateway when the node is connecting to the Internet. Here we assume the 16-bit subnet ID to be all 0's. So the site-local address is fec0::H(PK, rn). Such a design has two advantages. First, an adversary cannot arbitrarily claim the ownership of an IP address unless it finds a proper pair (PK′, rn′) such that H(PK′, rn′) = H(PK, rn). Even if the (PK′, rn′) pair is correct, the adversary may be challenged to prove its ownership of the corresponding SK′, which is difficult. Second, normal users may occasionally find collisions in the hashing results. So the random number rn provides a way to generate a new IP address while PK is kept unchanged.

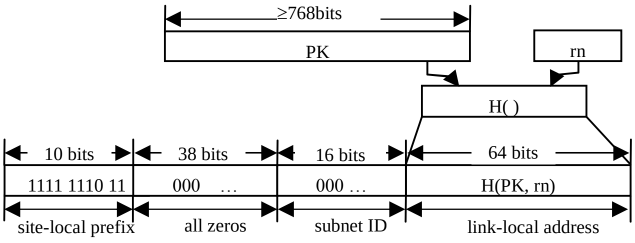

Figure 1 The CGA site-local IPv6 address

Table 2 Definitions of symbols and notations

Symbol	Description
X_{IP}	IP address of node X
X_{SK}	private key of host X
X_{PK}	public key of host X
X_{rn}	the random number used by host X to hash its IP address
DN	domain name
ch	a random number used as a challenge
seq	a unique sequence number generated by a message initiator
RR	route record to keep track of hosts traversed by AREQ/RREQ
SRR	secure route record (similar to RR except that information is added to verify each host's identity in the list)
$[msg]X_{SK}$	the ciphertext of message *msg* encrypted by host X's private key

After generating a new IP address, the host can verify its uniqueness and, if desired, register with the DNS server its domain name. We integrate the extended DAD and 6DNAR to achieve this goal. The NS and NA messages in the original DAD are extended to AREQ and AREP messages, respectively. The former can only reach one-hop neighbors, while the latter can be flooded to the MANET. All hosts will help verify the uniqueness of the IP address, and the DNS will verify the uniqueness of the IP address-to-domain name binding.

To perform the DAD procedure, a node S broadcasts an address request AREQ(S_{IP}, seq, DN, ch, RR). DN can be left empty if registration of a domain name is not desired. Every host should help verify the possible collision of S_{IP} with its own IP address and properly rebroadcast the AREQ. Duplicate AREQs will not be rebroadcast. When rebroadcasting AREQ, the host should append its address to the route record RR. When a node R receives an AREQ with S_{IP} equal to its own IP address, it unicasts[**] an address reply AREP(S_{IP} , RR, $[S_{IP} , ch]R_{SK}$, R_{PK}, R_{rn}) to S along the reverse direction of RR. When S with a pending address request receives the AREP message, it authenticates the integrity of the message as follows:

1. It verifies if the lower part of S_{IP} matches $H(R_{PK}, R_{rn})$.
2. It decrypts $[S_{IP}, ch]R_{SK}$ by R_{PK} and verifies if the decrypted result matches $[S_{IP}, ch]$, where ch is the challenge sent in S's earlier AREQ.

[**] Note that in the last hop when AREP is transmitted to S, the packet should be modified as a broadcast since S does not even have a legal IP address yet to receive packets.

The first step verifies that R does follow the CGA rule to generate its IP address. The second step ensures that R does own the corresponding private key R_{SK} for the public key R_{PK}. The inclusion of ch in AREQ serves as a challenge to R, while R's correctly encrypting ch serves as a response to S's challenge. Randomly selecting ch in each AREQ prevents replay attack. If both checks pass, the AREP message is considered valid, and S should generate a new IP address (with a new rn) and restart the DAD procedure again.

On detecting a duplicate IP address, host R should also unicast an AREP to DNS to warn DNS to not create a domain name-IP address entry for S. Again, the DNS can verify the AREP with the same checks as above. The only difference is that the challenge ch was issued by S. So the DNS should keep a copy of the ch associated with the AREQ that registered with it for a while for such secure duplication checking purpose. Figure 2 illustrates an example for the above procedure.

When a DNS server N receives an AREQ with a conflict domain name DN in its database, it unicasts a DREP(S_{IP} , RR, [DN, ch]N_{SK}) message to S. When S receives the DREP, it authenticates the message by decrypting [DN, ch]N_{SK} with DNS's public key N_{PK} and compare the result with [DN, ch] that it initiated recently. If the verification passes, S should choose another domain name and retry DAD. Again, N's correctly encrypting ch serves as a response to S's challenge.

If S receives no AREP or DREP after sending out an AREQ within a predefined period of time, it assumes that its address S_{IP} and domain name DN is unique. Similarly, if DNS receives no AREP after receiving an AREQ within a predefined period of time, it assumes that S_{IP} are unique and stores (DN, S_{IP}) in its domain name table.

3.2 Secure DNS Services

Let's consider the scenario that in an outdoor activity, we would like to establish a public server (such as *yahoo.com*) to provide services. If so, the corresponding domain name-IP address mapping should have been pre-established in the DNS. Our protocol requires each host know the public key of the DNS before entering the MANET. So a host can securely inquire the IP address of the web server via any well known secure communication protocol. Other hosts can not arbitrarily claim owning the server's IP address due to our secure address auto-configuration protocol in Section 3.1 (otherwise, the impersonating host will be challenged of owning the corresponding private key to generate the IP address).

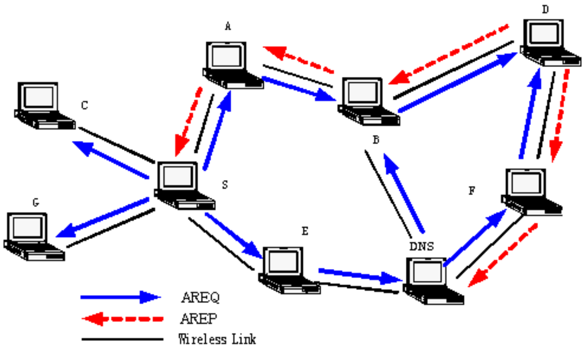

Figure 2 The secure DAD procedure to detect duplicate IP addresses and domain names

Once owning a domain name-IP address mapping in the DNS, a host can also request to change its IP address if necessary. Since we bind each IP address with a public-private key pair, the DNS can challenge the host which intends to change its IP address whether it does own the corresponding private key. Specifically, a challenge ch can be initiated by the DNS, and the replier, say X, must present its old IP address X_{IP} and new IP address X'_{IP} , together with X_{rn}(the random number to generate the old IP address), X'_{rn}(the random number to generate the new IP address), X_{PK}, and [X_{IP}, X'_{IP}, ch]X_{SK}. Note that the host does not need to change to a new key pair. The verification is similar to the earlier procedure, and correctly decrypting [X_{IP}, X'_{IP}, ch] means that X does own the secret key X_{SK}. After the verification, the DNS can switch to the new X'_{IP}.

3.3 Secure Route Discovery

Next, we present our route discovery protocol. The protocol is derived based on the DSR protocol [6]. For a source node S to search for a route to a destination D, it broadcasts a route request RREQ(S_{IP}, D_{IP}, seq, SRR, [S_{IP}, seq]S_{SK}, S_{PK}, S_{rn}). On receiving the message for the first time, each intermediate node I attaches its identity information to the route record SRR:

SRR := SRR|([I_{IP} , seq]I_{SK}, I_{PK}, I_{rn})

and rebroadcasts the RREQ. The information in SRR allows us to verify I's identity. When D receives the RREQ, it first verifies the correctness of the route as follows:

1. Check the validity of the source host by verifying: (i) if the lower part of S_{IP} is equal to H(S_{PK}, S_{rn}), and (ii) if the decrypted result of [S_{IP}, seq] S_{SK} by S_{PK} is equal to [S_{IP}, seq].
2. Check the validity of each intermediate host I in SRR by verifying: (i) if the lower part of I_{IP} is equal to H(I_{PK}, I_{rn}), and (ii) if the decrypted result of [I_{IP}, seq]I_{SK} by I_{PK} is equal to [I_{IP}, seq].

The above checks are similar to the above DAD procedure, which binds to how IP addresses are generated. Passing the checks implies that the source S and each intermediate node I are as they claimed. Then D unicasts a route reply RREP(S_{IP}, D_{IP} [S_{IP}, seq, RR]D_{SK}, D_{PK}, D_{rn}) to S along the reverse direction of RR, where RR is the route record extracted from SRR containing the intermediate nodes' IP addresses.

When S with a pending RREQ receives the RREP, it verifies the message by checking: (i) if the lower part of D_{IP} is equal to H(D_{PK}, D_{rn}), and (ii) if the decrypted result of [S_{IP}, seq, RR]D_{SK} by D_{PK} is equal to [S_{IP}, seq, RR]. If both checks pass, the RREP message is considered valid and S can start sending data packets to D via this route.

S can also cache the discovered route RR for future use. For example, when another host S′ intends to find a route to D and the RREP is received by S, S can directly reply a cached route reply CREP(S'_{IP} , S_{IP}, D_{IP}, $RR_{S'\to S}$, [S'_{IP}, seq′, $RR_{S'\to S}$] S_{SK}, S_{PK}, S_{rn}, [S_{IP}, seq, $RR_{S\to D}RR_{S\to D}$]D_{SK}, D_{PK}, D_{rn}) to S′, where $RR_{S'\to S}$ and $RR_{S\to D}$ are the routes from S′ to S and from S to D, respectively. Note that the sequence number seq′ is initiated by S′, while seq was initiated earlier by S when searching for a route to D. The verification of the route at S′ is similar to the earlier procedure. Figure 3 illustrates the transmission of RREQ, RREP, and CREP messages.

3.4 Secure Route Maintenance and Credit Management

While transmitting data packets, if an intermediate node I finds that its connection to its next hop I′ is broken, it can send a route error message RERR(I_{IP}, I'_{IP}, [I_{IP}, I'_{IP}]I_{SK}, I_{PK}, I_{rn}) to S. Again, the packet allows S to verify that the packet is sent from I. Under normal situations, S simply accepts the route error report and runs the route discovery procedure again to search for a new route. However, if the problem persists (i.e., S keeps on encountering that routes are either unusable or short-lived), S needs to determine if some nodes are malicious. S can collect the routes that it has found recently but encountered route breakage. If RERR messages are reported by the same host with a particularly high frequency, the RERR reporting node or the node next to the reporting node might be a hostile node. In this case, S should try to route around the hostile area. A hostile node may keep on changing its identity, which is allowed in IPv6. So S may not be able to find a node with a particularly high RERR reporting frequency. In this case, we suggest that S can maintain a credit for each host that has relayed data packets for it. Whenever a data packet is correctly acknowledged by D, the credit of each host in the route is increased by one. A new node should be given a low credit. If a host is found to misbehave, its credits are decreased by a very large amount. In a highly hostile environment, S should try to choose a route in which all hosts exhibit high credits.

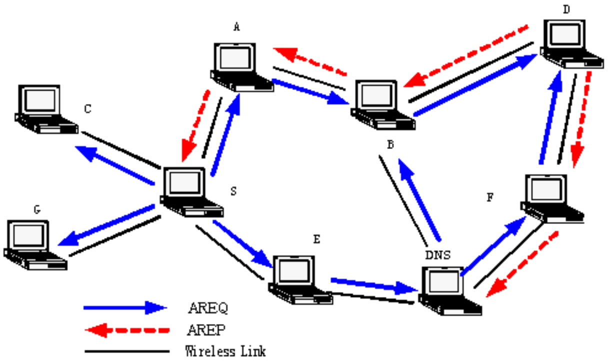

Figure 3 The secure route discovery, route reply,and cached route reply

Another frequently seen problem is the black hole problem, where a host simply accepts packets without forwarding them. Since hosts can not hide their identities in our protocol, the source host can traverse the route and test the integrality of each host. In this way, misbehave hosts are likely to be discovered.

4 Security Analysis

In this section, we discuss several possible attacks and how our protocol defends such attacks.

- Impersonation of DNS: A host may want to impersonate a DNS or replay DNS's earlier messages. Since we impose that every host knows DNS's public key prior to entering the MANET, such attacks can be easily defended by conventional authentication schemes (such as attaching a challenge in DNS query and response messages).
- Black hole attack: A malicious node may announce having good routes leading to all other hosts and thus attract all hosts choosing it as a relay node. When data packets arrive, the host may simply ignore them, thus causing the black hole problem. As discussed earlier, hosts can not easily hide their identities in our protocol. Further, with our credit management mechanism, such attacks are unlikely to succeed after the network is stable.
- Replayed or Forged AREP/DREP/RREP/CREP: Replaying AREP/DREP/RREP/CREP is unlikely because the attackers have to know how to encrypt either the challenge or the sequence number. An adversary can not forge a AREP/DREP/RREP/CREP because it does not know the private key of the host which it intends to pretend.
- Replayed or Forged RERR: Since we adopt source routing, a host can not easily forge a RERR unless it is a node in the routing path. Again, it has to present its identity to the source on reporting the RERR. In this case, the source has to accept this report because even if this is a false report, it still makes no sense to ask this

malicious host to relay packets. However, if the malicious host keeps on conducting such attacks, its identity will be tracked by the initiator. A replay of RERR is only possible after the corresponding route has been announced broken for at least once. In this case, replay attacks make no sense.

5 Conclusions

In this paper, we have proposed a secure bootstrapping and routing protocol for an open MANET that may be exposed to attacks. Our design bundles the generation of IPv6 addresses with the routing protocol so that a malicious node must present its identity when conducting routing attacks. Even though IPv6 allows a malicious host to keep on changing its identity, our credit management mechanism will discourage a host to choose routes passing low-credit or hostile areas. In particular, we adopt DSR as the basis of our routing protocol. This allows us to easily track the identities of misbehaving nodes. (Translating to other routing protocols is possible, but we may lose such tracking capability, which deserves further investigation.) Our protocol adopts DNS as the only security infrastructure in a MANET; therefore, maintaining a MANET would become an easy job. However, hosts do not need to always contact DNS unless it intends to play as a server with a permanent domain name.

Acknowledgement

This work is co-sponsored by the MOE Program for Promoting Academic Excellence of Universities under grant numbers A-91-H-FA07-1-4 and 89-E-FA04-1-4.

References

[1] Arkko, J., Nikander, P. and Mantyla, V., “Securing IPv6 Neighbor Discovery Using Crytographically Generated Address (CGAs),” *Internet Draft: draft-arrko-send-cga-00.txt*, Work in Progress, 2002.

[2] Bobba, R. B., Eschenauer, L., Gligor, V. and Arbaugh, W., “Bootstrapping Security Associations for Routing in Mobile Ad-Hoc Networks,” *Technical Report, TR2002-44*, Univ. of Maryland, 2002.

[3] Haas, Z. J., Pearlman, M. R. and Samar, P., “The Zone Routing Protocol (ZRP) for Ad Hoc Networks,” *Internet draft: draftietf-manet-zone-zrp-04.txt*, Work in Progress, 2002.

[4] Hu, Y. C., Perrig, A. and Johnson, D., “Ariadne: A secure on demand routing protocol for ad hoc networks,” *Proc. of the 8th ACM International Conference on Mobile Computing and Networking* , 2002.

[5] Hu, Y. C., Johnson, D. B. and Perrig, A., “Sead: Secure efficient distance vector routing in mobile wireless ad hoc networks,” *Proc. of Fourth IEEE Workshop on Mobile Computing Systems and Applications (WMCSA ’02)*, 2002, pp. 3–13.

[6] Johnson, D., Maltz, D. and Broch, J., “DSR: The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks,” *Ad Hoc Networking*, edited by Charles E. Perkins, chapter 5, Addison-Wesley, 2001, pp. 139–172.

[7] Kitamura, H., “Domain Name Auto-Registration for Plugged in IPv6 Nodes,” *Internet Draft: draft-ietf-dnsext-ipv6-nameauto-reg-00.txt*, Work in Progress, 2002.

[8] Lu, B. and Pooch, U., “Cooperative security-enforcement routing in mobile ad hoc networks,” *Proc. of the 4th International Workshop on Mobile and Wireless Communications Network*, 2002.

[9] Montenegro, G. and Castelluccia, C., “Statistically Unique and Cryptographically Verifiable (SUCV) Identifiers and Addresses,” *Proc. of 2002 Network and Distributed System Security Conference*, 2002.

[10] Narten, T., Nordmark, E. and Simpson, W., “Neighbor Discovery for IP Version 6 (IPv6),” *RFC2461*, 1998.

[11] Papadimitratos, P. and Haas, Z. J., “Secure Routing for Mobile Ad hoc Networks,” *Proc. of the SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002)*, San Antonio, TX, 2002.

[12] Park, J.-S., Kim, Y.-J. and Park, S.-W., “Stateless Address Autoconfiguration in Mobile Ad Hoc Networks using Site-local Address,” *Internet Draft: draft-park-zeroconf-manet-ipv6-00.txt*, Work in Progress, 2001.

[13] Park, S. D., Kim, P. and Kim, Y., “IPv6 Domain Name Auto-Registration (6DNAR),” *Internet Draft: draft-park-6dnar-01.txt*, Work in Progress, 2003.

[14] Perkins, C. and Bhagwat, P., “Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers,” *Proc. of ACM SIGCOMM’94 Conference on Communications Architectures, Protocols and Applications*, 1994, pp. 234–244.

[15] Perkins, C., Royer, E. and Das, S., “IP Address Autoconfiguration for Ad Hoc Networks,” *Internet Draft: draft-ietfmanetautoconf-01.txt*, Work in Progress, 2001.

[16] Perkins, C. and Royer, E., “Ad hoc on-demand distance vector routing,” *Proc. of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, New Orleans, LA, 1999.

[17] Thaler, D. and Hagino, J., “IPv6 stateless DNS

Discovery," *Internet Draft: draft-ietf-ipv6-dns-discovery-04.txt*, Work in Progress, 2002.

[18] Yi, S., Naldurg, P. and Kravets, R., "Security-aware ad hoc routing for wireless networks," *Proc. of ACM Int'l Symp. on Mobile ad hoc networking and computing*, 2001.

[19] Zapata, M. G., "Secure Ad hoc On-Demand Distance Vector (SAODV) Routing," *Internet Draft: draft-guerrero-manetsaodv-00.txt*, Work in Progress, 2002.

Biographies

Yu-Chee Tseng received his B.S. and M.S. degrees in Computer Science from the National Taiwan University and the National Tsing-Hua University in 1985 and 1987, respectively. He worked for the D-LINK Inc. as an engineer in 1990. He obtained his Ph.D. in Computer and Information Science from the Ohio State University in January of 1994. From 1994 to 1996, he was an Associate Professor at the Department of Computer Science, Chung-Hua University. He joined the Department of Computer Science and Information Engineering, National Central University in 1996, and has become a Full Professor since 1999. Since Aug. 2000, he has become a Full Professor at the Department of Computer Science and Information Engineering, National Chiao-Tung University, Taiwan.

Dr. Tseng served as a Program Chair in the *Wireless Networks and Mobile Computing Workshop*, 2000 and 2001, as a Vice Program Chair in the *Int'l Conf. on Distributed Computing Systems (ICDCS)*, 2004, as an Associate Editor for *The Computer Journal*, as a Guest Editor for *ACM Wireless Networks* special issue on "Advances in Mobile and Wireless Systems", as a Guest Editor for *IEEE Transactions on Computers* special on "Wireless Internet", as a Guest Editor for *Journal of Internet Technology* special issue on "Wireless Internet: Applications and Systems", as a Guest Editor for *Wireless Communications and Mobile Computing* special issue on "Research in Ad Hoc Networking, Smart Sensing, and Pervasive Computing", as an Editor for *Journal of Information Science and Engineering*, as a Guest Editor for *Telecommunication Systems* special issue on "Wireless Sensor Networks", and as a Guest Editor for *Journal of Information Science and Engineering* special issue on "Mobile Computing".

He is a recipient the Outstanding Research Award, 2001-2002, from the National Science Council, ROC, and a recipient of the Best Paper Award in Int'l Conf. on Parallel Processing, 2003. His research interests include mobile computing, wireless communication, network security, and parallel and distributed computing. Dr. Tseng is a Senior Member of the IEEE.

Jehn-Ruey Jiang received his Ph. D. degree in Computer Science in 1995 from National Tsing-Hua University, Taiwan. He joined Chung-Yuan Christian University as an Associate Professor in 1995. He is currently an Associate Professor of the Department of Information Management, Hsuan-Chuang University. He is a recipient of the Best Paper Award in Int'l Conf. on Parallel Processing, 2003. His research interests include distributed computing, mobile computing, distributed fault-tolerance, protocols for mobile ad hoc networks and wireless sensor networks.

Jih-Hsin Lee is a research assistant of Chung-Shan Institute of Science & Technology. He is also a master student of College of Electrical Engineering and Computer Science, National Chiao-Tung University, Hsin-Chu, Taiwan. His research interests include network administration and information system development.