Secure Dynamic MANET On-demand (SEDYMO) Routing Protocol

Helena Rifà-Pous and Jordi Herrera-Joancomartí
Universitat Oberta de Catalunya
Rambla del Poblenou 156
08018 Barcelona, Spain
{hrifa,jherreraj}@uoc.edu

Abstract

This paper describes the state of the art of secure ad hoc routing protocols and presents SEDYMO, a mechanism to secure a dynamic multihop ad hoc routing protocol. The proposed solution defeats internal and external attacks using a trustworthiness model based on a distributed certification authority. Digital signatures and hash chains are used to ensure the correctness of the protocol.

The protocol is compared with other alternatives in terms of security strength, energy efficiency and time delay. Both computational and transmission costs are considered and it is shown that the secure protocol overhead is not a critical factor compared to the high network interface cost.

Keywords: Network security, secure routing protocols, network attacks, digital signatures

1. Introduction

In mobile ad hoc networks (MANET), nodes cooperate to form a communication infrastructure that extends the wireless transmission range of every terminal without using any dedicated network device such as access points or base stations. Instead, the mobile nodes behave as routers and take part in route discovery and maintenance.

MANET protocols typically assume that all nodes cooperate in the coordination process. This assumption is unfortunately not true in a hostile environment. Because cooperation is assumed but not enforced in MANETs, malicious attackers can easily disrupt network operations by violating protocol specifications. The main network-layer operations in MANETs are ad hoc routing and data packet forwarding, which interact with each other and fulfill the functionality of delivering packets from the source to the destination. Ad hoc routing protocols exchange routing messages between nodes and maintain routing states at each node accordingly. Based on routing states, data packets are forwarded to the destination by intermediate nodes along an established route.

For a secure ad hoc routing protocol to be practical, transmission delays and energy consumption introduced by security measures must be low. We compare the proposed protocol with other two and analyze if the costs due to security are admissible for a handheld device.

1.1 Routing challenges

Multihop routing is the procedure to relay a message between two endpoints through a sequence of intermediate nodes. Routing protocols are designed to fulfill path discovery and maintenance of routing tables. The tradeoffs between routing strategies are quite complex since the best approach depends on many factors, such as network size, mobility and data traffic. We focus on reactive protocols since they are well suited for mobile ad hoc networks with little or medium traffic and no high requirements for real time services.

Nodes in an ad hoc network exchange neighborhood information to construct a virtual view of the network topology and allow data packet routing. This information has to be protected to avoid malicious nodes disrupting the network. Malicious nodes can be divided in external and internal attackers. External attackers can inject erroneous routing information, replay previous routing messages, or modify the valid routing information. Internal attackers, however, can usually cause more severe damages. Such nodes may have been trusted in the past but later they do not commit anymore to their initial promises or they have been compromised by external attackers.

From the standpoint of security, an optimal routing protocol should fulfill the following criteria [24]:

- **Certain discovery:** If a route between two points in a network exists, it should always be possible to find it and have the certainty that it is correct.
- **Isolation:** Misbehaving nodes should be identified and isolated from routing.

- **Location privacy:** Information about node location and network structure shall be protected from adversaries trying to destroy or capture nodes.
- **Self-stabilization:** The routing protocol should be able to automatically recover from any problem in a finite amount of time without human intervention. It must not be possible to permanently disable a network with a punctual attack.
- **Byzantine robustness:** The routing protocol should be able to function properly even if some participating nodes are turn out to become malicious or are intentionally damaged.

The enforcement of these requirement prevents either passive and active attacks like the ones presented below:

- **Impersonation:** By masquerading as another node, a malicious node can impersonate a legitimate node to misrepresent the network topology. One example of this attack is the spoofing attack.
- **Denial of service (DoS):** The objective of this attack is to crash or congest a resource so that no longer operates correctly. DoS attacks can be classified into two categories, namely routing-disruption attacks and resource-consumption attacks. In routing-disruption attacks legitimate packets are routed in dysfunctional ways. Some forms of these attacks are the wormhole, black hole, gray hole, selfish, routing loop and rushing attack. In resource consumption attacks packets are injected into the network in an attempt to consume valuable network and nodes resources.
- **Disclosure:** Routing control information such as specific status details of a node, node location, private or secret keys, passwords and so on can be revealed to unauthorized nodes. Location disclosure attack and passive attack are in this category.

1.2 Related work

Some secure reactive routing protocols have been proposed in MANET to prevent route discovery process. We briefly outline the most relevant characteristics of them.

Security-Aware ad hoc Routing (SAR) [23] introduces the idea of trust level as one of the metrics in path finding. Nodes are associated with security levels and every level owns a different key. Only nodes that share a level key can process and forward messages in a specific level. The main drawback of the protocol is the difficulty to manage dynamic security levels operating with symmetric keys.

ARIADNE [13] utilizes one-way hash chain to prevent malicious nodes from modifying sensible information as sequence number or hop counts. Nodes best way need to authenticate neighbors is using TESLA protocol, which requires clock synchronization and introduces a delay. Multiple replies are returned for each request. Internal attackers that do not relay messages are detected using a feedback approach similar to the IPv6 one.

Secure Routing Protocol (SRP) [20] requires that for every route discovery the source and the destination have a security association (SA) between them, which is used to calculate MAC codes to support data integrity and authenticity of route packets. Although it is an efficient protocol because only uses symmetric cryptography, the main weakness is the lack of a key management scheme for establishing SA between every two nodes of the network. Furthermore, it is vulnerable to fabricated route error messages attacks.

Secure Ad hoc On-demand Distance Vector (SAODV) [11] employs digital signatures and hash chains in order to provide data integrity, source authenticity and non-repudiation. Digital signatures, which are generated by the source node, are used to protect non-mutable fields of messages. Every intermediate node has to verify the signature. Hash chains are used to protect hop count information ensuring that this metric has not been decremented by an attacker.

Authenticated Routing for Ad hoc Networks (ARAN) [8] provides authentication and non-repudiation services. When a node generates a routing message must also sign it. Every intermediate node verifies the signatures of the source and the previous node, removes the latter, and signs the original message.

Security Aware Adaptive Dynamic Source Routing (SADSR) [9] uses trust level as a metric of the path finding, like SAR. All nodes sign routing packets that pass thought them in order the peer nodes can compute a trustworthiness level for each path.

Secure Route Discovery Protocol (SRDP) [14] introduces backward authentication to lighten the security overhead of the protocol. Route integrity, protected via aggregated MACs or multisignatures, is verified in the response messages, not in the broadcast discovery phase.

1.3 Contributions

Designing routing protocols that have strong security as well as performance its a challenging task. There does not exist an optimal protocol for all contexts, but for a specific scenario. The proposed protocol is an extension of DYMO, a routing protocol that offers quick adaptation to dynamic conditions, low processing and memory overhead, and low bandwidth. Its initialization phase may introduce a noticeable delay in large networks since in a route request discovery process all the nodes involved learn about the network topology. For this reason, DYMO is very suitable for mobile and multihop ad hoc networks with multiple cross-communications between parties, and is not recommended for large networks with few and sporadic communication lines that consume real-time services.

Up to our knowledge, there is only one proposal that tries to secure DYMO, SDYMO [10]. SDYMO exploits the fact that DYMO is based on AODV to adopt the security extension of that protocol -SAODV- to its data structure. However, SDYMO does not protect that part of DYMO protocol that differs from AODV (that is, routing information from intermediate nodes is not secured). Therefore, SDYMO can be considered equal to SOADV, and in the rest of the article references will be made only to SAODV.

Our main contribution includes: (1) a secure protocol for mobile ad hoc networks with multi-peer interactions, (2) an analysis of the cost overhead -time and energy- of the secure wrapper of the protocol, and (3) a comparison of our proposal with previous ones in terms of security vulnerabilities and energy and temporal costs.

The comparison is performed with SAODV [11] and ARAN [8], which are secure extensions of AODV protocol, the predecessor of DYMO. SAODV, ARAN and SEDYMO are all based on public key cryptography (PKI), not demand time synchronization, and are intended to meet similar requirements.

SEDYMO solves security attacks that could be committed in ARAN or SAODV. ARAN suffers selfish attacks, and SAODV is vulnerable to spoofing attacks from the nodes that are in the selected routing path. We will see that the security robustness of our proposal does not imply a major cost.

The remainder of this papers is organized as follows: In section 2 we review DYMO, the model our protocol is build upon pointing its weaknesses and threats. Section 3 describes SEDYMO, the proposed security protocol. Section 4 analyzes our proposal both in security terms and in energy and temporal costs, and compares it with two other secure reactive protocols. Finally, section 5 concludes the paper and outlines some ideas for future research.

2 DYMO

Dynamic MANET On-demand (DYMO) [5] is a simple and fast routing protocol for multihop networks. IETF MANET working group has selected it as the single standard reactive routing protocol ([6]).

The basic operations of the DYMO protocol are route discovery and route management.

On route discovery the originating node initiates dissemination of a Route Request (RREQ) throughout the network. During this process, each intermediate node records a route to the originating node and to all the other nodes in the path that have appended routing information in the message. Optionally, it itself adds routing information to be reached and forwards the RREQ to its neighbors. When the target receives a RREQ, it responds with a Route Replay (RREP) sent hop-by-hop toward the originating node. Intermediate nodes update again their records to the target and to the routing nodes in the path. When the originating node receives the RREP, routes have then been established between the originating node and the target node in both directions. Moreover, if intermediate nodes have appended its routing information to RREQ and RREP messages, communication paths have been established between any two peers in the path.

Nodes maintain a routing table with information collected in the discovery process. The routing table is labeled with a sequence number that identifies the update of the node's position within the network topology. When a node creates a routing message during the route discovery process, it increments if own table sequence number to indicate its position update.

A routing table contains a list of target addresses associated with the sequence number of the target routing table and its metric. The metric used in DYMO is hop count which determines the minimum number of nodes a packed has to pass through before arriving to the destination. The best route is the one with a minimum hop count value.

Nodes only update routing table information if the data they receive comes from a node with an equal or greater sequence number than the cached one. In this way, sequence numbers are used to avoid updates in the table with stale information.

2.1 Exploits allowed by DYMO

DYMO does not specify any special security measures although it states that messages must be protected by the use of authentication techniques to avoid impersonation and denial o service attacks.

Several forms of DoS attacks can assault DYMO. A selfish attack is possible by increasing the hop count field in route discovery messages. By setting the hop count field of the route request message to infinity, created routes will tend to not include the malicious node. Moreover, the protocol is unprotected against modification and fabrication attacks. A node can generate false routing elements and throw them in the network. Messages lack any integrity so it is easy to poison a routing table. DYMO is also susceptible to the wormhole attack since there is no way to check whether or not the packet has passed over a private network.

Finally, disclosure attacks are possible in DYMO since no confidentiality means are defined. The relative position of one node with respect to the other nodes of the network can be discovered.

3 SEDYMO

In this section we present SEDYMO, a new security protocol extension for DYMO that offers integrity, authenticity and non-repudiation. Its security mechanisms are based on digital signatures (simple, multiple or aggregate) and hash chains.

We assume a distributed Certificate Authority (CA) [24, 16] that issues authorization certificates to control the access to the resources of the network. Certificates are extended to authenticated users that hold a correct reputation history. Authentication is performed using identity certificates from recognized external CAs. Local authorization certificates bind the user identity with its IP address and cryptographic keys. They are meant to be renewed in short periods of time to guarantee malicious node ejection and isolation.

Comparing with other security protocols, SEDYMO deals with public key cryptography such that no preestablished secret keys are needed (i.e. SAR, SRP). On the other hand, SEDYMO differs from similar approaches in that it does not relay on synchronized clocks like needed in ARIADNE, SEAD or SADSR.

3.1 The proposed scheme

SEDYMO route discovery process is similar to DYMO, but it bounds it with two rules: intermediate nodes must always append routing information to the routing messages they forward, and data from previous routers can not be removed from the packet even if it is stale or disregarded. Security is endorsed in the protocol with the use of hash chains and digital signatures that protect the veracity, integrity and authenticity of the data.

3.1.1 Hash Chains

Hop count is the metric used in DYMO to generate routing tables. SEDYMO uses hash chains [17] in the route request messages to enforce each node to state the true hop distance it is from the origin, thus guaranteeing the shortest path between two peers can be selected. Hash chain mechanisms are not used in route replays since route discovery is completed when a route request reaches the target node, replays are only a mean of path dissemination.

DYMO routing messages are made up of a message header, message block, and an indefinite number of address blocks each of which can have an associated data block.

```
dymo_msg =   <msg_header>
             <msg_block>
             {<addr_block><data_block>}*
```

In order to include a hash chain mechanism in the protocol we need to modify one of the fields of the DYMO routing message and add some new ones (see table 1).

Field	Description
`Hash`	h, the hash value of the same previous field.
`HashFunc`	$H(\cdot)$, a hash function identifier.
`HopLimit`	A fixed value n which determines the maximum number of hops a packet can do in the network.
`TopHash`	$H^n(s)$, where s is the hash chain *seed*.
`HopCnt`	Number of hops from the origin.
`Node.HopCnt`	Number of hops between the origin and a particular node

Table 1. Hash chain fields

Three fields are introduced in the message block of a route request: `HashFunc` and `TopHash`, that contain static values, and `Hash`, whose value varies on each hop so every intermediate node has to update it. `HopLimit`, `HopCnt` and `Node.HopCnt` were already defined by DYMO, although the meaning of the `Node.HopCnt` is modified. Figure 1 depicts in a loosely manner the data structure of a DYMO route request message with the additional hash chain fields (in white) introduced by SEDYMO.

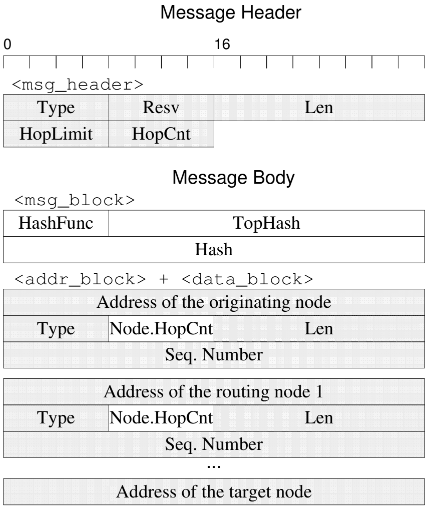


Figure 1. DYMO message with hash chain

When the originating node initiates a route discovery process it has to perform the following steps:

1. Generating a random value s that will be used as the *seed* of the hash chain.
2. Initializing the `Hash` field of the routing message.
$$\texttt{Hash}_0 = s$$
3. Stating the hash function algorithm identifier used to compute every element of the hash chain.
$$\texttt{HashFunc} := H(\cdot)$$

4. Computing the top has value

$$\texttt{TopHash} := H^{\texttt{HopLimit}}(\texttt{Hash}_0)$$

When an intermediate node i forwards a route request message it has to:

1. Store in the routing table the incoming $\texttt{Hash}_{i-1}$ value of the received message. This value may be later requested to proof node honesty throughout the routing process.

2. Update some fields of the route request message

$$\texttt{Hash}_i := H(\texttt{Hash}_{i-1})$$

$$\texttt{HopCnt} := \texttt{HopCnt} + 1$$

3. Append routing information in a data block of the routing message and state its hop distance from the originating node.

$$\texttt{Node.HopCnt} := \texttt{HopCnt}$$

The `Node.HopCnt` field in a data block of SEDYMO overwrites the one defined in DYMO. Note that in DYMO, `Node.HopCnt` field identifies the number of hops that a particular data block has traversed since it was appended in the message. It is initialized with 0 and its value has to be incremented at each intermediate hop. On the other hand, in SEDYMO `Node.HopCnt` is a static parameter that provides the same information as the original one but from another point of view: it states the hop distance between the originating node and that particular router. The advantage is that the proposed `Node.HopCnt` field can be included in the intermediate node's digital signature and becomes a routing map evidence.

When a node receives a routing element it has to check the hop count of the latest node of the routing list,

$$H^{\texttt{HopLimit}-\texttt{HopCnt}}(\texttt{Hash}) = \texttt{TopHash}$$

then honesty of the players is ensured and the packet can be updated and forwarded.

3.1.2 Digital Signatures

In order to guarantee authenticity, integrity and non-repudiation of the non-mutable routing data contained in a routing message, SEDYMO uses digital signatures. The inclusion of a digital signature mechanism in the protocol requires the introduction of some new fields in the routing message (see table 2). Figure 2 loosely depicts the data structure of a SEDYMO routing message with the introduced digital signature fields in white.

Field	Description
`SignFunc`	Signature algorithm
`SignValue`	Signature value
`CertId`	Certificate issuer and serialNumber
`Cert`	Certificate

Table 2. Digital signature information fields

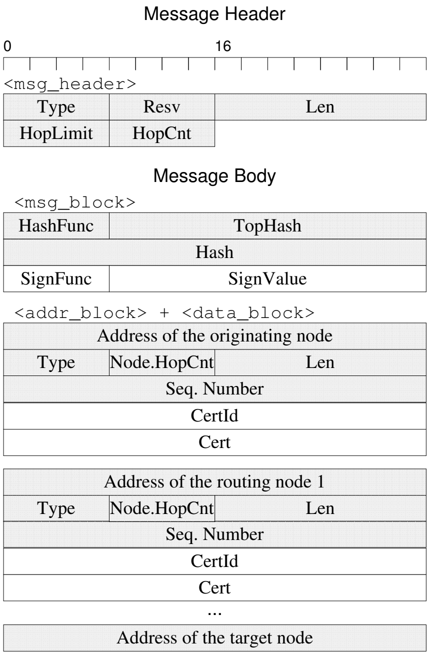


Figure 2. Example of a SEDYMO message

In a route request message, the static information generated by the originating node must be signed by this first node. Subsequent nodes have to sign the concatenation of three portions of the routing message:

- The routing information they append in the message, thus stating its authenticity.
- All present address blocks, in order to rapidly detect modifications in the routing path.
- The `TopHash` parameter of the message block, thus binding the signature with its container to prevent reusing attacks.

The signature information fields `SignFunc` and `SignValue` are located in the common message block of the routing message. This is because we use aggregate signatures to combine the signatures of all nodes into a single one. Signature aggregation allows to join n signatures on n distinct messages from n distinct users into a single one, that way reducing the size of the message. There are two main constructions for aggregate signatures. The first are based on the short signature scheme of Boneh, Lynn, and Shacham [2] and supports general aggregation (GAS). The second, based on a multisignature scheme of Micali, Ohta, and Reyzin, [22] are built from any trapdoor permutation but only support sequential aggregation (SAS) [19].

Although GAS is more powerful than SAS, the fact that sequential aggregation can be built from standard primitives such as RSA has its benefits, such as software implementations turn out better performances. In SAS the aggregation is done during the signing process so the generation costs are equivalent to that of a plain RSA signature. On the other hand, the verification process requires the evaluation of n nested signatures from n users and the cost increases linearly in the number of signatures.

When an intermediate node receives a routing element it must verify all the signatures it contains. Furthermore, the value of `HopCnt` fields in the data blocks have to be checked to assert its consistency along all blocks.

A target node, after validating a route request message, must generate a route replay in the following way:

1. The maximum number of hops the message is allowed to traverse is limited to the length of the path.

$$\texttt{HopLimit} := \texttt{HopCnt}_{RREQ}$$

2. Address blocks must contain an ordered list of the intermediate nodes to reach the target.

3. Data blocks associated with address blocks, must contain the routing table sequence number of the node with which they are bounded.

4. A signature is generated of the above fields.

These parameters enforce the use of the optimal path in route replay messages. Intermediate nodes must check if the received routing information is coherent with the values stored in their routing tables. If not, a routing error message against their previous node have to be generated.

Intermediate nodes do not contribute with new information in a route replay message so they all sign the same data. A multisignature scheme is used for this purpose because of its potential efficient batch verification. The signature length and verification time for multisignatures based on GDH [1] is independent of the number of signers and is almost de same as for the base signature scheme. However, it not make the most in RSA schemes, with a similar behavior than SAS.

On the other hand, SEDYMO may reduce its costs by following two different strategies. First of all, is recommended that maintenance routing messages do not include user certificates thus reducing the required bandwidth.

Moreover, SEDYMO can work in a lax configuration that lightens the protocol but makes it a little more vulnerable to some attacks (see section 4.1. Aggregate signatures and multisignatures are not used, but every data block holds a simple signature of the information appended by an intermediate node plus the `TopHash` parameter of the message block. Address and data blocks from other routers can be removed if that information is stale or considered of no relevance.

In this mode, nodes do not verify all signatures in a route request but the first, the last, and the ones which routing data block can cause an update in their routing table. First node's signature verification is mandatory to preserve the headers integrity of the routing message (i.e. a flag indicating lax config.) while the last one guarantees that the hop count value to the originating node has not been faked.

In route replay messages, only the target is requested to sign the message. Intermediate nodes must check the path stated in the element is coherent with the values gathered in the route request, but are not required to verify the signature if they are not to update the target node entry in their routing table. This mode of operation assumes the originating node trusts the target node.

SEDYMO routing error messages must include a digital signature to determine the issuer node and as an evidence of problem. In a routing error message, a digital signature is included for every unreachable node. In the same way, unsupported-element errors have to be signed by the issuer of the message.

4 Results

4.1 Security Analysis

In this section we present how SEDYMO may resist DYMO exploits described in section 2.1 and we compare SEDYMO with other secure routing protocols.

Impersonation attacks are detected since the routing information source is always authenticated and integrity check is preformed on data. For example, fabrication of false routing error messages is not possible because all types of routing elements have to be signed. Black hole attacks are also prevented with the use of hash chains that avoid claiming a hop distance lower than the real one. Replay attacks can also be detected and proved because a node must keep an evidence of the metrics computation and because of the path check that is done in the route replay process.

Selfish attacks are prevented with the mandatory inclusion of the previous nodes signatures in the routing element. However, in the lax mode of operation a node could state the path between it and the originating node is longer that it really is. The more signatures the protocol forbids to remove, the bigger the coalition of nodes that is required to successfully perform a selfish attack.

SEDYMO is unprotected to wormhole attacks. The protocol does not have a mechanism to reveal if some nodes communicate themselves through a private network and block the forwarding packets of other users. Nodes can learn it from the experience but not from the protocol itself. Once a misbehaving node is detected, its identity is broadcasted throughout the network so that no certificate renewal will be archived. Without a valid certificate, a node cannot participate in the network.

SEDYMO does not deal with confidentiality then disclosure attacks are possible. However, its routing packets do not carry critical information such as keys or passwords.

SEDYMO improves other secure routing protocols based on asymmetric cryptography and presented in section 1.1. ARAN suffers from not being able to record the shortest or quickest path between two nodes, making selfish attacks possible. SEDYMO gets the shortest path and can prevent selfish attacks. In SAODV a malicious intermediate node can spoof its identity. This vulnerability is not present in SEDYMO because intermediate nodes have to authenticate its data. SADSR manages multiple routes to each destination in a way that introduces a reasonable network load in the DSR protocol. Despite the overhead, SADSR does not prevent selfish attacks.

4.2 Energy costs

Mobile ad hoc networks are usually comprised of wireless devices with power constraints, so routing protocols have to be designed to accommodate to these characteristics. In this section we analyze how security affects the energy costs of a protocol. The testbed consists of a PDA with 206MHz StringARM processor and a connection to a wireless LAN through a network interface with a radiated power of 15dBm. We first expose computational costs and then we review the energy spent in the network interface.

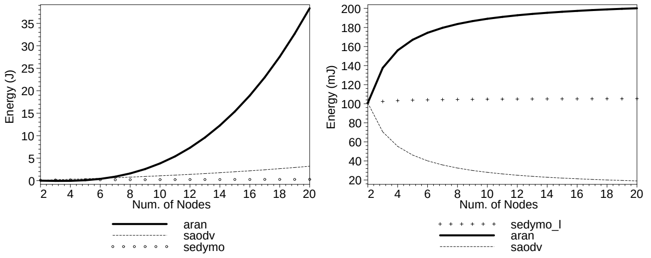
Table 3 shows the number of cryptographic operations carried out by the nodes in a routing path to establish a route between two endpoints separated N hops.

Protocol	Sign.	Verif.	Hash
ARAN	$2(N-1)$	$2(2N-3)$	-
SAODV	2	$2(N-1)$	$2(\sum_{i=1}^{N}(mhc-i+1))$
SEDYMO	$2(N-1)$	$2(\sum_{i=1}^{N-1} i)$	$\sum_{i=1}^{N}(mhc-i+1)$
SEDYMO_l	N	$2(N-1)$	$\sum_{i=1}^{N}(mhc-i+1)$

Table 3. Number of cryptographic operations

In ARAN each node generates a signature and verifies two of them in a route request process, and does the same in a response process. In SAODV only the source and destination nodes sign a message when they initiate the request or response transmission. Intermediate nodes verify one signature for each routing message they relay. Moreover, each node has to check the value of a hash chain. In SEDYMO every node generates a signature and verifies all the previous ones. Yet, the lax configuration of SEDYMO is lighter and only requires the verification of 2 signatures per node if the intermediate nodes are not interested in a full update of its routing table.

We evaluate the protocol using 512-RSA digital signatures and sha1 hash chains, which is secure enough for an environment of short-live certificates. Based on the study of Nachiketh [21], the consumed power to generate a signature is $96mJ$, the signature verification costs $4,8mJ$ while the computation of a hash takes $0,76\mu J/Byte$.



(a) Average energy (J) per node to create a complete Routing table

(b) Average energy (mJ) two discover a route between two peers

Figure 3. Energy consumed in cryptography

Figure 3(a) shows the results of energy consumption per node to build routing tables for all the nodes in a communication path. The maximum hop count parameter mhc of SAODV and SEDYMO protocols is set to the size of the route, and the number of bytes to hash is set to 20 bytes since this element belongs to a hash chain. Note that SEDYMO is a path accumulation protocol so its very efficient to create a complete view of the network topology.

On the other hand, if cross-routing paths among the nodes that form a network are not required because communication is always between 2 fixed end-points, the lax configuration of SEDYMO can be employed. Figure 3(b) shows the results of energy consumption per node versus network diameter to create a single route between to peers.

Energy spent in network interface is a function of the size of the transmission packet. Feeney [18] measured energy costs of a wireless 802.11 interface at 11Mbps with a radiation power of 15dBm[1]. Table 4 shows the results. Route request messages are sent in broadcast while route responses are transmitted point-to-point. It is worth noting that during transmission and reception states, idle interface is also consuming $741mW$.

The routing packets size of the analyzed protocols -in plain mode and secure mode- are depicted in table 5. We have considered the protocols interchange certificate iden-

[1] The maximum permitted output power is 1W in the United States, and 100mW in Europe (IEEE, 1997). Transceivers for mobile applications typically radiate less than 50mW.

	Transmission	Reception	Idle
Unicast	$(0.48S + 431)\mu W$	$(0.12S + 50)\mu W$	$741mW$
Broadcast	$(2.1S + 272)\mu W$	$(0.26S + 50)\mu W$	

Table 4. Net interface power costs per byte S

tifiers instead of the certificates themselves. The size of SEDYMO messages is increasing in each hop because every intermediate node appends its data.

	Plain		Secure	
	RREQ	**RREP**	**RREQ**	**RREP**
1	12		$\begin{cases} 101 & \text{if } N = 1 \\ 185 & \text{if } N > 1 \end{cases}$	
2	18		144	
3	$10 + 10N$		$116 + 29N$	$116 + 8T + 21N$
4	$\begin{cases} 10+10N & \text{if } N < 3 \\ 40 & \text{if } N => 3 \end{cases}$		$\begin{cases} 52+96N & \text{if } N < 3 \\ 316+8N & \text{if } N => 3 \end{cases}$	$137 + 8T$

Protocols: 1=ARAN, 2=SAODV, 3=SEDYMO, 4=SEDYMO_l (lax config)
Parameters: T=total hops, N=hop count

Table 5. Size in bytes of routing packets

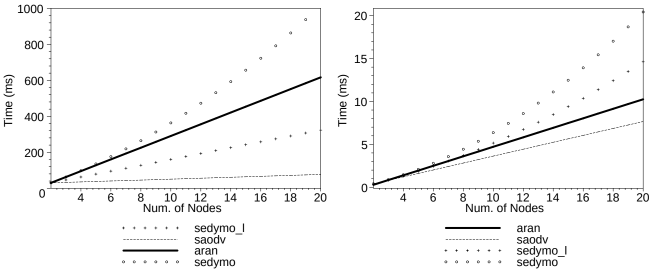
Although the message size increase due to security data is appreciable, the transmission energy derived from it is irrelevant compared to the high consume of the network interface in the basic idle state. For example, the energy spent in a DYMO network of 20 nodes is 5,027mJ/node, while in SEDYMO is 5,028mJ/node (the elapsed time is considered equal in both cases). Consequently, secure transmission cost overhead is depreciable in front of processing one.

Moreover it has to be emphasized that the minimum energy a device consumes in its network interface its really high Only one life second of a network interface card (nic) consumes more energy than the total cryptographic operations needed for a path discovery service of any of the studied protocols. To reduce consumption the nic can be set to doze or sleep state that costs $186mW$, yet increasing the response time of the device.

4.3 Time delay

Introducing security in a routing protocol reduces its performance because of the computation overhead that entails, and transmission delays due to the bigger messages of the protocol.

First of all we analyze the delay introduced by signature generations and verifications. Hash chain operations are not considered because they are fast constructs. Executing a 512-RSA signature takes $13.7ms$; the verification is faster, it lasts $1.3ms$ [7]. With this data we have generated a figure 4(a) that resumes the behavior of the analyzed protocols.



(a) Cryptography computations time delay

(b) Transmission time delay

Figure 4. Time delay

On the other hand, data transmission also introduces retards. Figure 4(b) shows time delay due to the transport of security data in a network with and effective rate of 5Mbps. The total time it takes to a node obtaining the response of a path discovery is approximately a 10% greater.

Table 6 sums up the resulting delay for a network with a diameter of 20 nodes. It is clear that the bottleneck is the processor speed and not the network. SEDYMO involves delays due to cryptographic processing that can not be underestimated. In the conclusions section we sketch possible solutions.

	Crypto proc.	Sec. data trans.	Total sec. overhead
ARAN	$616,8ms$	$10,2ms$	$627,0ms$
SAODV	$76,8ms$	$7,7ms$	$84,5ms$
SEDYMO	$1014,6ms$	$20,4ms$	$1035,0ms$
SEDYMO_l	$323,4ms$	$14,6ms$	$338,0ms$

Table 6. Example of time delays

5 Conclusions

In this paper we presented SEDYMO, a security enhancement of DYMO protocol that solves most of its security flaws and prevents and detects a vast range of misbehaviors. Up to our knowledge, its the first proposal to secure the whole functionality of DYMO, the IETF selected solution to become the standard reactive routing protocol for ad hoc networks.

SEDYMO fixes the exploits allowed by DYMO using public key cryptography, as other secure routing protocols that are reviewed in the article, but SEDYMO is more efficient than others because one single route discovery process updates the cross-routing paths of all intermediate nodes.

The security flaw of SEDYMO is the wormhole attack. It is difficult to guarantee the integrity of path lengths with a software-only approach. Nevertheless, specific countermeasures to this attack can be included in the network with the use of protocols such as SECTOR [4] or packed leashes [12].

The energy overhead owing to secure properties introduced in wireless ad-hoc routing protocols is predominated by the computing energy spent in cryptography operations. SAODV is the lighter one in terms of energy costs per path discovery process. However SAODV have and important security flaw that allows spoofing attacks and thus the corruption of routing tables. On the other hand ARAN and SEDYMO are more costly but robust protocols. SEDYMO works with a path accumulation model meaning that a single discovery process creates complete routing tables for all route intermediate nodes. In this sense, SEDYMO is very effective and optimal for networks where nodes communicate only with a portion of other nodes.

The results also show that despite the energy overhead caused by security, this consume is not relevant compared to other device expenses, i.e. the network interface. That is, computational power is not really limiting the battery lifetime.

On the other hand, the paper has analyzed time delays induced by cryptographic processing and data transmission. The first ones are the most significant in a 206MHz cpu. Although the performance of handheld devices tends to be weak and restricted, even relatively expensive schemes like the analyzed protocols are nowadays conceivable, with bearable delays in small and medium networks (about 20 nodes in diameter). Moreover, RSA signature generation speed can be improved around 2.5 times with the use of modified variants of the algorithm [3] that have better performances and are backwards-compatible with standard RSA.

Hardware implementations of aggregate signature and multisignature algorithms based on elliptic curves have to be considered in future work. Elliptic curves signature verification uses pairing-based cryptosystems. Pairing efficient solutions have been presented ([15]) and integration in smart cards is feasible.

References

[1] A. Boldyreva. Efficient threshold signature, multisignature and blind signature schemes based on the gap-diffie-hellman-group signature scheme. *Cryptology ePrint*, 2002.

[2] D. Boneh, C. Gentry, H. Shacham, and B. Lynn. Aggregate and verifiably encrypted signatures from bilinear maps. In *Proc. of Advances is Cryptology. Eurocrypt*, 2003.

[3] D. Boneh and H. Shacham. Fast variants of RSA. *RSA Laboratories Cryptobytes*, 5(1):1–8, 2002.

[4] S. Capkun, L. Buttyan, and J. Hubaux. SECTOR: Secure Tracking of Node Encounters in Multi-hop Wireless Networks. In *Workshop on Security of Ad Hoc and Sensor Networks (SASN)*, pages 21–32, 2003.

[5] I. Chakeres and C. Perkins. Dynamic MANET On-demand (DYMO) Routing Rrotocol. *IETF Internet Draft, v.06*, October 2006. (Work in Progress).

[6] I. D. Chakeres and J. P. Macker. Mobile ad hoc networking and the ietf. *SIGMOBILE Mob. Comput. Commun. Rev.*, 10(3):79–81, 2006.

[7] C. D.Westhoff, B.Lamparter and A.Weimerskirch. On digital signatures in ad hoc networks. *Wiley Journal European Transactions on Telecom.*, 16(5):411–425, October 2005.

[8] K. et al. A secure routing protocol for ad hoc networks. In *International Conference on Network Protocols (ICNP)*, pages 78–87, November 2002.

[9] S. Ghazizadeh, O. Ilghami, E. Sirin, and F. Yaman. Security-aware adaptive dynamic source routing protocol. In *IEEE Conference on Local Computer Networks (LCN)*, page 0751, Washington, DC, USA, 2002. IEEE Computer Society.

[10] M. Guerrero. Secure dynamic manet on-demand (sdymo) routing protocol. *IETF Internet Draft, v.00*, February 2006. (Work in Progress).

[11] M. Guerrero. Securing ad hoc routing protocols. *IETF Internet Draft, v.06*, september 2006. (Work in Progress).

[12] Y. Hu, A. Perrig, and D. Johnson. Packet leashes: A defense against wormhole attacks in wireless ad hoc networks. Technical report, Rice University, December 2001.

[13] Y. Hu, A. Perrig, and D. Johnson. Ariadne: A secure on-demand routing protocol for ad hoc networks. In *ACM Conference on Mobile Computing and Networking*, 2002.

[14] J.Kim and G. Tsudik. Srdp: Securing route discovery in dsr. In *MobiQuitous*, pages 247–260. IEEE Computer Society, 2005.

[15] T. Kerins, W. P. Marnane, E. M. Popovici, and P. Barreto. Efficient hardware for the tate pairing calculation in characteristic three. *Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, August 2005.

[16] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang. Providing robust and ubiquitous security support for mobile ad-hoc networks. In *International Conference on Network Protocols (ICNP)*, pages 251–260, 2001.

[17] L. Lamport. Password authentication with insecure communication. *Commun. ACM*, 24(11):770–772, 1981.

[18] L.Feeney and M.Nilsson. Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In *INFOCOM*, pages 1548–1557, 2001.

[19] A. Lysyanskaya, S. Micali, L. Reyzin, and H. Shacham. Sequential aggregate signatures from trapdoor permutations. *Cryptology ePrint Archive, Report 2003/091*, 2003.

[20] P. Papadimitratos and Z. Haas. Secure routing for mobile ad hoc networks. In *Commun. Networks and Distributed Systems Modeling and Simulation Conference (CNDS)*, 2002.

[21] R.Nachiketh, R.Srivaths, A.Raghunatan, and J.Niraj. Analysing the energy consumption of security protocols. *ACM ISLPED Conference*, pages 30–35, 2003.

[22] S.Micali, K.Ohta, and L.Reyzin. Provable subgroup signatures. Unpublished manuscript, 1999.

[23] S. Yi, P. Naldurg, and R. Kravets. Security-aware ad hoc routing for wireless networks. In *MobiHOC*, October 2001.

[24] L. Zhou and Z. J. Haas. Securing ad hoc networks. *IEEE Network*, 13(6):24–30, 1999.