BISS: Building Secure Routing out of an Incomplete Set of Security Associations

Srdjan Čapkun and Jean-Pierre Hubaux
Laboratory for Computer Communications and Applications (LCA)
School of Information and Communication Sciences (I&C)
Swiss Federal Institute of Technology Lausanne (EPFL)
CH-1015 Lausanne, Switzerland
srdan.capkun@epfl.ch, jean-pierre.hubaux@epfl.ch

ABSTRACT

We investigate secure routing in ad hoc networks in which security associations exist only between a subset of all pairs of nodes. We focus on source routing protocols. We show that to establish secure routes, it is in general not necessary that security associations exist between all pairs of nodes; a fraction of security associations is sufficient. We analyze the performance of existing proposals for secure routing in such conditions. We also propose a new protocol, designed specifically for ad hoc networks with an incomplete set of security associations between the nodes. We call this protocol BISS: a protocol for Building Secure Routing out of an Incomplete Set of Security Associations. We present a detailed analysis of this protocol, based on simulations, and show that it can be as secure as the existing proposals that rely on a complete set of security associations.[1]

Categories and Subject Descriptors

C.0 [**Computer-Communication Networks**]: [Security and protection]; C.2.2 [**Network Protocols**]: [Routing Protocols]

General Terms

Security

Keywords

Security, Ad Hoc Networks, Mobility, Security associations, Routing

1. INTRODUCTION

By definition, a mobile ad hoc network [31, 22, 23] does not rely on any fixed infrastructure; instead, all networking functions (e.g., routing, mobility management, etc.) are performed by the nodes themselves in a self-organizing manner. For this reason, designing routing protocols for mobile ad hoc networks is challenging and securing these protocols even more so.

So far, the problem of routing in ad hoc networks has been mainly studied in a non-adversarial setting, and only recently has the focus of research shifted to the design of secure routing protocols; researchers have already devised a number of proposals to secure both reactive (on-demand) and proactive routing protocols [18, 20, 19, 17, 29, 32, 16]. The authors of the most robust (but also of the most demanding) solutions generally assume that, prior to network operation, security associations exist between *all* pairs of nodes in the network. This means that either symmetric keys are shared between all nodes, or that the nodes know each others' authentic public or Tesla [30] keys; this assumption is important for avoiding the routing-security dependency loop. We elaborate this in more detail in Section 2.2.

Several solutions have been proposed for the initial key setup. One solution consists in pre-loading pairwise keys in all nodes to create all the security associations at the initialization. However, this approach makes the insertion of new nodes in the network very difficult. Hu, Perrig and Johnson [18] propose a solution to this problem. Their approach makes use of an on-line key distribution center and is thus very effective, although it requires a costly initialization phase and relies on the availability of (and connectivity to) the key distribution center.

In [11], we propose a system for the self-organized establishment of security associations based on mobility. We show that mobility can be used to set up security associations between nodes, including in order to secure routing. This mobility-based approach enables a more flexible setup of the security associations and requires only an off-line authority; the drawback, with respect to other approaches, is that the establishment of the security associations requires some time. As we will see, this problem is dramatically alleviated by the findings in this paper.

In many scenarios, it is unrealistic to assume that security associations have been established between all pairs of nodes prior to network operation. In practice, node mobility, network partitioning, and sporadic connectivity to other nodes

---

[1] The work presented in this paper was supported (in part) by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322 (`http://www.terminodes.org`).

WiSE'03, September 19, 2003, San Diego, California, USA.

or to key distribution centers will prevent nodes from establishing or timely renewing security associations with other nodes.

In this paper, we show that even if only a fraction of the security associations are established between nodes, routing can still be secured. We focus on on-demand routing protocols, in which a node attempts to discover a route to some destination only when it has a packet to send to that destination; more specifically, we assume a source routing protocol and consider DSR as an example. First, we show how Ariadne [18] can cope with an incomplete set of security associations (although it was not designed with this objective in mind). Second, we propose a new protocol that we call BISS: (**B**uilding Secure Routing out of an **I**ncomplete **S**et of **S**ecurity Associations). We present a detailed analysis of these protocols, based on simulations.

The work presented in this paper is a part of the Terminodes Project [4, 21].

The organization of the paper is the following. In Section 2, we survey the related work. In Section 3, we provide the model of our system. In Section 4, we describe our solution and we analyze it in Section 5. Finally, we conclude the paper in Section 6.

2. STATE OF THE ART

2.1 Setting up security associations

Several solutions have been proposed specifically to set up security associations for secure routing in ad hoc networks.

In [35], Zhou and Haas propose a distributed public-key management service for ad hoc networks. The service, as a whole, has a public/private key pair K/k, that is used to verify/sign public-key certificates of the network nodes. The private key k is divided into n shares using a $(n, t+1)$ *threshold cryptography* scheme, and the shares are assigned to n arbitrarily chosen nodes, called servers. Signatures are then generated by a collaborative action of the servers. The application of threshold cryptography ensures that the system can tolerate a certain number $t < n$ of compromised servers, in the sense that at least $t+1$ partial signatures are needed to compute a correct signature. Unfortunately, the proposal has two major drawbacks: First, it requires an authority to empower the servers. Second, it assumes that some of the nodes must behave as servers, which does not seem to be realistic, at least in civilian applications.

Kong et al. [24] propose a system where a single private network key is shared between all network nodes and is used to sign certificates to network nodes. This system allows *any* node to carry a share of the private key of the service. The advantage of this system is availability of the service, and an interesting novelty is that any node not yet possessing a share can obtain a share from any group of at least $t+1$ nodes that already possess a share. The disadvantage is that the first $t+1$ nodes must be initialized by a trusted authority; it is also unclear how the value of t can be changed in case the overall number of nodes significantly increases (or decreases). Furthermore, the system seems to be vulnerable to the Sybil attack [12]: an attacker can take as many identities as necessary to collect enough shares and reconstruct the system's private key.

A different approach, proposed by Asokan and Ginzboorg [2], is based on a shared password. In this approach, nodes willing to establish a secure session must share a *prior context*. The proposed solution is the following: A fresh password is chosen and shared among users (e.g., it is written on a blackboard). To prevent *dictionary attacks* [26], this password is not used directly; instead, the authors propose to make use of *password-authenticated key exchange* by which the parties derive a strong shared key starting from only a weak secret (i.e., the password). This approach has the drawback of being somewhat cumbersome, as it requires the users to type the password in their personal device and to be present in the same room.

Another approach, designed for the address ownership problem in Mobile IPv6, is described by Montenegro and Castelluccia in [27] and by O'Shea and Roe in [28]. Their idea is to derive the IP address of the node from its public key: first, the public key is hashed with a cryptographic hash function, and then, (part of) the hash value is used as a part of the IP address of the node. The advantage is that there is no longer need for certificates that bind the node's address to its public key, since one is derived from the other in a cryptographically verifiable way. In [5], Bobba et al. use SUCV identifiers to implement a secure binding between IP addresses and keys that is independent of any trusted security service. They illustrate their solution with the Dynamic Source Routing (DSR) protocol and argue that the solution is applicable to other protocols such as SEAD and Ariadne. In that approach, however, it is unclear how the network handles node membership.

Hu, Perrig and Johnson [18] propose to make use of Tesla [30] authentication mechanism and to distribute Tesla keys to the nodes by means of an on-line key distribution center. This approach is very effective, although it requires a costly initialization phase.

In [15] Eschenauer and Gligor propose a random key predistribution scheme for sensor networks. Its operation is briefly described as follows. A random pool of keys is selected from the key space. Each sensor node receives a random subset of keys from the key pool before deployment. Any two nodes able to find one common key within their respective subsets can use that key as their shared secret to initiate communication. This approach is extended by Chan, Perrig and Song in [8].

In [11], we have proposed a key establishment technique that benefits from mobility and uses node encounters to establish security associations. In that work, we proposed protocols that allow the implementation of our system with both symmetric and public-key cryptography. We observed the rate of the establishment of security associations both analytically and by simulations with various mobility models.

In [10], we have proposed a self-organized public-key management solution for mobile ad hoc networks. In this proposal, each node maintains a repository of public-key certificates, and the authentication is performed by merging nodes' repositories and by finding appropriate chains of certificates between the nodes' public keys within their merged repositories.

2.2 Secure routing

The assumptions about security associations and the way that they are used to secure routing vary a lot from one protocol to another. Here, we give an overview of the ways security associations are used to secure routing in ad hoc networks. We focus on secure on-demand routing protocols.

In [29], Papadimitratos and Haas assume that to securely route, it is sufficient to establish a security association only between the sender and the receiver. They show that their proposal prevents a number of attacks, but the proposed protocol is still vulnerable to some *active* attacks [17].

Sanzgiri et al. [32] consider a different scenario, in which nodes authenticate routing information coming from their neighbors, but the sender and the receiver do not authenticate all the nodes on the routing path. This approach effectively protects networks from passive attackers and attackers that cannot compromise legitimate nodes, but is still vulnerable to a number of attacks if one or more nodes get compromised.

Hu, Perrig and Johnson [18] propose a more robust protocol, which is more demanding in terms of security associations. In their approach, they assume that security associations exist between all pairs of nodes (through authentic public or Tesla [30] keys, or by shared secret keys). This allows both the sender and the receiver to authenticate all the nodes on the chosen routing path.

We follow this latter approach, but we assume that each node has security associations established only with a fraction of the other nodes. This assumption is desirable, as full distribution of keys in ad hoc networks cannot always be complete before network operation, either due to unavailability of servers, network partitioning, or simply because the establishment of security associations takes some time [11]. Moreover, a partial establishment of security associations, in some applications, can also be a lasting network characteristic. Further incompleteness in the key distribution might be introduced by rekeying delays, or by probabilistic key-distribution schemes [15, 10].

An example of key distribution system in which security associations are established between only a fraction of pairs of nodes is our already mentioned mobility-based approach [11]. The speed of establishment of security associations with this technique depends on the rate of node encounters; thus, given specific mobility patterns, some nodes will only rarely meet and exchange cryptographic material necessary for the establishment of security associations. Furthermore, if security associations are limited in time, if some associations expire, they might stay "broken" for a while before network conditions allow them to be renewed.

More work in the area of ad hoc network security has been reported, notably in [33, 9, 34, 1, 25, 3, 10, 6, 7].

3. SYSTEM MODEL

We consider an ad hoc network of mobile nodes, controlled by an *off-line* central authority. The authority controls network membership and decides which nodes can join the network. We assume that each node has a unique identity (e.g., assigned to it by the authority); Furthermore, each node holds a certificate signed to it by the authority, which binds the node's identity and its public key. We also assume that each node holds a correct public key of the authority, so that it can verify the correctness of the certificates presented by other nodes. Each node is able to generate cryptographic keys, to check signatures, and more generally to accomplish any task required to secure its communications (including to agree on cryptographic protocols with other nodes).

We use the following notation: we denote node ids by capital letters (e.g., U); the private and public keys of a node U by K_U and PK_U, respectively; a shared secret key between two nodes U and V by k_{UV}; the signature on a message m with private key K_U by $(m)_{K_U}$; the message authentication code with a key k_{UV} over a message m by $MAC_{k_{UV}}(m)$; and the certificate, signed by the central authority, binding PK_U with U, by c_U.

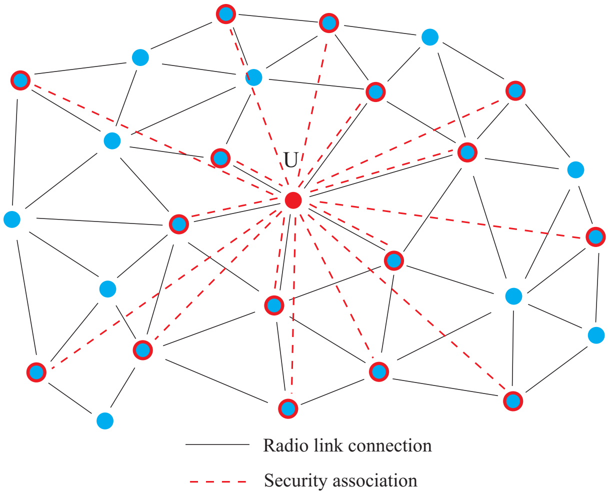


Figure 1: Local and long-range security associations of node U.

If a node U holds a certificate signed by the central authority that binds node V with its (V's) public key, and node V holds a certificate signed by the central authority that binds node U with its (U's) public key, then we say that there exists a security association from U to V. In our system, we further assume that in parallel with the exchange of their public-key certificates, the nodes establish a shared secret key k_{UV} between them that they will use for later communication. Hence, security associations are always symmetric.

We assume that two nodes that are in the power range of each other exchange certificates that contain their public keys and establish a security association. This one-hop establishment of security associations breaks the well-known routing-security interdependence cycle [18, 11]: Security associations cannot be established over multiple hops as the routing protocol does not operate securely (because security associations are not established yet). This means that if two nodes want to establish a security association over a multi-hop route, their packets are at risk to be sent through false routes, or simply dropped. Besides these "local" security associations, nodes have security associations established with other, more distant, nodes in the network. These "long distant" security associations are established either by the mobility-based approach or by some other means (e.g., by pre-loading keys through a key distribution center). This is illustrated on Figure 1.

By p^U we denote the percentage of security associations that node U established with other network nodes. We define the percentage p of security associations in the network as the average of nodes' percentages of security associations. The percentage p of the security associations depends on numerous factors, like node mobility [11], or connectivity of the nodes with the key distribution center.

4. BUILDING SECURE ROUTING OUT OF SECURE LINKS

It is important to notice that to perform route discovery, both the initiator and the target node need to be able to authenticate the nodes on the chosen route; otherwise, route discovery fails.

Our main goal is to analyze if the network is *securely* connected, meaning if any pair of nodes can find at least one *secure* route between themselves in the network. By a secure route we mean the route on which all the nodes can be authenticated, both by the sender and the receiver. We denote by $\mathcal{N}$ the set of network nodes, where $n = |\mathcal{N}|$, by A the size of the node deployment area, and by $\lambda = n/A$ the node density.

We denote by θ the fraction of pairs of nodes between which there exists a route in the network. Similarly, we define θ_s as the fraction of pairs of nodes between which there exists a *secure* route in the network. By $d(U,V)$ and $d_s(U,V)$ we denote the lengths of the shortest route and the shortest secure route between nodes U and V in the network, respectively.

We define the average secure routing capability c and the average secure routing length ratio r as follows

$$c = \frac{\theta_s}{\theta}$$

$$r = \frac{\sum_{U,V \in \mathbb{S}} \frac{d_s(U,V)}{d(U,V)}}{|\mathbb{S}|}$$

where $\mathbb{S} = \{(U,V) \in \mathcal{N} : d_s(U,V) < \infty, d(U,V) < \infty\}$. Both the average secure routing capability and the length ratio depend on the node density λ, the node power range, and the percentage p of the security associations.

4.1 Direct route authentication

As we already mentioned, the most common security assumption in the existing secure routing protocols is that security associations are established between all pairs of network nodes prior to the run of the routing protocol. Thus, it is assumed that through these security associations, both the initiator of the route request and the target are able to *directly* authenticate all the nodes on the secure route that they establish between them. We call this way of route authentication the *direct route authentication.* One example of the direct route authentication protocol is Ariadne [18].

In Ariadne, when an initiator broadcasts a `ROUTE REQUEST`, this request is authenticated and then forwarded by each node that receives it, provided that the node hasn't received the same request before. If the node cannot authenticate the `ROUTE REQUEST`, it drops the packet and does not forward the request. This mechanism also enables nodes to prevent excessive `ROUTE REQUESTS`. Each forwarding node adds itself to the list of nodes within the `ROUTE REQUEST`. This operation is secured such that no node can remove another node from the list and that the target node can authenticate all the nodes in the `ROUTE REQUEST` that it receives. Out of the received `ROUTE REQUEST`s, the target node chooses the most favorable (authentic) route and sends a `ROUTE REPLY` to the initiator. This reply is then routed back along the chosen route to the initiator, which authenticates all the nodes in the `ROUTE REPLY` list (the ones on the chosen path).

In Ariadne, the authors explain that route data authentication can be based on three techniques: Tesla [30], *digital signatures* and *message authentication codes (MACs).* Route maintenance is secured by the authentication of `ROUTE ERROR` messages on the side of the sender. This prevents malicious nodes from issuing false `ROUTE ERROR` messages.

In Section 5.1, we analyze the performance of direct route authentication protocols in networks with an incomplete set of security associations. We show that in such networks, these protocols can ensure some level of secure routing capability and we further explore how the performance changes with different node densities and percentages of security associations.

4.2 Indirect route authentication

Direct route authentication protocols are very effective in networks with a complete set of security associations, but they are not optimal in networks where the set of security associations is incomplete.

We thus propose a slightly different approach to secure routing, specifically designed for networks with an incomplete set of security associations. We illustrate this approach with a new protocol that we call BISS (**B**uilding Secure Routing out of an **I**ncomplete **S**et of **S**ecurity Associations).

In BISS, the sender and the receiver can establish a secure route, even if, prior to the route discovery, only the receiver has security associations established with all the nodes on the chosen route. Thus, the receiver will authenticate route nodes directly through security associations (as in the direct route authentication approach). The sender, however, will authenticate directly the nodes on the route with which it has security associations, and indirectly (by exchange of certificates) the nodes with which it does not have security associations.

As we show in Section 5.1, this change from the direct to indirect route authentication increases the secure routing capability of the network and reduces its secure routing length ratio. In this section, we also assess the increase of the communication and computational cost.

In BISS, we use two route data authentication mechanisms: message authentication codes (MACs) and digital signatures.

We now describe the basic operation of the BISS protocol. The operation of BISS `ROUTE REQUEST` relies on mechanisms similar to direct route authentication protocols. When an initiator sends a `ROUTE REQUEST`, it signs the request with its private key and includes its public key PK_I in the request along with a certificate c_I signed by the central authority binding its id with PK_I. This enables each node on the path to authenticate the initiator of the `ROUTE REQUEST`. The `ROUTE REQUEST` message contains the id of the target node. The node that receives this `ROUTE REQUEST` authenticates the initiator (by verifying the signature on the message), and tries to authenticate the target directly through security associations that it has. Only if a node can successfully authenticate both the initiator and the target will the node broadcast the message further. Here again, note that the node will process the `ROUTE REQUEST` only if it did not receive the same request before.

In BISS, we use similar route request data authentication mechanisms as in Ariadne. This means that each node must have a security association with the target and computes a MAC over the received message with the key that it shares

ROUTE REQUEST

$$\begin{array}{rl} \ldots & \\ W \rightarrow T: & \langle REQUEST, I, T, rid, PK_I, H, c_I, \sigma_I, \{U, V, W\}, \{M_{UT}, M_{VT}, M_{WT}\}\rangle \end{array}$$

ROUTE REPLY

$$\begin{array}{rl} T: & \text{authenticate } W, V, U \text{ through security associations; check } \sigma_I \\ : & \text{choose the most favorable route (in this case } \{U, V, W\}) \\ : & \sigma_T = (REPLY, I, T, rid, \{U, V, W\})_{K_T} \\ T \rightarrow W: & \langle REPLY, I, T, rid, \{\sigma_T\}, \{U, V, W\}, \{PK_T\}, \{c_T\}\rangle \\ & \\ W: & \sigma_W = (REPLY, I, T, rid, \{U, V, W\})_{K_W} \\ W \rightarrow V: & \langle REPLY, I, T, rid, \{\sigma_T, \sigma_W\}, \{U, V, W\}, \{PK_T, PK_W\}, \{c_T, c_W\}\rangle \\ & \\ V: & \sigma_V = (REPLY, I, T, rid, \{U, V, W\})_{K_V} \\ V \rightarrow U: & \langle REPLY, I, T, rid, \{\sigma_T, \sigma_W, \sigma_V\}, \{U, V, W\}, \{PK_T, PK_W, PK_V\}, \{c_T, c_W, c_V\}\rangle \\ & \\ U: & M_{UI} = \text{MAC}_{k_{UI}}(REPLY, I, T, rid, \{U, V, W\}) \\ U \rightarrow I: & \langle REPLY, I, T, rid, \{\sigma_T, \sigma_W, \sigma_V, M_{UI}\}, \{U, V, W\}, \{PK_T, PK_W, PK_V\}, \{c_T, c_W, c_V\}\rangle \\ & \\ I: & \text{authenticate } T \text{ with } \sigma_T, W \text{ with } \sigma_W, V \text{ with } \sigma_V \text{ and } U \text{ with } M_{UI} \end{array}$$

Figure 2: An example of run of the BISS route discovery protocol. In this example the route request is initiated by the node I to the target node T, through the intermediary nodes U, V and W. We assume that all three intermediary nodes have a security association established with the target, but only node U has a security association with the initiator. We also assume that the initiator and the target do not have a security association established between them.

with the target. The node then adds the computed MAC to the message and broadcasts it further. To prevent the removal or replacement of the nodes from the route, we use per-hop hashing [18].

When the ROUTE REQUESTs reach the target, the target chooses the most favorable route and sends back the ROUTE REPLY through the chosen route (with the reverse order of nodes). Notice that only the routes containing legitimate (authenticated) nodes will be considered by the target, and that the target can easily check their authenticity because of the security associations.

The ROUTE REPLY message contains the list of nodes of the chosen route; it is protected either with the MAC that is computed with the shared key k_{IT} between the initiator and the target, or with the signature on the message, signed by the target's private key K_T. This enables the initiator to authenticate the ROUTE REPLY data.

However, until this protocol stage, only the target authenticated the nodes on the route (upon receiving the ROUTE REQUEST). Whether it is necessary for the source to do the same, depends on the security assumptions about the system. In Ariadne, the assumption is that the initiator directly authenticates these nodes.

In BISS, if the initiator can *trust* the target for having authenticated the nodes of the route, it is not necessary that the initiator authenticates the nodes itself. If the security requirements of the system are such that the initiator also must authenticate each node on the route, the following mechanism is used: Each node along the ROUTE REPLY path checks if it has a security association established with the initiator. If it does, it computes a MAC over the received message and attaches it to the reply along with its id (the same as in the ROUTE REQUEST). If the route node does not have a security association established with the initiator, it signs the message with its private key and attaches this signature to the message, along with its public-key certificate. When the initiator receives the ROUTE REPLY message, it verifies the MACs and the signatures and accepts or rejects the route, depending whether the verification succeeded or failed.

The BISS route maintenance mechanism uses ROUTE ERROR authentication by the sender. A node that cannot forward a received packet will return the ROUTE ERROR message to the packet source along the reversed route through which it received the packet. This ROUTE ERROR message contains either the MAC computed over the message with the key shared between the node and the source of the original packet, or the signature of the message, signed with the public key of the node and authenticated by the certificate from the central authority.

In Figure 2 we illustrate the flow of the ROUTE REPLY from the target back to the route request initiator. This example shows the execution of BISS, initiated by the node I to the target node T, through the intermediary nodes U, V and W. We assume that all three intermediary nodes have a security association established with the target, and that only node U has a security association with the initiator. We further assume that the initiator and the target do not have a security association between them. The figure illustrates the content of the BISS ROUTE REQUEST and ROUTE REPLY messages. The ROUTE REQUEST message contains the request field, the ids of I and T, the route id (rid), the public key PK_I of I, the certificate c_I that certifies this key, the signature σ_I by I on the message, the hash value H for the per-hop hashing verification, the node id list, and the message authentication code list $\{M_{UT}, M_{VT}, M_{WT}\}$. When it reaches I, the ROUTE REPLY message contains: the reply field, the initiator id I, the target id T, the route id *rid*, the list of signatures and MACs $\{\sigma_T, \sigma_W, \sigma_V, M_{UI}\}$, the id*s* of the nodes on the route, the list of public keys $\{PK_T, PK_W, PK_V\}$ of the nodes which do not have security associations with I, and the list of public-key certificates $\{c_T, c_W, c_V\}$ certifying these keys.

What differs between the direct and indirect route authentication is the authentication of the relaying nodes by the initiator. In the direct approach, even if a single node in the `ROUTE REPLY` cannot be directly authenticated (through security associations) by the initiator, the route is discarded. In BISS, each node on the chosen route authenticates itself to the initiator by signing a message with its private key and by attaching a certificate for the corresponding public key to the route reply message (nodes W and V in the example).

By allowing the set of security associations in the network to be incomplete, BISS enables, in terms of secure routing capability, a more effective routing than direct route authentication protocols. This is simply because in BISS two nodes can establish a route between themselves even if only the target has security associations established with the nodes on the secure route. In direct route authentication protocols, prior to route establishment, both the initiator and the target need to have security associations established with the nodes on the secure route.

An important, beneficial side-effect of BISS is that the route discovery protocol naturally increases the number of security associations; for example, in Figure 2, when node I receives the `ROUTE REPLY`, it receives the public keys of nodes T, W and V; conversely, these three nodes have previously obtained the public key of I by means of a `ROUTE REQUEST`. Therefore, node I can establish a security association with T, W and V (e.g., through the same route). As a result, if some of these nodes happen to be involved in a secure route establishment in the future, they will be able to rely on the MAC (through security associations) rather than on the signature scheme, at least until the end of the next rekeying period.

The presented implementation of BISS uses public-key cryptography. However, this protocol can be equally implemented with symmetric-key cryptography only (e.g., this can be achieved by replacing the nodes' public keys of nodes with authentic Tesla keys). As BISS is meant to be used with an off-line central authority, public-key certificates are still necessary (to certify nodes' Tesla or public keys). If BISS were implemented with an on-line central authority, public-key cryptography could be avoided altogether.

5. EVALUATION OF BISS

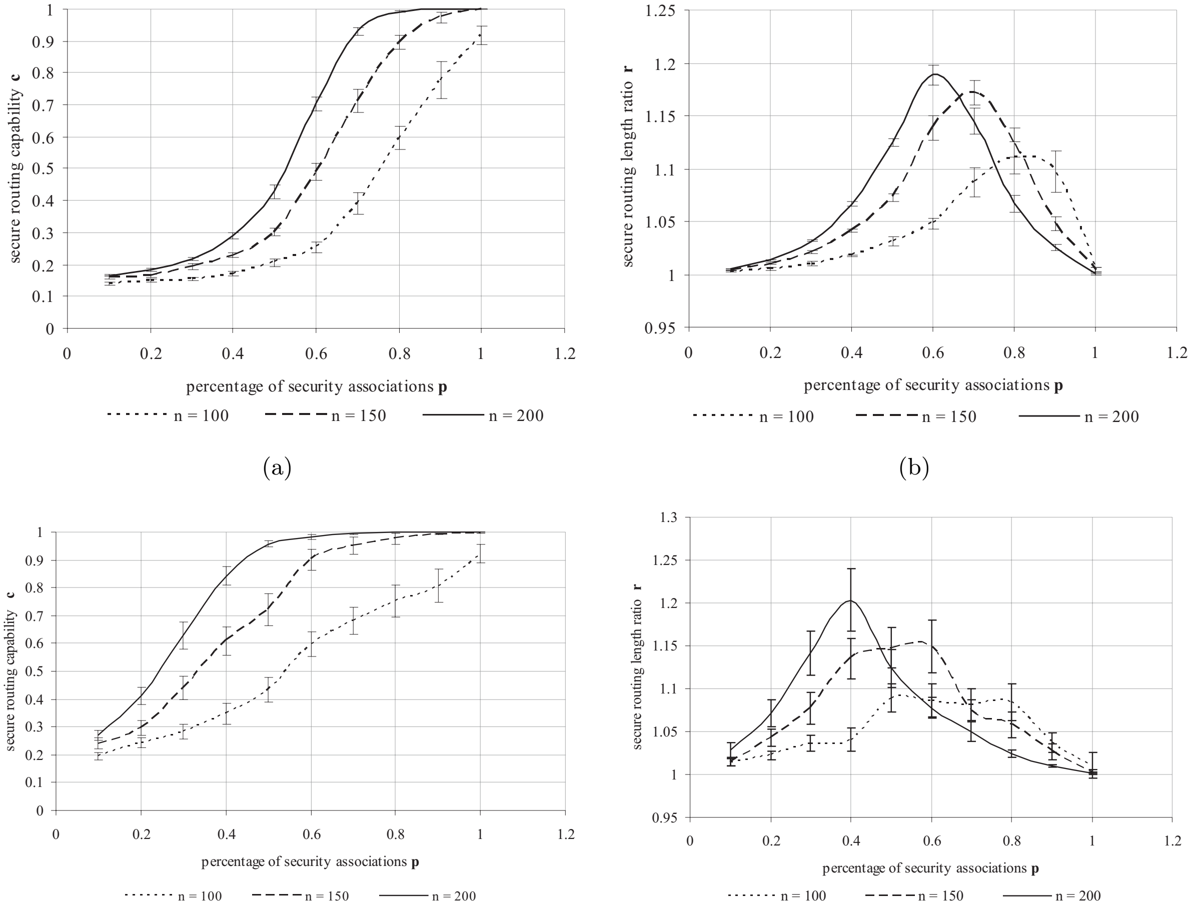
5.1 Simulation results

In this section, we show our simulation results. We observe two values: the secure routing capability c and the secure routing length ratio r. We consider the following scenarios: area size of 1000m×1000m; node densities of 100/km^2, 150/km^2, 200/km^2; nodes are uniformly placed; power range of 150m. We assume that the security associations established between the nodes are randomly assigned and independent. We simulated the behavior of both direct route authentication and BISS protocols. The results of our simulations are shown on Figure 3. Figure 3a shows the secure route capability with direct route authentication protocols for various values of security associations; Figure 3c shows the same for BISS. Figure 3b shows the secure route length ration with direct route authentication protocols for various values of security associations; Figure 3d shows the same for BISS. From these figures, we observe that the BISS protocol performs better than the direct route authentication protocols (both in terms of secure routing capability and length ratio). Nevertheless, it should be noted that with direct route authentication protocols, network nodes can still route securely even if as much as 15% of security associations between nodes are not established (for $n = 200$).

In Figures 3b and 3d, we observe the secure routing length ratio with both direct route authentication protocols and with BISS. This ratio represents the increase in the route length with respect to insecure routes. In networks with a complete set of security associations ($p = 1$), there is naturally no increase in route length with respect to non-secured routes ($r = 1$). In networks with an incomplete set of security associations, this increase depends on the node density and the percentage of security associations. Our simulations show that for the percentages of security associations for which the secure routing capability $c = 1$, the route length increase is higher in the case of direct route authentication protocols than with BISS. The peak values of the secure routing length ratios (Figures 3b and 3d) are the highest values of the route length increase. These peak values occur when the percentage of the security associations is such that the secure routing capability is between 0.7 and 0.8. This is expected as if the secure capability is lower, fewer paths will be found, and if it is higher, the paths that are found have shorter (almost shortest path) lengths.

These results indicate that for networks with an incomplete set of security associations, BISS is indeed more appropriate than the direct route authentication protocols. BISS introduces some additional communication and computational overhead with respect to direct route authentication protocols, but only in the cases when direct authentication cannot be used; otherwise, BISS incurs the same overhead as direct authentication protocols. In BISS, unlike in direct route authentication protocols, route reply messages need to be signed by the nodes on the route (in the case that they do not have a security association with the initiator) and each signing node needs to attach its public key ($\approx$ 1024bits) and public-key certificate ($\approx$ 500bytes) to the message. We note, however, that some nodes on the route may share a security association with the initiator, in which case these nodes do not introduce any additional communication overhead.

We observe that the node density is an important factor for secure routing, meaning that the higher the node density, the lower the fraction of security associations required to securely route. Following the work of Dousse, Baccelli, Thiran and Hasler [14, 13] on connectivity and critical density of ad hoc networks, we observe the secure critical density of an ad hoc network, for various percentages of security associations. By critical density of a network, we mean the lowest node density at which the network is fully connected. Equivalently, we define the secure critical density λ_{sc} of a network as the lowest node density at which the network is securely connected. The secure critical density of a network depends on the percentage of security associations established in the network and on the underlying secure routing protocol. The secure critical node density is thus the node density at which the fraction θ_s of pairs of nodes between which there is a secure route is equal to 1. On Figure 4 we show the dependency of θ_s on the node density, for various



(a) (b)

(c) (d)

Figure 3: Secure routing performance with direct route authentication and with BISS, as a function of the percentage of security associations p (the figures are shown with 95% confidence intervals); a) secure routing capability c with direct authentication routing; b) secure routing length ratio r with direct authentication routing; c) secure routing capability c with BISS; d) secure routing length ratio r with BISS.

percentages of security associations, with BISS. We notice that if the density is 200/km^2, then the percentage of security associations must be 80% in order for all nodes to be able to securely communicate; however, with a density of 400/km^2, this proportion can be as low as 30%. These results are very encouraging, especially if security associations are established with the mobility-based approach [11]. As an example, we consider a scenario in which only 40% of security associations need to be established for nodes to route securely (node density $\lambda = 350$, Figure 4). In this scenario, the time until the network is fully operational, with the mobility based approach, reduces from 10,000s (around 3h) (the time needed to establish all security associations), to 1000s (around 15min) (the time needed to establish 40% of security associations) [11].

These results are not surprising, but they do show that routing in networks with an incomplete set of security associations is indeed possible, provided that the network is sufficiently dense.

5.2 Security analysis

In this section, we analyze the security of BISS, showing how it resists a number of attacks. We consider several types of attacks. The first type are passive attacks, in which a malicious node tries to gain some advantage by observing routing information of data communicated between the nodes. Most passive attacks can be prevented in BISS by protecting the content of the packets by strong encryption. However, some attacks are still possible, such as attacks against privacy, traffic analysis, etc. We do not consider these attacks in this paper.

The second type of attacks are active attacks. An active attacker eavesdrops and injects packets into the network, in order to disrupt or control communication between other nodes. We differentiate active attackers according to the number of nodes they control and the number of honest nodes they compromised. We assume that if an attacker has compromised an honest node, then it can perform all cryptographic operations as the honest node, and it possesses all its private/shared keys.

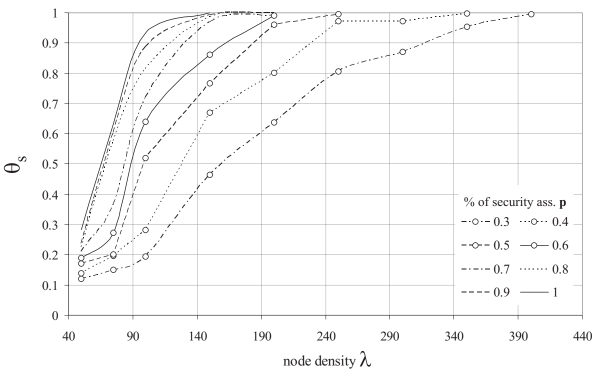


Figure 4: The fraction θ_s of pairs of nodes between which there is a secure route as a function of the node density, for various percentages of security associations, with BISS. Size of the area $A = 1000m \times 1000m$, power range$= 150m$.

BISS effectively prevents most active attacks from attackers that do not control any of the legitimate network nodes. This is achieved by mutual authentication of the initiator, target, and network nodes during the route establishment.

Attacks from attackers that control one or more legitimate network nodes are much more difficult to thwart and range from simple extensive route requests to wormhole attacks. With respect to these attacks, BISS exhibits the same resilience as Ariadne, as the security of the route establishment in both protocols assumes authentication between the same entities at the same stages of protocol execution, but performed with different cryptographic primitives and communication assumptions. A detailed analysis of Ariadne security can be found in [18].

6. CONCLUSION

In this paper, we have investigated secure routing in mobile ad hoc networks with an incomplete set of security associations, assuming a source routing protocol. We have shown that with existing protocols such as Ariadne, secure routing is still possible even with an incomplete set of security associations, provided that the percentage of security associations is sufficiently high.

We have proposed an optimization of the existing protocols that we call BISS. The main novelty of BISS is that it is the first protocol designed specifically for networks with an incomplete set of security associations. Thus, in BISS, the route request initiator authenticates the route nodes not only by means of security associations, but also by exchanging certificates with the nodes of the route. Whereas in the existing protocols (e.g., Ariadne), both the initiator and the target need to have security associations established with all the nodes on a secure route, in BISS, only the target node needs to have security associations with the route nodes. Nevertheless, BISS features the same level of security as Ariadne (albeit at a higher cryptographic and communication cost for the first established secure routes).

In our analysis, we have observed the percentage of the pairs of nodes that can communicate securely for a given percentage of security associations and for certain node densities, with the direct route authentication and the BISS protocols. We have shown that all conditions being equal, a higher percentage of nodes can route securely with BISS than with direct route authentication protocols. Moreover, we have observed that the routing paths are shorter with BISS than with direct route authentication protocols.

Our analysis also illustrates the influence of the node density on the secure routing capability of the network. Our findings are in line with our intuition, as they show that the higher the node density, the higher the secure routing capability of the network. We have shown that with BISS, all nodes can route securely, even if as little as 30% of the security associations are established, provided that the node density is sufficiently high (in this case 400/km^2, with a power range of 150m).

Moreover, we have shown that BISS transforms the problematic routing - security dependency loop into a virtuous circle: the more routes are established with BISS, the more security associations are created, and the easier becomes the later establishment of secure routes.

Recently, we have shown [11] that mobility can be exploited to establish the security associations between the nodes; the drawback, however, is that this operation requires a certain amount of time, which depends notably on the size of the network and on the speed of the nodes. Not surprisingly, most of the security associations are established rather quickly, but a few ones require much more time. As it is able to cope with only a partial set of the security associations, BISS substantially reduces the relevance of this drawback: the number of security associations BISS requires to be operational can be established in an amount of time which is *more than one order of magnitude* smaller than for the complete set.

In the future, we intend to study in more detail the cryptographic and communication overheads of BISS. We also aim to devise a solution to the same problem for other on-demand routing protocols (e.g., AODV), and for proactive routing protocols.

7. ACKNOWLEDGEMENTS

We thank Adrian Perrig for his valuable comments on this work. We also thank EPFL diploma student Mehdi Touati for his contribution to the simulation results.

8. REFERENCES

[1] Y. Amir, Y. Kim, C. Nita-Rotaru, and G. Tsudik. On the Performance of Group Key Agreement Protocols. In *Proceedings of ICDCS*, 2002.

[2] N. Asokan and P. Ginzboorg. Key Agreement in Ad Hoc Networks. *Computer Communications*, 23:1627–1637, 2000.

[3] N. Ben Salem, L. Buttyán, J.-P. Hubaux, and M. Jakobsson. A charging and rewarding scheme for packet forwarding in multi-hop cellular networks. In *Proceedings of MobiHoc*, 2003.

[4] L. Blažević, L. Buttyán, S. Čapkun, S. Giordano, J.-P. Hubaux, and J.-Y. Le Boudec. Self-Organization in Mobile Ad Hoc Networks: The Approach of Terminodes. *IEEE Communications Magazine*, June 2001.

[5] R.B. Bobba, L. Eschenauer, V.D. Gligor, and W. Arbaugh. Bootstrapping Security Associations for

Routing in Mobile Ad-Hoc Networks. Technical Report TR 2002-44, University of Maryland, May 2002.

[6] L. Buttyán and J.-P. Hubaux (Eds). Report on a Working Session on Security in Wireless Ad Hoc Networks. *Mobile Computing and Communications Review*, 7(1), 2003.

[7] L. Buttyán and J.-P. Hubaux. Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks. *ACM/Kluwer Mobile Networks and Applications (MONET)*, 8(5), October 2003.

[8] Haowen Chan, Adrian Perrig, and Dawn Song. Random key predistribution schemes for sensor networks. In *IEEE Symposium on Security and Privacy*, May 2003.

[9] M. Corner and B. Noble. Zero-interaction authentication. In *Proceedings of MobiCom*, 2002.

[10] S. Čapkun, L. Buttyán, and J.-P. Hubaux. Self-Organized Public-Key Management for Mobile Ad Hoc Networks. *IEEE Transactions on Mobile Computing*, 2(1), January-March 2003.

[11] S. Čapkun, J.-P. Hubaux, and L. Buttyán. Mobility Helps Security in Ad Hoc Networks. In *Proceedings of MobiHoc*, 2003.

[12] J. Douceur. The Sybil attack. In *Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS)*, 2002.

[13] O. Dousse, Francois Baccelli, and P. Thiran. Impact of interferences on connectivity in ad hoc networks. In *Proceedings of Infocom*, San Francisco, April 2003.

[14] O. Dousse, P. Thiran, and Martin Hasler. Connectivity in ad-hoc and hybrid networks. In *Proceedings of Infocom*, pages 1079–1088, New York, June 2002.

[15] L. Eschenauer and V. Gligor. A key-management scheme for distributed sensor networks. In *Proceedings of the 9th ACM conference on Computer and Communications Security*, 2002.

[16] M. Guerrero Zapata and N. Asokan. Securing Ad Hoc Routing Protocols. In *Proceedings of the ACM Workshop on Wireless Security (WiSe)*, September 2002.

[17] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Packet leashes: A defense against wormhole attacks in wireless networks. In *Proceedings of IEEE Infocom*, April 2003.

[18] Y.-C. Hu, A. Perrig, and D. B. Johnson. Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks. In *Proceedings of MobiCom*, September 2002.

[19] Y.-C. Hu, A. Perrig, and D. B. Johnson. Efficient Security Mechanisms for Routing Protocols. In *Proceedings of NDSS*, February 2003.

[20] Y.-C. Hu, D. B. Johnson, and A. Perrig. Secure efficient distance vector routing in mobile wireless ad hoc networks. In *Proceedings of (WMCSA)*, June 2002.

[21] J.-P. Hubaux, Th. Gross, J.-Y. Le Boudec, and M. Vetterli. Toward Self-Organized Mobile Ad Hoc Networks: The Terminodes Project. *IEEE Communications Magazine*, January 2001.

[22] D. B. Johnson. Routing in Ad Hoc Networks of Mobile Hosts. In *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications*, December 1994.

[23] J. Jubin and J.D. Turnow. The DARPA Packet Radio project. *Proceedings of the IEEE*, 1987.

[24] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang. Providing robust and ubiquitous security support for mobile ad hoc networks. In *Proceedings of the 9th International Conference on Network Protocols (ICNP)*, November 2001.

[25] L. Lazos and R. Poovendran. Energy-Aware Secure Multicast Communication in Ad-hoc Networks Using Geographic Location Information. In *Proceedings of IEEE International Conference on Acoustics Speech and Signal Processing*, 2003.

[26] A. Menezes, P. van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997.

[27] G. Montenegro and C. Castelluccia. Statistically Unique and Cryptographically Verifiable (SUCV) Identifiers and Addresses. In *Proceedings of NDSS*, 2002.

[28] G. O'Shea and M. Roe. Child-proof authentication for MIPv6 (CAM). *ACM Computer Communications Review*, April 2001.

[29] P. Papadimitratos and Z.J. Haas. Secure Routing for Mobile Ad Hoc Networks. In *Proceedings of CNDS*, January 2002.

[30] A. Perrig, R. Canetti, J.D. Tygar, and D. Song. The TESLA Broadcast Authentication Protocol. *RSA CryptoBytes*, 5(Summer), 2002.

[31] C. E. Perkins. *Ad Hoc Networking*. Addison Wesley Professional, December 2000.

[32] K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields, and E. M. Belding-Royer. A Secure Routing Protocol for Ad hoc Networks. In *Proceedings of the International Conference on Network Protocols (ICNP)*, November 2002.

[33] F. Stajano. *Security for Ubiquitous Computing*. John Wiley and Sons, February 2002.

[34] Y. Zhang and W. Lee. Intrusion Detection in Wireless Ad-Hoc Networks. In *Proceedings of MobiCom*, 2000.

[35] L. Zhou and Z. Haas. Securing Ad Hoc Networks. *IEEE Network*, 13(6):24–30, 1999.