Consiglio Nazionale delle Ricerche

Introducing Authenticated Information in a Reliable Multicast Protocol for Mobile Computing

N. De Francesco, M. Petrocchi

IIT TR-11/2002

Technical report

Giugno 2002

Istituto di Informatica e Telematica

Introducing Authenticated Information in a Reliable Multicast Protocol for Mobile Computing

Nicoletta De Francesco[1], Marinella Petrocchi[2]
[1]Dipartimento di Ingegneria dell'Informazione, Università di Pisa, Italy
nicoletta.defrancesco@iet.unipi.it
[2]Istituto di Informatica e Telematica, C.N.R. Pisa, Italy
marinella.petrocchi@iit.cnr.it

Abstract

We consider a known protocol for reliable multicast in distributed mobile systems where mobile hosts belonging to a group communicate with a wired infrastructure by means of wireless technology. The original specification of the protocol does not take into consideration any notion of computer security. In this paper it is shown how an adversary may eavesdrop on communications between legitimate members and inject packets over the wireless links, pretending to be a legitimate member belonging to the group. We suggest a revised version of the protocol providing authenticity and integrity of packets over the wireless links.

Keywords: wireless communication, authentication of origin, cryptographic protocols, distributed systems security.

Contents

1 Introduction

Technological developments in computer and communication are enabling the deployment of computing systems based on portable computers and wireless networking. Users may be equipped with hand-held computing devices and roam around freely while maintaining connectivity with a wired infrastructure. Such architectures may be exploited for novel applications and services spread out in a variety of directions.

With the considerable spread of wireless access both undisputed advantages and new security problems arise. Indeed, broadcasting messages over radio channels makes traffic eavesdropping and packets' injection relatively easy for adversaries in possession of adequate resources. The practicality of security attacks in wireless environments has been discussed and shown recently, [9, 24].

We consider the multicast protocol for mobile computing developed in [2, 8]. Design issues are considered in the cited papers in order to support reliable and totally ordered communication within a group of processes, (*group members*), running on mobile hosts. The protocol is concerned with *reliable multicast communication*, where *reliability* means, very informally, that all packets (messages) are delivered and that duplicates are discarded. Each process at stake is able to detect possible losses of packets. Lost packets are then recovered by using a mechanism based on *nack* messages and retransmission.

The protocol is not intended to support real time applications.

The design of the protocol does not take in any account security issues and the presence of possible adversaries has not been considered. In this paper we assume that only *group members* are authorized to participate through the protocol. Unauthorized mobile hosts will be treated as potential adversaries. Provided such a diversification between *group members* and the rest of the mobile world, authentication mechanisms will be built to assure the capability for the members to authenticate a packet as originated by a member itself. The protocol was originally designed to guarantee a set of properties. The security mechanisms proposed in this paper will allow the original properties to hold even when the presence of mobile adversaries is considered.

Some concepts coming from recent proposals by Balfanz *et al.*, [7], seem to fit our context quite well. Further, an analogy between digital streams considered in [11] and the messages exchanged through the protocol will be highlighted.

The following section presents an overview of the protocol. Section 3 considers the protocol in a security environment and informally shows how the protocol could be vulnerable to attack by adversaries. In Section 4 we suggest an new version of the protocol, enriched with security mechanisms to authenticate the exchanged packets as coming from *group members*. In Section 5 we discuss related works in the wireless authentication area. Section 6 offers some conclusions and lists future works.

2 Protocol Overview

A very intuitive notation will be used throughout the paper. We consider a set of agents able to send and receive messages. Basically, we represent the sending and reception of a message msg from a single sender A to a single receiver B in the following way:

$$c_j \quad A \xrightarrow{u} B \quad : \quad msg$$

where msg is the exchanged message, c_j is the j-th communication channel, on which the exchange takes place. A and B are the sender and the receiver of msg. Raised u stands for "unicast" modality (a point to point connection).

We represent the multicast of a message (from a sender A to a set $\mathcal{B}$ of multiple but defined receivers, with only one sending action) as follows:

$$c_j \quad A \xrightarrow{m} \mathcal{B} \quad : \quad msg$$

where $\mathcal{B}$ is a set of processes. Raised m stands for "multicast" modality.

We represent the broadcast of a message (from a sender A to multiple receivers $\mathcal{B}$, with only one sending action) as follows:

$$c_j \quad A \xrightarrow{b} \mathcal{B} \quad : \quad msg$$

where A possibly broadcasts msg to each process B belonging to the set $\mathcal{B}$. Contrary to the multicast modality, here the set $\mathcal{B}$ may change dynamically. Prime b stands for "broadcast" modality.

We provide a sketch of the multicast protocol for distributed mobile systems that is considered in the paper. We refer to a simplified version of the protocol, [4]. The full version can be found in [8], along with a detailed discussion of its motivation and advantages. The protocol has been later extended to support weaker ordering guarantees that can be selected by each sender, e.g. FIFO and causally ordered delivery [5]. A comparison with a similar reliable multicast protocol for distributed mobile systems has been performed in [3]. A formal analysis of the properties of the protocol, concerned with total order delivery, no duplicates and losses recovery, has been performed in [4].

The system model on which the protocol is defined is as follows. It consists of mobile hosts MHs and stationary hosts SHs, called *gateways*. The gateways are connected both to a wired network (that provides reliable and FIFO-ordered communication) and to a wireless link that covers a spatially limited *cell* nearby each gateway. Cells provide only incomplete coverage and wireless communication is unreliable. MHs communicate through wireless links and may move. Movements are unpredictable, in the way that a MH may leave a cell without prior negotiation and reenter any other cell or even remain out of coverage for some time. The protocol establishes that MHs may only exchange messages with the gateway of the cell where they happen to be located in and with a special stationary host acting as the coordinator. The gateways may broadcast messages to all MHs in their cell and send messages to a specific MH in their

$$
\begin{array}{lll}
c_1 & MH \xrightarrow{u} C & : \ new \\
c_2 & C \xrightarrow{m} \{G_1, G_2, ..., G_N\} & : \ new, seq \\
c_3 & G_i \xrightarrow{b} \{MH \mid MH \, is \, in \, cell \, i\} & : \ new, seq
\end{array}
$$

Figure 1: Transmitting a *new* message.

cell. The resulting scenario is quite general since it can accommodate contemporary wireless LANs, infrared networks, picocellular wireless networks where cells coincide with rooms in a building, physical obstructions and long-range movements.

The protocol works as follows. A dedicated SH acts as the *coordinator*, denoted as C. A mobile host may receive messages from the application layer and send them to the group. Such messages are sent by the mobile host as *new* messages to the coordinator C that processes incoming *new* messages in sequence. C constructs a message containing the payload and an increasing sequence number. C then transmits the resulting message to all gateways through a FIFO-multicast. Gateways broadcast this message in their respective cells.[1]

Due to their movement across cells and uncovered areas and to the unreliability of the wireless links, MHs could receive duplicates or could miss packets.

The exchange of a *new* message can be formalized as in Fig. 1 and the procedure can be explained as follows:

1. A mobile host MH, wishing to communicate a *new* message to others, sends the message to the coordinator C.

2. The coordinator multicasts *new* to only all the static hosts $\{G_1, .., G_N\}$ on the wired link. It adds to the message the tag *seq*, containing the sequence number of the current *new*. Each gateway G_i maintains a list of messages recently received from C.

3. Each gateway G_i, responsible for cell i, broadcasts what it previously received from C in the cell and the mobile hosts currently present in cell i receive the message. mobile hosts as well as

By maintaining a history of the received sequence numbers, a mobile host discards duplicates and sends the gateway a proper *nack* message upon receiving an out-of-order message. Upon receiving a *nack*, the gateway sends MH a copy of the missing multicasts. Each gateway stores a copy of each multicast previously sent until it knows that the multicast has been delivered to every mobile host.

[1] Actually, the full version of the protocol is based on a set of coordinators whereas here a global synchronization structure among the coordinators has been considered.

The protocol does not use any notion of hand-off (unlike similar protocols for distributed mobile systems, [1]), i.e. it does not require any data exchange between the old gateway and new when a *group member* moves from one cell to another. Each gateway manages cell switchings autonomously without interacting with the other gateways.

3 Considering a hostile environment

Considering security communication protocols, cryptographic functions are introduced in the structure of messages in order to guarantee the fulfillment of certain security properties. Given the sensitive nature of information possibly exchanged in a run of a protocol it appears reasonable to consider the presence in the net of potential adversaries: unauthorized hosts may try to interfere with the normal execution of a protocol in order to achieve advantages in their interest. Hereafter, we consider a potential adversary to have the adequate technical equipment to eavesdrop traffic and actively inject packets over the wireless links. For details about the practicality of such interferences we refer to [9, 24].

The protocol was originally designed to guarantee a set of properties. We highlight two of them:

Authenticity (P_1) Any packet received by a *group member* has been originated by a *group member*.

No Duplicate (P_2) No *group member* accepts duplicate packets, i.e. duplicates are discarded.

These properties by themselves are clearly not sufficient since they would be satisfied even by a protocol in which *group members* do not accept any packet. A similar trivial solution is ruled out by another property, *Non Triviality*, i.e. *group members* indeed accept packets. The properties of the protocol have been formally analyzed in [4].

However, properties P_1 and P_2 might not hold in a classical context of security analysis where the presence of an adversary has to be considered.

In the following we suppose the presence of unauthorized mobile hosts, i.e. hosts that are not *group members*. To distinguish between authorized and unauthorized hosts, we write GM to denote a *group member*. Each GM is also a mobile host MH but the opposite is not true: a generic MH is not necessarily an authorized host.

An unauthorized host X can eavesdrop on and inject traffic over the wireless links in the following way.

$$\begin{array}{llll} (1) & c_j & X(A) \longrightarrow B & : \quad msg \\ (2) & c_j & A \longrightarrow X(B) & : \quad msg \end{array}$$

Notation (1) describes X that sends a message msg to party B pretending to be party A (unauthorized injection); (2) denotes: msg, intended for B, is eavesdropped on by X (eavesdropping).

The original protocol has been specified including no notion of cryptography. All messages are exchanged between parties as cleartexts and X may simply force *group members* to accept a forged message:

$$\begin{array}{lll} c_1 & X(GM) \xrightarrow{u} C & : \ false\_new \\ c_2 & C \xrightarrow{m} \{G_1, G_2, ..., G_N\} & : \ false\_new, seq \\ c_3 & G_i \xrightarrow{b} \{MH \mid MH \, is \, in \, cell \, i\} & : \ false\_new, seq \end{array}$$

Without any form of authentication of a *group member* to the coordinator in the message over channel c_1, X may be able to impersonate GM and transmit a capriciously generated payload $false\_new$[2], causing the downfall of Property P_1.

With regard to property P_2, duplicates are discarded by the receivers if the related packets carry an already received sequence number.

$$\begin{array}{lll} c_1 & GM \xrightarrow{u} C & : \ new \\ c_2 & C \xrightarrow{m} \{G_1, G_2, ..., G_N\} & : \ new, seq \\ c_3 & G_i \xrightarrow{b} X(\{MH \mid MH \, is \, in \, cell \, i\}) & : \ new, seq \\ c_{3bis} & X(G_i) \xrightarrow{b} \{MH \mid MH \, is \, in \, cell \, i\} & : \ new, seq + 1 \end{array}$$

X eavesdrops on the broadcasted message over channel c_3 and successively sends a false packet containing the same payload and a sequence number greater than seq. GMs in cell i do not discard the packet as a duplicate therefore Property P_2 does not hold anymore. This event is a consequence of the absence of a mechanism to authenticate the broadcasted message as indeed coming from a stationary host belonging to the set $\{G_1, G_2, ...G_N\}$.[3]

4 Adding security properties

The invalidity of Properties P_1 and P_2 is a consequence of no mechanism abling authorized hosts to authenticate each other.

We now define two properties regarding authentication between *group members* and stationary hosts:

- (P_A) Capability for the coordinator to authenticate the sender of a *new* message as a *group member*.
- (P_B) Capability for all the *group members* to authenticate the received broadcasts as indeed originated from the stationary hosts $\{G_1, G_2, ...G_N\}$.

[2] Receivers of the broadcasted message over channel c_3 are denoted as generic mobile hosts MHs, including both authorized and unauthorized hosts.

[3] The invalidity of the properties has been shown with two simple examples. They are not the only ones and other examples may be reported in order to break such properties.

Following, security procedures to make the listed properties hold will be presented and added to the original protocol.

Note 1. We do not require that a gateway correctly identifies a GM when it asks for a lost packet sending a *nack* message. The authentication of origin of the *nack* message is unimportant given that the contents of the packets do not have to be kept secret.

With reference to asymmetric cryptography [22], the digital signature is the typical mechanism to guarantee authentication of origin and integrity. In our context, unfortunately, to digitally sign each *new* message may cause an infeasible computational overload for mobile hosts which have intrinsic limited resources. They already have to cope with severe constraints in terms of power consumption and bandwidth (the wireless bandwidth is typically one order of magnitude smaller than wired bandwidth) and may not have the resources for performing public key operations in their completeness. Hence, we look for solutions with thrifty use of classical digital signature schemes as in [21].

We assume that the multicast over channel c_2 in Fig. 1 can not be compromised. We trust the coordinator C as well as the stationary hosts G_i on the wired link. Furthermore, we do not consider an adversary able to tamper with the communication on the wired link. If there is an injection of data coming from unauthorized hosts, we assume it to occur on the wireless link.

We mainly strive towards two goals: i) since we use public key cryptography, we need a method for guaranteeing the ownership of the public keys at stake. Common solutions rely on Public Key Infrastructures (PKIs) and digital certificates, [13]. In a wireless environment, the management of digital certificates could result in a bottleneck for the whole system. Subsection 4.1 presents an alternative method for bootstrapping authentication without the need of a PKI; ii) authentication in a broadcast environment presents different features with respect to a traditional point to point connection. To build security mechanisms leading to the fulfillment of Property P_B, we inherit a procedure originally developed to sign digital streams during live broadcasts (similarities between this context and our environment will be highlighted in Subsection 5.2).

4.1 Pre-Authentication and Location Limited Channels

Inspired by a recent work by Balfanz *et al.* [7], we make use of a method for bootstrapping authenticated and integral communication between group members participating in the protocol. A *pre-authentication* phase, in which a certain amount of information is exchanged between GMs and SHs over a privileged channel will be inserted. Information exchanged during the *pre-authentication* phase will be used through the main wireless link.

We inherit the concept of *Location-Limited Channels* from [7]. A *Location-Limited Channel* (hereafter LLC) is separated from the main wireless link and exploits security properties by virtue of the media over which data are sent. In

order to be used for *pre-authentication*, LLCs must support *physical identification*, i.e. human operators must be able to visibly control which devices are communicating with each other during a transmission over the LLC. Hence audio and infrared channels could be good LLCs given the physical limited range of their transmissions, [7, 14, 16].

Taking advantage of the group members capability to move, the human operator managing a GM can:

1. recognize that a communication on the LLC has successfully started with the intended stationary host (by means of visible clues, e.g. a light on the intended stationary host lights up);
2. be reasonably aware that only GM itself and the intended stationary host are trying to transmit on that LLC (in this case, the more direct the channel, the easier it is to monitor);
3. eventually conclude that GM indeed communicated with the right device.

We plan to use LLCs to exchange information about the public keys of the stationary hosts. Making use of public key cryptography rises the problem of how to authenticate the origin of a public key, i.e. the association between the public keys and the identities with which they are associated must be authenticated in a secure manner. Common solutions rely on a Public Key Infrastructure (PKI), a set of Certification (and Registration) Servers and security policies to manage the secure emission, renewal and revocation of digital certificates. A digital certificate is an electronic document that declares a legitimate link between an identity (person or machine) and a public key. However, the management of digital certificates run by a PKI could result in a bottleneck in the whole system under investigation. The *physical identification* property of the transmission over LLCs is a smart loophole to bypass the need of a PKI to guarantee the public keys at stake. The physical proximity of the hosts during the transmission over the LLC (and the consequent monitoring) is a way out to delegate the hosts themselves as guarantors for the benign nature of data exchanged over the LLC. Subsequent communications over the main wireless link will be accepted as "well-originated" if they refer to the data exchanged over the LLC.

5 The revised protocol

The design of a revised protocol will be introduced in this section by adding cryptographic mechanisms in order to guarantee the security properties considered in Section 4.

Assumption 1. Of all the mobile hosts, it is assumed that *only* the *group members* can transmit over the LLC.

Assumptions on the stationary hosts are: (i) they can transmit over the LLC; (ii) they hold a pair of public/private keys to perform regular signature schemes as in[21].

The environment under examination consists of both wired and wireless links. Communications necessarily pass through the stationary hosts on the wired link. This architecture allows us to separate the authentication mechanism into two distinguished parts: the first part concerns authenticating a *group member* to the coordinator, while the second is concerned with the authentication of the gateways G_i to the *group members* currently present in cell i. Even though we loose the precise identity of the sender of a new message, we are interested only in *group authenticity* (each group member can recognize whether a message was sent by a *group member*) rather than *source authenticity* (the capability to identify the single party within a group).

5.1 Authenticating the mobile sender

We require a mobile host to prove its group membership in order to send to the group. In the pre-authentication phase the *group member* has to be physically located close to the coordinator.

$$\begin{array}{llll} LLC & GM \xrightarrow{u} C & : & Hash\{Nonce_{GM}\} \\ LLC & C \xrightarrow{u} GM & : & pk_C \end{array}$$

The pre-authentication phase takes place over a selected LLC (e.g. exploiting infrared technology). First, the mobile host transmits the digest of a randomly generated number $Nonce_{GM}$ to the coordinator over the LLC. The coordinator replies transmitting its public key pk_C. An adversary able to listen over the LLC does not add any useful information to his knowledge, given the public nature of the information exchanged from C to GM. (The non-reversibility of one-way hash functions is implicitly assumed too.)

$$c_1 \quad GM \xrightarrow{u} C \quad : \quad \{Nonce_{GM}\}_{pk_C}, Hash\{new, Nonce_{GM}, Ndup\}, new, Ndup$$

Communication continues over the main wireless link. The contents of the message over channel c_1 in Fig. 1 have been changed by adding the encryption of $Nonce_{GM}$ with C's public key $\{Nonce_{GM}\}_{pk_C}$ and by applying a one-way hash function to the 3-tuple consisting of the payload *new* , the nonce and another nonce $Ndup$ to be used only once.

How can the coordinator have guarantees about the origin of the message? C can decrypt the first part with its private key and retrieve $Nonce_{GM}$. Then, it computes the digest of the nonce and compares it with that received over the LLC. If the two digests match, C may be reasonably sure that whoever sent $\{Nonce_{GM}\}_{pk_C}$ over channel c_1 is the same mobile host that previously transmitted over the LLC. Further, the whole message is authenticated as coming from the same mobile host, since $Nonce_{GM}$ is introduced as an argument of a

one-way function together with *new* (and $Ndup$). In this way, *new* is tied to $Nonce_{GM}$. From *Assumption 1* in this Section, it follows that the mobile host that originated the message over c_1 is indeed a *group member*.

The nonce $Ndup$ is inserted to avoid replay attacks: unauthorized hosts could eavesdrop on channel c_1 and simply transmit the same message later. Each time GM sends a *new* message, he should randomly generate a nonce $Ndup$ to insert in the packet both as plaintext and as an argument of the hash function. C should record the $Ndup$ he receives and should not to accept any message with the same $Ndup$ in the future.

In the construction above, GM performs a public key encryption only once. Further, there is no connection between this construction and the movement of *group members* from one cell to another: the transmission over LLC happens once only before the first packet is transmitted. There is no relation to the gateways of a single cell.

5.2 The broadcast environment.

Contrary to above, what will be proposed now is a sort of *authentication on demand*. Each mobile host maintains its capability to receive broadcasted messages apart from the fact that the gateways authenticate themselves to it. It is reasonable to suppose that a GM decides to trust a packet as sent by a legitimate gateway or to willingly ask for an authenticity proof.

In the latter case we suggest to exploit part of a mechanism originally developed to sign digital streams, [11]. A digital stream is a long (potentially infinite) sequence of bits. Usually, applications that deal with streams require the user to consume the data he receives at almost the input rate, without excessive delay. For this reason, authenticating digital streams represents a different problem compared with the authentication of finite messages. Traditional digital signature schemes do not fit properly because they require the receiver to process the entire message in order to verify the signature. For the intrinsic nature of some kinds of streams (e.g. live broadcasts), the sender itself does not know the entire sequence to be sent in advance.

Similarities between digital streams and the finite packets exchanged through our protocol are straightforward to highlight: i) with regard to authentication techniques they both require little use of traditional signature schemes (the stream receiver has to check the signature as the packets arrive, the mobile hosts may not have the resources to perform public key operations in their completeness); ii) since each gateway is devoted to simply forwarding packets coming from the coordinator, it does not know the contents of the packets in advance, as in live broadcast.

The idea is to apply the *off-line* solution of [11], where each forwarded packet is treated as a block belonging to a digital stream.

In the pre-authentication phase, LLCs can be used to transmit a first "1-time" public key from the gateway to the petitioning GM. 1-time signature

schemes are a special kind of signature scheme, introduced in [15, 17], much faster to compute and verify than regular signatures. These schemes can be used to sign only one packet. We assume that each gateway can generate an arbitrary number of 1-time public keys.

$$\begin{array}{llll} LLC & GM \xrightarrow{u} G_i & : & request \\ LLC & G_i \xrightarrow{u} GM & : & 1pk_{G_i}^{seq} \end{array}$$

A *group member* that wants authenticated packets asks for the transmission of the first 1-time public key of the gateway responsible for the cell in which the *group member* happens to be located. This transmission happens over the LLC. (For the transmission over the LLC the *group member* is assumed to be close to the gateway.) With notation $1pk_{G_i}^{seq}$ we indicate the seq-th 1-time public key of G_i, where seq is the sequence number of the packet the gateway is to broadcast in the cell (the same seq as in the original protocol, Fig. 1).

$$\begin{array}{llll} c_3 & G_i \xrightarrow{b} \{MH | MH\, is\, in\, cell\, i\} & : & new, seq, 1pk_{G_i}^{seq+1}, \\ & & & Sig^{-1}_{1pk_{G_i}^{seq}}\{Hash\{new, seq, 1pk_{G_i}^{seq+1}\}\} \end{array}$$

G_i broadcasts in its cell the (new, seq) as in the original protocol in Fig.1 along with a 1-time signature of its hash based on the 1-time public key sent over the LLC. (With notation $Sig^{-1}_{1pk^j_{G_i}}\ \{msg\}$ we mean: "msg is signed with the private key corresponding to the j-th 1-time public key of G_i.") A new 1-time public key $1pk_{G_i}^{seq+1}$ is also transmitted and will be used to verify the signature of the $seq+1$ broadcasted message. This structure is repeated for all the packets gateway G_i broadcasts in its cell.

Contrary to what proposed in Subsection 5.1, there is no need for the insertion of a nonce to prevent replay attacks. seq plays the role of the nonce $Ndup$ in the previous construction. seq can not be manipulated since it is an argument of the 1-time signature.

Broadcast communication is received by everybody in the cell but the verification of the 1-time signature is likely to be taken into consideration only by the members who have previously requested the first 1-time public key over the LLC. The other GMs do not take into account the signature and simply consider the payload new and the sequence number seq.

To work correctly, the whole mechanism requires that no packet is lost. The sending of "nack messages" already considered by the protocol under investigation guarantees such a requirement. Suppose a *group member* receives a packet containing a sequence number greater than expected: according to the original protocol, GM asks for the re-transmission of the lost packets (see Section 2). GM can re-build the correct order for verifying the signature because each 1-time public key is strictly related to the sequence number of the packets: the seq-th packet contains the $(seq + 1)$-th public key, to be used to verify the signature of the $(seq + 1)$-th packet, and so on.

6 Related works

The Wired Equivalent Protocol (WEP) has been included in the 802.11 standard [18] for wireless LANs as an attempt to solve security problems of wireless connectivity. The primary goal of WEP is to protect the confidentiality of user data from eavesdropping. A related goal concerns with access control, i.e. how to prevent the injection of new traffic from unauthorized mobile hosts. To this aim, the 802.11 standard includes an optional feature to discard all packets not encrypted according to WEP. In reality we are not interested in the secrecy of the exchanged packets, but rather reverting to WEP in order to achieve authenticity of origin (at least in searching solutions that satisfy Property 1 in Section 4). Unfortunately, WEP contains security flaws that give rise to a number of vulnerabilities prone to attacks, [6, 9, 10, 24].

The use of out-of-band channels to bootstrap authentication in wireless networks was first proposed by Anderson and Stayano in [23]. Their *Resurrecting Duckling* protocol sets up a relationship between two devices, in their terminology a mother and a duckling. In the initial phase of the protocol the two devices exchange a secret key over a LLC established through *physical contact*. Successively, the duckling uses the secret key to recognize its mother over the wireless link.

In [7] Balfanz *et al.* extend the concept of LLCs not only to set up a master-slave relationship but they consider LLCs to be generally used for ad-hoc wireless networks. In this more general context new candidates to become LLCs are introduced, like sound and infrared.

To build up authentication mechanisms for the protocol under investigation, we have chosen to avoid the restrictive condition of physical contact for LLCs in order to exploit in the pre-authentication phase the wireless capability of both the stationary and the mobile hosts.

Analogously to [7], we do not require our LLC to be resistant to eavesdropping. On the contrary, the Resurrecting Duckling protocol of [23] expects a shared secret key to be exchanged over the LLC. This makes the LLC vulnerable to eavesdropping. Being the shared key compromised, all subsequent communications on the main wireless link could be compromised, i.e. an adversary could obtain the information necessary to impersonate someone else. The usage of public key cryptography renders the channel cold to passive eavesdropping over the LLC because of the public nature of the information exchanged. When the goal is the exchange of the public keys over the LLC, the participants will be able to authenticate each other on the main wireless link proving possession of their corresponding private keys.

Contrary to [7, 23], it does not properly follow the definition of an *"ad-hoc wireless network"*: transmissions of meaningful payloads do not take place entirely over wireless links, nor are mobile routers present in the system to forward messages to final mobile receivers. Communications necessarily pass through the stationary hosts on the wired link. Bootstrapping authentication through pre-authentication over LLCs can be applied to more general scenarios

than peer to peer authentication in ad-hoc wireless networks

We found some similarities between the packets exchanged through the protocol under investigation and the potentially infinite sequence of bits denoted as digital streams.

In the original work by Gennaro and Rohatgi, [11], bootstrapping authentication of digital streams is obtained by applying a single traditional digital signature in combination with 1-time signatures. The digital stream is divided into blocks and each block carries a public key, which is used in a 1-time signature scheme to sign the following block. Only the first block needs to be signed with a traditional signature scheme. In our work, the first digital signature has been substituted by the transmission of a first 1-time public key over the LLC. The transmission over the LLC initializes the authentication chain as well as a traditional digital signature and removes the need of a digital certificate (and a related PKI) to certify about the origin of the public key.

The difficulty in the approach of [11] is that if a block is missing, the authentication chain is broken and subsequent packets can not be authenticated. Efficient constructions to solve the problem of authenticating streamed data over channels with packet loss have been recently proposed, [12, 19, 20].

We have not relied on similar constructions due to the intrinsic nature of the protocol for mobile computing under investigation. The original specifications of the protocol were drawn with the specific intention to recover lost packets.

7 Concluding remarks

Starting from a known protocol for distributed mobile systems, we have shown how adversaries can break the authenticity/integrity requirements of the protocol. We have added authentication mechanisms over the wireless links. The mechanisms rely on two different techniques: *Location-Limited Channels* and "1-time" signature schemes. These techniques have been mainly chosen to avoid the need for a Public Key Infrastructure and for the low complexity of the underlying encryption/decryption algorithms. As future work, we plan to i) formally prove the security of the enhanced protocol by using a formal verification environment; ii) introduce data authentication in the extended version of the protocol.

References

[1] A. Acharya and B. Badrinath. A Framework for Delivering Multicast Messages in Networks with Mobile Hosts. *ACM/Baltzer Mobile Networks and Applications*, 1(2):199–219, 1996.

[2] G. Anastasi and A. Bartoli. Group Multicast in Distributed Mobile Systems with Unreliable Wireless Network. In *18th IEEE Symp. Reliable Distributed Systems (SRDS 99)*. IEEE Computer Society Press, 1999.

[3] G. Anastasi and A. Bartoli. On the Structuring of Reliable Multicast Protocols for Mobile Wireless Computing. Technical Report 1/00, Information Engineering Dept., University of Pisa, Italy, January 2001.

[4] G. Anastasi, A. Bartoli, N. De Francesco, and A. Santone. Efficient Verification of a Multicast Protocol for Mobile Computing. *The Computer Journal*, 44(1), 2001.

[5] G. Anastasi, A. Bartoli, and F. Spadoni. A Reliable Multicast Protocol for Distributed Mobile Systems: Design and Evaluation. *IEEE Transactions on Parallel and Distributed Systems*, 12(10):1009–1022, October, 2001.

[6] W. Arbaugh, N. Shankar, and Y.C. Justin Wan. Your 802.11 Wireless Network has No Clothes. In *IEEE International Conference on Wireless LANs and Home Networks*. World Scientific e-proceedings, March, 2001.

[7] D. Balfanz, D.K. Smetters, P. Stewart, and H. Chi Wong. Talking to Strangers: Authentication in Ad-Hoc Wireless Networks. In *Proc. of the 2002 Network and Distributed Systems Security Symposium (NDSS'02)*. The Internet Society, San diego, CA, February 2002.

[8] A. Bartoli. Group-based Multicast and Dynamic Membership in Wireless Networks with Incomplete Spatial Coverage. *ACM/Baltzer Mobile Networks and Applications*, 3(2):175–188, 1998.

[9] N. Borisov, I. Goldberg, and D. Wagner. Intercepting Mobile Communications: the Insecurity of 802.11. In *Proc. of the Seventh Annual International Conference on Mobile Computing and Networking (MOBICOM 2001)*, pages 180–189. ACM, Rome, Italy, July 2001.

[10] S. Fluhrer, I. Mantin, and A. Shamir. Weaknesses in the Key Scheduling Algorithm of RC4. In *Proc. of Eighth Annual Workshop on Selected Areas in Cryptography*, August 2001.

[11] R. Gennaro and P. Rohatgi. How to Sign Digital Streams. *Information and Computation*, 165(1):100–116, 2001.

[12] P. Golle and N. Modadugu. Authenticating Streamed Data in the Presence of Random Packet Loss. In *Proc. of the 2001 Network and Distributed Systems Security Symposium (NDSS'01)*. The Internet Society, San Diego, CA, February 2001.

[13] R. Housley, W. Ford, W. Polk, and D. Solo. Internet X.509 Public Key Infrastructure Certificate and CRL Profile. IETF - Network Working Group., January, 1999.

[14] M. Lamming, M. Eldridge, M.Flynn, C. Jones, and D. Pendlebury. Providing Access to any Document, Any time, Anywhere. *ACM Transactions on Computer-Human Interaction*, 7(3):322–352, 2000.

[15] L. Lamport. Constructing Digital Signatures from a One-Way Function. Technical Report CSL 98, SRI Intl, 1979.

[16] C. Lopes and P. Aguiar. Aerial Acoustic Communications. In *Proc. of the 2001 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2001.

[17] R. Merkle. A Digital Signature based on a Conventional Encryption Function. In *Advances in Cryptology - Crypto '87*, volume LNCS 293, pages 369–378. Springer-Verlag, 1988.

[18] L.M.S.C. of the IEEE Computer Society. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. In *IEEE Standard 802.11, 1999 Edition*, 1999.

[19] A. Perrig. The BiBa One-Time Signature and Broadcast Authentication Protocol. In *Proc. of CCS'01*. ACM, Philadelphia, Pennsylvania, November 2001.

[20] A. Perrig, R. Canetti, D. Song, and D. Tygar. Efficient and Secure Source Authentication for Multicast. In *Proc. of the 2001 Network and Distributed Systems Security Symposium (NDSS'01)*. The Internet Society, San Diego, CA, February 2001.

[21] R. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public Key Cryptosystems. *Comm. of the ACM*, 21(2):120–126, 1978.

[22] F.B. Schneider. *Applied Cryptography*. J. Wiley & sons, Inc, 1996.

[23] F. Stajano and R.J. Anderson. The Resurrecting Duckling: Security Issues for Ad-Hoc Wireless Networks. In *Proc. of 7th Security Protocols Workshop*, volume LNCS 1796, pages 172–194. Springer-Verlag, 1999.

[24] A. Stubblefield, J. Ioannidis, and A.D. Rubin. Using the Fluhrer, Mantin and Shamir Attack to Break WEP. In *Proc. of the 2002 Network and Distributed Systems Security Symposium (NDSS'02)*. The Internet Society, San Diego, CA, February 2002.