Authenticity in a Reliable Protocol for Mobile Computing*

Nicoletta De Francesco
Dipartimento di Ingegneria dell'Informazione
Università di Pisa
via Diotisalvi 2 - 56122 Pisa, Italy
nicoletta.defrancesco@iet.unipi.it

Marinella Petrocchi
Istituto di Informatica e Telematica
Consiglio Nazionale delle Ricerche
via Moruzzi 1 - 56124 Pisa, Italy
marinella.petrocchi@iit.cnr.it

ABSTRACT

We consider a known protocol for reliable multicast in distributed mobile systems where mobile hosts 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: an adversary may eavesdrop on communications between hosts and inject packets over the wireless links. We suggest a secured version of the protocol providing authenticity and integrity of packets over the wireless links. The secure mechanisms introduced rely on two different techniques: *Location-Limited Channels* and "1-time" signature schemes. Further, we outline the formal verification of part of the secured protocol.

Keywords

Authentication of origin, distributed systems security, wireless communications, formal analysis.

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.

*We gratefully acknowledge funding support for this research. This research was partially supported by CNR project "Tecniche e Strumenti Software per l'Analisi della Sicurezza delle Comunicazioni in Applicazioni Telematiche di Interesse Economico e Sociale"; by MIUR project "Strumenti, Ambienti ed Applicazioni Innovative per la Società dell'Informazione"; by CSP project "SeTAPS: Strumenti e Tecniche per l'Analisi di Protocolli di Sicurezza". The view and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the opinions or policies, either expressed or implied, of CNR, MIUR, CSP, University of Pisa and Istituto di Informatica e Telematica.

SAC 2003, Melbourne, Florida, USA

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, [7, 24].

We consider a simplified version of the multicast protocol for mobile computing developed in [1, 6]. Design issues are considered in the cited papers in order to support reliable and totally ordered communication within a set of processes, 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 a set of *legitimate hosts* are authorized to participate through the protocol. Unauthorized mobile hosts will be treated as potential adversaries. We propose an extension of the original protocol with authentication mechanisms to assure the capability for the legitimate hosts to authenticate a packet as originated by a legitimate host. The proposed protocol could be fruitfully used in environments where authenticity and integrity are the main concerns, whereas secrecy is less important. For example, taking into consideration a university environment, a professor and his assistants, whilst exchanging students grades, are more worried about the integrity rather than the secrecy of the information exchanged. We base the extension on some concepts coming from proposals by Balfanz *et* al., [5], which seem to fit our context quite well. Further, an analogy between digital streams considered in [10] 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. In Section 4 we propose a new version of the protocol, enriched with security mechanisms to authenticate the exchanged packets. Section 5 gives a sketch of our ongoing research regarding the formal verification of the secured pro-

tocol. In Section 6 we discuss related works in the wireless authentication area. Section 7 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 and it is not fixed in advance. 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, [2]. The full version can be found in [6], along with a detailed discussion of its motivation and advantages. A formal analysis of the properties of the protocol, concerned with total order delivery, no duplicates and losses recovery, has been performed in [2].

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 cell. The resulting scenario is quite general since it can accommodate contemporary wireless LANs, infrared networks, picocellular wireless networks where cells coincide

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

Figure 1: Transmitting a *new* message.

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 other hosts. 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.

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.

The protocol does not use any notion of hand-off, i.e. it does not require any data exchange between the old gateway and new when a host moves from one cell to another.

3. CONSIDERING A HOSTILE ENVIRONMENT

Considering security communication protocols, cryptographic functions are introduced in the structure of messages in

[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.

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 set of *legitimate hosts*, denoted as LMHs, authorized to participate through the protocol plus an unauthorized mobile adversary with the adequate technical equipment to eavesdrop on traffic and actively inject packets over the wireless links. For details about the practicality of such interferences we refer to [7, 24].

The protocol was originally designed to guarantee a set of properties, among which: i) *Integrity*, requiring that any packet received by a legitimate host has been originated by a legitimate host; ii) *No Duplicate*, stating that no legitimate host accepts duplicate packets, i.e. duplicates are discarded.

The properties of the protocol have been formally analyzed in [2]. However, these properties might not hold in a classical context of security analysis where the presence of an adversary has to be considered. Without any authentication mechanism, an unauthorized host can eavesdrop on and inject traffic over the wireless links.

We now define two properties regarding authentication between legitimate hosts and stationary hosts:

(P_A) Capability for the coordinator to authenticate the sender of a *new* message as a legitimate host.

(P_B) Capability for all the legitimate hosts to authenticate the received broadcasts as indeed originated from the stationary host responsible for the cell in which they happen to be located.

We do not require that a gateway correctly identifies a LMH 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.

We propose to add security procedures to the original protocol to make properties P_A and P_B hold.

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 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, [12]. In a wireless environment, the management of digital certificates could result in a bottleneck for the whole system. Subsection 3.1 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 4.2).

3.1 Bootstrapping authentication

We make use of a method for bootstrapping authenticated and integral communication between mobile hosts participating in the protocol. A *pre-authentication* phase, in which a certain amount of information is exchanged between legitimate hosts 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 [5]. 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, [13, 15].

We 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. The practicability of such pre-authentication schemes in wireless environments has been successfully shown in [5].

4. THE SECURED PROTOCOL

The design of a secured protocol will be introduced in this section by adding cryptographic mechanisms in order to guarantee security properties. In the following we suppose the presence of unauthorized mobile hosts, i.e. hosts that

are not legitimate. To distinguish between authorized and unauthorized hosts, we write LMH to denote a legitimate host. Each LMH is also a mobile host MH but the opposite is not true: a generic MH is not necessarily a legitimate host.

We assume that the multicast over channel c_2 in Fig. 1 can not be compromised. We trust the agents 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 use LLCs with the following assumptions: i) of all the mobile hosts, *only* the legitimate hosts can transmit over the LLC; ii) the stationary hosts can transmit over the LLC and 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 legitimate host to the coordinator, while the second is concerned with the authentication of the gateways G_i to the legitimate hosts currently present in cell i. Even though we loose the precise identity of the sender of a new message, we are interested only in *g*eneric authenticity (each legitimate host can recognize whether a message was sent by a legitimate host) rather than *s*ource authenticity (the capability to identify the single party within a set).

4.1 Authenticating the mobile sender

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

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

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 $Nonce_{LMH}$ 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 LMH. (The non-reversibility of one-way hash functions is implicitly assumed too.)

$$\begin{array}{llll} c_0 & LMH \xrightarrow{u} C & : & \{Nonce_{LMH}\}_{pk_C} \\ c_1 & LMH \xrightarrow{u} C & : & Hash\{new, Nonce_{LMH}, Ndup\}, \\ & & & new, Ndup \end{array}$$

Communication continues over the main wireless link. Communication over channel c_0 has been added compared with the original protocol: LMH sends the encryption of $Nonce_{LMH}$ with C's public key $\{Nonce_{LMH}\}_{pk_C}$. The contents of the message over channel c_1 in Fig. 1 have been changed 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 message over c_0 with its private key and retrieve $Nonce_{LMH}$. 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 $\{Nonce_{LMH}\}_{pk_C}$ over channel c_0 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 $Nonce_{LMH}$ is introduced as an argument of a one-way function together with *new* (and $Ndup$). In this way, *new* is tied to $Nonce_{LMH}$. From assumption i) on LLCs (see this section), it follows that the mobile host that originated the message over c_1 is indeed a legitimate host.

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 LMH 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 accept any message with the same $Ndup$ in the future.

We get into the issue of defining the expiry period for the information sent from LMH to C over the LLC. We suggest that LMH generates an explicit request to invalidate the nonce $Nonce_{LMH}$. This request may be sent over the LLC and may contain the nonce as a clear text seeing that once the nonce has been invalidated its knowledge on behalf of an adversary is irrelevant. On the other hand, the very first request reasonably comes from the legitimate LMH, the one (except the coordinator) to know $Nonce_{LMH}$.

In the construction above, LMH performs a public key encryption only once. Further, there is no connection between this construction and the movement of hosts from one cell to another: the transmission over LLC and channel c_0 happens once only before the first packet is transmitted. There is no relation to the gateways of a single cell.

4.2 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 LMH 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, [10]. 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.

Each forwarded packet will be treated as a block belonging to a digital stream, see [10].

In the pre-authentication phase, LLCs is used to transmit a first "1-time" public key from the gateway to the petitioning LMH. 1-time signature schemes are a special kind of signature scheme introduced in [14], 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.

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

A legitimate host 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 host happens to be located. This transmission happens over the LLC. (For the transmission over the LLC the host is assumed to be close to the gateway.) With notation $1pk_{G_i}^{seq}$ we indicate the $s$eq-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).

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

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: "$m$sg 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 4.1, 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 hosts who have previously requested the first 1-time public key over the LLC. The other LMHs 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 host receives a packet containing a sequence number greater than expected: according to the original protocol, LMH asks for the re-transmission of the lost packets (see Section 2). LMH 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.

We give an informal justification on the correctness of the secured construction. Suppose a generic LMH, at the n-th point of the computation, accepts the following as an authentic message:

$$\begin{array}{l} new^X, n, 1pk_{G_i}^{n+1}, \\ Sig^{-1}_{1pk^n_{G_i}} \{Hash\{new^X, n, 1pk_{G_i}^{n+1}\}\} \end{array}$$

where new^X is originating from an unauthorized mobile host. If the generic LMH accepts this message as being authentic, it means that i) the adversary knows the private key corresponding to the n-th 1-time public key of the gateway and consequently he is able to reproduce valid signatures (which shouldn't be possible seeing that private information of the stationary hosts are never exchanged during the protocol) or ii) the adversary was able to forge the authentication chain by inserting in message (n-1)-th his own 1-time public key. This is possible only if the unauthorized host is able to insert his own 1-time public key at the very first communication over LLC. Taking into consideration the assumption on transmission over LLCs, it follows that this possibility, in reality, is infeasible.

The adversary could be able to inject over cell i a broadcast coming from a stationary host responsible for another cell. A generic LMH in cell i does not accept this message as an authentic message since he does not have the right 1-time public key to verify the signature (the last authentication chain in which he has involved is the chain related to the stationary host in which he happens to be located).

5. A FRAMEWORK FOR A FORMAL VERIFICATION

This section gives an outline of our present research regarding the formal verification of the secured protocol. Our preliminary analysis consisted in verifying property (P_B) listed in Section 3.

We formally analyzed if property (P_B) holds using the software tool PAMOCHSA, [17], which automatically analyses cryptographic protocols and reveals, at a conceptual level, attacks against security procedures. A detailed description of the theory behind the development of the tool can be find in [16].

Making use of model checking techniques to analyze protocols like the one under investigation represents an interesting challenge given the diversity of such protocols from standard classed cryptographic schemes in the way that a sender broadcasts a continuous (and possibly unbounded) stream of messages. Receivers use information retrieved in

earlier packets to authenticate later packets. Furthermore, our particular scenario includes mobility, i.e. mobile hosts can move from one cell to another. Currently, the successful outcome of this kind of analysis applied to these classes of protocols is in question. In [4] Archer states such an analysis is not feasible, on the other hand Broadfoot and Lowe show their successful results derived applying model checking techniques on a well known stream authentication protocol, motivating, even though informally, several steps of the analysis, [8].

We concentrated our analysis on the second part of the secured protocol, see Subsection 4.2, which concerns a broadcasting environment. We considered an environment consisting of 2 cells (and therefore 2 gateways). A special process *environment*, which is a simple sequential process responsible for the sending of *new* messages to the gateways, has been included. The reason for introducing this last fictitious process is to cut off the first part of the protocol. Our analysis may appear limited since it takes into account only a finite number of LMHs and messages and treats an incomplete model of the hosts' mobility. However, we can argue that the analysis over a reduced specification set does not fail to detect attacks respect to analysis over a specification set which include the following:

1. an unbounded (and possible dynamically changeable) number of LMHs;
2. an unbounded number of messages broadcast by the gateways;
3. the LMHs' capability to move infinitely from one cell to another.

We leave the formal discussion about the correctness of our analysis for a future work. Regarding our reduced specifications set, no attack was found. This implicates that the analyzed property holds.

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. Unfortunately, WEP contains security flaws that give rise to a number of vulnerabilities prone to attacks, [3, 7, 9, 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 [5] 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. 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 [5], 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. Contrary to [5, 23], our environment 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, [10], 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 [10] 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, [11, 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 added authentication mechanisms over the wireless links. The mechanisms rely on two different techniques: secure wireless channels to initialize the communications 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) complete the formal security proof of the secured protocol making use of model checking techniques; ii) compare the performances of the original protocol and the secured protocol in terms of the end-to-end delay (or *latency*), i.e. the time experienced by a message along its way from a sending host to a receiving host.

8. ACKNOWLEDGMENTS

We are very grateful to Claud Anticoli for his valued collaboration as proof reader. Special thanks also go to Fabio Martinelli especially for clarifying and fruitful discussions of the formal verification of secure multicast. We would like to thank Giuseppe Anastasi for his careful feedback and advice on wireless multicast protocols. We would also like to thank the anonymous referees for their useful comments. We are indebted to Giordano Fusco for his helpful editorial advice.

9. REFERENCES

[1] G. Anastasi and A. Bartoli. Group Multicast in Distributed Mobile Systems with Unreliable Wireless Network. In *Proc. of SRDS 99*. IEEE Computer Society Press, 1999.

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

[3] W. Arbaugh, N. Shankar, and Y. J. 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.

[4] M. Archer. Proving Correctness of the Basic TESLA Multicast Stream Authentication Protocol with TAME. In *Proc. of ACM SIGPLAN and IFIP WG 1.7 WITS'02*, Portland, USA, January 2002.

[5] D. Balfanz, D. Smetters, P. Stewart, and H. C. Wong. Talking to Strangers: Authentication in Ad-Hoc Wireless Networks. In *Proc. of NDSS'02*. The Internet Society, San Diego, CA, February 2002.

[6] 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.

[7] N. Borisov, I. Goldberg, and D. Wagner. Intercepting Mobile Communications: the Insecurity of 802.11. In *Proc. of MOBICOM'01*, pages 180–189. ACM, Rome, Italy, July 2001.

[8] P. Broadfoot and G. Lowe. Analysing a Stream Authentication Protocol using Model Checking. In *Proc. of ESORICS'02*, volume LNCS 2502, pages 146–161. Springer, October 2002.

[9] 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.

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

[11] P. Golle and N. Modadugu. Authenticating Streamed Data in the Presence of Random Packet Loss. In *Proc. of NDSS'01*. The Internet Society, San Diego, CA, February 2001.

[12] 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.

[13] 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.

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

[15] C. Lopes and P. Aguiar. Aerial Acoustic Communications. In *Proc. of WASPAA '01*. IEEE Computer Society Press, 2001.

[16] F. Martinelli. Analysis of Security Protocols as Open Systems. *Theoretical Computer Science (to appear)*. A preliminary version in ICTCS, World Scientific, pages 304-315, 1998.

[17] F. Martinelli, M. Petrocchi, and A. Vaccarelli. Automated Analysis of Some Security Mechanisms of SCEP. In *Proc. of ISC'02*, volume LNCS 2433, pages 414–429. Springer, October 2002.

[18] L. 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 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. Schneider. *Applied Cryptography*. J. Wiley & sons, Inc, 1996.

[23] F. Stajano and R. 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. Rubin. Using the Fluhrer, Mantin and Shamir Attack to Break WEP. In *Proc. of NDSS'02*. The Internet Society, San Diego, CA, February 2002.