SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks. 2008; **1**:17–24
Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/sec.1

Fast distributed bi-directional authentication for wireless sensor networks

Romano Fantacci*,†, Francesco Chiti and Leonardo Maccari
Telecommunication Network Lab, Department of Electronics and Telecommunications, University of Florence, Florence, Italy

Summary

In this paper, we present the comparison between a distributed and a centralized authentication protocol for wireless sensor networks (WSN). We outline the difference between authentication and key-agreement schemes and we propose a novel approach based on the use of polynomial functions to produce a distributed bi-directional authentication. The advantages of this approach are: speed of operation that can allow multiple subsequent authentications, thus support to mobility; balanced energy consumption if compared with the imbalanced centralized approach and the prevention of partition attacks. Copyright © 2008 John Wiley & Sons, Ltd.

KEY WORDS: sensor networks; security; authentication

1. Introduction

Wireless sensor networks (WSN) are distributed, large scale networks comprised of nodes with minimal hardware capacity. They are purpose-specific networks, normally used to monitor large geographical areas; typical applications are environmental monitoring, health-care, surveillance. Being large area network, the nodes are normally unattended; this introduces major challenges in two fields:

- The nodes should be energy efficient, since human intervention is needed to substitute batteries. This imposes strict limits on the bitrates and on the computational power of devices.
- The nodes could be stolen. This is a major issue when dealing with WSN security, since an attacker might have the possibility of stealing a device from the network, extract from the device sensitive data (such as cryptographic keys) and perform an attack behaving as an internal attacker.

WSN may be composed of thousands of nodes, so the cost of each individual device should be kept to the minimum; under this assumption it is hardly conceivable to use anti-tampering trusted hardware. Nevertheless, in applications in which security takes great importance, dealing with stolen and possibly re-programmed nodes is a very delicate task.

One more issue that makes the management of security in WSN even more challenging is the possible presence of mobile nodes in the network. Mobility can be introduced for various reasons, i.e., mobile sinks (nodes that collect information) can be used to control

*Correspondence to: Romano Fantacci, Telecommunication Network Lab, Department of Electronics and Telecommunications, University of Florence, Florence, Italy.
†E-mail: fantacci@lart.det.unifi.it

the state of the network or even the nodes in certain environment can be mobile, for example, to be able to change the density of the network in locations where a more complete coverage is needed. Mobility introduces the concept of handover, that is the phase in which a mobile node changes its point of attachment from a node of the network to another. During the handover various operations have to be performed, some of which are security-oriented, i.e., the roaming node should use some kind of authentication protocol with its new neighbor, to be able to set-up a secure link.

In this paper, we review the problems related with the use of security services, such as authentication and access control to be able to guarantee network security even when some nodes are mobile ones. We take into special consideration the difference between authentication protocols and key-negotiation protocols, that are widely used in WSN. As we will show here, key-negotiation is not enough to guarantee robustness against certain attacks, so that some form of stronger security measure is needed. Two means are adopted in other kinds of network to assure bi-directional authentication: public/private key security schemes or a shared key protocol that uses a centralized server. While the first one is inapplicable in WSN for computational inadequates we will show that also the latter is hardly usable.

We will then introduce a novel approach based on a distributed algorithm to provide lightweight authentication service with support to mobile nodes. Our model is based on the delegation of the authentication and access control phase to a coalition of nodes, geographically close, that take the responsibility of admitting a newcomer (or simply a roaming node). Using a modified form of a well-know secret sharing algorithm we guarantee that up to the take over of a certain threshold of nodes in the network the authentication is bi-directional and secure, without the need of a centralized authentication server (AS).

2. Bi-directional Authentication in WSN

Access control is the security service that is supported by networks that wish to have a tight control on the entrance of new nodes (see Stallings [1]). One of the key features of access control is the authentication of the node that wants to join the network, based on some cryptographic credentials that are embedded in the device. One feature that is desired for any access control policies is that the authentication should be bi-directional. Bi-directional authentication guarantees to the entering node that it is in contact with the correct network, in order to avoid the creation of rogue networks where a roaming node can be attracted. With such an attack it is possible to create partitions into the network, isolating entire areas and thus preventing an efficient monitoring. A rogue network can be set-up in two cases, when the authentication is not bi-directional, or when the attacker is in possession of the credentials to be identified as part of the network. How can a *network* authenticate itself with a node? Generally this function is delegated to a dedicated machine, the *authentication server* (AS), that is contacted at any new entrance or handover. The roaming node shares credentials with the AS and performs with it a bi-directional authentication, thus it is assured to enter the correct network. In distributed networks, such as *ad hoc* networks or mesh networks, a new client never enters directly in contact with the AS, but with another machine that is placed on the border of the network and it is able to route traffic to the AS. The authentication is so performed using a multi-hop path to the AS. An attacker that wants to perform a partition attack should be in possession of the credentials of the AS, since robust authentication protocols do not suffer from man in the middle attacks.

An alternative to such an approach is given by the use of public/private key cryptography. If each node is equipped with a valid certificate, released by the manager of the network, then the newcomer even without performing authentication with an AS can be sure that the node that is communicating with is part of the correct network, since it is in possession of a valid certificate.

Public key cryptography is hardly applicable to WSN, since the computational power needed is beyond the possibility of the devices, so the only viable approach to have bi-directional authentication considered so far is the centralized one. There are a few issues that make the centralized approach unsuitable for large area WSN:

- Routing: if the WSN are large enough, to perform a bi-directional authentication that needs the exchange of frames in both direction a unicast routing protocol is needed. In WSN, data are normally conveyed by the nodes to a gateway, and there is no need of a routing protocol to convey data from the gateway to the nodes.
- Delay of the authentication: a bi-directional authentication based on shared keys involves the exchange of at least four frames between the two parties [2]. If the network is composed of thousands

DOI: 10.1002/sec

of nodes the path can be made up of tens of hops. This implies that the authentication can take several seconds to be performed.
- Asymmetry of energy consumption: every authentication, and re-authentication produces the generation of packets in the network, but the load is not equally distributed. The nodes that are geographically close to the AS are involved with higher probability into the authentication procedure.

Using a centralized authentication scheme can lead to inefficiencies, especially, the concentration of energy consumption in the neighborhood of the AS may provoke the isolation of the AS from the rest of the network. Note that if mobile nodes are present, the number of re-authentications depends on the number and type of mobile nodes, so that the prediction of the energy consumption is hard to perform in advance.

For these reasons the majority of the techniques present in literature do not deal with authentication, but introduce methods for key-agreement between nodes. Key agreement is normally a wanted side effect of a successful authentication. Once completed the handshake that constitutes authentication, the new node should be in possession of a fresh symmetric key shared with the node that worked as intermediary to the network. Specific key agreement protocols are based on the hypothesis that two nodes share some common keying material and are able to derive new keys from that (e.g., applications see [3,4]). Once completed the procedure, each participant is assured that its correspondent was originally in possession of the same shared secret. This is enough to state that node has been originally part of the network, but has no use against attacks based on node theft. The node could have been re-programmed, and the newcomer can be lead to enter in a rogue network. Note that if the WSN is, for example, a surveillance network, it might be difficult to steal a big amount of nodes, but once only one of them has been stolen, the attacker might introduce into the network more devices under its control, equipped with the stolen key. Under this scenario the partition attacks represents a huge threat.

3. Node Authentication: A Distributed Approach

A different approach, that we introduce with this paper, is based on the participation of a group of nodes that cooperate to perform the authentication of the new coming node. The idea behind this approach is that whenever a node A is joining the network, or simply performing an handover from another point of attachment, it will chose a node B, that will be its intermediary to the WSN. The node B will contact all its one-hop neighbors asking for a contribution to the authentication decision, if the number of nodes responding is lower than a threshold, it will contact all its two-hop neighbors, and so on until the threshold is reached. All the neighbors will answer with a single packet, that will be directed to node B or will be routed through the one-hop neighbors if it is coming from nodes two-hop away. From now on, recalling IEEE 802.1X terminology, we will call the entering node a *supplicant*, the intermediary the *authenticator* and the other nodes involved *distributed authentication servers* (dAS). Before entering into details of the algorithm, we outline that this procedure has the following advantages:

- All the authentications' frames exchanged are confined in a limited geographical zone, even when multi-hop is involved, requests are sent in broadcast frames, so they are processed in parallel by the authentication servers.
- No routing is required, even in multi-hop exchanges, each node sends responses only to the node from which it received the request.

We expect such a protocol to offer higher performance than the centralized one in terms of time needed to perform the authentication, to be less prone to malfunctioning due to loss of frames and to offer a comparable level of security.

The algorithm is based on Shamir's polynomial secret sharing scheme [5], briefly, we recall the algorithm:

In Shamir's scheme the secret is a number (a y value of the polynomial) that could be used as a symmetric key to unlock a larger set of data. It works as follows:

- At start-up a trusted party generates a polynomial $q(x)$ of degree $K-1$ in a way that the secret S is given by $S = q(0)$.
- A set of N couples $(x_i, q(x_i))$ is generated and each actor receives one couple (a *share* of the secret).
- If the secret has to be revealed, at least K actors have to join their shares to make interpolation of the original polynomial possible.
- Knowledge of a set of less than K shares gives no information about the secret.

DOI: 10.1002/sec

Efficient functions for polynomial interpolation are present in literature and can be easily ported to sensor nodes. A naive porting of Shamir's secret sharing to be used as an authentication protocol could be done distributing a share of the secret to each node, and whenever a supplicant joins the network the authenticator collects from its neighbors K shares to interpolate the polynomial, if also the supplicant sends its share of the secret to the authenticator, the authenticator can verify that all the shares belong to the same polynomial. In that case the node is admitted.

This simple protocol needs many corrections to be used as an authentication protocol, some are:

- As said, the authentication should be bi-directional, so that also the entering node has to be guaranteed that the intermediary is not part of a rogue network, but the decision has been taken based on the participation of at least K nodes.
- Once the polynomial is interpolated, the authenticator node contains all the data needed to forge new credentials. If the node is stolen, or the node is controlled by the attacker during the authentication phase, then the security of the whole network is threatened. The authentication should then be based on a fresh polynomial derived from the original one.

This said, our idea is to generalize Shamir's scheme adding three more features:

- At start-up a set P of M polynomials of degree $K-1$ is generated, and each polynomial is sampled at values x_i; each node in the network is equipped with a vector $V_i = [P_0(x_i), P_1(x_i) \dots P_{M-1}(x_i)]$ and the correspondent x_i value.
- When the protocol starts, the first two frames are needed to generate a random vector of size M. The coefficients of the vectors are used to generate a new fresh polynomial with linear combination, so no private data are revealed.
- To overcome the limit due to the numbers of neighbors each node that receives a request can re-broadcast the request to its neighbors; this procedure is called *delegation*.

Referring to Figure 1, we give more details of the protocol: a simple scheme for random vector generation is the following, where a *Nonce* is a fresh random value and H( ) is a one-way hash function:

(1) $N_1 \rightarrow N_2$: [*Nonce*1]
(2) $N_1 \leftarrow N_2$: [*Nonce*2, H(*Nonce*1)]

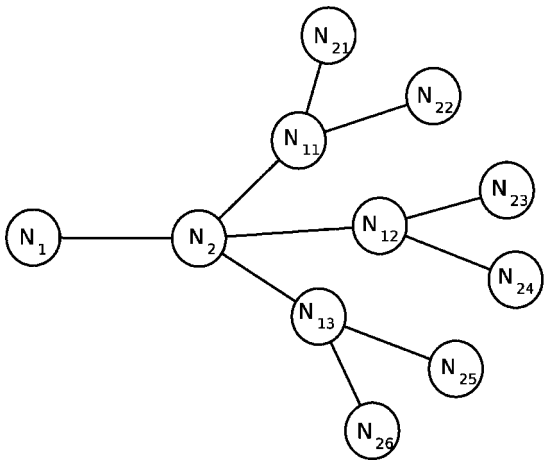

Fig. 1. N_1 is entering the network, N_2 is chosen as an intermediary, the others are nodes already authenticated to the network so we assume the communication between them and N_2 is secured by a key generated in the previous authentication.

(3) both nodes can generate $r = H(Nonce1, Nonce2)$

From r a vector $R = [r_0, r_1 \dots r_{M-1}]$ of arbitrary length can be derived with multiple hashes, discarding zero values.

Once N_1 enters the network it is in possession of vector V_1, it chooses an intermediary (N_2) and both r and R are generated.

The value r is transmitted to N_2's neighbors, which are able to recreate R. Each dAS computes $v_{ir} = \vec{R}\vec{V_i}$ and forwards back the couple (v_{ir}, x_i) to N_2 which in turn is able to compute the whole fresh polynomial $C(x)$, which is a random linear combination of the M initial polynomials. N_1 computes v_1, and sends (x_1, $H(v_1)$) to N_2 which can compute $C(x_1)$ and verify that $H(v_1) = H(C(x_1))$. If this stands, then the authentication had success and the result is that N_1 and N_2 should now be in possession of a fresh secret value $C(x_1)$, but the intermediary has not interpolated any of the initial polynomials but a random linear combination of them. Carefully choosing the size of the integers avoids the re-use of the same linear combination in successive authentications, since R is chosen randomly. From that moment on, N_1 and N_2 can use $C(x_1)$ as a key for encryption and message authentication, thus, also key agreement has been performed. This is an important point in the protocol, the communications between the authenticator and the dAS's needs to be ciphered, or an external attacker could try to collect enough data to re-interpolate the linear combination of the polynomials, and so derive $C(x)$.

DOI: 10.1002/sec

If the authenticator has not enough neighbors, it can use the delegation function, implemented as a counter in the frames. If a dAS receives a frame with the counter set to a greater value than zero, it will forward the request to its neighbors, with the counter decremented, thus reaching all the nodes at distance of two hops from the authenticator.

3.1. Security Considerations

Upon completion of the authentication, the authenticator node is assured that the supplicant is in possess of correct credentials to enter the network. At the same time, the supplicant node can be sure that the authentication has been performed not only with its authenticator, but with the participation of at least K nodes of the network. This is what we wanted to achieve, the supplicant has a proof that it is entering a network that is composed of at least K well-behaving nodes and is what we mean with *distributed bi-directional authentication* . K is a hard limit that has to be reached in order to assure the correct authentication, note that with delegation the limit of K can always be reached. This assumption stands up to the compromission of K nodes, if an attacker is able to steal K nodes then he can derive all the polynomials and insert into the network rogue authenticators. This is an intrinsic limit of threshold cryptography that represents the price to pay whenever a centralized algorithm cannot be used.

On the other side there is no reason why the authenticator should limit the request to K neighbors, they can be forwarded to any number greater than K neighbors. This can neutralize denial of service attacks on the authenticator: imagine that a node under control of the attacker is involved as a dAS in an authentication procedure, it could send wrong results to make authentication fail, producing a DoS. If more than K shares are collected, upon failure of an authentication a cheating detection algorithm can be used, calculating the polynomial using subsets of size K of shares, if some of them do not fail, then the cheater can be detected. This algorithm has been implemented and is under evaluation in real mica motes, it must be noted that once performed the first interpolation, repeating the interpolation changing only one share is computationally lighter for the re-use of already calculated factors.

Note also that with a centralized authentication, if the attacker is able to control a few nodes close to the AS, a high percentage of authentications can be interrupted with a black-hole attack. Even if the nodes are far from the AS, routing protocols can be diverted in order to deviate packets to the black hole. With the distributed approach this attack is not applicable.

4. Simulation Results

The distributed authentication protocol introduced in the previous section has been partly tested and compared to a centralized protocol, on the model described in Section 2. The simulations have been performed using Omnetpp simulator with the Castalia package [6], the considered scenario is a squared grid of 30×30 nodes, in open space without shadowing effect. The physical parameters have been chosen so that each node can communicate with all its closest neighbors but cannot reach nodes two-hop away. The MAC layer is a simple CSMA.

The centralized authentication protocol has been modeled using a geographical routing, each node is aware of its position and can calculate the closest one among its neighbors in the correct direction towards the final destination. Each frame is acknowledged up to the second re-transmission, then dropped. Authentications are not repeated, after a timeout they are considered failed. A total of 35 authentication have been performed, with 3 seconds of interval, the AS is placed at coordinates (0,0), each authentication is composed of the total exchange of four frames, two in each direction.

In Figure 2 is reported the 3D profile of one of the simulations run. It is evident that the number of transmitted frames concentrates in proximity of the AS. Since the routing protocol is geographic, nodes in the average shortest paths to the AS are more used. The average time for a full authentication procedure is 1.0160 s, the interval is between 0.1651 s and 1.8510 s.

In Figure 3 is reported the number of packets sent by nodes depending on their geographical position. Even if the average is a low value (less than one packet per authentication), the distribution is extremely steep.

A few issues need to be outlined:

- The only traffic present in the simulation scenario is the authentication traffic, that are never contemporary, we expect that with more traffic a higher number of collisions would lead to more failures.
- Authentication time is below 1.8510 s, but no computation time is considered. Failed authentications are not repeated, but in real scenarios each one will be repeated until success, increasing average delay.

DOI: 10.1002/sec

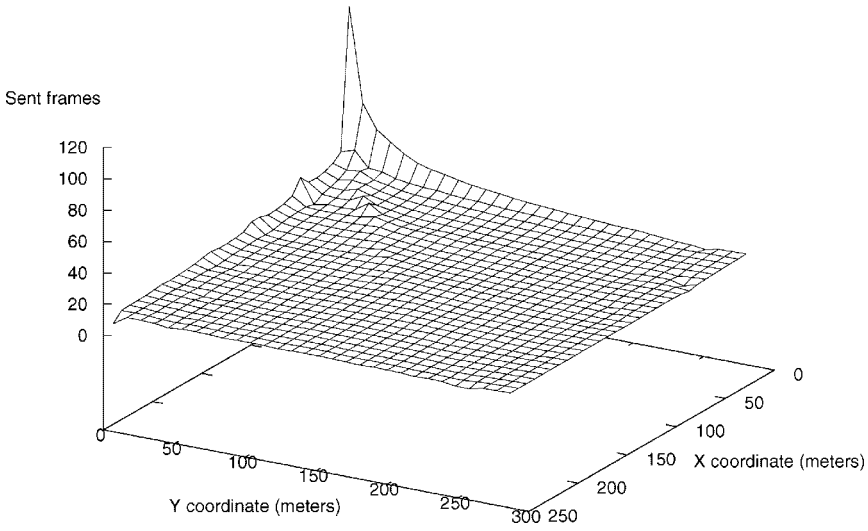

Fig. 2. The geographical distribution of sent packets in a single run of the centralized model simulation. Nodes that are close to the AS (placed in the origin of the coordinates), or on average shortest paths are more subject to energy drain due to packet sending.

- As expected, nodes close to the AS are highly stressed.

The distributed authentication scenario is again, a 30×30 grid of nodes, each authentication is performed using only the first-hops neighbors, with the same parameters of the centralized scenario. The geographical distribution of sent frames is reported in Figure 4. A total of 822 nodes have been involved in at least one authentication, the average of sent packets is 3 per node. Some nodes have been involved in more than one authentication or repetition of each authentication, up to a maximum of 18 packets sent. In Figure 5, the frequency of the sent packets per node is represented.

We see that the average value of packets sent per node is much lower than the previous case and, as expected, there is no real geographical dependency. The average

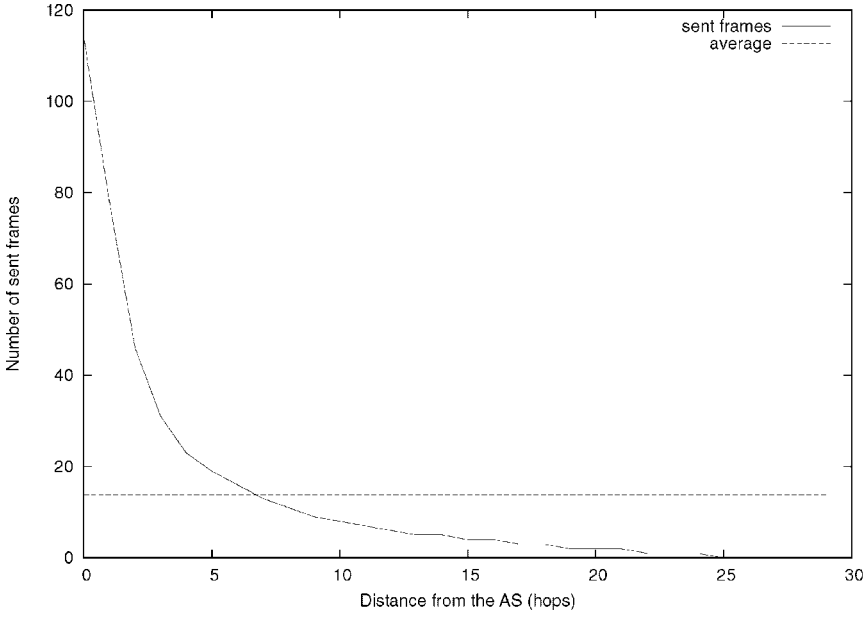

Fig. 3. The average number of packets sent by each node in function of the distance from the AS in terms of hops, after a run with 35 authentications with two failures.

DOI: 10.1002/sec

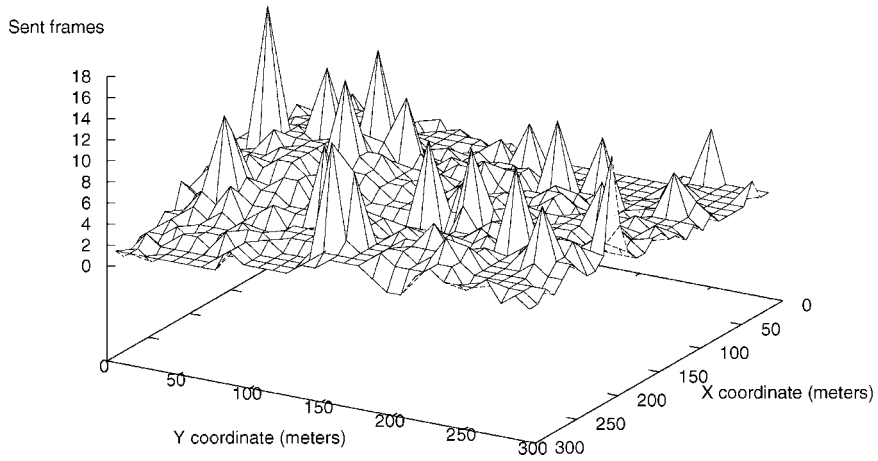

Fig. 4. The geographical distribution of sent packets in a single run of the distributed model simulation. Authenticators send and receive more frames, but there is no geographical dependency.

duration of each authentication is 0.4899, while the interval is between 0.1570 and 1.0660 seconds. These values strictly depend on the simulation parameters:

- The K parameter (the number of shares to be received) has been set to 7, which is the maximum number of neighbors a node could reach in a single hop (excluding the supplicant). An average of 15 authentications have to be repeated once in order to complete all 30 authentications, each repeated authentication introduces greater delays.
- The interval between the first and second retry is relevant in the distributed authentication. We did not explore the best value, but we believe it can be lowered.
- The delay is independent from the distance from the AS, while in a centralized model the number of hops determine the time needed for the authentication.

Our simulation model does not still support mobility, but the authentication time is short enough to support

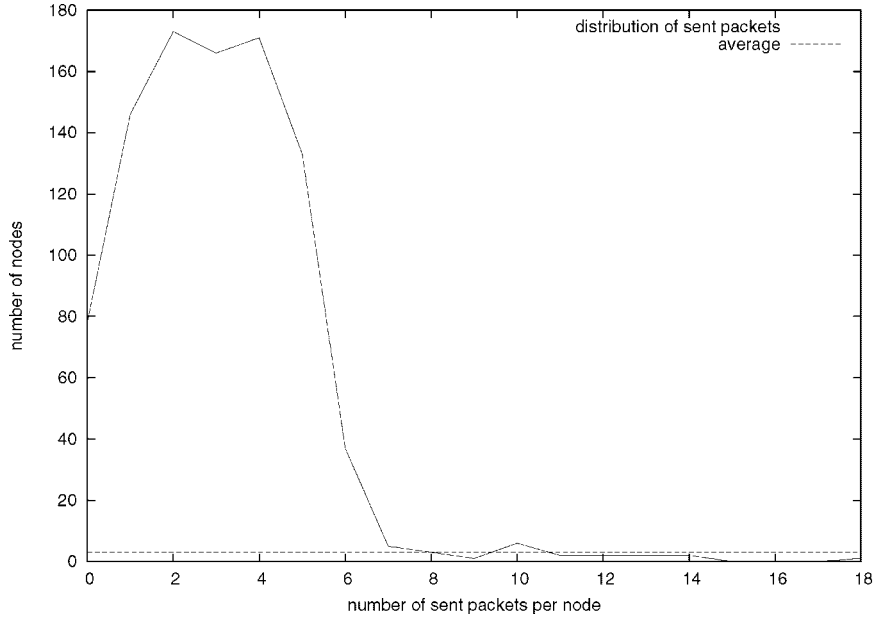

Fig. 5. Distribution of the number of sent frames per node. On the X-axis, the number of sent packets, on the Y -axis the frequency.

DOI: 10.1002/sec

multiple subsequent authentications. The algorithm has been implemented on real WSN devices, the performances are still under analysis but the first results are promising, since the computational overhead of polynomial interpolation and hash functions is sustainable by common sensor hardware without adding significant delay.

Also note that the payload size of the frames sent to and by each dAS are of limited size. Each dAS receives a frame containing a hash value (16/20 bytes, as needed by a standard MD5/SHA hash function) and answers with a packet containing a couple of integers, of the size defined for the polynomial representation. Since the points of the polynomials represent a cryptograhic key these values should be large enough to be considered secure (depending on the crypto algorithm used, keys starting from 5 bytes of length should suffice). If the sensors do not have processor support for large size integers, an 8 bytes value can be used as two standard 4 bytes integer and the same procedure can be applied using the combination of two polynomials, instead of a single one. Even if our comparison is not focused on any specific algorithm, in a centralized system the size of the frames will be carrying at least a hash function applied to some pre-shared key, and nonce or counter values, thus making the size of frames comparable if not larger than the ones used for polynomial authentication.

5. Conclusion and Future Developments

In this paper, we have presented an ongoing work to produce a lightweight bi-directional authentication protocol that can support mobile nodes in a wireless senor network. The aim of the protocol is to produce a bi-directional authentication, in order to avoid partition attacks and to support mobility. The protocol has been compared to a centralized authentication system, which is the common measure to produce bi-directional authentication and has shown good performances in terms of distribution of efforts between the nodes and of speed of authentication. The research on this field will continue to complete the authentication protocol, next important steps will be:

- The study of an algorithm to be added to the existing one, in order to set up a network from bootstrap. The simulations that have been performed up to now are targeted at an already bootstrapped network, the most simple solution seems to be to add an AS that is in possession of the whole polynomial definition, and can give to its neighbors a sufficient number of shares at start-up, before the authenticated nodes are more than K.
- implementation of the algorithm in a real testbed to verify performance with mobile nodes. This effort is ongoing, the right dimensions for the polynomials and the impact of delegation need to be carefully studied.

Acknowledgment

This work is partially supported by MIUR-FIRB Integrated System for Emergency (InSyEme) project under the grant RBIP063BPH.

References

1. Stallings W. *Cryptography and Network Security*: Principles and Practice (3rd Edition). Prentice Hall: Upper Saddle River, 2006.
2. Bird R, Gopal I, Herzberg A, *et al*. Systematic design of a family of attack-resistant authentication protocols. *Selected Areas in Communications, IEEE Journal* 1993; **11**(5): 679–693.
3. Delgosha F, Ayday E, Fekri F. MKPS: a multivariate polynomial scheme for symmetric key-establishment in distributed sensor networks. *Proceedings of the 2007 International Conference on Wireless Communications and Mobile Computing*, pp. 236–241, 2007.
4. Chan H, Perrig A, Song D. Random key predistribution schemes for sensor networks. *Proceedings of 2003 Symposium on Security and Privacy*, pp. 197–213, 2003.
5. Shamir A. How to share a secret. *ACM Communications* 1979; **22**(11): 612–613.
6. [Online]. Available: http://castalia.npc.nicta.com.au/index.php

DOI: 10.1002/sec