Mobility Helps Peer-to-Peer Security

Srdjan Čapkun, Jean-Pierre Hubaux, and Levente Buttyán*

Laboratory for Computer Communications and Applications (LCA)

EPFL (Swiss Federal Institute of Technology - Lausanne)

CH-1015 Lausanne, Switzerland

srdan.capkun@epfl.ch, jean-pierre.hubaux@epfl.ch, buttyan@hit.bme.hu

Abstract

We propose a straightforward technique to provide peer-to-peer security in mobile networks. We show that far from being a hurdle, mobility can be exploited to set up security associations among users. We leverage on the temporary vicinity of users, during which appropriate cryptographic protocols are run. We illustrate the operation of the solution in two scenarios, both in the framework of mobile ad hoc networks. In the first scenario, we assume the presence of an off-line certification authority and we show how mobility helps to set up security associations for secure routing; in this case, the security protocol runs over one-hop radio links. We further show that mobility can be used for the periodic renewal of vital security information (e.g., the distribution of hash chain/Merkle tree roots). In the second scenario, we consider *fully self-organized* security: users authenticate each other by visual contact and by the activation of an appropriate secure side channel of their personal device; we show that the process can be fuelled by taking advantage of trusted acquaintances. We then show that the proposed solution is generic: it can be deployed on any mobile network and it can be implemented either with symmetric or with asymmetric cryptography. We provide a performance analysis by studying the behavior of the solution in various scenarios.

Index Terms: Mobile ad hoc networks, Network-level security and protection

*Now with Budapest University of Technology and Economics Department of Telecommunications Magyar tudosok krt. 2, H-1117 Budapest, Hungary. e-mail: buttyan@hit.bme.hu

1 INTRODUCTION

Peer-to-peer security is considered to be more difficult to achieve than traditional security based on central servers. One would expect the problem to become even more challenging when users are allowed to move around and to be connected only sporadically. Indeed, according to common belief, wireless communication and mobility are at odds with security: jamming or eavesdropping is easier on a wireless link than on a wired one, notably because such mischief can be perpetrated without physical access or contact; likewise, a mobile device is more vulnerable to impersonation and to denial of service attacks.

The security architectures of existing mobile networks are highly centralized (as are their static, wireline counterparts). For example, the security of GSM relies on a key, shared by the subscriber and the operator, which is established at the time the contract is signed; the security of Third Generation cellular networks is based on the same principle. Another example is the *Wireless Transport Layer Security* (WTLS) protocol, aimed at providing secure Web access from a mobile device: the servers are authenticated by a certificate of their public key, delivered by a well-established certification authority.

Both examples show that the driving security concern has been to serve the interest of specific organizations: In the first case, the security system guarantees an operator that only legitimate subscribers can make use of the communication service it provides; in the second case, it lets an e-business company claim to its own customers that they are connected to the right Web server and that the message exchange is protected.

So far, nothing has been proposed for *peer-to-peer* security in mobile networks. We will show that, far from being a hurdle, mobility can in fact help security by enabling basic functions such as authentication and key establishment. We will illustrate the principles of our solution in two scenarios, both in the area of mobile ad hoc networks.

The first scenario corresponds to situations where an (off-line) authority provides the authorization to each mobile node to join the network, but it does so only at the initialization of each node; when in each others' radio range, nodes mutually authenticate and set up shared keys; this approach allows the nodes

to join the network at different times (in general, the authority does not even know how many nodes will eventually be present in the network). An important use of this approach is to secure routing, as the direct (one-hop) establishment of security associations avoids relying on routing for the establishment of security associations (i.e., this approach breaks the routing-security interdependence cycle [21, 4]). Another important application of this approach is to enable the periodic renewal of vital security information (e.g., the distribution of hash chain/Merkle tree roots).

In the second scenario, we consider *fully* self-organized security: in such a setting, there is no central authority whatsoever, and the establishment (and release) of security associations is purely based on mutual agreement between users; when they are close to each other, users can activate a *secure side channel* between their personal devices to authenticate each other and set up shared keys.

In [12], we have developed this initial idea by quantifying the benefits of mobility and by introducing a mechanism called “friends” that supports the establishment of security associations even between nodes that do not meet physically. Here, we further extend this work by proposing protocols that allow the implementation of our system with both symmetric and public-key cryptography.

The organization of the paper is the following. In Section 2, we survey the related work. In Section 3, we explain how security associations are created based on encounters and we provide the cryptographic protocols. In Section 4, we propose several applications of our approach. In Section 5, we study the pace at which the security associations are created. Finally, we conclude the paper in Section 6.

2 State of the art

In [29], Zhou and Haas propose a distributed public-key management service for ad hoc networks in which the functionality of the central authority is distributed over a subset of nodes through a threshold cryptography scheme. A more recent proposal by Luo et al. [22] describes a similar approach, that provides a more fair distribution of the burden by allowing *any* node to carry a share of the private key of the service.

A different technique, proposed by Asokan and Ginzboorg [1] is based on a shared password. In this approach, nodes willing to establish a secure session must share a prior context (in the example they "share" a room in which they are located). A fresh password is chosen and shared among users (e.g., it is written on a blackboard). From this weak password, the users derive a strong key, by making use of a password-authenticated key exchange.

Another approach, originally designed for the address ownership problem in Mobile IPv6, is described by Montenegro and Castelluccia in [23] and by O'Shea and Roe in [24]. 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 part of the IP address of the node.

In [15], Eschenauer and Gligor propose a random key pre-distribution 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 [6].

In [9], we propose a self-organized public-key management system for ad hoc networks, which is similar to PGP in the sense that users issue certificates for each other based on their personal acquaintances. In that system, each user maintains a local certificate repository and users' mutual authentication is performed through certificate chaining.

To the best of our knowledge, the only research published so far that shows that proximity of devices can help to set up security associations and to perform authentication is presented in [3, 7]. However, these proposals deal with the application level security and do not show how the mobility of the nodes can be used to progressively reinforce the security of the network (e.g., by increasing the security of routing).

Finally, we must mention the works of Grossglauser and Tse [16], and Dubois-Ferriere, Grossglauser and Vetterli [17, 14]; these papers show that mobility can help to increase the per-user throughput in ad hoc networks and to disseminate destination location information without incurring any communication

overhead to the network.

3 MECHANISMS TO ESTABLISH SECURITY ASSOCIATIONS

In this section, we first describe our system model and then we propose the mechanisms for the establishment of security associations.

3.1 System Model

In this work, we consider and discuss two scenarios: the first assumes the presence of a trusted authority, whereas the second is fully self-organized.

In the first scenario, we consider an ad hoc network of mobile *nodes*, controlled by an (off-line) central authority. This means that the authority controls network membership. 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 that 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 that other nodes hold.

In the second scenario, we consider an ad hoc network of mobile *nodes*, where each node represents a *user* equipped with a personal mobile *device*. We assume that each legitimate user has a single device. In this scenario, we consider that the network is *fully* self-organized, meaning that there is no infrastructure (hence no PKI), no central authority, no centralized trusted third party, no central server, and no secret share dealer, *even in the initialization phase*. A fundamental assumption is that each node is its own authority domain.

We assume, in both scenarios, that 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).

3.2 Mechanisms

If a node u possesses a certificate signed by the central authority that binds node v with its (v's) public key, then we say that there exists a one-way security association from u to v. Two one-way security associations between nodes u and v (one in each direction) constitute a two-way security association between the nodes. Equally, if u and v share a secret key k_{uv}, we say that there exists a two-way security association between u and v.

If public-key cryptography is used, a (two-way) security association between two nodes u and v is represented by triplet $(U,\ k_u,\ a_u)$ at the side of v and triplet $(V,\ k_v,\ a_v)$ at the side of u, where U and V are the names of the users that are associated with nodes u and v, k_u and k_v are the public keys of u and v, and a_u and a_v are the node addresses of u and v, respectively. Once nodes u and v have established a security association between them, they can set up secure communication channels that protect the integrity and confidentiality of the exchanged messages. In fact, for efficiency reasons, u and v may want to use symmetric key cryptography for the protection of their messages; in this case, they establish short-term symmetric keys (session keys) using the public keys in the security association. In this way, the nodes establish short-term symmetric-key security associations, which they can use for efficient secure routing [19].

Similarly, if symmetric-key cryptography is used, a security association between nodes u and v is represented by triplet $(U,\ k_{uv},\ a_u)$ at the side of v and triplet $(V,\ k_{uv},\ a_v)$ at the side of u, where k_{uv} is a symmetric key shared by u and v. In the symmetric-key based approach, we consider security associations to be always two-way; it is not possible to establish a one-way security association[1].

The establishment of security associations differs in the authority-based and in the fully self-organized scenario.

In the authority-based scenario, when two nodes move into the power range of each other, they will

[1]In practice, the nodes may derive sub-keys from the shared symmetric key of the security association, where each sub-key is used in one direction only and perhaps only for a specific security service (e.g., either for integrity or for confidentiality, but not for both); this is a policy issue, out of the scope of our discussion.

exchange certificates that contain their public keys, and establish a security association.

In the self-organized scenario, when they meet, users are obviously given the possibility to visually identify each other. The decision to set up a security association between two nodes is based on this physical encounter. To support this mechanism, we assume that each device is equipped with a short range connectivity system (e.g., infrared or wire). We call a channel established by this mechanism a *secure side channel*. A secure side channel can only be point-to-point and works only when the nodes are within a "secure range" of each other. We consider this assumption to be realistic, as almost all personal mobile devices are equipped with infrared interfaces. We assume that the activation of the side channel is made by both users consciously and simultaneously. When activating the side channel, the users simultaneously associate the name (or the face) of the other person to the established security association. This operation is very similar to the exchange of business cards; in fact, it can even be transparently combined with the exchange of *electronic* business cards (e.g., exchange of vCards[2] between PDAs). If a user wants to establish a security association with a user-independent device (e.g., a printer), she will visually identify the device and bind its identity to the context in which the device operates. In this paper, however, we focus on the establishment of security associations between users' personal communication devices. These encounters make it possible for a user to associate a face to a given identity (and to a given public key), thus solving many of the classical problems of security in distributed systems (e.g., impersonation attacks and Sybil attacks [13]).

We assume that an adversary can eavesdrop on all radio links and can manipulate messages in all kinds of ways. However, the adversary cannot modify messages transmitted over the secure side channel. Note that we do not require the secure side channel to protect the confidentiality of exchanged information. Finally, we consider that an adversary can have at his disposal as many fake devices as he wants.

The major difference between the fully self-organized and the authority-based approach stands in user involvement: In a fully self-organized approach, users need to establish security associations consciously,

[2]http://www.imc.org/pdi/

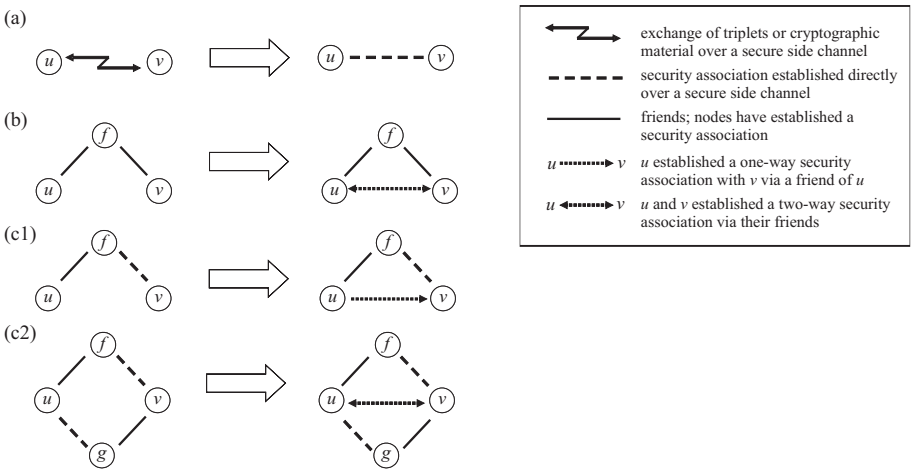


Figure 1. Three mechanisms to create new security associations using (a) the secure side channel, (b) a common friend, and (c1,c2) the combination of the first two approaches (mechanism (c1) is used only in the public-key based approach).

whereas in the authority-based approach, users do not need to be aware of the establishment of the security associations, as it is done automatically by their nodes. The use of either of these approaches strongly depends on the purpose of the network. Typically, the self-organized approach is useful in securing personal communications on the application level, whereas the authority-based approach is used to secure networking mechanisms such as routing. We will now address the public-key approach, and then the symmetric one.

3.2.1 Public-key approach

We focus on the establishment of security associations in the fully self-organized model. In the authority-based model, two nodes can establish a security association by exchanging their certificates; this is rather straightforward and we do not detail it further.

Three mechanisms support the establishment of new security associations (Figure 1). Mechanism (a) is used when two nodes u and v are in the vicinity of each other, and it consists in u and v exchanging their triplets using the secure side channel. Since the secure side channel ensures the integrity of the exchanged messages, it precludes the possibility of a man-in-the-middle attack. This guarantees a secure binding between the received user name, public key, and node address. In addition, the user can easily

verify the validity of the received name because the name should correspond to the person present at the encounter. The node can also verify that the other node indeed possesses the private key that belongs to the received public key by executing a simple challenge-response protocol. Finally, the node address can be verified against the public key. The verification of the node address against the public key is necessary, notably for secure routing. One possible solution is to generate the node address from its public key[3] by making use of a technique similar to CAM [24] or SUCV [23]. In this way, node addresses are bound to public keys in a verifiable way. Note, however, that a malicious node may generate several public keys and the corresponding node addresses and distribute them to other nodes. Whether this is a problem very much depends on how the routing protocol is secured; a thorough study of this issue is left for future work. In the authority-based model, this is not a problem, as the node address is bound to the public key of the node by the certificate issued by the authority; this removes the need for CAM, SUCV, or similar mechanisms.

Protocol 1 shows a possible implementation of the direct establishment of security associations.

Protocol 1: Direct Establishment of a Security Association

msg1 (secure side ch.)	$u \rightarrow v:$	$a_u \mid \xi_u = h(r_u \mid U \mid k_u \mid a_u)$
msg2 (secure side ch.)	$v \rightarrow u:$	$a_v \mid \xi_v = h(r_v \mid V \mid k_v \mid a_v)$
msg3 (radio ch.)	$u \rightarrow v:$	$r_u \mid U \mid k_u \mid a_u$
msg4 (radio ch.)	$v \rightarrow u:$	$r_v \mid V \mid k_v \mid a_v$
	$u:$	$h(r_v \mid V \mid k_v \mid a_v) = \xi_v?;\ V?;\ match(k_v, a_v)?$
	$v:$	$h(r_u \mid U \mid k_u \mid a_u) = \xi_u?;\ U?;\ match(k_u, a_u)?$
msg5 (radio ch.)	$u \rightarrow v:$	$\sigma_u(r_v \mid U \mid V)$
msg6 (radio ch.)	$v \rightarrow u:$	$\sigma_v(r_u \mid V \mid U)$

In Protocol 1, u and v first generate random numbers r_u and r_v, respectively, and exchange, through the secure side channel, their addresses a_u and a_v and the cryptographic hash values $\xi_u = h(r_u \mid U \mid k_u \mid a_u)$

[3]If the node has several public keys, then the node address is generated from a designated one.

and $\xi_v = h(r_v \mid V \mid k_v \mid a_v)$ of their random numbers and triplets. After this initial exchange, u and v send messages to each other through the radio interface (since they have obtained each other's node address in the first two messages). They exchange their random numbers and triplets, and each of them verifies if the hash value of the received random number and triplet is equal to the received hash value ξ_u (or ξ_v). If this is the case, they can be sure that they have received the random number and the triplet from the party with which they exchanged the first messages through the secure side channel. The random numbers serve as nonces and guarantee the freshness of the subsequent messages. Now, both users can verify if the received user name corresponds to the other party and both nodes can verify if the received node address matches the received public key. Finally, the nodes generate and send to each other a signature ($\sigma()$) on the received random number and the user names in order to prove that they possess the private keys that belong to the exchanged public keys.

With Mechanism (b), two nodes u and v can establish a security association if they have a common friend f. A simple solution is the following: Since f knows the triplets of both u and v, it can issue (on request from u and/or v) fresh certificates for both triplets and send them to v and u, respectively, via the network. Both u and v know the public key of f and they also trust f, therefore they can both verify the received certificates and will accept the information therein if the verification is successful.

Mechanism (c) is a combination of the friendship relationships and the encounters, and it establishes only a one-way security association: If nodes u and f are friends and f has obtained the triplet of v in an encounter with v, then f can issue (on request from u) a fresh certificate for the triplet of v, and send this certificate to u via the network. Since u knows the public key of f, and also trusts f, she can verify the received certificate and accept the received triplet if the verification is successful. A two-way security association between nodes u and v is then established as a combination of two one-way security associations (from u to v and from v to u).

The protocols corresponding to Mechanisms (b) and (c) are straightforward and we do not detail them.

3.2.2 Symmetric-key approach

The mechanisms used in the symmetric-key approach are similar; they can be applied to both fully self-organized and to the authority-controlled networks.

The first mechanism (Figure 1, Mechanism (a)) is the direct establishment through the side channel: When the nodes are in the vicinity of each other, they can exchange, through the side channel, their user names and node addresses, and additional data that allow them to compute a shared secret. It is important to note, however, that in a pure symmetric-key approach, setting up a shared secret between two parties always requires a confidential channel between them. This means that in this case, the side channel must ensure not only the integrity but also the confidentiality of messages. Like in the public-key implementation, the users can verify the received names through personal encounters. The node addresses, on the other hand, can be verified against the received (and verified) names.

Mechanism (b) supports the establishment of security associations between two nodes u and v via a common friend f. By assumption, f already has a security association with both u and v, meaning that it has symmetric keys established with them. In addition, f is trusted by both u and v. Therefore, to establish a session key between u and v, well-known symmetric-key protocols can be used, where f plays the role of the trusted (key) server. The session key can be generated either by f who would send it to both u and v (like in the Kerberos protocol), or by u or v, in which case f would be used as a trusted relay (like in the Wide-Mouth-Frog protocol [5]).

Finally, Mechanism (c) can be used when two nodes u and v do not have a common friend, or have a common friend f but do not want f to know their shared secret key. Like in the public-key based approach, Mechanism (c) combines the first two mechanisms (encounters and friends). Let us assume that u has a friend f who has already set up a security association with v using the first mechanism. Similarly, let us assume that v has a friend g who has set up a security association with u using the first mechanism. Now u and v can set up a security association using f and g by u generating key contribution k_u and sending it to v via g, v generating key contribution k_v and sending it to u via f, and then both u and v

computing a common value k_{uv} from k_u and k_v. Protocol 2 illustrates this in more detail. In this protocol, nodes u and v first exchange the names of their friends to be used in the protocol as trusted relays, and two nonces r_u and r_v that are used to guarantee the freshness of subsequent messages. Then, u generates some random key k_u and sends it to v via g (msg3 and msg4), and v generates some random key k_v and sends it to u via f (msg3' and msg4'). Here, $d_{x\to y}$ is a direction bit that indicates that the message goes from x to y (and not from y to x)[4]. req and rep are bits that indicate that the message is a *request* to a friend or a *reply* from a friend, respectively. We need these bits because every node can play either the role of a requesting node (u and v) or the role of a friend (f and g), and thus we must indicate not only that this is a message from x to y but also that x is the requesting node and y is the friend (or vice versa). Finally, u and v compute a common value k_{uv} from k_u and k_v using a publicly known pseudo-random function h (e.g., a hash function).

Protocol 2: Friend-Assisted Establishment of a Security Association

msg1	$u \to v:$	f, r_u
msg2	$v \to u:$	g, r_v
msg3	$u \to g:$	$u, \{d_{u\to g}, req, v, k_u, r_v\}k_{ug}$
msg4	$g \to v:$	$g, \{d_{g\to v}, rep, u, k_u, r_v\}k_{vg}$
msg3'	$v \to f:$	$v, \{d_{v\to f}, req, u, k_v, r_u\}k_{vf}$
msg4'	$f \to u:$	$f, \{d_{f\to u}, rep, v, k_v, r_u\}k_{uf}$
	$u, v:$	$k_{uv} = h(k_u, k_v)$

An interesting feature of the protocol is that it replaces a single trusted party with two parties trusted by one entity each. If f and g are not colluding, then none of them has enough information to compute k_{uv}. In addition, both u and v trust at least one of them for not colluding.

[4]Note that since messages are always encrypted with a symmetric key k_{xy}, the only ambiguity could be the direction.

4 APPLICATIONS

In this section, we describe several applications of our approach.

4.1 Key establishment

The mobility-assisted establishment of security associations can be notably used to secure routing, or simply to protect the confidentiality of user personal communication.

In the first case, this can help to establish sufficient security associations to enable secure routing, thus by breaking the routing-security interdependence cycle [21]: Security associations cannot be established over multiple hops as the routing protocol does not operate securely (because security associations are not established yet).

A conventional solution to the routing-security interdependence includes pre-loading pairwise keys in all nodes to create all the security associations at the initialization. However, this approach prevents the insertion of new nodes in the network and is demanding in terms of storage. Other solutions [21] rely on an on-line key distribution center to initially distribute keys and to handle new nodes. Although effective, this approach requires a costly initialization phase and the use of complex security protocols. Another possible solution is to make use of cryptographically verifiable identifiers such as SUCV or CAM [4]. The advantage of this and similar approaches is that all public keys can be simply verified against nodes' identifiers. The disadvantage is that the nodes cannot change (revoke) their public keys without changing the identifiers. Other approaches to key distribution may include: network-wide or local broadcast or multicast of public or TESLA [25] keys, either by the servers or by the nodes themselves. These approaches are very effective, but costly in terms of communication overhead. The main drawback of these schemes is that even if public keys are distributed to all nodes, routing will not be efficient due to the high cost of public key cryptography, or at least until symmetric keys are established.

Our mobility-based approach is different in the sense that it enables a flexible setup of security associations, simplifies the introduction of new nodes in the network, requires at most an off-line authority

and enables the establishment of secret keys between all pairs of nodes; a drawback is that the establishment of the security associations requires some time, as detailed in Section 5. Because of this, we see our approach as a standalone solution in some applications, but also as a complementary solution to other key management solutions in other applications. Furthermore, as we showed in our previous work [10], in some scenarios, it is sufficient that only a small percentage (less than 40%) of security associations are established between the nodes to enable secure routing.

If used to secure users' personal communication, our mobility-based scheme can enable mutual authentication of users that have already met. Moreover, it can also be used to fuel the creation of certificate graphs; in that case, authentication can be performed through certificate chaining, as described in [9].

4.2 Periodic distribution of security material

Besides key establishment, mobility can also help to periodically distribute security material. Notably, it can be used for the distribution of hash chain and Merkle tree roots [26]. Hash chains and Merkle trees enable authentication of values and are typically used to ensure message freshness and authentication. They have been used to secure various aspects of routing. Hauser, Przygienda and Tsudik [18] present an efficient mechanism for the authentication of link state routing updates. Zhang [28] improves this mechanism and presents a chained Merkle-Witnernitz one-time signature. Hu, Perrig and Johnson [19] propose a set of efficient security mechanisms for ad hoc networking, which make use of hash chains and Merkle hash trees. They also use hash chains to efficiently secure routing in ad hoc networks: to secure distance vector routing updates in SEAD [20] and to prevent malicious changes of hop count in Ariadne [21]. Finally, hash chains and Merkle hash trees are used to prove encounters between the nodes in mobile ad hoc networks [8].

All mentioned mechanisms require each node to distribute its hash chain or hash tree roots in an authentic way to every other node in the network. Moreover, they require the root of a new chain/tree to be distributed to every node, before the values from the old chain/tree are exhausted. This problem is

especially important in mobile ad hoc networks, in which broadcasts of roots/chains may be costly, or infeasible (if only symmetric-key cryptography is used for authentication). We thus propose a mobility-based approach similar to our key establishment scheme that works as follows.

Each node maintains two hash chains: an *active* and a *pending* one. We assume that the root of the active hash chain/tree has already been distributed; thus the active hash chain/tree can be used for the security application of interest as described earlier. In contrast, the root of the pending hash chain/tree has not been distributed yet, and the node has to distribute it to every other node in the network before the active hash chain/tree runs out of elements. When the root of the pending hash chain/tree has been distributed, the node can turn the pending hash chain/tree into an active state. At the same time, the node would generate a new pending hash chain/tree and begin distributing its root. Putting in place the pending hash chains/trees while using the active ones ensures continuous operation of the system.

In our scheme, the root of the pending hash chain/tree is disseminated in an authentic way when nodes encounter each other. Together with the root of the pending hash chain/tree, the nodes also disseminate a time t in the future. The value of t should be estimated in such a way that the active hash chain/tree does not run out of elements by t and the root of the pending hash chain/tree is distributed to all other nodes by time t. Then, at t, the pending hash chain/tree becomes active, and a new pending hash chain/tree is generated; the process is then repeated.

5 Performance Evaluation

In this section, we provide an estimate of the pace at which security associations are created. We assume that, initially, each node established security associations only with its friends; we further assume that each node has the same number of friends.

In our analysis, we will observe the following values: the convergence $r(t)$, which represents the fraction of the security associations established until time t, and the convergence (meeting) time t_M, which is the time needed to establish all the desired security associations. One additional value of interest is the average

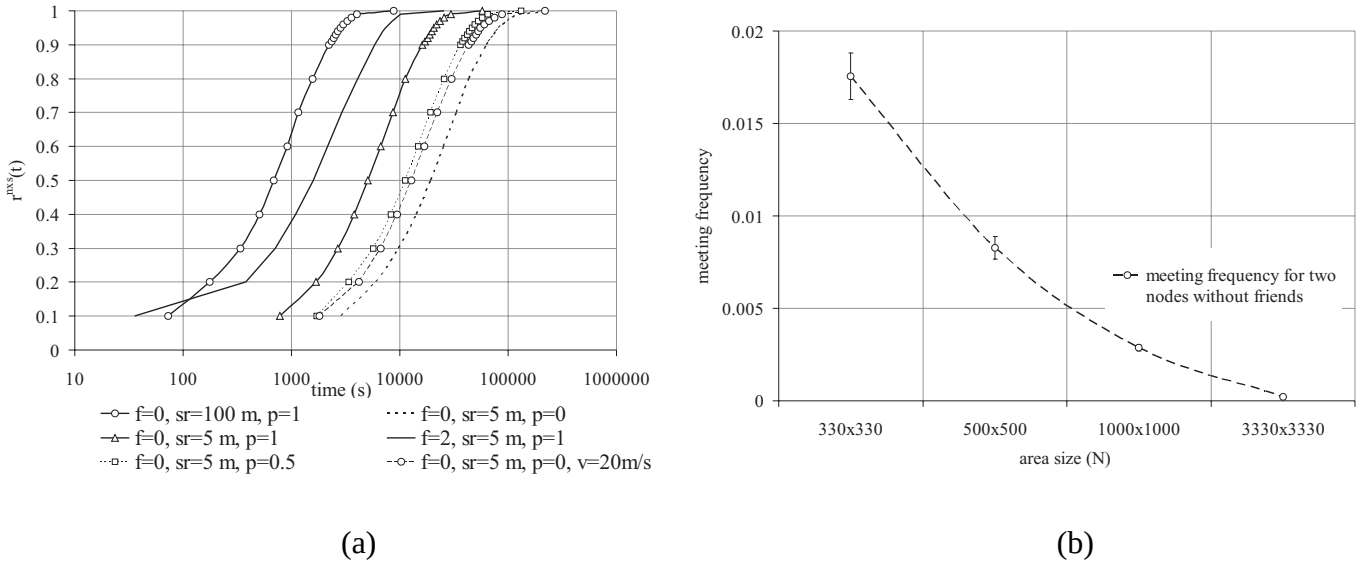
meeting frequency $1/t_{IM}$ of nodes. Here, t_{IM} is the node inter-meeting time. This value is important for assessing the frequency of rekeying and the time necessary to perform key revocation.

In our simulations, we make use of the Random Waypoint mobility model and we extend this model with some new features; we call this new model the Restricted Random Waypoint. In the conventional Random Waypoint model, a node chooses its destination and its speed towards this destination randomly. After arriving at the destination, the node pauses for a certain period of time, and then chooses its new destination and its speed. In the *Restricted Random Waypoint* model the nodes move in the same way as in the Random Waypoint model, but their choice of destination points is restricted to a number of fixed points on the plane with some probability p . This means that with probability p, a node randomly chooses a point from a finite set of destination points, and with probability $1 - p$, it chooses as its destination a random point on the plane. We call this model the *Restricted Random Waypoint* mobility model. This model is closer to reality in the sense that users normally do not randomly choose any point on a plane as their destination, but they rather move to some meeting points (e.g., meeting rooms, lounges, restaurants) where communication between users takes place. If $p = 1$ and if the set of destination points is small, the convergence time will be very small. On the contrary, if $p = 0$, we have the standard Random Waypoint mobility model and the convergence time will be longer.

In this mobility model, two nodes can establish a security association if they are in the *security range* of each other (for the fully self-organized network) or in each others' power range (in the authority based network). The security range is significantly smaller than the power range of mobile nodes and is the maximum range that is sufficient for the secure side channel to be set up.

In all simulations, we use the same simulation area, (a 1000m$\times$1000m square) and we set the number of nodes to $n = 100$. When the nodes hit the area border, they bounce off under the same angle under which they hit the border. The node maximum speed is set to 5 m/s (except in one case on Figure 2a, where it is 20 m/s), and the minimum speed to 1 m/s [27]. The pause time is set to 100 s.

On Figure 2 we observe the convergence $r^{n\times s}(t)$ and the node meeting frequency. Figure 2a illustrates



(a) (b)

Figure 2. Restricted Random waypoint simulation results; (a) average convergence, (b) meeting frequency. Here, f is the number of node's friends, p is the restriction probability, v is the node speed, and sr is the range within which the nodes can establish security associations. The results are shown with 95% confidence intervals.

that the friends mechanism speeds up convergence proportionally to the number of friends. Furthermore, this shows that, as expected, a higher average speed of nodes results in a faster convergence (and therefore a shorter convergence time). The same figure also illustrates another very intuitive result: The convergence is faster if the nodes gather at and around meeting points. It is also interesting to observe that, in the most favorable scenario (in which the security range is 100 m and the network is controlled by a central authority), 40% of security associations are established in less than 1000 seconds (17 minutes). This is an important result, given that, as we have shown in [10], this percentage of security associations is sufficient to support secure routing. Figure 2b shows the node meeting frequency (two nodes), in areas of various sizes. Here, we observe that the meeting frequency is inversely proportional to the size of the area. These results are in line with our analysis of the convergence time and meeting frequency of random walks [11], where we show that the convergence time is proportional to $N \log N$ and the meeting frequency is proportional to $\frac{2}{N}$. Here, N is the size of the area in which the nodes move.

6 CONCLUSION

In this paper, we have shown that mobility can help to provide security in mobile networks. We have illustrated our approach on two application scenarios in the area of mobile ad hoc networks: networks with an off-line authority and fully self-organized networks. In the first scenario, a direct establishment of security associations over the (one-hop) radio link solves the well-known security-routing interdependency problem. In the second scenario, we have shown that the solution is intuitive to the users, as it mimics real life concepts (physical encounters and friends), and solves some classical problems of security in distributed systems.

We have shown that our solution works both with public-key and with symmetric cryptography and we have provided the related protocols. We have studied the pace of establishment of the security associations under various mobility scenarios. In particular, we have extended the Random Waypoint model by introducing the concept of meeting points in order to be closer to human behavior. We have shown that in self-organized scenarios, the set-up of security associations can take several hours, while in the case of networks controlled by central authorities, this time can be as low as 20 minutes. We have further shown that the vast majority of the security associations are set up in much shorter time than the full set of security associations. This is an important observation, as we have recently shown [10] that secure routing is also possible in networks in which only 40% of security associations are established. Moreover, if the users are willing to set up security associations, they can decide to move close to people of their interest.

To the best of our knowledge, this research effort is the first that shows how mobility can help to secure peer-to-peer mobile networks.

In the future, we plan to study even more realistic and more sophisticated mobility models, including those with correlated mobility patterns [2]. We will further study how an even *incomplete* set of security associations can be exploited to perform crucial security operations. We also intend to further investigate rekeying and key revocation schemes. Finally, we intend to analyze the burden of the cryptographic

functions on the processing units, especially in the public-key case.

7 ACKNOWLEDGMENTS

We are thankful to Mario Čagalj, Mathias Grossglauser, Adrian Perrig, Markus Jakobsson and Milan Vojnović for useful comments and discussions. The authors also thank EPFL students David Schmalz and Olivier Roy for their contribution in producing simulation results. 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`).

REFERENCES

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

[2] F. Bai, N. Sadagopan, and A. Helmy. IMPORTANT: A framework to systematically analyze the Impact of Mobility on Performance of RouTing protocols for Adhoc NeTworks. In *Proceedings of Infocom*, April 2003.

[3] D. Balfanz, D. Smetters, P. Stewart, and H. Wong. Talking to strangers: Authentication in ad hoc wireless networks. In *Proceedings of NDSS*, 2002.

[4] R.B. Bobba, L. Eschenauer, V.D. Gligor, and W. Arbaugh. Bootstrapping Security Associations for Routing in Mobile Ad-Hoc Networks. In *Proceedings of IEEE Globecom*, December 2003.

[5] M. Burrows, M. Abadi, and R. Needham. A Logic of Authentication. In *William Stallings, Practical Cryptography for Data Internetworks, IEEE Computer Society Press.* 1996.

[6] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In *IEEE Symposium on Security and Privacy*, May 2003.

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

[8] S. Čapkun, L. Buttyán, and J.-P. Hubaux. SECTOR: Secure Tracking of Node Encounters in Multi-hop Wireless Networks. In *Proceedings of the ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN)*, Washington, USA, October 2003.

[9] 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.

[10] S. Čapkun and J.-P. Hubaux. BISS: Building Secure Routing out of an Incomplete Set of Security Associations. In *Proceedings of WiSe*, 2003.

[11] S. Čapkun, J.-P. Hubaux, and L. Buttyán. Mobility Helps Peer-to-Peer Security. Technical report, EPFL-IC-LCA, no. IC/2003/81, 2003.

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

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

[14] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli. Age Matters: Efficient Route Discovery in Mobile Ad Hoc Networks Using Encounter Ages. In *Proceedings of MobiHoc*, 2003.

[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. Grossglauser and D. Tse. Mobility Increases the Capacity of Ad-Hoc Wireless Networks. In *Proceedings of Infocom*, 2001.

[17] M. Grossglauser and M. Vetterli. Locating Nodes with EASE: Mobility Diffusion of Last Encounters in Ad Hoc Networks. In *Proceedings of Infocom*, 2003.

[18] R. Hauser, A. Przygienda, and G. Tsudik. Reducing the Cost of Security in Link State Routing. In *Proceedings of NDSS*, February 1997.

[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. SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks. In *Proceedings of the Fourth IEEE Workshop on Mobile Computing Systems and Applications*, June 2002.

[21] 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.

[22] H. Luo, J. Kong, P. Zerfos, S. Lu, and L. Zhang. URSA: Ubiquitous and Robust Access Control for Mobile Ad Hoc Networks. *IEEE/ACM Transactions on Networking*, to appear, 2004.

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

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

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

[26] R. C. Merkle. Protocols for Public Key Cryptosystems. In *Proceedings of the IEEE Symposium on Security and Privacy*, 1980.

[27] J. Yoon, M. Liu, and B. Noble. Random Waypoint Considered Harmful. In *Proceedings of Infocom*, 2003.

[28] Kan Zhang. Efficient Protocols for Signing Routing Messages. In *Proceedings of NDSS*, March 1998.

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