SECURITY DESIGN SCHEME FOR USER AUTHENTICATION ON WIRELESS SENSOR NETWORKS[1]

***Deepak Choudhary, #Prof. Rakesh Kumar, $Neeru Gupta**

**Ph.D Research Scholar, CMJ University, Shillong*

#Director, K.P. Jain Engg. College, Ghaziabad

$ CoD, CSE Deptt. In Manav Bharti University, Solan

ABSTRACT

In this paper, we consider User Authentication (UA) for wireless sensor networks. UA is a fundamental issue in designing dependable and secure systems. Imagine that a wireless sensor network is deployed in an intelligent building, a hospital, or even a university campus, to allow legitimate users to send queries and retrieve the respective result at any of the sensor nodes. Importantly, the system needs to provide a means of user authentication to verify if the user is valid. We propose a dynamic strong-password based solution to this access control problem and adapt it into a wireless sensor network environment. The proposed strong-password authentication approach imposes very light computational load and requires simple operations, such as one-way hash function and exclusive-OR operations. We present the design of the proposed scheme and discuss how to make use of the security features on MAC sub-layer (Medium Access Control) based on the IEEE 802.15.4 specification. Analysis on security and communication costs is presented to evaluate the effectiveness of the proposed scheme.

Keywords: *User Authentication; Sensor Networks; Security; IEEE 802.15.4 Specification*

INTRODUCTION

Wireless Sensor Networks (WSNs) are developed to collect data about the monitored environment over a geographic area [1]. The data will be sent and presented, probably after some processing, to users either in an ad-hoc queries manner or upon event detection. Many different kinds of WSN applications could be proposed due to the ubiquitous nature of WSN and its easy deployment. This means environmental data will be available almost in everywhere in near future. For example, using a WSN for an intelligent building, the current temperature, humidity in a particular location area will be available on demand. In general, most of queries in WSN applications are issued at the points of base stations or at the backend of the application systems. However, we could foresee that there should have great needs to access the real-time data inside WSNs. Therefore, real-time data may no longer be accessed at the base station or the gateway node only, rather, they could be accessed anywhere from a sensor node in a WSN in an ad-hoc manner. In general, the collected data may not be so critical, such as the query of the current temperature in a location within a building. However, for some applications, the data collected is valuable and confidential. Security measures should be provided to protect the access to these critical data as well as to restrict non-authorized users from

[1] How to cite the article:
Choudhary D., Kumar R., Gupta N., Security Design Scheme for User Authentication on Wireless Sensor Networks, *International Journal of Advances in Engineering Research*, March 2012, Vol 3, Issue 3, 58-67

gaining the access the data. Access control is a classical problem in many existing computer systems and applications. Normally user authentication (UA)is used as a basic solution to safeguard the access control issue.

Many examples of UA measures can be found in our daily life, such as login to our office's local area network, mobile phone's device authentication, down to a password-based authentication for our account transactions on banks' ATM machines; and the like. Unfortunately, a review of current studies on WSN reveals that user authentication has not been adequately addressed, although many researchers have been reported on WSN security issues. This may be due to the resource- constrained nature of WSNs, where computation, storage, and battery power are quite limited on each sensor node. Given the resource constraints, it is difficult to apply traditional UA solutions in WSNs.

In this paper, we study the UA problem in the context of a WSN where legitimate user is allowed to query and collect the data at any sensor node of the network. We propose a UA solution based on the strong-password authentication approach [4] which requires much less computation and thus is feasible to be adapted into the WSN environment. The remaining part of this paper is organized as follows. In Section 2, we briefly review the related work on UA. In Section 3, we propose our system model and describe the design of the proposed UA protocol. An analytical evaluation with recommendation on IEEE 802.15.4 security features are presented in Section 4 and 5 respectively. Finally, we conclude the paper in Section6.

RELATED WORK

There has not been much work published on user authentication schemes in WSNs. However, it is quite interesting to examine various works on smart cards-based UA schemes for mobile communications or remote networking environments. Some properties, types of attacks, and protocol handshakes from these works could serve as a good framework for developing UA solutions for the WSN environment.

Smart Card-based UA Schemes

Solutions for remote password authentication with smart cards have been firstly proposed [2]. A smart card is physically issued to the user who first registers to a system. Each user possesses a smart card for later login and authentication. A smart card is an IC processor which can efficiently perform computational operations. For examples, it can perform a one-way hash function; generate a random vector or signature. In fact, a tiny sensor node could perform parts of these operations; although its computational power is not as much as the smart card. Therefore, examining the smart card-based UA scheme will help design a UA scheme for WSNs.

A number of UA schemes using smart cards can be found in [3, 4, 5, 6, and 7]. The scheme [6] is based on El Gamal cryptosystem [16], which belongs to a public key cryptosystem and a signature scheme based on discrete logarithms. However, the scheme can be broken by creating a valid pair of (userID, PassWord) without knowing the secret key of the system. Thus, a legitimate user could compute some other's password. To address this problem, a modified UA scheme was introduced

[18] to prevent the forgery attack. Also, an enhanced smart-card based remote UA scheme with check digits was introduced to remove the threats of impersonating other legal users. Detailed descriptions of these works can be found in [3].

Dynamic ID-based Remote UA Schemes

Password-based authentication schemes are the most widely used methods for remote UA [4]. Existing schemes could be categorized into two types. One uses weak-password approach, while the other uses strong password approach. The weak-password authentication approach is based on El Gamal cryptosystem. The advantage of this scheme is that the remote system does not need to keep a user ID- password table to verify the validity of the user login. However, such a weak-password authentication approach leads to heavy computational load on the whole system. Thus, this scheme cannot be applied to a WSN environment, as remote sensor nodes cannot afford to do this heavy computation. Unlike the weak-password approach, strong password authentication is mostly based on one-way hash function [17] and exclusive-OR operations (XOR). It requires much less computation and needs only simple operations. With this in mind, this scheme may have advantages when it is applied to a WSN environment. Das et al. proposed a dynamic ID-based scheme [4], which is based on the strong-password authentication approach. The scheme allows the users to choose and change their userIDs and passwords freely. The system has no need to assign a password to a particular userID. This feature will be incorporated into our proposed UA scheme for wireless sensor networks as well. The algorithms in [4] are claimed to be secure against ID theft; and able to resist the replay and forgery attacks, as well as insider attacks. However, some of the algorithms were proved by Awasthi [5] to have loop-holes in the process of password verification. These loopholes are already enough to make the whole system insecure, as an intruder is able to use any random password to get into the system.

UA Scheme for Mobile Communication Environment

El-Fishway et al. [8] proposed an effective authentication scheme for mobile users. This scheme could even be applied to the existing GSM mobile networks. It is assumed that there is no central certification authority and there is no trust between the communicating entities. The protocol also makes use of public keys, secret keys, and one-way hash function concepts. One of the merits for this protocol is that it never allows the user's secret (i.e., password or secret key) to be transmitted out of the user's home domain with respect to the mobile network infrastructure. In our proposed scheme, the user's password is never passed out to the sensor networks too. One of the drawbacks of this scheme is that it introduces an extra communication flow between the user's home domain and remote domain; to make it have four handshake flows in the protocol. Therefore, the communication overhead might increase as well.

Most of the existing UA schemes require high computation cost caused by exponentiation operations; and not suitable for mobile devices (e.g., PDAs, mobile phones, sensor nodes etc.). Lee et al. [7] also proposed an improved UA scheme with low computation cost by using smart cards and one-way hash functions. Only three phases are used in this scheme, namely, Registration Phase, Login Phase, and Authentication Phase. This scheme can resolve the attacks of forgery, replay, and modified login

message. Our proposed solution in Section 3 makes use of Lee's framework having three phases as above; but adapts it for a wireless sensor network environment.

UA Scheme in Sensor Networks

Very few works on UA in WSN can be found. Benenson et al. [9] proposed a scheme against sensor node capture attacks. The protocol is based on Elliptic Curve Cryptography (ECC) [10], The idea of this scheme is using the Public Key Infrastructure (PKI) approach, Base Station acts as a central Certificate Authority (CA), i.e., CA (priv keyCA, pub_keyCA). A legitimate User's certificate (U) is signed by the CA with user's public key, certU= signCA (pub_keyU). The scheme requires more overhead for encryption and signature verification than decryption and signing. The authors claimed that ECC is still feasible for sensor nodes. However, it could possibly become a bottle neck for sensor nodes to perform the verification process during a high traffic load of the whole network. The notion of n-authentication is introduced in Benenson et al. [11], which means that the whole authentication succeeds if the user can successfully authenticate with any subset of sensors out of a set of n sensors. n-could be the average number of the sensors within a unit broadcast distance of a particular sensor or the user. The protocol works in the following manner. A user tries to send his/her ID and certificate to a group of n sensor nodes. Each sensor node will send back the user a challenge, i.e., Msg (sensor-id, nonce); and the user needs to respond all the challenges from the set of sensor nodes. The response from user to a Sensor(i) is: sighU(hash(U, Sensor(i), noncei)). Now each sensor node will verify the user's reply of response in the following:

Sensor(i): verify(cert U) := pub_keyU

Sensor(i): verify(SignU (hash(U, Sensor(i), noncei)))

If the user is successfully authenticated for one sensor node, this sensor node will then broadcast a Yes-vote to other nodes in the group within the broadcast unit. If the user is not successfully authenticated, this node sends nothing out, other nodes wait for the timeout. Each sensor node in the group (out of n nodes) collects the Yes-votes. Recall that n-authentication was designed to against the number (t) of sensor nodes that can be compromised. The value of t was suggested by the authors to be less than n/2. Therefore, the protocol will terminate if either (n-t) Yes-votes are collected, i.e., successful authentication; or (t+1) or more Yes-votes fail to be received before the sensor times out, i.e., unsuccessful authentication. Some weaknesses were pointed out that an adversary might have a bogus certificate and a bogus signature in sending the challenge-response. There was still a chance of having an impersonation attack. Also, Denial-of-Service (DoS) attacks could occur by sending either many bogus certificates to make sensor nodes' memory exhausted, or bogus signatures to make sensor nodes running out of energy in verifying them.

Phase 1 - Registration

Step 1.	A registration interface is launched on a user's mobile device, and a User submits his/her ID (*userID*) and a chosen password *PW*.
Step 2.	Sensor Gateway-node (GW) for registration with its private key or shared secret key (i.e. key) computes:
2.1	A = hash(userID ‖ key); //output 512-bit (e.g. use SHA-512 as the one-way hash function
2.2	B = hash(A ‖ hash(PW))
Step 3.	The GW-node replies to the user for successful registration.
Step 4.	GW-node then passes the dataset of (userID, PW, A, B, TS) in clear texts to the GW-PC's database engine. *TS* represents the Timestamp that the Gateway recorded before when a user was doing the registration. String A and B are the outputted 512-bit hex-strings of hash operations and are used to cover the contexts of userID and PW respectively. A subset (userID, A, TS) is then distributed over the sensor network in encrypted mode during transmission. This dataset(userID, A, TS) is assumed to be stored on a particular set of sensor nodes, which are able to provide a login interface to users' devices in order to perform the login service at later time

Figure 1. Steps of operations for phase 1

THE PROPOSED USER AUTHENTICATION SCHEME

A wireless sensor network is deployed in a confined area, which is divided into different zones. Authorized users can access the WSN somewhere in the network using mobile devices, say a Notebook PC. The mobile device is assumed to have the ability to communicate with the sensor nodes within the WSN (e.g., through an embedded sensor node). Before issuing any queries into the system, a user must register with a name and a password, probably at the sensor gateway (GW) node. Upon successful registration, the user can submit a query to the sensor network system at any time within a predefined or administrative configurable period. This configurable time period could be set differently depending on the nature of applications. During a particular querying process, the user has to remain in place, login to a nearest sensor login-node in a zone, issue the queries and get back the result. Once the predefined time period has expired, the user may need to restart a new cycle by doing the registration again if he/she foresees that more queries need to be performed.

Protocol Description

The proposed scheme is divided into three phases: the Registration phase, the Login phase, and the Authentication phase. The operations of the three phases are described below. Assume a sensor node has already installed the registration and login interface. The sensor node is then attached/embedded to a user's own mobile device; say a PDA or a Notebook PC. A sensor gateway-node connecting to a PC server (here collectively called GW-node) is also assumed to be connected to the WSN. The steps to be performed in this phase are illustrated in Figure1.

Phase 2 - Login

If the User wants to do some queries of sensory information, he/she needs to login to a dedicated sensor login-node. The user submits the userID* and password PW*. Note that this login-node already has the list of datasets (userID, A, TS) if this record has not yet been expired. The steps to be performed in this phase are illustrated in Figure 2.

Phase 3 - Authentication

Now, the registration GW-node has received an input of (userID*, C2, C1, T). The following steps of operations are listed in Figure 3.

```
Step 1.  At GW-node:
    1.1 IF userID* exists on its table list
        THEN retrieve parameters of dataset(userID,A,B);
    1.2 ELSE send Msg(REJ_LOGIN) to login-node for
        rejecting the login;
Step 2.  Verify timestamp T with current time T* if it is
        within the expected time interval for transmission
        delay;
    2.1 IF ((T* - T) >= delta_T)
        THEN respond Msg(REJ_LOGIN) to login-node;
    2.2 Compute C2* = (B XOR A);
    2.3 Compute C1* = hash(B XOR T);
    2.4 IF (C2* != C2) or ( C1* != C1)
        THEN respond Msg(REJ_LOGIN);
    2.5 ELSE send Msg(ACC_LOGIN) to login-node for
        accepting the login;
        // If C2*= C2, it means B* has been verified and
        then PW* is also verified; if C1*= C1, it means the
        timestamp T has been verified
Step 3.  Login-node sends login result back to User.
```

Figure 3. Steps of operations for phase 3

Note, there is one-hop communication between a user's mobile device and the sensor GW-node during the registration. Also, there is only one-hop communication between the device and the sensor login-node, since the user will go to the nearest login node area when performing this login and subsequent queries. For the communication scenario between the login-node and the GW-node, multiple hops may be required. The overall handshake of the proposed protocol for user authentication is illustrated in Figure 4.

ANALYTICAL EVALUATION

In this section, we present the analysis of security features of the proposed protocol and the comparison of the cost overhead.

Step 1. User submits (userID*, PW*) to login-node.
Step 2. **IF** userID* is contained in the login-node's list of dataset(userID, A, TS)
THEN the corresponding A is known
2.1 **ELSE** send Msg(REJ_LOGIN) to User;
2.2 Compute B* = hash(A ‖ hash(PW*))
2.3 Compute C2 = (B* XOR A)
// use C2 to seal the content of B*
2.4 Compute C1 = hash(T XOR B*)
// T is the current timestamp at that moment
Step 3. Login-node sends Msg(userID*, C2, C1, T) to the registration GW-node for final authentication process

Figure 2. Steps of operations for phase 2

Security Analysis

Security Scenario Cases:

Assume that the registration process in Phase 1 is carried out in a secure mode. For example, the registration place can be in an area where only persons carrying a staff card are allowed to enter. This will minimize the possibility of eavesdropping over the air when users are doing the registration. The following cases can be identified by the system.

1) Valid user ID, fake password PW?
The system may identify it in Authentication state, at step 1 in Fig. 4, check C1 value. (Where C1 indirectly covers B*, and B* indirectly covers PW*).
2) Invalid userID, valid/fake PW? The system may identify it quickly in Login state, at step 2 inFig.4
3) Replay login-message attack without packet modification, i.e., reuse the packet of Msg (userID* C2, C1, T)? his could be identified by delta_T in Authentication state at step 1 in Fig. 4.

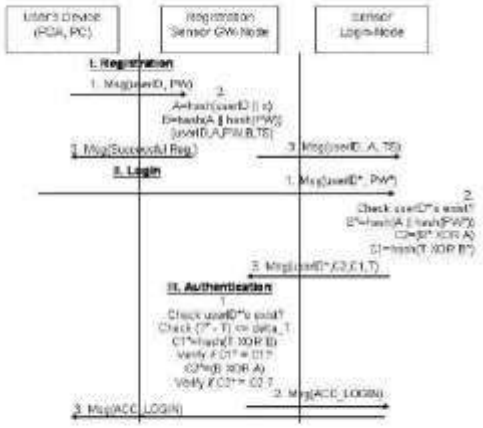

Figure 4. Communication handshakes for the three phases

4) Replay login-message attack with modification of value T, i.e., Msg (userID*, C2, C1, Te)? Te is modified to current date and time. This replay could be identified by either IEEE 802.15.4 enabled security mode of AES-CCM-128 (which stands for Advanced Encryption Standard, combined encryption and authentication) at MAC level [12]; or C1* in Authentication state at step 1 in Fig. 4 (if no security mode is enabled); since C1* will not equal to C1 (even T is modified as Te, C1 value in fact directly protects the value of (T).
5) Replay login-message attack with modifications of T and C1, i.e., Msg (userID*, C2, C1e, Te)?In this case, an intruder has to be recomputed a correct value of C1e=hash (TeXORB*);

However, the intruder does not have the value of B* on hand That means when the GW-node tries to verify C1e, it will not match with the value of C1. Therefore, the replay login-message could also be identified during Authentication process at step 1 in Fig.4 The attacks of login-message replay and login message forgery can be protected from the above scenarios 3, 4 and 5. However, this proposed UA scheme has a weakness for protecting from insider attacks, which is a very difficult to handle. *'Insider*

Attack'.

Apparently, legitimate users are assumed to be honest and they will not disclose his/her userID and password to his/her colleagues for using. However, if there is a user breaching this honesty, intruders can still be able to access the WSN. The only backdoor that a system administrator needs to check up with is the history log of user query. From that log, the same the same or overlapping querying times occurred. This legitimate user could be then put into a black list. However, this kind of checking has no guarantee to find out if the times for multiple queries are in different. In addition, this kind of checking is very time consuming and laboring in reality, this insider attack perhaps could be minimized by enforcing the security policies imposed to all authorized users, such that they are highly responsible of keeping his/her userID and password securely. Furthermore, the duration of registration cycle could be administrative adjusted, since this will affect the effective timestamp (TS) parameter. When the next registration comes, the (userID, PW) pair could be enforced to be different for the same user. This will prevent the user from using the same value of (userID, PW) as a habit. In doing so, this may also help preventing the intruders or attackers attempt to reuse the old value of (userID,PW).

Cost Overhead Comparisons

In this section, we use the computational overhead (the computation time required by sensor nodes, denoted by T) and communication cost (denoted by C) as the metrics to evaluate the performance of the proposed protocol. Some notations are further defined as follows:

T H : the time for performing a one-way hash function hash().
TXOR : the time for performing an XOR operation.
T EXP : the time for performing a modular exponential computation.
CMH : the delay time for the communication taken place between the login-node and the GW-node in multi hops.

Table 1 shows the overall cost of the proposed dynamic UA scheme. The total cost overhead is the sum of computation and communication costs for all the three phases. For comparison, the cost overhead for Benenson's [9] n-authentication approach is listed in Table 2. Although a direct comparison might not be appropriately due to the different approaches used in each setting, we can see the costs are dramatically different for the two schemes.

RECOMMENDATIONS

Based on the overall conceptual framework and security protocol as proposed above, in this section we consider the implementation issues. Although a practical and experimental implementation is beyond the scope of this paper, we give the recommendations of using security options at MAC sublayer based on IEEE 802.15.4 specification [13]. In particular, Access Control List (ACL) and secured security modes [13] will be incorporated into our scheme in order to provide confidentiality on frame level at MAC sublayer for all of the three-phase protocol. The major effect and objective of

this approach is to cover the password or user related information during its transit within the UA's handshakes. Regarding security specifications for the IEEE 802.15.4 standard, there are still many limitations and deficiencies that need to be revised for later version of the specification [12, 13, 14]. Not all of the security features could be supported and accessed by application level through the use of security API. For example, there are inadequate supports in the number of ACL entries by the TinySec [15]. The specification allows a maximum of 255 ACL entries to be supported; but TinySec supports none at the moment. Developers need to implement the equivalent ACL entries by themselves if this security feature is required. Within the ACL, there are no supports for group keying and pairwise keying in the 802.15.4 specification. However, these two features are supposed to be paramount for our practical implementation.

Recall that the MAC sublayer provides the following security mode: unsecured mode, ACL mode, and secured mode [13]. Unsecured mode is the default security mode for the MAC sublayer; this means no security (encryption or authentication) is provided at all. ACL mode provides a means for a particular group of sensor node devices to filter received frames according to the source address in the frame. If the sender of the data frame was not found in the ACL entries on the receiving node side, then the frame will be eventually filtered out or passed to the next higher layer for further processing. Format of an ACL entry is shown in Figure 5. The destination address of an outgoing packet is matched with the address field in an ACL entry. The packet is then processed using the specified security suite with the key field and IV (Initialization Vector) field listed in the ACL entry. For incoming packets, the source address is matched with the address field in the ACL entry; and the replay counter field acts as a reference to detect the occurrence of any packet replay. While in secured mode, it provides a mechanism for the MAC sublayer to both use the ACL functionality and provide encryption or/and authentication functions (depending on which security suite be enabled) on incoming and outgoing frames.

Address	Security Suite	Key	Last IV	Replay Counter

Figure 5. Format of an ACL entry

In our proposed UA scheme, the ACL mode combined with the secured mode could be set up on login-nodes during the Phase 1 Registration at Step3, since the sensor GW-node will distribute the Msg (userID, A, TS) to the group of sensor login-nodes. All the login-nodes will have their ACL entries recording the source address of GW-node. For other ordinary sensor nodes, they do not have this ACL entry, and that they will not retain this data frame which has already been filtered out at MAC sublayer. Also, during this data frame distribution with the sensor network, no password information will be disclosed, since data value of password and B are not distributed outside the GW-node in this case. Similarly, the above combined mode (i.e. ACL plus secured mode) could also be set up on GW-node at Phase 2 Login, step 3 (in Fig. 4) where login-nodes will send Msg(userID*,C2,C1,T) back to GW-node waiting for authentication in next phase. Now, it is the GW-node's turn to examine from its ACL address entries to verify if the data frames' source addresses match with its stored address list. The static addresses of all the login-nodes have been pre-installed before the deployment on the gateway node side.

CONCLUSION

In this paper, a light-weight user authentication has been introduced to address the access control problem in a WSN environment. An effective dynamic UA scheme was proposed based on strong-password authentication approach. The proposed UA scheme was further justified through the security and cost analysis, and discussion on the implementation issues with the recommendations of using security features of the IEEE 802.15.4 MAC sublayer. In our future work, an implementation of the proposed UA scheme will be carried out on our WSN test-bed and experimental tests will be conducted.

REFERENCES

1. C.Y. Chong and S. Kumar (August 2003); Sensor Networks: evolution, opportunities and challenges; *Proceedings of IEEE*, Vol. 91, No. 8, pp. 1247-1256.
2. C.C. Chang, and T.C. Wu (May1991); Remote Password Authentication with Smart Cards; *IEEE Proceedings*, vol. 138, no. 3, pp. 165-- 168.
3. A. Awasthi and S. Lal (2005); A New Remote User Authentication Scheme Using Smart Cards with Check Digits; *Manuscript*.
4. M.L. Das, A. Saxena, and V.P. Gulati (2004); A Dynamic ID-based Remote User Authentication Scheme; *IEEE Transactions on Consumer Electronics*, Vol. 50, No.2.
5. A. Awasthi, (Sep 2004); Comment on Adynamic ID-based Remote User Authentication Scheme; *Transaction on Cryptology*, Vol. 01, Issue 02, Page 15-17.
6. M.S. Hwang and L.H. Li, (2000); A New Remote User Authentication Scheme Using Smart Cards; *IEEE Transaction Consumer Electronic*, vol. 46, No. 1, pp.28 -30.
7. C.Y. Lee, C.H. Lin, and C.C. Chang, (March 2005); An Improved Low Communication Cost User Authentication Scheme for Mobile Communication; *Proceedings of the IEEE 19th International Conference on Advanced Information Networking and Applications (AINA 2005)*, Taiwan.
8. N. El-Fishway, M. Nofal, and A. Tadros, (May 2002); An Effective Approach for Authentication of Mobile Users; *IEEE 55th Vehicular Technology Conference (VTC)*.
9. Z. Benenson, N. Gedicke, and O.Raivio (June 2005), Realizing Robust User Authentication in Sensor Networks, *Workshop on Real-World Wireless Sensor Networks*, Sweden.
10. D. Malan, M. Welsh, and M. Smith, A Public-key Infrastructure for Key Distribution in TinyOS based on Elliptic Curve Cryptography; First *IEEE International Conference on Sensor and Ad Hoc Communications*.
11. Z. Benenson, F. Gartner, and D. Kesdogan (September 2004), User Authentication in Sensor Networks (Extended Abstract); *Lecture Notes informatics Proceedings of Informatik 2004, Workshop on Sensor Networks*, Ulm, Germany.
12. N. Sastry and D. Wagner, (Oct 2004); Security Considerations for IEEE 802.15.4 Networks, *ACM Workshop on Wireless Security* (WiSe 2004), Philadelphia, PA, USA.
13. IEEE Standards for 802.15.4; Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR- WPANs),

Version of 1 October 2003, http://standards.ieee.org/getieee802/download/802.15.4- 2003.pdf

14. SmartRF CC2420 Datasheet (rev 1.3), 2005-10-03 (for Mica-z sensor mote), Chipcon AS, www.chipcon.com/files/CC2420_Data_Sheet_1_3.Pdf
15. C. Karlof, N. Sastry, and D. Wagner, (June 2004); *TinySec: User Manual*, http://www.tinyos.net/tinyos1.x/doc/tinysec.pdf
16. T. ElGamal, (July 1985); A public key cryptosystem and a signature scheme based on discrete logarithms; *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp.469 – 472.
17. B. Schneier, (1996); Applied cryptography, *John Wiley & Sons Inc.*, New York, 2nd edition.
18. C.C. Chang and K.F. Hwang, (2003); Some Forgery Attack on a Remote Authentication Scheme Using Smart Cards, *Informatics*, vol. 14, no. 3, pp. 189 – 294.