A secure and lightweight ad-hoc routing algorithm for Personal Networks

Assed Jehangir, Sonia Heemstra de Groot

Faculty of Electrical Engineering, Mathematics and Computer Science,
University of Twente, Enschede, The Netherlands

{jehangira, heemstra}@cs.utwente.nl

***Abstract*—Over the past few years, there has been increasing interest in utilizing Personal Area Networks (PANs) to offer users innovative and personalized services. This interest is a consequence of the widespread use of mobile devices such as laptops, mobile phones, PDAs, digital cameras, wireless headsets, etc. to carry out a variety of user-centric tasks. The PAN itself is built upon an ad-hoc network where devices trust their neighbors to route their packets. The cooperative nature of ad-hoc networks allows malicious nodes to easily cripple the network by inserting false route information, replaying old messages, modifying messages of other nodes, etc. An applicable area still under research, and the focus of this paper, is *secure routing protocols* for ad-hoc networks. To achieve availability in the PAN, the routing protocol used must be robust against both dynamically changing topology and malicious attacks. However, the heterogeneous nature of Personal Network (PN) devices means that traditional security mechanisms are too resource intensive to be sufficient by themselves. This paper describes a new ad-hoc secure routing protocol for Personal Networks (PNs), suitable in a *limited* multi-hop scenario. This protocol is based on ADOV and relies on efficient cryptographic primitives to safeguard the security and privacy of PN users. Following that, a number of attacks in the area of ad-hoc networks are discussed, and it is shown that the new algorithm protects against multiple *un-coordinated* active attackers, in spite of compromised nodes in the network.**

Key words: Ad-hoc routing, secure routing, AODV, PAN, hash chains.

1. INTRODUCTION

Personal Networks (PN) [1] is a new concept related to the field of pervasive computing. A PN comprises a core consisting of a PAN, and can potentially be extended to include all of the person's devices, both in his vicinity and those at remote locations, such as the home and office. This on-demand and transparent extension of the PAN, will physically be made via infrastructure-based networks such as an organizations intranet, other ad-hoc networks, etc. The aim of this work is to propose a routing protocol that solves the problems of securing multi-hop routing in the core PAN.

Devices belonging to the PN can have one or more wireless interfaces such as Bluetooth, IEEE 802.11, UWB, ZigBee, etc. These devices are expected to be mobile and may enter or leave the PN at any time. They can also change their geographic location while continuing to be part of the PN. There are convincing reasons for enabling multi-hop routing in the core PAN. For example, in order to ensure network connectivity amongst different wireless technologies, nodes with more than one type of wireless interface will need to route packets between the different radio domains. Additionally, due to the limited range of wireless signals, multiple hops may be needed for communication. Finally, it may be more energy efficient to transmit packets over multiple small hops, even if end to end communication is possible using one large hop.

The cooperative nature of ad-hoc networks, in which individual nodes cooperate by forwarding packets for each other, make PANs highly vulnerable to attack. A security mechanism requiring all participating nodes to know a "shared secret" (which is regularly updated) can successfully protect against attacks from external attackers who are unable to "crack" this secret. Such a solution, although quite relevant in the context of PNs (where all nodes fall under one administrative domain), is ineffective in the presence of compromised PN nodes advertising incorrect routing information to other PN nodes. Node compromise is a real possibility and can occur in different ways, the most common of which is likely to be through software viruses, worms, trojan software etc.

One advantage of using a shared key to secure communication is that per-link (pair-wise) keys limit passive participation and local broadcast. In a PN context, where devices can be very heterogeneous in nature, using asymmetric cryptography based on public/private keys, imposes substantial requirements on the devices, and is therefore not realistic. Therefore one challenge of any computationally lightweight security mechanism, not based on shared secrets, is to implement broadcast authentication using only symmetric cryptographic primitives.

2. RELATED WORK

Our work, carried out in the context of PNs, addresses the issues related to secure communication in a PAN. Security of ad-hoc routing protocols can be divided into the two main areas of secure routing and secure data forwarding. In this paper, we propose a new and efficient secure routing protocol for PANs, called SL-AODV (Secure Lightweight AODV). The design of SL-AODV is based on the Ad-hoc On-demand Distance Vector (AODV) [2] routing protocol, and uses efficient symmetric cryptographic primitives. SL-AODV

prevents external attackers and compromised nodes from tampering with routes consisting of uncompromised nodes. It also protects against a variety of denial-of-service (DoS) attacks.

A security extension of the AODV routing protocol, known as S-AODV (Secure AODV), has been proposed in [3]. The S-AODV scheme assumes that each node has certified public keys of other network nodes, so that intermediate nodes can validate all in-transit routing packets. The originator of a routing packet appends to it a signature and the last element of a hash chain. As the packet traverses the network, intermediate nodes cryptographically validate the signature and the hash value, thus validating the integrity of the packet and the number of hops to the sender. Unfortunately, the use of public key cryptography in S-AODV imposes a high processing overhead on intermediate nodes and is unrealistic for many PN scenarios.

SL-AODV being based on the basic operation of the AODV protocol, is also a *reactive*, or an on-demand routing protocol. Other well known reactive secure routing protocols include ARAN [4], ARIADNE [5] and SRP [6]. Authenticated Routing for Ad-hoc Networks (ARAN), like S-AODV, is based on asymmetric cryptography and uses authentication certificates to protect the routing protocol. Before joining an ad-hoc network, a node must authenticate itself using valid certificates generated by a certificate authority (CA). However, a CA can not always be guaranteed in an ad-hoc environment. More importantly, the use of asymmetric cryptography imposes a high processing overhead that we believe is not realistic for many PN scenarios.

ARIADNE is a security extension of the DSR protocol and uses only efficient symmetric cryptography. The routing protocol messages are protected by message authentication codes (MACs) and are authenticated at each hop using TESLA [8]. TESLA is a broadcast authentication scheme requiring loose time synchronization. During route discovery, each hop authenticates new information in the route request. The destination node buffers the route reply until intermediate nodes can release the corresponding TELSA keys. Once the TESLA security condition is verified by the destination, it includes a MAC in the route reply to certify that the security condition was met. TESLA requires the source of the route request to estimate a maximum end-to-end delay in the ad-hoc network. The choice of this value does not affect the security of the protocol, however values that are too small may cause the route discovery to fail. Additionally, there must be a mechanism to distribute one authentic public TESLA key for each node and shared secret keys between pairs of communicating nodes. Furthermore, network links must be bi-directional and there must be loose time synchronization between all the nodes.

The design of the Secure Routing Protocol (SRP) is also based on the basic operation of the DSR protocol. The strength of SRP lies in the fact that the correctness of discovered routes can be verified from the route geometry itself. The destination validates the incoming route request using a shared key, and constructs a route reply that is protected using a MAC. The route reply is then returned to the source over the reversed path. False or corrupted routing information is discarded by the end nodes using an existing end-to-end security association. SRP, like ARIADNE, requires network links to be bi-directional and requires existing shared secret keys between pairs of communicating nodes. However, it does not have the other requirements of ARIADNE, making it a good candidate for use in PNs. The main weakness of SRP is due to the fact that routing messages are not authenticated in intermediate nodes. As a result the protocol is very vulnerable to denial-of-service (DoS) attacks such as repeatedly flooding the ad-hoc network with route requests. Also, because the MAC can not be verified by intermediate nodes, certain DSR optimizations can not be applied.

3. REQUIREMENTS AND ASSUMPTIONS

Most attacks against the routing protocol are caused by malicious injection of incorrect routing information, or modification of messages from other nodes. To prevent these attacks, it is necessary for each intermediate node to verify the origin and integrity of routing messages that it forwards. Therefore the security requirements for SL-AODV include the ability to: perform hop-by-hop authentication of routing packets, ensure routing packet integrity and resist replay attacks.

The new security mechanisms provide a defense against multiple uncoordinated attackers creating incorrect routes, in spite of compromised nodes. SL-AODV requires nodes to forward routing protocol packets only of authenticated neighboring nodes. Therefore, unauthenticated attackers cannot take part in the routing process and are restricted to a limited number of types of DoS attacks. SL-AODV is also able to withstand node compromise (authenticated nodes acting maliciously) by restricting their ability to disrupt the routing protocol.

The authentication mechanism must also have a low communication and computation overhead. An inefficient authentication mechanism could be exploited by malicious nodes by flooding the network with invalid messages and overwhelming nodes with the cost of verifying authentication. Additionally, to support nodes that have limited processing power, the security mechanisms of SL-AODV use efficient one-way hash chains [7] and do not use computationally expensive asymmetric cryptography. To use a one-way hash chain for authentication, we assume a mechanism for nodes to distribute commitment of their generated hash chains to all other nodes. This is necessary as routing messages generated by a node N have to be authenticated by each intermediate node that forwards these messages. Therefore, intermediate nodes need to know an authentic element from the hash chain of node N.

Lastly, our protocol does not aim to provide confidentiality and privacy for the sender of the routing message.

4. SL-AODV

Each node uses a specific element from its hash chain in each route request (RREQ) and route reply (RREP) packet it generates. Based on this hash element, the one-way hash chain provides authentication for the sequence number and the (lower bound value of the) hop count in each routing protocol message. In other words, when such a routing protocol message is received at a downstream node, that node is able to authenticate both the sender of the message and the minimum hop count to the sender. This is done using information contained in the *Hash* field of the RREQ and the RREP packets. Figures 1 and 2 show the modified packet format of

the RREQ and the RREP packets. Section 5 will analyze the reasons behind these modifications.

SL-AODV requires an upper bound to be assumed on the diameter of the ad-hoc network; we use *m*-1 to denote this bound. Thus, all hop count values in routing packets must be less than *m*. If a node's hash chain is the sequence of values c_0, c_1, c_2, $c_3 \ldots c_n$ where *n* is divisible by *m*, then for a sequence number *i* of some routing packet, let $k = n/m - i$. An element from the group of elements c_{km}, $c_{km+1}, c_{km+2} \ldots c_{km+m-1}$ from the hash chain will be used to authenticate the routing packet *for that sequence number*.

When a node sends a RREQ or a RREP packet, it sets the hop count field to 0 and the hash value field to the first element in the group of its own hash chain elements corresponding to that sequence number. In the above example for sequence number *i*, the node sets the hash value to c_{km}. Nodes receiving any routing packets can easily authenticate the hash value, given any earlier authentic hash chain element. Based on the sequence number and the hop count of the received packet, and the sequence number and the hop count of the latest prior authentic hash value for that source, the node hashes the hash value received in the packet the correct number of times (according to the description above as to which hash value must be used for any given sequence number and hop count) to confirm that the resulting value equals the prior authentic hash value. If so, the entry is authentic and the node processes it in the routing algorithm, otherwise the node drops the packet. Before this packet is retransmitted, the intermediate node will increase the hop count by one and update the hash value in the packet to the hash of the hash value it originally received (i.e. the first hop node will hash c_{km} to c_{km+1} , the second c_{km+1} to c_{km+2} etc.). Additionally, nodes will only accept routing packets with sequence numbers higher than what they have on record for that source.

Each time a node generates a routing update, it will use a one larger sequence number and use a specific (as explained above) next element from its hash chain. The receiving nodes will verify that they have not seen a packet from the sender with that sequence number or higher, and then verify that the hash value corresponds to the sequence number and the hop count.

This use of a hash value corresponding to the sequence number and hop count in a routing packet prevents any node from advertising a route to some destination claiming a greater sequence number than that destination's own current sequence number, due to the one-way nature of the hash chain. Likewise, no node can advertise a route better than that for which it has received an advertisement, since the hop count in an existing route cannot be decreased. Lastly, routing packets received at a later time, using the same sequence number are automatically dropped, so malicious nodes can not carry out replay attacks.

To safeguard against replay attacks, nodes wishing to authenticate a routing packet need to know the latest sequence number used by the sender of that packet. Therefore, unlike the AODV expanding ring approach for disseminating RREQs, SL-AODV requires the RREQs to be disseminated through the entire ad-hoc network. This ensures that all the nodes are aware of the latest publicly available hash value of the sender (corresponding to the latest sequence number used). Assuming that network diameter is not large, this should not result in perceptible performance degradation.

For the same reason, the RREPs from the destination also need to be disseminated through the entire network, because they too contain the latest publicly available hash value of the destination. If these RREPs are not widely disseminates, an attacker overhearing the RREP can not only replay that packet in another part of the network, but also use the hash value to construct RREQ packets.

When a node discloses a new hash chain value (each time it generates a RREQ or a RREP packet), all parties potentially have access to that value. Since attackers can create bogus messages using previously disclosed hash values, receivers must verify that the hash value used in each new routing message is based on a safe chain element. A safe chain element is one that was only known to the sender. Receivers must discard any message that is unsafe and may have been forged. Therefore, attackers are unable to change their source address and are thus much more restricted in their ability to disrupt the routing protocol and launch DoS attacks.

As RREQ packets are broadcasted through the entire ad-hoc network, they have a major potential for abuse. Nodes have, in principle, a limit for the number of RREQ packets they are willing to forward for any given node (over a period of time). To prevent attackers from changing their source address, intermediate nodes authenticate the original sender of the RREQ and not merely the last hop sender. Additionally, the original sender of the RREP packet is also authenticated, to prevent an attacker from advertising invalid routes. However, as route error (RERR) packets are only transmitted one hop, they are authenticated using security associations existing between neighboring nodes, and do not need additional protection. A simple check to verify that routes listed in RERR packets do indeed go through the sender of the RERR packet, will provide sufficiently robust security against malicious RERR packets.

5. SL-AODV PACKET FORMAT

One of the security requirements for SL-AODV was that the source of the RREQ can only trust RREPs that are sent by the destination itself. In standard AODV intermediate nodes with fresh routes to the destination can also generate RREPs on behalf of the destination. Since intermediate nodes are no longer trusted to reply to RREQs on behalf of the destination, there is no need to have the 'D' (Destination only flag) and the 'G' (Gratuitous RREP flag) in the AODV routing packets. Details on the AODV packet format can be found in [2].

In standard AODV the destination sequence number in a RREQ packet has two uses. Firstly, it lets intermediate nodes understand how up to date the sender requires the RREP to be. Secondly, it helps the destination choose a new sequence number when it reboots. After rebooting, a node does not remember its sequence number and trusts anybody that sends to it a RREQ with the correct number. However, such functionality is not acceptable since a malicious node can put a much bigger destination sequence number than the real one. This allows a very easy attack that consists in setting the destination sequence number to the maximum value. The next time the node increments the sequence number, its counter will overflow and this will cause unexpected results. Additionally, the first functionality is also not necessary since intermediate nodes are no longer allowed to reply to route request packets and the destination is expected to reply with the most up to date route. Therefore, there is no need to have

a destination sequence number field and the 'U' flag in the RREQ packet. Lastly, as there is no support for multicasting, the 'J' and 'R' flags are also no longer needed.

As a result of the expanding ring approach in AODV, senders often needed to send multiple RREQs before receiving a RREP back. To ensure that sequence numbers are not used too fast, each new broadcast increments the RREQ ID value. However, since SL-AODV does not use the expanding ring approach, senders always increment their sequence number before retransmitting RREQs. Intermediate nodes can now use the sequence number and source address in each RREQ packet, instead of RREQ ID, to drop duplicates. Figure 1 and 2 show the new SL-AODV RREQ and RREP packet formats respectively.

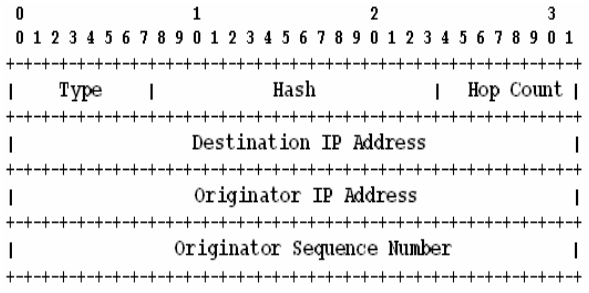

Figure 1: SL-AODV RREQ packet format

Type: 1

Hop Count: The number of hops from the Originator IP Address to the node handling the request.

Destination IP Address: The IP address of the destination for which a route is desired.

Originator IP Address: The IP address of the node which originated the Route Request.

Originator Sequence Number: The current sequence number to be used in the route entry pointing towards the originator of the route request.

Hash: Hash value from the senders hash chain corresponding to the sequence number and hop count of the packet.

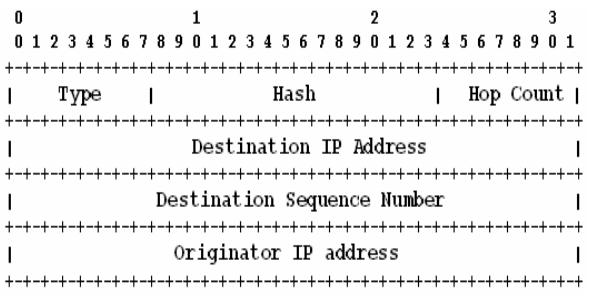

Figure 2: SL-AODV RREP packet format

Type: 2

Destination IP Address: The IP address of the destination for which a route is supplied.

Destination Sequence Number: The destination sequence number associated to the route.

Originator IP Address: The IP address of the node which originated the RREQ for which the route is supplied.

Hash: Hash value from the senders hash chain corresponding to the sequence number and hop count of the packet.

6. EVALUATION

An obvious disadvantage of fully disseminating routing messages is the extra number of transmissions necessary. However, this dissemination means that other nodes in the network also discover a path to the two communicating nodes. It is conceivable that these active nodes may be involved in future communication, and in many cases new path discovery will not be necessary. As AODV is designed for potentially hundreds of nodes, the reverse route made during route discovery is deleted if it is not chosen during the route reply. This is done in order to reduce the number of entries in local routing tables. PANs on the other hand, are made of much smaller number of nodes so caching these routes for a longer period could be advantageous. Therefore if both the routes in the RREQ and the RREQ are cached, all the nodes in the PAN will have routes to these communicating nodes. Over time, all the active nodes will find themselves in the routing tables of other nodes. This is advantageous because node wishing to communicate with others may already have route to those node. However, if that route is broken, then the standard AODV route error packets will be returned and a new route discovery can be started.

However, as certain devices may be very battery/memory constrained so frequent broadcasts are not feasible for them. These low powered nodes should not take part in routing packets belonging to other nodes. Consequently, they also do not need to know the hash key commitments of other nodes as well (only of those they want to communicate with).

Malicious nodes can attempt to reduce the amount of routing information available to other nodes, by not advertising certain routes or by destroying routing packets that pass through them. This shows the unwillingness of the malicious node to forward packets for those destinations. We do not attempt to defend against this attack, since the attacker could also otherwise drop data packets sent to those destinations.

Additionally, our protocol does not aim to prevent attackers from injecting data packets into the network. Injecting data packets only results in a DoS attack if it floods the network. However, attackers can attempt to degrade the performance of the routing protocol by repeatedly sending RREQ packets that flood the network. To thwart malicious route request floods, intermediate nodes which authenticate each routing message, must filter out excessive route request packets coming from one node.

An attacker can also modify a routing packet by changing its destination, source address and the hop count. For example, an attacker advertising a zero hop count for all destinations can cause all nodes around it to route packets for all the

destinations to itself. SL-AODV provides defense against this type of attack by authenticating the sender of the routing packet as well as the number of hops to the sender.

In another attack, malicious routing packets claiming a large sequence number can attempt to force the receiving node to perform a large number if hash operation in order to authenticate the routing packet. In order to guard against such attacks the receiving node should limit the number if hashes it is willing to perform for each authentication, discarding packets that do not meet that criteria.

A wormhole attack is carried out by a pair of colluding attackers in the network, linked via a tunnel. In such a case, every routing packet received by an attacker A, is sent over the tunnel to attacker B, to then be forwarded normally by B. Similarly, B sends every routing packet it receives to A. This means that the hop count field in these routing packets will not be changed, so most routes between nodes towards the two ends of the network will pass through A and B, thus creating a virtual vertex cut of the nodes in the network. This type of attack is very difficult to detect as no false packets are being created or maliciously modified, and can not be protected against by SL-AODV.

Lastly, compromised nodes can also be used to launch attacks. Since SL-AODV requires nodes to authenticate the sender of each routing message, compromised nodes can not spoof their source address, and are restricted to attempting DoS attacks. SL-AODV protects against such DoS attacks by limiting the number of route requests from any one source. So neighbors of malicious nodes may temporarily or permanently block any further packets from that address. As the source is not able to authenticate itself as any other address, it can not perform any more attacks.

7. CONCLUSION AND FUTURE WORK

This paper has presented the design and evaluation of SL-AODV, a new ad-hoc routing protocol that relies only on efficient symmetric cryptography to ensure the security of PN users. SL-AODV operates on demand; the design being based on the basic operation of AODV. We have made a comparison with other existing secure routing protocols like ARIADNE, SRP and ARAN and discussed why none of them can insure complete security while using simple procedures and lightweight mechanisms. Due to the use of hash chains and network wide broadcast of routing messages SL-AODV is suitable in a limited multi-hop scenario, such as that of a PAN. It requires nodes to forward routing protocol packets only of authenticated nodes. We have shown that the new protocol protects against multiple un-coordinated active attackers, in spite of compromised nodes in the network.

The next step to be done in the evolution of this proposal is to implement SL-AODV in the ns-2 simulator and evaluate its network performance against standard AODV.

ACKNOWLEDGMENT

The authors gratefully acknowledge that the work presented in this paper has been performed in the context of the IOP Gencom QoS for PN@home and the ES IST project MAGNET.

REFERENCES

[1] Ignas G. M. M. Niemegeers, Sonia M. Heemstra de Groot, "Research Issues in Ad-Hoc Distributed Personal Networking", *Wireless Personal Communications: An International Journal*, Volume 26, Issue 2-3, Pages 149-167, Kluwer Academic Publishers, August 2003.

[2] Charles E. Perkins, Elizabeth M. Royer, Samir R. Das, "Ad-hoc On-Demand Distance Vector (AODV) Routing" (work in progress), draft-ietf-manet-aodv-13.txt, February 2003.

[3] Manel Guerrero, "Secure ad-hoc on-demand distance vector (SAODV) routing", draft-guerrero-manet-saodv-01.txt, August 2004.

[4] C.Shields, B.Dahill, K.Sanzgiri, B.N.Levine and E.M.Belding-Royer, "A secure routing protocol for ad-hoc networks", in *Proceedings of the 10th IEEE International Conference on Network Protocols (ICNP)*, November 2002.

[5] Y. Hu, A. Perrig, D. Johnson, "Ariadne: A Secure On-Demand Routing Protocol for Ad-Hoc Networks", *MobiCom '02*, September 23-26, 2002, Atlanta, Georgia, USA.

[6] Z. J. Haas, P.Papadinitratos. "Secure routing for mobile ad-hoc networks", in *Proceedings of SCS Communication Networks and Distributed Systems Modeling and Simulation (CNDS)*, January 2002.

[7] Y. Hu, D. Johnson, A. Perrig, "SEAD: secure efficient distance vector routing for mobile wireless ad-hoc networks".

[8] A. Perrig, R. Canetti, D. Song and J.D. Tygar, "Efficient and Secure Source Authentication for Multicast, in *Network and Distributed System Security Symposium, NDSS '01,* pages 35-46, February 2001.