Two's Company, Three Is a Crowd: A Group-Admission Protocol for WSNs

Joao Girao and Miquel Martin*

NEC Europe Ltd.
Kurfuersten Anlage 36
69115 Heidelberg
Germany
{joao.girao,miquel.martin}@netlab.nec.de

Abstract. Once a wireless sensor network (WSN) is stable and has been running for a while, sensors start to fail due to hardware problems, battery exhaustion or even due to their physical destruction. In any case, the administrator of the network may wish to replace the damaged nodes with new ones to reinforce the coverage area. In this paper we make use of an out of band channel (OOB) to bootstrap an authenticated symmetric key. The protocol ensures that the new sensor nodes are currently part of the region covered by the network before negotiating sensitive key material and making them a part of the system and its operations. We describe a novel approach to group admission for wireless sensor networks using an OOB secure channel and perform a security evaluation over this protocol.

1 Introduction

Wireless Sensor Networks (WSNs) are considered by many to be a new hot research topic where the focus lies on solving the problems of routing, clustering, security, etc... with the minimum amount of processing and message transmissions. More than that, WSNs are different from other types of networks due to their unique traffic patterns, topology and restricted functionality.

Applications cover a wide scope, ranging from monitoring of environmental data (e.g. quality control in farming), accident prevention on the road, animal tracking, and even people in border controls, as well, as a number of military applications.

Some security protocols in WSNs make use of symmetric keys which are dynamically assigned [1]. These keys are usually agreed on during a bootstrap phase which may or may not be considered attacker free. The problem appears when, after the network has been stable for quite some time, the owner wishes to add

* The work presented in this paper was supported by the European Commission within the STReP UbiSec&Sens of the EU Framework Program 6 for Research and Development (IST-2004-2.4.3). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the UbiSec&Sens project (http://www.ist-ubisecsens.org) or the European Commission.

L. Buttyan, V. Gligor, and D. Westhoff (Eds.): ESAS 2006, LNCS 4357, pp. 70–82, 2006.

more nodes to the sensor network. Since most protocols assume some common knowledge, such as a key shared with the reader or a pool of keys distributed amongst the nodes, they cannot be extended, since most times this knowledge disappears from either the network or the owner side. Even in case the knowledge is still present, the cost of programming custom made sensors creates a scalability and cost issues.

For the class of sensors considered in this paper, we assume that radio transmission is two orders of magnitude more expensive than computations, in terms of power consumption. Therefore, message transmission is to be minimized, since it is the main reason for the network's limited life span.

We provide a simple and feasible mechanism with which nodes may be added to a sensor network by creating a common base of knowledge, using an Out of Band channel (OOB), which is used by a sensor to prove to its neighbors it is spatially part of the network and vice-versa. This protocol can be used with a number of different OOB channels and is flexible in its operation, allowing for its application in many different key distribution schemes.

Although there has been a number of papers and work related to security in WSNs, the problem of group admission, or rather, of adding new sensors to a pre-existing sensor network, has not been throughly addressed.

Our motivation stems from protocols such as [2], [1], [3], [4] and [5], where extending the network becomes complex and impractical.

In [1] and [3] the problem appears when adding new nodes: so that an attacker cannot retrieve information on keys which are not being used between the communicating nodes, but which might be used somewhere else in the network, these keys should be erased from the sensor's. After this, one cannot add more nodes and expect to get the deployment-time probability that two nearby nodes share a key. The case is similar in [4] and [5] where the master key should be erased from the sensors after the individual keys are agreed upon, or an attacker might obtain this key and break the system. Inherent to all these protocols, including [2], is that whatever method we choose, we must first re-program all the new nodes to contain information about the network. When buying new nodes where we simply want to extend small parts of the network, this process will become expensive and impractical.

Several other papers have looked at how OOB channels and human interaction can be used to enhance security protocols. In [6] the author describes the pairing problem where two nodes have contact for the first time and wish to exchange a strong secret having only the usual wireless channel and a very low bandwidth, authenticated channel. This problem statement also defines our scenario. In [7], the authors automate the OOB channel by using visual mechanisms and a camera in order to reduce the actual human interaction. [8] formalizes multi-channel protocols design and presents a number of variations on [9] and [7] with different security objectives.

Contribution: In this paper we propose a protocol using an Out of Band channel (OOB) that bootstraps a group key which is used amongst the new sensors

and the sensors which belong to the network to prove these are valid sensors and in the area covered by the OOB. This protocol can be used to extend or reinforce certain parts of the network by allowing other sensors to join in.

Organization: In the next section, section 2, we discuss the network and attacker models, after which we define clear security objectives we intend to cover by the protocol described in section 3. The following section 4 proposes different OOBs according to their applicability. The scheme itself is proposed in section 5 and the security analysis of the scheme in section 6. Finally, we end our contribution with a discussion on performance and our conclusions in sections 7 and 8.

2 Network and Attacker Models

In this section we will derive a network model consistent with the problem space and a threat model based on the characteristics of the network and application considered in this paper.

2.1 Network Model

We consider a network composed of sensors, $S_i \in \mathcal{S}$ and readers, $R_i \in \mathcal{R}$. Although we make no assumptions on the traffic patterns which occur between these entities, we assume it is possible to have bi-directional communication between the sensors.

We term neighbor of S_j ($N_i \in \mathcal{N}_{S_j}$) the sensor nodes within radio range of S_j and, since we consider the radio to be symmetrical, the sensor S_j is itself a neighbor of each of its N_i.

While the description above is true for the radio channel, we further extend this model to comprise an out of band channel, which is secure by nature. This channel has different characteristics from the radio channel in that it's unidirectional but can still be considered as a broadcast medium scoped in range. The sensors are always at the receiving end and the insertion entity, I, is the sender.

2.2 Attacker Model

Sensors are meant to be cheap and therefore may not comprise a tamper resistance unit. With this in mind, the sensors are subject to attacks which consist in capturing the actual sensor and reading its memory. Solutions which consist of using a unique key are therefore excluded due to the security risk of having a network wide key stored in easy to capture nodes.

There are a few solutions which deal with the key distribution problem in such networks. Most of them ([1], [3]) consider a pool of initial keys which are used to find a common key with the neighbors or a common key ([4], [5]), which should be erased once the bootstrapping phase finishes. In the first case it is required to either program the pool of keys in the new sensors or even impossible to find

the common ground on which to build a security association. Programming a pool of keys in new sensors may also prove unfeasible for most commercial applications since it adds complexity on the side of the buyer and the storage of the initial pool of keys.

Our attacker wishes to add sensors of his own in the sensor network. His motivation is either to provide false readings, eavesdrop or simply discover the keys used by the neighbors' sensors. He has physical access to the network location and, unless supervised, may interfere with any network protocol by either using the radio channel or physical means.

It is not our aim to consider denial of service (DoS) attacks, although some importance is given on how to protect the OOB channel from such attacks.

3 Security Objectives

There are two main security objectives to be fulfilled by this protocol:

1. Provide a mechanism for the nodes in a network to recognize new sensor nodes as valid, in the absence of a pre-shared secret or trust relationship, but using a seed provided by an external actor over a secure channel.
2. Design a key agreement protocol that adds the new nodes into the network, by bootstrapping the network key with the authentication information received in the previous step.

4 Out of Band Channel

In communications, an Out Of Band Channel refers to a separate, dedicated channel, different from that used in normal transmissions. In the scope of WSN's, we consider the normal traffic of the network to be "in-band", and propose an external channel to transmit a key that bootstraps the in-band channel security.

The bootstrap key is transmitted over the OOB channel to both the nodes we want to add, and the ones already in the network. Because these two groups of nodes have not yet established any security relation the key must be sent in the clear.

For this reason, the OOB channel must have a reduced scope that ensures only the intended nodes can receive it. We achieve this by choosing an OOB channel which is geographically confined by nature, and therefore assume it to be secure. Some possible examples of such an OOB channel follow:

- *Light beam:* A device like a flash light is turned on and off intermittently. The nodes apply a preset sampling to their light sensors and extract a binary sequence from the light intensity. The scope is limited to the flash light beam spot. Figure 2 illustrates this scenario.
- *Buzzer:* A sound emitting device with a clearly defined output spectrum broadcasts a series of short tones. The nodes use a microphone and a pass-band filter to extract binary sequences from the sound (in the case of an on/off buzzer) or longer sequences if the buzzer emits multiple tones. The scope is limited to the hearing range of the listening device.

- *Local measurement:* The nodes use a predefined function to extract a key from their measurements. For instance, temperature sensors use the range of their reading (e.g. between 30 and 33 degrees) to infer the secret used to bootstrap the key.
- *Vibration measurement:* The old and new nodes are held in a container and shook together (possible shook in the hand). The resulting vibration is interpreted as a key, as explained in [10].

All of these channels use one or more sensors already present in the sensor node to read bit stream from their environment. Because of power saving and security concerns, listening for the OOB message and renegotiating the network keys should not be done constantly. In our approach, nodes already present in the network receive a message from the sink, which triggers the OOB channel monitoring and renegotiation. New nodes are only activated when we are prepared to deliver the OOB message. Section 4.1 illustrates a practical situation where sensors are added using a flashlight as OOB.

Certain applications might have more stringent security requirements, which render our OOB security insufficient; In a possible attack scenario, the attacker plants nodes next to the already existing ones. As we add new sensors and initiate the key bootstrap, the infiltrated sensors have access to the OOB message and can potentially become part of the network. In such cases, the delivery of the OOB message should be done using methods are specific to the channels nature. The easiest way would require a container which confines the channel, since only those nodes which we choose to put into the container could see the message. In the flash light example, one could pick up some of old nodes and hold them in the hand together with the new ones, or inside a dark bag, together with the new ones; this way, the OOB message would only reach our new nodes and our trusted hand picked old nodes).

In the container confined method, the new nodes are assumed to be trusted, but one must carefully choose the old nodes. If any old node is a disguised attacker node, it would gain access to the network. For this reason, we recommend delivering the OOB message in a container that holds any number of new nodes, and only one old node. If the old node already belonged to the network, the key bootstrap will succeed. If, on the other hand, it was an attacker node, the new sensors will never access the WSN, since the attacker node does not have access to it either, and thus can not act as a bridge. When using a single old node, all new nodes must initially communicate through it for the purposes of bootstrapping the key, but once the nodes are securely in the network, further keys can be negotiated, eliminating the single point of failure.

Finally, it is important that the rate at which sensors sample the OOB channel is comparable to the frequency of the changes in the OOB medium. In the flashlight example, sampling frequency should be comparable to the switch-flick rate of the flashlight, and the duration of the message would be given by this rate and the key size.

4.1 Example Secure OOB Channel

Let us analyze the specific case of the light beam. The network owner has decided to increase node density in a given area, and so, purchases a number of blank sensors. Without any further pre-configuration, he activates the nodes and scatters them in the desired area, as seen in Fig. 1. Next, he sends a message through the sink, requesting the old nodes to monitor the OOB channel. He can now transmit the key using a normal flashlight: flicking the switch on and off at random intervals generates variations in the light intensity perceived by the nodes, which is, in turn transformed into a binary sequence, which will be used as bootstrap key (see Fig. 2). Since the owner can visually verify whether someone is watching and whether an attacker is interfering with the process, the channel can be said to be protected against eavesdroppers and man-in-the-middle attacks. It is therefore private and provides message integrity.

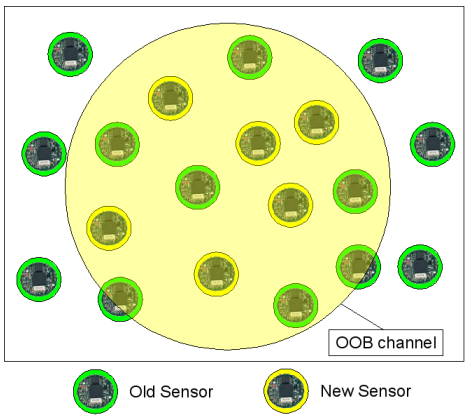

Fig. 1. Sample network and OOB channel coverage

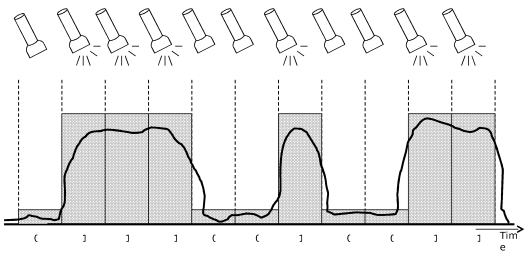

Fig. 2. Shared secret over a secure out of band channel using a flash light

5 The Scheme

Once the OOB channel has been used to securely establish a shared secret, the protocol ensures that the secret is known by the several entities and makes use of this small short-lived key to bootstrap the key agreement protocol. We make use of a combination and variation of two well known protocols: MANA [9] and SPEKE [11].

In the following protocols we consider the interaction between Alice (A) and Bob (B) and then extrapolate for the case of n sensors. For the examples considered, the process is the same whether the sensor is new or was already part of the network so any sensor may either take the role of Alice or Bob. In case the sensors already know each other, the protocol is unnecessary.

5.1 The Toolbox

MA-3. The Manual Authentication Protocol (MANA) [9] allows two devices to pair by allowing a user to input a shared password in both devices, as seen in Fig. 3.

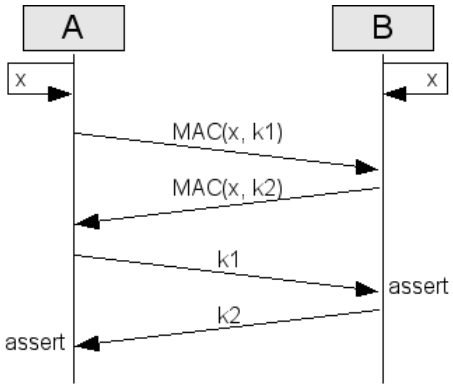

Fig. 3. MA-3

The shared input is used to generate two commitments, one by Alice and the other by Bob, on two pseudo-random numbers, $\{k_1, k_2\} \in \mathbb{Z}$. First the commitments, in this case $\mathrm{MAC}(x, k_1)$ for Alice and $\mathrm{MAC}(x, k_2)$ for Bob, are exchanged. Once both parties have received the commitment they can open the commitment by sending k_1 and k_2 respectively. Since x is never sent over the wire, it can be used to confirm the commitment and the short-lived key and one-time nature of the protocol ensures its security even with small $|x|$. The typical size for the shared secret x, recommended by the authors of MANA, is on the order of 20 bits.

SPEKE. Simple Password Authenticated Exponential Key Exchange (SPEKE) [11], and also depicted in Fig. 4, is a key agreement protocol which describes a

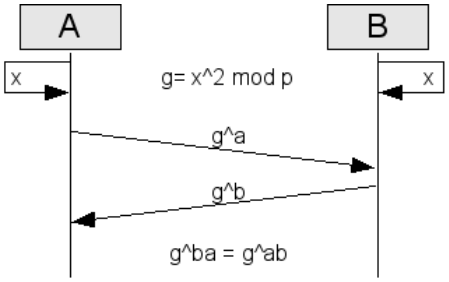

Fig. 4. SPEKE

way to use a shared secret to bootstrap an authenticated Diffie-Hellman (DH) [12] key exchange.

Let x be a member of $\mathbb{Z}$ and $\mathbb{Z}_p$ a multiplicative group where p is prime, then $g = x^2 \mod p$ is a generator for a subgroup of the multiplicative group.

We then use g and $\mathbb{Z}_p$ as the parameters for the DH key exchange such as g^a is the public parameter for Alice, g^b the public parameter for Bob, with both pseudo-random numbers $\{a, b\} \in \mathbb{Z}$, and $s = g^{ba} = g^{ab}$ the shared secret resulting from the exchange.

Since g is secret, x acts as a shared secret which bootstraps an authenticated DH. Please note that, contrary to the previous scheme, x musn't be a small value. It should be in the range of 80 bits.

5.2 Our Contribution

SPEKE with ECDH. This variation of the SPEKE protocol simply makes use of an Elliptic Curve Diffie-Hellman (ECDH) [13] since our main concern is the size of the operands. In this case, the secret is not used to determine a generator of the group but rather used a secret multiplier of the DH agreed key. The secret itself is never transmitted on the wire. An illustration of the scheme can be seen in Fig. 5.

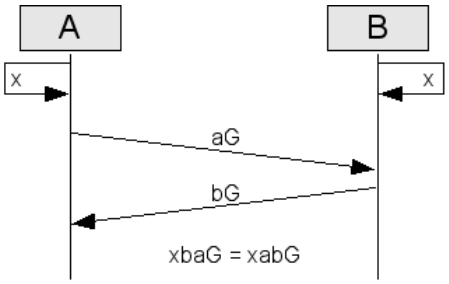

Fig. 5. EC-SPEKE

Let G be a generator in the elliptic curve E with domain parameters T known to all parties.

As in the usual EC-DH, both parties exchange their public parameters aG, for Alice, and bG, for Bob, with a and b pseudo-random numbers in $\mathbb{Z}$. The resulting secret becomes the combination of the previously shared secret $x \in \mathbb{Z}$ with the EC-DH exchange such that $S = xbaG = xabG$ is the agreed key known only by Alice and Bob.

Group Admission and Shared Secret Agreement. This protocol focuses on the communication between the introducer, I, the old, S, and the new, S^*, sensor nodes. I acts like the trusted party to both sides which bootstraps the security association between old and new nodes.

As a first step, I, S and S^*, agree on a common OOB channel. The characteristics of this channel must conform to the ones proposed in section 4, so that we may consider the channel secure. Once the channel is established, I distributes a shared secret to both S and S^* sensors, simultaneously. This small shared secret, x, will be the basis for the steps that follow.

S^* and S sensors will broadcast a commitment to a k_sG value, where $k_s \in \mathbb{Z}$ is random and G a generator for a previously agreed, and secure in terms of ECDLP, elliptic curve E. The resulting value is a random point in E. Finally, the commitment which is broadcasted can be calculated as $\text{MAC}(x, k_sG)$.

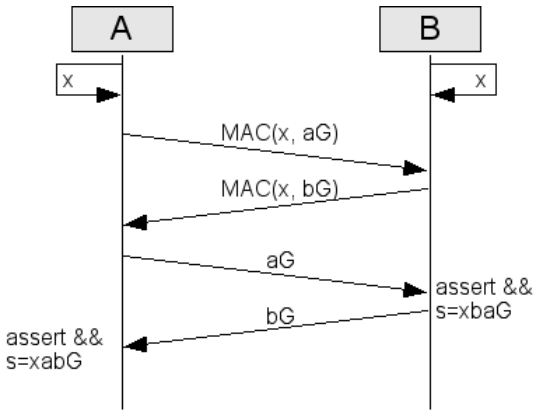

Fig. 6. Group Admission with Key Agreement

Once all commitments have been sent[1], all sensor nodes will open their commitments by sending their values k_iG. When these values are received, the commitment is confirmed and, if valid, an entry for that particular sensor node is created in memory. The entry will contain the id of the sensor and the authenticated shared key $s = xk_sk_iG$.

[1] This can be achieved by simply setting a timer and ignoring further commitments once a node's commitment has been opened.

The aforementioned scheme, depicted in Fig. 6, ensures that the shared secret s is authenticated via the trusted third party I. Please note that the algorithm is symmetric for S and S^* and in principle either can commit to their value first.

Algorithm 1. Scalable Algorithm for Group Admission with Key Agreement

1: S: Receive $x \in \mathbb{Z}$ from OOB channel.
2: $S \rightarrow N_i$: MAC(x, k_sG)
3: $N_i \rightarrow S$: MAC(x, k_iG)
4: $S \rightarrow N_i$: k_sG
5: **for all** $i \in \{N_1, N_2, \cdots, N_n\}$ **do**
6: **if** don't know N_i yet **then**
7: $N_i \rightarrow S$: k_iG
8: Assert MAC(x, k_iG)
9: Store, for N_i, xk_sk_iG
10: **end if**
11: **end for**

Algorithm 1 uses the protocol defined above, and depicted in Fig. 6, in an environment with several sensors.

6 Security Analysis

In this section we briefly summarize the security of the protocols on which our proposal is based and then show that the combination of these protocols does not hinder the security level of the scheme. This is not a security proof but rather work in the direction of the proof.

The security of the DH key agreement protocol is based on the discrete log problem. While exponentiation is still considered as a one-way function, or in the case of ECDH the difficulty of inverting a point multiplication, the strength of the agreed key can be correlated to the underlying primitive. The choice of the parameterization of the group and selection of the generator still play important roles in the overall security of the system since the discrete log problem does not apply equally to all groups and the selection of the generator might reveal partial information. All these issues are addressed in [12] and [13].

Since we are using ECDH, it is also important to note that this is a relatively new area and, although we can consider the ECDLP to hold, it might be this is proven solvable in the future. The security level of the resulting key is dependant on this. The DLP in general, and also the ECDLP, are mathematical NP-hard problems and therefore we can achieve *provable security*.

SPEKE is also provable secure since it is based on the same mathematical assumptions as the DH protocol and we can further infer to the security level of EC-SPEKE which is based on ECDH.

The MANA protocol, as also described in [14], is computationally secure. The one-time use of the commitment and key link the attack to a very short time window which allows for short keys. Furthermore, the fact the commitment is

only open once both sides have committed to their values disallows a man-in-the-middle attack. Since we use the MANA protocol in the same way, and if we consider that the result of the first point multiplication in the ECDH is a pseudo-random value which falls under the random oracle model (in the sense it cannot be predicted), then the authentication part of our protocol should supply the same security properties as the MANA protocol.

We can therefore divide the key agreement component of our protocol, which should be provable secure, and the authentication part of the protocol, which should be computationally secure. The overall security of this protocol takes the shape of the weakest of its components and, our protocol, should be computationally secure.

It is important to note that no assumptions can be made about this protocol until a formal security analysis of the protocol, which instantiates the line of thought provided in this section, is performed.

Note: Once the commitment is opened, an attacker can easily brute-force the value of the secret key x, since we assume it's size is around 20 bits[2]. However, the importance on the security of this value is limited in time since we only require that it is kept secret from the time the commitments are sent to the point at which they are revealed. Once this phase has passed, the secrecy of x is no longer required. x is added to the agreed key only as a way to link the weak authentication key with the final key. In the case where the protocol is extended to multiple parties, all parties should commit to their values prior to the first open message. This will ensure the security of the protocol still holds true.

7 Performance and Discussion

In [15] the authors provide promising results on the calculation of point multiplication in EC with the same micro-controller as that used in most commercial sensors. Although these results are not acceptable for continuous use in the network, they are quite reasonable for one-time use both in terms of computation intensity, power consumption and time, making this a viable solution.

Also in terms of bandwidth the scheme fits the WSN scenario. The transmission size of the commitments can be as short as 8 bytes, using UMAC-64 [16], with a reasonable security level. The transmission of the opening of the commitments is of one point. If we assume a 163 bit ($\approx$ 20 bytes) generator point to use as base for the ECDH, we would need 21 bytes (which fit 164 bits) to transmit the point. In the overall, both these values fit in one packet of all packet formats so far proposed for sensor networks (so far the lower bound has been a previous proprietary TinyOS [17] Medium Access Control (MAC) protocol with 29 bytes payload).

7.1 Discussion

During the OOB example we suggested that the channel be applied directly to the network. Although this is possible and one way to perform the protocol,

[2] It would take an attacker on average 2^{19} tries to obtain the value of x, which is perfectly feasible.

it is not very secure. To minimize the risk of an attacker introducing nodes in the network before the procedure, as a means to authenticate his nodes at the same time as new nodes are added, we suggest that the introducer I captures a small number of already authenticated nodes, which he chooses one by one, and performs the procedure using those nodes and the new ones in a controlled environment. In more practical terms, should the number of sensors be small, we can even foresee that I simply picks up a sensor from the same area that he wishes to replenish and puts all the sensors, new and old, on his hand, where he performs the procedure.

8 Conclusion

In this paper we present a mechanism to extend a sensor network by using an OOB channel to convey a short secret which is then used to authenticate a key agreement protocol. We show how this scheme is theoretically secure and feasible for implementation under the restrictions of the sensor nodes.

We believe this protocol to be generic enough to be applied with a number of encryption and authentication protocols.

References

1. Pietro, R.D., Mancini, L., Mei, A.: Random key-assignment for secure wireless sensor networks. In: 1st ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'03). (2003) 62–71
2. Perrig, A., Szewczyk, R., Wen, V., Culler, D., Tygar, J.D.: SPINS: Security protocols for sensor networks. Wireless Networks **8** (2002) 521–534
3. Chan, H., Perrig, A., Song, D.: Random key predistribution schemes for sensor networks. In: IEEE Symposium on Security and Privacy. (2003)
4. Zhu, S., Setia, S., Jajodia, S.: Leap: efficient security mechanisms for large-scale distributed sensor networks. In: CCS '03: Proceedings of the 10th ACM conference on Computer and communications security, New York, NY, USA, ACM Press (2003) 62–72
5. Lai, B., Kim, S., Verbauwhede, I.: Scalable session key construction protocol for wireless sensor networks (2002)
6. Hoepman, J.: The ephemeral pairing problem. In: 8th Int. Conf. Finantial Cryptography, Key West, FL, USA (2004)
7. McCune, J.M., Perrig, A., Reiter, M.K.: Seeing-is-believing: Using camera phones for human-verifiable authentication. In: SP '05: Proceedings of the 2005 IEEE Symposium on Security and Privacy, Washington, DC, USA, IEEE Computer Society (2005) 110–124
8. Wong, F.L., Stajano, F.: Multi-channel protocols. In: Proceedings of Security Protocols Workshop, LNCS (2005)
9. Gehrmann, C., Mitchell, C.J., Nyberg, K.: Manual authentication for wireless devices. Cryptobytes **7** (2004) 29–37
10. Holmquist, L., Friedemann, M., Schiele, B., Alahuhta, P., Beigl, M., Gellersen, H.: Smart-its friends: A technique for users to easily establish connections between smart artefacts. Lecture Notes in Computer Science **2201** (2001) 116

11. Jablon, D.: Strong password-only authenticated key exchange. Computer Communication Review, ACM SIGCOMM **26** (1996) 5–26
12. Diffie, W., Hellman, M.E.: New directions in cryptography. IEEE Transactions on Information Theory **IT-22** (1976) 644–654
13. Research, C.: Standards for efficient cryptography, SEC 1: Elliptic curve cryptography (2000) Version 1.0.
14. Laur, S., Asokan, N., Nyberg, K.: Efficient mutual data authentication using manually authenticated strings. Research Report in the IACR ePrint archive (2005) http://eprint.iacr.org/2005/424.
15. Gura, N., Patel, A., Wander, A., Eberle, H., Shantz, S.: Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs. Cryptographic Hardware and Embedded Systems (CHES) (2004) 119–132
16. Black, J., Halevi, S., Krawczyk, H., Krovetz, T., , Rogaway, P.: Umac: Fast and secure message authentication. In: Advances in Cryptology - CRYPTO '99. Lecture Notes in Computer Science. Volume 1666. (1999) 216–233
17. Hill, J., Levis, P., Madden, S., Woo, A., Polastre, J., Whitehouse, C., Szewczyk, R., Sharp, C., Gay, D., Welsh, M., Culler, D., Brewer, E.: TinyOS: http://www.tinyos.net (2005)