DASSR: A Distributed Authentication Scheme for Secure Routing in Wireless Ad-hoc Networks

Phu H. Phung[1(✉)] and Quang Tran Minh[2]

[1] Department of Computer Science, University of Dayton, Dayton, OH, USA
phu@udayton.edu
[2] Faculty of Computer Science and Engineering,
Ho Chi Minh City University of Technology, VNU-HCM, Vietnam
quangtran@hcmut.edu.vn

Abstract. Secure routing is vital in wireless ad-hoc networks for establishing reliable networks and secure data transmission. However, most routing security solutions in wireless ad-hoc networks make assumptions about the availability of key management infrastructures that are against the very nature of ad-hoc networks. In this paper, we propose DASSR scheme, a new secure routing approach based on a fully distributed authentication and self-organized public key management scheme without any central authorizing entity. In DASSR, routing messages are authenticated between neighboring nodes (hop-by-hop) and between source and destination nodes (end-to-end) by using nodes' signatures. Once authenticated, messages are guaranteed for integrity and non-repudiation, hence the scheme could prevent potential routing attacks from malicious nodes. We evaluate our proposed scheme DASSR by applying it to the AODV routing protocol, a representative of reactive ad-hoc routing protocols, and demonstrate the effectiveness and security properties of the proposed approach. A comprehensive review of related secure routing protocols is presented and compared with the proposed scheme DASSR.

Keywords: Mobile · Wireless ad-hoc · Security · Secure routing protocol · Distributed · MANET

1 Introduction

A wireless ad-hoc network is a network of nodes, commonly mobile nodes, communicating to each other by self-organizing without a fixed or centralized infrastructure [7,26]. Mobile ad-hoc networks (MANETs) and wireless sensor networks (WSNs) are two instances of a wireless ad-hoc network that are widely deployed and used in practice such as in military, vehicular networks, disaster recovery [24], and many other domains. Wireless ad-hoc networks have been also integrated with Internet of Things (IoT) to carry out more powerful applications to real-world [2].

In a wireless ad-hoc network, the connectivity is ad-hoc in the sense that each node can create and join a network "on-the-fly" by performing basic networking

T.K. Dang et al. (Eds.): FDSE 2016, LNCS 10018, pp. 219–236, 2016.
DOI: 10.1007/978-3-319-48057-2_16

functions such as routing, forwarding, and service discovery. Nodes in a wireless ad-hoc network participate in routing processes to establish data forwarding policies for end-to-end communications. In order to realize multi-hop communications, effective multi-hop routing protocols, such as AODV [22], OLSR [25], HWMP [5], must be implemented in each node [6,31].

Ad-hoc network routing protocols mainly focus on providing the convenience for nodes to join the networks, improving collaborations between nodes in an end-to-end multi-hop communication fashion. These protocols normally assume that every node performs and follows the protocol and have not considered security aspects. However, there is no mechanism in a routing protocol to ensure that every node in an ad-hoc network follows the protocol. For example, a malicious node can modify a field in a routing message illegitimately to falsify the routing information in a network, which cannot be detected and prevented in wireless ad-hoc network standard routing protocols. This makes wireless ad-hoc networks be vulnerable with various security attacks including data forwarding attacks (e.g., denial of service, data fabrication, packet delay, data dropping and spoofing, etc.,) and network control attacks (route fabrication, making loop on networks, changing network topology) [1,3,16,29] (c.f. Sect. 2 for detailed attacks).

In addition, in the dynamic environments of ad-hoc networks, nodes are dynamically issue control messages for network establishment and management. Specifically, in the reactive routing protocol such as AODV [22], a node can issue control messages, e.g., for route request, whenever it wants to transfer data. This creates a great chance for malicious nodes to attack the network. For example, a malicious node can spoof the address of its neighbor nodes to send a false routing message to break an active routing path [1]. As a result, security for wireless ad-hoc network routing protocols is a great challenge attracting many researchers recently.

There have been a number of works proposing secure routing protocols in wireless ad-hoc networks [1,12,16,23,28,30,33,34]. Most routing security solutions such as SAODV [34], ARAN [28] assume the availability of key management infrastructures. This assumption is impractical in wireless ad-hoc networks as it violates the nature of this network architecture that does not have a fixed infrastructure. Furthermore, in our security analysis, there is no previous security scheme examining the integrity of transactions between neighbor nodes, which create security flaws for fabrication attacks. Some other works attempt to resolve this issue by proposing a cryptographic model [3] or a public-key scheme using MAC addresses on layer 2 based routing protocols [8,10]. However, the cryptographic approach needs a complex algorithm embedded in a routing protocol, while the later one cannot be able to work on layer 3 routing protocols which are commonly used in wireless ad-hoc networks.

In this work, we aim at filling the aforementioned gaps by proposing a simple yet efficient authentication scheme for secure routing based on a self-organized public-key mechanism in wireless ad-hoc networks. Our proposed scheme, namely DASSR (Distributed Authentication Scheme for Secure Routing), is different from previous approaches that it is fully distributed without requiring a trusted

server while still can defense the network against the identified attacks with low overhead. In summary, the main contributions of this paper are:

- A fully distributed authentication scheme is proposed to ensure the integrity and non-repudiation of routing messages between neighbors' nodes (hop-by-hop communications), and between the source and destination nodes (end-to-end communications) in wireless ad-hoc networks. The proposed distributed scheme is based on a key exchange and a revised self-organized public-key mechanism without a certification server.
- The proposed approach does not rely on MAC addresses for identifying public keys so that it can work on layer 3 protocols which are commonly used in ad-hoc networks. An application of the proposed scheme DASSR on AODV, a reactive routing protocol has been performed to illustrate the effectiveness of the approach.
- The proposed authentication method is based on signatures, which is simple for implementation and introduces low overhead compared to hashing or cryptography-based counterparts.
- A deep security analysis is performed to demonstrate that the proposed scheme DASSR can prevent identified attacks. We also present a comprehensive review of existing solutions and compare their security properties with DASSR.

The rest of this paper is organized as follows. Section 2 presents background of this work including routing protocols in wireless ad-hoc networks, security flaws in routing protocols, and examines current approaches to securing routing protocols. In Sect. 3, we detail our proposed distributed authentication scheme DASSR and key exchange for reactive routing protocols. We present an implementation of the proposed scheme in the AODV routing protocol in Sect. 4. Section 5 analyzes the security properties and advantages of the proposed scheme DASSR, and performs a comparison of DASSR and related secure routing schemes. We conclude our contributions and address further work in Sect. 6.

2 Background and Related Work

In this section, we review routing protocols in wireless ad-hoc networks, analyze and discuss the security issues in the existing routing protocols, and present related work.

2.1 Routing in Wireless Ad-Hoc Networks

In wireless ad-hoc networks, network topology is dynamically changed with frequent joining or moving out of mobile nodes. To realize multi-hop communications, effective multi-hop routing protocols must be implemented in each node. Two main branches of routing protocols, namely the proactive and reactive routing protocols have been proposed. The proactive protocols statically build routing tables for mobile nodes in advance (before the routes/paths are used) and periodically update those routing tables. This approach is suitable for small

networks, but it is inefficient for large networks involving a huge number of control packages traveling through. Representatives of proactive routing protocols are OLSR (Optimized Link State Routing Protocol) [25], DSDV (Destination-Sequenced Distance-Vector Routing Protocol) [17], STAR (Source-Tree Adaptive Routing) [11]. In contrast, the reactive counterparts, such as AODV (Ad hoc On-Demand Distance Vector Routing Protocol) [22], DSR (Dynamic Source Routing Protocol) [13,15], DYMO (Dynamic Manet on Demand Routing Protocol) [18], TORA (Temporally Ordered Routing Algorithm) [19], examine routes on-demand when a node needs a route/path for data forwarding.

With the flexibility nature in ad-hoc networks as mobile nodes can actively issue control messages to establish routing processes for data forwarding, specifically in reactive routing protocols, the networks are exposed to attacks by malicious nodes. Typical types of attacks are described in the following sub-section.

2.2 Typical Attacks in Wireless Ad-Hoc Routing Protocols

As mentioned earlier, security issues have not been considered in ad-hoc network routing protocols. Any node can join a network, and read, forward, and send routing messages to neighbor nodes in a network without authentication. This design allows malicious nodes to launch serious attacks. In this subsection, we detail three types of attacks that are common in reactive routing protocols in wireless ad-hoc networks. The proposed scheme DASSR aims to detect and prevent these attack types.

Impersonation Attacks. A malicious node misrepresents its identity in the network so that it will break route discovery or path maintenance processes. A malicious node listens to its neighbor nodes to identify their identities and then modify its identity such as MAC or IP address in outgoing packets to generate falsified routing information: (1) a malicious node impersonates the source node, (2) a malicious node impersonates the destination node or neighbor of destination by forging a `Route Reply` with its address as a destination node, and (3) a malicious node forms a loop by spoofing nodes to change an existing route to a circle so that the message is relayed in the loop continuously without reaching the real destination.

Modification Attacks. When a malicious node is in the route discovery path, it might modify the route request or route reply. As a consequence, the discovered path causes the source node to transmit data wrong. The modification can happen for the following things: (1) route sequence numbers and (2) hop count. As for case (1), when a malicious node $\boldsymbol{M}$ receives a route request, that is destined to node $\boldsymbol{D}$ from source node $\boldsymbol{S}$, from its neighbor node $\boldsymbol{N}$, the malicious node $\boldsymbol{M}$, after re-broadcasting the message, redirects transactions toward itself by unicasting to $\boldsymbol{N}$ a `Route Reply` containing a much higher destination sequence number for $\boldsymbol{D}$ than the value last advertised by $\boldsymbol{D}$. In consequence, on receiving valid `Route Reply` from $\boldsymbol{D}$, $\boldsymbol{N}$ will discard this message. As for case

(2), malicious nodes can modify the hop count field of a `Route Request` message by resetting this value to zero or setting this value to infinity. This modification leads the route discovery process wrong.

Fabrication Attacks. In reactive routing protocols in ad-hoc network, when a node in an active path moves, the path is broken. Routing protocols such as AODV has a route maintenance mechanism to recover such broken paths. This is implemented by the node upstream of the broken link, broadcasting a `Route Error` message to all active upstream neighbors [28]. However, this mechanism is vulnerable as a malicious node may falsify an existing route by generating a `Route Error` message that in fact is not true, resulting in a denial-of-service attack in the network as nodes receiving falsified `Route Error` message cannot verify the correctness and thus delete the active path.

2.3 Related Work

In the following paragraphs we summarize related work for security in wireless ad-hoc networks.

Data Forwarding Security. Several works have been dedicated for data forwarding security in distributed systems like wireless sensor networks and MANETs, where there is no centralized element to manage the security policy. Rezvani *et al.*, proposed a collaborative-based reputation method to which the credibility of each node is evaluated by other nodes in the network [27]. Based on the credibility, the trustworthiness of mobile nodes is measured. This allows the network to detect malicious or untrusted nodes, protecting network nodes from receiving data from attackers. The accuracy of the propagated credibility is validated using the variances of sensors whereby the distribution of noise in sensors is modeled by Gaussian distribution which is not always correct in the real wireless environments. In addition, the credibility propagation may also include judgments from untrusted/malicious nodes.

Routing Security. Beside data forwarding security, because of its nature, wireless ad-hoc network is significantly vulnerable with routing security as routing establishment and management are essential and these processes are conducted frequently. Various researches have been dedicated for routing security methods which mainly rely on cryptography [3]. Ben-Othman *et al.*, proposed an Identity Based Cryptography (IBC) method for node identity in the Hybrid Wireless Mesh Protocol (HWMP) [8–10] for IEEE 802.11s mesh network [5]. As HWMP is a layer 2 routing protocol, MAC addresses of mobile nodes are used as the public keys for the control messages such as route request (RREQ), route reply (RREP). As a result, this approach does not need a centralized entity to verify the authentication of public keys. Therefore, it is suitable for security routing in infrastructure-less ad-hoc networks. The essential issues in this method, however,

are that (i) IEEE 802.11s is not the only standard protocol for ad-hoc networks, meanwhile more commonly used routing protocols work on layer 3 where IP address is used instead of MAC address; (ii) theoretically, MAC address can also be faked by malicious nodes thus the system needs a secured scheme to protect MAC address fabrication. Our work in this paper is different which focuses on layer 3 secure routing protocols which are commonly used in ad-hoc networks.

In another aspect, there have been a number of solutions for securing routing protocols working on layer 3 in wireless ad-hoc networks such as [1,4,12,16,20,23,28,30,33,34]. These solutions both have advantages and disadvantages. The most common disadvantage is that they assume a fixed infrastructure, which is against the nature of ad-hoc networks, and is complex to implement in practice. In a previous work [23], we proposed a hash-based authentication scheme among two nodes to authenticate the messages without introducing a fixed infrastructure. While that approach can ensure the integrity of messages, it still be open to fabrication attacks as there is no end-to-end authentication between the source and destination nodes. The proposed scheme DASSR in our work overcomes these weaknesses by introducing a fully distributed authentication mechanism without a fixed server while providing end-to-end authentication. We present DASSR in detail in the next section and perform a comprehensive comparison of the proposed scheme DASSR and related solutions in Sect. 5.2.

3 Distributed Authentication Scheme for Reactive Routing Protocols

Reactive routing protocols demonstrate the effectiveness in wireless ad-hoc networks as it works on-demand, reducing the broadcasting messages for updates. However, this feature makes the network vulnerable to attacks since there is no mechanism to authenticate the messages from neighbor and source nodes. Without authentication, reactive routing protocols are vulnerable to three main attack categories: impersonation attacks, modification attacks, and fabrication attacks as analyzed in Sect. 2. Our approach to preventing these potential attacks is to authenticate all messages in a routing protocol. Using authentication, routing messages are guaranteed two main properties:

Integrity. This property ensures that the content of routing messages from an untrusted node cannot be altered or modified by malicious/unauthorized nodes thanks to the signature verification. The integrity of routing messages are guaranteed by a hop-to-hop and end-to-end authentication mechanism.

Non-repudiation. Routing messages are signed using a private key by the sending node, and will be validated by the receiver using public key of the sender. The successful validation guarantees the non-repudiation of the messages, which ensures that the messages are sent by the node signed the messages and cannot be spoofed by other nodes.

The authentication process in DASSR is performed in 2 steps: hop-by-hop authentication at intermediate nodes and end-to-end authentication at the destination node. The authentication uses RSA Public-key crypto-system: messages

will be signed by the sender using its private key and verified by the receiver using the sender's public key. The original message with signature from a sender will be forwarded to the destination receiver so that the receiver can verify its integrity. Thus in this scheme, each node needs to store a public-key repository for the authentication process. In the following subsections, we present the process in detail.

3.1 Overview of the Proposed Scheme

The overview of the proposed scheme DASSR is depicted in Fig. 1 and explained as follow. Before a source node $\boldsymbol{S}$ sends/broadcasts a routing message $\mathcal{M}$ according to a routing protocol, it first signs $\mathcal{M}$ with its private key to create a signature $signature\_S$ and attach to $\mathcal{M}$. Then $\boldsymbol{S}$ broadcasts the signed message [$\mathcal{M}$,$signature\_S$]. When its neighbor node $\boldsymbol{n}$ receives the signed message, $\boldsymbol{n}$ uses the public key of $\boldsymbol{S}$ to verify $signature\_S$. If the verification succeeds and $\boldsymbol{n}$ is not the destination, it additional signs the message with its private key then forwards the double signed message [$\mathcal{M}$,$signature\_S$,$signature\_n$] further. At any intermediate node $\boldsymbol{i}$, once it receives a double signed message, it verifies the second signature (which is signed by a neighbor node). The verification ensures the integrity of the message from the neighbor node. If the destination address does not match with $\boldsymbol{i}$'s address, it signs the message and generates its signature $signature\_i$, then replaces the $signature\_n$ by its own signature $signature\_i$, and finally forwards the new double signed message further. This process is similar at a destination node $\boldsymbol{d}$ except when checking if the destination address matches with $\boldsymbol{d}$'s address, $\boldsymbol{d}$ needs to verify the signature of $\boldsymbol{S}$ $signature\_S$.

In summary, routing messages are authenticated among intermediate nodes (hop-by-hop) and from the source to destination nodes (end-to-end authentication). The authentication is based on signature using RSA Public-key Cryptosystem [14]. Thus, each node in a network generates its own pair of public key $\boldsymbol{PuK}$ and private key $\boldsymbol{PrK}$. For hop-by-hop authentication, each node keeps track of a list of neighbor nodes and for each neighbor, maintains its neighbors'

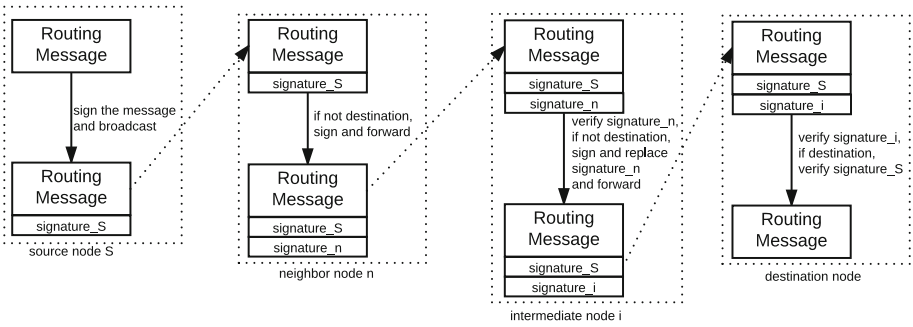

Fig. 1. Overview of the proposed distributed authentication scheme DASSR for secure routing.

address and public key. For end-to-end authentication, a destination node needs to keep the public key of the sender in order to verify the signature. The big challenge in this scheme is how each node can keep public keys of other nodes for the authentication. Using or assuming a centralized element or certification server to distribute public keys or certifications is not suitable for wireless ad-hoc networks. Our approach for this issue is that each node when joining the network first exchange its public key with the neighbor nodes. Each node in the network exchanges its public key repository to neighbor nodes so that eventually, any node in the network will have public keys of the other nodes in a self-organized and distributed manner without having a central element. These steps are presented in detail below.

3.2 Public Key Exchange Process Between Neighbor Nodes

A node that wants to join a network sends a join request message to its neighbors to exchange their `public keys`. The node broadcasts a message requesting the key exchange to its neighbor nodes. The receiving node responds with a join reply message that includes its public key. The pseudo-code algorithm is given in Fig. 2.

Sender i:

```
1. Generate RSAPairKey = (PuK, PrK);
2. Broadcast Join.Req = (AGREEMENT_REQ, request_id, sender_addr, PuK);
```

Receiver j:

```
1. Receive a message;
2. If packetType == AGREEMENT_REQ then
   Send Join.Rep =
       (AGREEMENT_REP, request_id, sender_addr, neighbor_addr, PuK);
Else if packetType == AGREEMENT_REP then
   Store it to public key list;
```

Fig. 2. Public key exchange process between neighbor nodes

A sender ***i*** before joining the network first generates its RSA private and public key pair ***PuK, PrK***, then it broadcasts a join message with the packet type ***AGREEMENT_REQ***, together with the request id, its address and public key ***PuK***.

Once a node receives a message with a packet type ***AGREEMENT_REQ***, it will unicast back the key exchange agreement message with the packet type ***AGREEMENT_REP***, together with the request id, its address, and the neighbor address and its public key ***PuK***. If the packet type is ***AGREEMENT_REP***, it will extract the neighbor public key and store in its neighbors' public key list.

3.3 Public Key and Certificate Repository Exchange

In our proposed scheme DASSR, an end-to-end authentication must be performed to ensure the integrity and non-repudiation of the original message from the source node. Therefore, a node must keep the public key of other nodes in the network to be able to verify the signature of the source node. To this end, we adopt and revise self-organized public-key management scheme proposed by Capkun *et al.,* [32], which is suitable for wireless ad-hoc networks as the management scheme does not rely on any trusted authority or fixed server.

Overview of Self-organized Public-Key Management Scheme [32]. This public key management scheme works based on the following principle:

> *"If a user* $\boldsymbol{u}$ *believes that a given public key* $\boldsymbol{PuK\_v}$ *belongs to a given user* $\boldsymbol{v}$*, the user* $\boldsymbol{u}$ *can issue a public-key certificate in which* $\boldsymbol{PuK\_v}$ *is bound to* $\boldsymbol{v}$ *by the signature of* $\boldsymbol{u}$*."*

Based on that principle, nodes that receive the newly issued certificate from a neighbor add it to their own certificate graph and further distribute the updated certificate graph. When a node $\boldsymbol{u}$ wants to verify the authenticity of another node $\boldsymbol{v}$'s public key, it merges its local certificate repositories and then evaluates the authenticity of $\boldsymbol{PuK\_v}$ from the merged repository.

The scheme also provides the way that detects misbehaving users and resolves the conflicting certificates during operation. The solution requires users conscious involvement in creating their public/private key pairs and issuing certificates; all other operations (including certificate exchange and construction of certificate repositories) are fully automatic.

Revised Scheme for Public Key and Certificate Repository Exchange. We adopt and revise the aforementioned self-organized public key management scheme to apply in our distributed authentication scheme to ensure end-to-end authentication. The modified scheme is detailed below.

Nodes exchange their certificate graph and construct their updated certificate repository by following the certificate exchange process given in [32]. The scheme is applied to our scheme with a modification in which after certificate repository exchange, nodes perform public key exchange with its neighbors.

Upon receiving a public key exchange request, a node validates the public key by looking up its certificate repository. If found, it exchanges the public key and then store the tuple (node id, public key) in its trusted neighbor list. If not found, it waits for the convergence time T_{CE} (that is the expected time after which, when issued, a certificate reaches all the nodes in the network [32]) and then looks up its latest updated certificate graph again. If still not found, it refuses to exchange public key.

Through this exchange process, a node will have a up-to-date certificate repository and the list of trusted neighbors' public-keys. The certificate graph

is used for authentication of end-to-end transactions in a network; whereas the list of neighbors public-key is used for hop-by-hop authentication.

According to this scheme, at least one node in the network has to issue the certificate for inclusion of a new node. If the certificate is issued, its public-key certificate is distributed throughout the network during the convergence time. Otherwise, it means that no node in the network know the new node. Thus, the new node is not allowed to take part in the further networking activities.

4 An Implementation of the Proposed Scheme DASSR for the AODV Routing Protocol

To demonstrate how our proposed authentication scheme works in a particular reactive routing protocol, we deploy the scheme in the AODV (Ad hoc On-Demand Distance Vector) routing protocol. In this section, we first review the AODV protocol, then we present a secure AODV protocol using our distributed authentication scheme DASSR.

4.1 AODV Protocol

As mentioned earlier, AODV protocol is a routing protocol for wireless ad-hoc networks using a reactive routing approach, which does not keep every node in the network on a routing table but builds a path on-demand. The routing protocol has three main different packets: `Route Request` ($\mathcal{RREQ}$), `Route Reply` ($\mathcal{RREP}$), and `Route Error` ($\mathcal{RERR}$) [21,22]. When a node wants to send a message to a destination that is not cached in the routing table, it issues a `Route Request` ($\mathcal{RREQ}$). When a node receives a `Route Request`, it forwards further or issues a `Route Reply` if it is the destination node or it has a fresh-enough route to the destination. When a node issues a `Route Reply`, it constructs $\mathcal{RREP}$ message and unicasts back to the neighbor node in the reverse path. A `Route Error` message will be issued and broadcasted when there is an error in a discovered path.

Similar to other routing protocols in wireless ad-hoc networks, AODV was designed without security consideration. Hence it is also vulnerable to typical attacks such as impersonation, modification, fabrication attacks presented in Sect. 2.2. In the next subsection, we present the implementation of our distributed authentication scheme DASSR in AODV to secure the routing protocol.

4.2 Secure AODV Using the Proposed Distributed Authentication Scheme DASSR

In the following revised AODV protocol, we assume that the public key exchange process presented in the previous section have been performed and completed. Thus, each node has neighbor nodes' public keys and a certificate repository of other active nodes in the network.

Route Request. A node having a packet to send, so-called a source node $\boldsymbol{S}$, initiates a $\mathcal{RREQ}$ message. Eventually, this message arrives at the destination node through the forwarding of zero or more intermediate nodes. In our scheme DASSR, the source node $\boldsymbol{S}$ attaches its signature signed with its private key $\boldsymbol{PrK\_S}$ to the $\mathcal{RREQ}$ message as follows:

```
Message = (RREQ, signature_S)

where
signature_S = [bcastId, destAddr, destSeq, srcAddr, srcSeq]PrK_S
```

Note that the signature *signature_S = [bcastId, destAddr, destSeq, srcAddr, srcSeq]PrK_S* indicates that it is signed by the private key $\boldsymbol{PrK\_S}$ on the content in [..], which are broadcast ID, destination address, destination sequence number, source address, source sequence number. These are non-mutable fields in a $\mathcal{RREQ}$ message. The content is not encrypted so that any receiving node can read to perform the routing protocol.

On receiving $\mathcal{RREQ}$ with a single signature from the source node $\boldsymbol{S}$, a neighbor node $\boldsymbol{n}$ first assures the integrity of the message by validating the source's signature with the source's public key. If the message is valid, the node continues the steps in the AODV routing protocol such as updating the hop count in the $\mathcal{RREQ}$.

The neighbor node $\boldsymbol{n}$ generates its own signature and appends this signature to the message before forwarding. The new message contains:

```
Message = (RREQ, signature_S, signature_n)
where
signature_n = [bcastId, destAddr, destSeq, srcAddr, srcSeq, hopcount]PrK_n
```

Continuing with this revised AODV protocol, any intermediate node $\boldsymbol{i}$ (except the neighbor node $\boldsymbol{n}$ which is one-hop from the source node) will receive a double signed $\mathcal{RREQ}$ message. Upon receiving the double signed $\mathcal{RREQ}$ message, node $\boldsymbol{i}$ validates the signature of the forwarding node only. This authentication process follows a hop-by-hop authentication that uses the exchanged and trusted public-keys. If the validation succeeds, node $\boldsymbol{i}$ signs the $\mathcal{RREQ}$ message similar to node $\boldsymbol{n}$ above, and replaces the forwarding node's signature with its own signature, and then rebroadcasts the message:

```
Message = (RREQ, signature_S, signature_i)
```

The signatures of intermediate nodes help preventing spoofing attacks. In this way, the authentication process is performed in a hop-by-hop manner based on the list of trusted neighbors' public-keys, without accessing the local certificate repository.

Repeating this procedure, the authenticated $\mathcal{RREQ}$ message arrives at the destination node. Note that all intermediate nodes do not validate the signature of the source node $\boldsymbol{S}$ in our scheme, except the neighbor node $\boldsymbol{n}$ and the destination node or an intermediate node initiating a `Route Reply`. At the destination, it first validates the forwarding (neighbor) node's signature and then validates

the signature of the source node $signature\_S$. If the validation succeeds, the message is ensured its integrity (content is unaltered by unauthorized nodes) and non-repudiation (it was actually sent by the source node $\boldsymbol{S}$). The destination node validates the signature of its neighbor node as the same procedure as that in the intermediate nodes presented in the previous section.

For the signature of the source node, after successfully authenticating the neighbor nodes signature, the destination node verifies the signature of the source node to authenticate the original route request from the source node. In our public key exchange mechanism presented in Sect. 3.3, the destination node should have the public key of the source node. Thus, the destination node of a $\mathcal{RREQ}$ can validate the signature of the source node. If the validation is successful, the $\mathcal{RREQ}$ message is guaranteed the integrity and non-repudiation from source node to destination node and among intermediate nodes. This end-to-end authentication process can prevent the modification attacks and impersonation attacks that cannot been solved in a hop-by-hop authentication method such as in [23].

In the case that the destination node cannot find the public-key of the source node in its repository, it still can apply the authentication process proposed in [32]. According to [32], when a user $\boldsymbol{u}$ wants to authenticate a public key $\boldsymbol{PuK\_v}$ of another user $\boldsymbol{v}$, both nodes merge their updated certificate repositories and $\boldsymbol{u}$ tries to find a certificate chain to $\boldsymbol{v}$ in the merged repository. If the certificates are both valid and correct, $\boldsymbol{u}$ authenticates $\boldsymbol{PuK\_v}$. Here again, $\boldsymbol{u}$ performs the certificate correctness check locally. If node $\boldsymbol{u}$ cannot find any certificate chain to $\boldsymbol{PuK\_v}$, it aborts the authentication.

Route Reply. In the AODV routing protocol, a route reply message ($\mathcal{RREP}$) is initiated by either the destination or intermediate nodes which have a fresh-enough route to the destination.

In this secure AODV protocol, the node initiating a route reply $\mathcal{RREP}$ message signs the message by its own private key and unicasts back to the neighbor node in the reverse path. The neighbor node of the initiating node validates the signature of source node (physical neighbor) and then attaches its signature to the message and forwards back to the next hop in the reverse path. Each node along the reverse path back to the source, on receiving the $\mathcal{RREP}$ message, validates the signature of the senders by using their trusted neighbors public key list, replaces the signature of neighbor node by its own one and forwards back to the next hop. When the source node receives the $\mathcal{RREP}$ message, it validates the two signatures. This process is similar to the destination node validates the $\mathcal{RREQ}$ message presented previously.

Route Error. `Route Error` ($\mathcal{RERR}$) message in the route maintenance process is another target for attacks; hence, it needs to be authenticated. The procedure for authentication of route error is the same as $\mathcal{RREP}$ authentication process.

In route reply and route error processes, if all validations succeed, the $\mathcal{RREP}$ message is guaranteed the integrity and non-repudiation for end-to-end transactions; therefore, this solution could prevent possible attacks mentioned above.

5 Evaluation and Comparison

5.1 Security Analysis of the Proposed Scheme DASSR

As presented and discussed above, our proposed authentication scheme DASSR is fully distributed without any fixed server. Since the validation of signatures does not need any central server, the authentication process imposes less overhead on the network because it does not need to communicate with a server for verification. In addition, the messages themselves are not encrypted, thus reduce the computation overhead at nodes.

As discussed in Sect. 3, using the message authentication in DASSR scheme, the integrity and non-repudiation of routing messages are guaranteed among nodes that include source, destination, and intermediate nodes. The integrity of messages ensures that the content of messages is unaltered by a malicious node. The non-repudiation guarantees that a received message came from the node did construct and sent the message, a malicious node cannot spoof another node to send a message thanks to signature verification. Therefore, our DASSR scheme can prevent potential routing attacks including impersonation, fabrication, and modification attacks. We present the detailed analysis as follows.

Impersonation attacks: By using hop-by-hop and end-to-end signature validation, our DASSR scheme can prevent any malicious node from spoofing the MAC or IP address of other nodes. If a malicious node constructs a falsified routing message using a spoofed address, the signature validation is failed because the address does not match with its public key thanks to the signature. If the signature validation is failed, the received messages are dropped.

Fabrication attacks: Any malicious node can generate a wrong *route error* message to falsify the network. However, our DASSR scheme authenticates any type of message in the network. Therefore, malicious nodes cannot spoof other nodes address to falsify route errors. Nevertheless, any trusted node can initiate wrong information to do the network harm. Since the scheme ensures the non-repudiation of messages, a trusted node that continues to inject false messages into the network can be detected and thus deleted from trusted list of neighbors, being excluded from future routing activities.

Modification attacks: Modifications such as source ID or destination sequence number are detected by the end-to-end authentication. However, the falsified modification of hop-count field can not be detected. For this case, we just rely on the transitively trusted relationship in which all nodes in the network are trusted directly or indirectly via some other nodes.

5.2 Comparison

In this subsection, we review the state-of-the-art on secure *reactive* routing protocols in wireless ad-hoc networks and compare our DASSR scheme with the existing secure routing protocols in literature.

ARAN [28]. ARAN (Authenticated Routing for Ad-hoc Networks) uses cryptographic certificates to achieve authentication, message integrity and non-repudiation in the route discovery process. It assumes the existence of a trusted certificate server which forms a center element.

SAODV [34]. SAODV (Secure Ad-hoc On-Demand Vector) routing protocol guarantees security based on a key management scheme in which each node must have certificated public keys of all nodes in the network. This protocol uses public key distribution approach. Therefore, it is difficult to deploy and it costs high since it requires both asymmetric cryptography and hash chains in exchanging messages.

OSR [4]. OSR, stands for On-demand Secure Routing Protocol Resilient to Byzantine Failures, floods route request and reply messages to prevent Byzantine failures. It uses digital signature to authenticate the source, however it requires a public key infrastructure.

Ariadne [12]. Ariadne, stands for Secure On-Demand Routing Protocol, provides point-to-point authentication of routing messages using MAC (Message Authentication Code) based on a shared key between two nodes. It assumes that sender and receiver establish the shared key before exchanging routing messages.

IBC [10]. IBC (Identity Based Cryptography) uses MAC addresses and cryptography to secure routing messages. MAC addresses are used as public keys, therefore, the mechanism does not require a centralized entity. However, as discussed earlier, this protocol is applicable for layer 2 routing protocols while more commonly used routing protocols work on layer 3 where IP address is used.

ESARP [33]. ESARP (Efficient Security Aware Routing Protocol) uses an asymmetric encryption to encrypt routing messages. It uses a key exchange scheme to distribute public keys so that it does not require a centralized server. However, the encryption introduce high overhead in computation.

In summary, existing schemes for secure routing are either based on the assumptions of the availability of key management infrastructures, which are against the very nature of ad-hoc networks [4,12,28,34], or not applicable for every ad-hoc network protocols [10], or high overhead due to complex cryptographic algorithms [10,33]. Our scheme DASSR authenticates routing messages

Table 1. Comparison of secure routing protocols with DASSR.

Scheme	Security	Verification mechanism	Fixed infrastructure required
ARAN [28]	Encryption	Public Key Cryptography	Trusted certificate server
SAODV [34]	Authentication	Digital Signature	Key Distribution System
OSR [4]	Authentication	Digital Signature	Public Key Infrastructure
Ariadne [12]	Authentication	MAC[a]	Key Distribution Center
IBC [10]	Encryption	Cryptography	None
ESARP [33]	Encryption	Cryptography	None
DASSR	Authentication	Digital Signature	None

[a] Message Authentication Code.

using digital signature without encryption, therefore it creates less overhead. The public key exchange in DASSR is fully distributed without any centralized element or fixed infrastructure. DASSR can prevent identified routing attacks in ad-hoc networks as analyzed in Sect. 5.1. Table 1 shows a comprehensive comparison of our DASSR scheme compared with existing secure routing solutions.

6 Conclusion and Future Work

In this work, we proposed DASSR, a fully distributed hop-by-hop and end-to-end authentication scheme for reactive routing protocols in wireless ad-hoc networks. Its advantages are two-fold: (1) It uses an efficient hop-by-hop authentication scheme, which prevents impersonation, modification, and fabrication attacks, during path discovery without resorting to any central entity. (2) The proposed scheme also provides an end-to-end authentication mechanism by adapting a self-organized public-key management scheme. In this way, our DASSR scheme can ensure the integrity and non-repudiation of original messages from the source node, thus can prevent modification attacks without relying on a certificate server. Our scheme can work on layer 3 protocols (as it does not rely on MAC addresses) which are widely used in wireless ad-hoc networks. We demonstrate the security properties and the effectiveness of the proposed scheme by deploying it to the AODV protocol, a representative of reactive ad-hoc network routing protocols. Secure routing protocols adopted the proposed scheme DASSR do not use cryptography or rely on a central server, therefore the overhead is low. In the future work, we will implement the DASSR scheme in other reactive routing protocols and compare its overhead and performance with other related secure routing protocols to confirm the effectiveness as well as the efficiency of the proposed approach.

Acknowledgments. This material is based on research sponsored partially by grants from University of Dayton Research Council and the Swedish Research Council (through Chalmers University of Technology). The authors would also like to thank the three anonymous reviewers for their helpful feedbacks.

References

1. Abdelaziz, A.K., Nafaa, M., Salim, G.: Survey of routing attacks and countermeasures in mobile ad hoc networks. In: 15th International Conference on Computer Modelling and Simulation (UKSim 2013), pp. 693–698, April 2013
2. Alcaraz, C., Najera, P., Lopez, J., Roman, R.: Wireless sensor networks and the internet of things: do we need a complete integration. In: 1st International Workshop on the Security of the Internet of Things (SecIoT 2010). IEEE, December 2010
3. Andel, T.R., Yasinsac, A.: Surveying security analysis techniques in MANET routing protocols. IEEE Commun. Surv. Tutorials **9**(4), 70–84 (2007)
4. Awerbuch, B., Holmer, D., Nita-Rotaru, C., Rubens, H.: An on-demand secure routing protocol resilient to byzantine failures. In: Proceedings of the 1st ACM Workshop on Wireless Security, WiSE 2002, NY, USA, pp. 21–30 (2002). http://doi.acm.org/10.1145/570681.570684
5. Bahr, M.: Proposed routing for IEEE 802.11s WLAN mesh networks. In: Proceedings of the 2nd Annual International Workshop on Wireless Internet, WICON 2006, NY, USA (2006). http://doi.acm.org/10.1145/1234161.1234166
6. Bakht, H.: Survey of routing protocols for mobile ad-hoc network. Int. J. Inf. Commun. Technol. Res. **1**(6), 258–270 (2011)
7. Baryun, A., Al-Begain, K., Villa, D.: A hybrid network protocol for disaster scenarios. In: Fifth IEEE International Conference on Next Generation Mobile Applications, Services and Technologies, pp. 129–136, September 2011
8. Ben-Othman, J., Benitez, Y.I.S.: On securing HWMP using IBC. In: 2011 IEEE International Conference on Communications (ICC), pp. 1–5, June 2011
9. Ben-Othman, J., Mokdad, L., Benitez, Y.I.S.: Performance comparison between IBC-HWMP and Hash-HWMP. In: Global Telecommunications Conference (GLOBECOM 2011), pp. 1–5. IEEE, December 2011
10. Ben-Othman, J., Saavedra Benitez, Y.I.: IBC-HWMP: a novel secure identity-based cryptography-based scheme for hybrid wireless mesh protocol for IEEE 802.11s. Concurr. Comput. Pract. Exp. **25**(5), 686–700 (2013). http://dx.doi.org/10.1002/cpe.1813
11. Garcia-Luna-Aceves, J.J., Spohn, M.: Source-tree routing in wireless networks. In: Seventh International Conference on Network Protocols (ICNP 1999), pp. 273–282, October 1999
12. Hu, Y.C., Perrig, A., Johnson, D.B.: Ariadne: a secure on-demand routing protocol for ad hoc networks. Wirel. Netw. **11**(1–2), 21–38 (2005). http://dx.doi.org/10.1007/s11276-004-4744-y
13. Johnson, D.B.: Routing in ad hoc networks of mobile hosts. In: The Workshop on Mobile Computing Systems and Applications, pp. 158–163. IEEE Computer Society (1994)
14. Jonsson, J., Kaliski, B.: Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1 (2003)

15. Kanthe, A.M., Simunic, D., Prasad, R.: Comparison of AODV and DSR on-demand routing protocols in mobile ad hoc networks. In: 1st International Conference on Emerging Technology Trends in Electronics, Communication and Networking (ET2ECN 2012), pp. 1–5, December 2012
16. Karlof, C., Wagner, D.: Secure routing in wireless sensor networks: attacks and countermeasures. Ad Hoc Netw. **1**, 293–315 (2003)
17. Khan, K.U.R., Zaman, R.U., Reddy, A.V., Reddy, K.A., Harsha, T.S.: An efficient DSDV routing protocol for wireless mobile ad hoc networks and its performance comparison. In: Second UKSIM European Symposium on Computer Modeling and Simulation (EMS 2008), pp. 506–511, September 2008
18. Kum, D.W., Park, J.S., Cho, Y.Z., Cheon, B.Y.: Performance evaluation of AODV and DYMO routing protocols in MANET. In: 7th IEEE Consumer Communications and Networking Conference (CCNC 2010), pp. 1–2, January 2010
19. Kuppusamy, P., Thirunavukkarasu, K., Kalaavathi, B.: A study and comparison of OLSR, AODV and TORA routing protocols in ad hoc networks. In: 3rd International Conference on Electronics Computer Technology (ICECT 2011), vol. 5, pp. 143–147, April 2011
20. Lee, Y.H., Kim, H., Chung, B., Lee, J., Yoon, H.: On-demand secure routing protocol for ad hoc network using ID based cryptosystem. In: Proceedings of the Fourth International Conference on Parallel and Distributed Computing, Applications and Technologies, PDCAT 2003, pp. 211–215, August 2003
21. Perkins, C., Belding-Royer, E., Das, S.: Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561 (Experimental) July 2003. http://www.ietf.org/rfc/rfc3561.txt
22. Perkins, C.E., Royer, E.M.: Ad-hoc on-demand distance vector routing. In: Proceedings of Second IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 1999), pp. 90–100, February 1999
23. Phu, P.H., Yi, M., Kim, M.-K.: Securing AODV routing protocol in mobile ad-hoc networks. In: Hutchison, D., Denazis, S., Lefevre, L., Minden, G.J. (eds.) IWAN 2005. LNCS, vol. 4388, pp. 182–187. Springer, Heidelberg (2009). doi:10.1007/978-3-642-00972-3_15
24. Quang, T.M., Yoshitaka, S., Cristian, B., Shigeki, Y.: On-site configuration of disaster recovery access networks made easy. Ad Hoc Netw. **40**, 46–60 (2016). Elsevier
25. Rousseau, S., Benbadis, F., Lavaux, D., San L.: Overview and optimization of flooding techniques in OLSR. In: WoWMoM 2011, pp. 1–7, June 2011
26. Ray, N.K., Turuk, A.K.: A framework for disaster management using wireless ad hoc networks. In: Proceedings of the 2011 International Conference on Communication, Computing and Security, ICCCS 2011, NY, USA, pp. 138–141 (2011). http://doi.acm.org/10.1145/1947940.1947970
27. Rezvani, M., Ignjatovic, A., Bertino, E., Jha, S.: A collaborative reputation system based on credibility propagation in WSNs. In: IEEE 21st International Conference on Parallel and Distributed Systems (ICPADS 2015), pp. 1–8, December 2015
28. Sanzgiri, K., LaFlamme, D., Dahill, B., Levine, B.N., Shields, C., Belding-Royer, E.M.: Authenticated routing for ad hoc networks. IEEE J. Sel. Areas Commun. **23**(3), 598–610 (2005)
29. Sen, J.: Routing security issues in wireless sensor networks: attacks and defenses. In: Sustainable Wireless Sensor Networks (2011)
30. Sivakumar, M., Jayanthi, M.K.: Reliability analysis of link stability in secured routing protocols for MANETs. Eng. J. **18**, 66–76 (2014)

31. Taneja, S., Kush, A.: A survey of routing protocols in mobile ad hoc networks. Int. J. Innov. Manag. Technol. **1**(3), 279–285 (2010)
32. Čapkun, S., Buttyán, L., Hubaux, J.P.: Self-organized public-key management for mobile ad hoc networks. IEEE Trans. Mobile Comput. **2**(1), 52–64 (2003). http://dx.doi.org/10.1109/TMC.2003.1195151
33. VinothKumar, K., Rajaram, A.: An efficient security aware routing protocol for mobile ad hoc networks. Int. J. Comput. Sci. Netw. Secur. **14**(12), 66–73 (2014)
34. Zapata, M.G.: Secure ad hoc on-demand distance vector routing. SIGMOBILE Mob. Comput. Commun. Rev. **6**(3), 106–107 (2002). http://doi.acm.org/10.1145/581291.581312