Two Factor user Authentication in Wireless Sensor Networks

Ajeena A
Student,Mtech CSE
Younus College of Engineering and Technology,
Vadakkevila, Kollam-691010

Muneera Hashim
Asst.Professor ,CSE
Younus College of Engineering and Technology,
Vadakkevila, Kollam-691010

Abstract: **Wireless sensor networks (WSN) are typically deployed in an unattended environment, where the legitimate users can login to the network and access data as and when required . Consequently, user authentication is a primary concern in this resource-constrained environment before accessing data from the sensor/gateway nodes. User authentication is essential for customized services and privileged access control in wireless sensor network. Designing a user authentication protocol for wireless sensor networks is a difficult task because wireless networks are susceptible to attacks and sensor nodes have limited energy, processing and storage resources. In this letter, we present a two-factor user authentication protocol for WSN, which provides strong authentication and achieves efficiency.**

1. INTRODUCTION

Advance in wireless communication technology are enabling the deployment of networks of small sensors. A wireless sensor network is a collection of sensor nodes organized into a cooperative network. wireless sensor network (WSN) of spatially distributed automated sensors to monitor physical or environmental conditions, such as temperature, sound, pressure etc. and to cooperatively pass their data through the network to a main location. Generally Wireless sensor networks (WSNs) are large scale, usually slow moving or static. A Wireless Sensor Network typically consists of a large number of tiny, low- power and multifunctional sensor nodes that are deployed in a region of interest The sensor nodes (motes) in such networks are designed to sense the environment and collect data. A sensor node, also known as a mote that is capable of performing some processing, gathering sensory information and communicating with other connected nodes in the network. A mote is a node but a node is not always a mote. In a WSN, each sensor node is able to independently perform some processing and sensing tasks. Furthermore, sensor nodes communicate with each other in order to forward their sensed information to a central processing unit or conduct some local coordination such as data fusion. The sensor nodes are equipped with sensors, embedded micro-processors and radio transceivers for sensing ,data processing and communicating capabilities. They communicate over short distance via a wireless medium and collaborate to accomplish a common task.

Fig 1. Sensor Node

Over the years, Wireless Sensor Networks (WSN) have attracted in increasing interest from researchers due to its ubiquitous nature, easy deployment, and the range of applications they enable. Networks of thousands tiny sensor devices, which have low processing power, limited memory and energy play important roles for an economical solution to some of the challenging problems. Wireless Sensor Networks (WSNs) have many promising applications including environmental monitoring, traffic monitoring, fire alarming, logistics, military sensing and tracking, Health monitoring and so on. In general, most of the queries in WSN applications are issued at the points of base stations or Gateway (GW) nodes of the network. Gateway is a router or a proxy server that routes between networks. That is, gateway is a network point that acts as an entrance to another network. However, one can foresee that there should have great needs to access the real-time data inside WSN, where

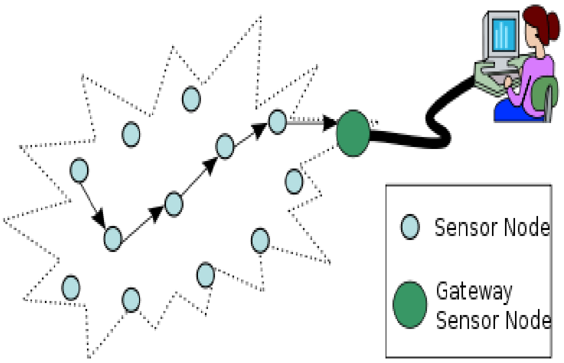


Fig 2.Wireless Sensor Network

real-time data from the sensor nodes may no longer be accessed through the GW-node only, instead, the data are to be accessed directly by the external party (user) as and when demanded. If the data in WSN are made available to the user on demand, then authentication of the user must be ensured before allowing the user to access data.

Fig 3. Gateway Nodes

The efficient user authentication in WSN application layer has not been addressed adequately in comparisons with the network and link layers protocol in WSN. One of the possible factors could be the resource-constrained WSN environment. User authentication is essential for customized services and privileged access control in wireless sensor network. Designing a user authentication protocol for wireless sensor networks is a difficult task because wireless networks are susceptible to attacks and sensor node has limited energy, processing and storage resources .This letter, present an efficient user authentication protocol for WSN. The protocol uses the two-factor authentication concept and resists many logged in users with the same login identity, stolen-verifier, guessing, impersonation and replay threats. A two-factor authentication is a concept used to describe an authentication mechanism, where more than one factor (e.g., password and chip card) is required to authenticate the communicating party.

II. RELATED WORKS

There have been significant progresses in WSN for link layer security and network layer security . However, the application layer security in WSN has not been addressed effectively. Benenson et al. proposed a protocol for WSN, “User authentication in sensor networks” where user can successfully authenticate with any subset of sensors out of a set of *n* sensors. Subsequently, Watro et al. proposed a user authentication protocol, named TinyPK, using the RSA and Diffie-Hellman algorithms . We observe that the TinyPK protocol suffers from the “masquerade as sensor node to an unknowing user” attack explained as follows. On having user’s public key, the intruder encrypts a session key along with other parameters and sends the encrypted string to the user. Upon receiving the encrypted string, the user would assume that it has come from the sensor node, though it has come from the intruder. Consequently, the user decrypts the received string using her/his private key and uses the session key for subsequent operations with the intruder. Wong et al. proposed an efficient user authentication protocol for WSN using only hash function, named “A dynamic user authentication scheme for wireless sensor networks ”which is based on user’s password. We observe a security flaw in Wong et al.’s protocol as explained below. The protocol is vulnerable to many logged in users with the same login-id threat, that is, who has a valid user’s password can login to the sensor network. The protocol also suffers from stolen-verifier attack, because both the GW-node and login-node maintain the lookup table of registered users’ credentials. This letter aims to devise a user authentication protocol that eliminates the weaknesses of Wong et al.’s protocol and provides better security.

III.THE PROPOSED PROTOCOL

WSN are deployed in a confined area, which could be divided into different zones. Authorized users can access WSN using their mobile devices (e.g., Notebook PC, PDA). Before issuing any queries to or access data from sensor network, the user has to register with the GW-node of the network. Upon successful registration, the user can submit query to the WSN at any time within a predefined or administrative configurable period. The basic idea of the protocol is that a user will receive a personalized smart card from the GW-node at the time of the registration process and then, with the help of user’s password and smart card the user can login to the sensor/GW node and access data from the network. The protocol is divided into two phases: Registration phase and Authentication phase.

U	User
ID	Identity of U
PW	Password of U
DID	Dynamic login identity of U
GW	Gateway node of WSN
K	Symmetric key of GW-node
h(・)	Cryptographic hash function
S1 ⊕ S2	String S1 is XOR-ed with string S2

Table 1: Notations used in the protocol

A.Registration Phase

This phase is invoked when a user, Ui, wants to register with the WSN. Ui submits his/her identity (IDi) and password (PWi) to the GW-node in a secure manner. Upon receiving the registration request, the GW-node computes Ni = h(IDi || PWi) ⊕h(K), where K is a symmetric key known to only GW-node, and‘ || ’ is bit-wise concatenation operator. Then the GW-node personalizes a smart card with the parameters h(•), IDi, Ni, h(PWi) and xa, where h(•) is a cryptographically secure hash function. Here, xa is a secret parameter generated securely by the GW-node and stored in some designated sensor nodes before deploying the nodes in the field, who are responsible to exchange data with users (we assume that a node is responsible for many applications. If the WSN is built for only one application then this secret parameter is known to all nodes). The GW-node now sends the personalized smart card to Ui in a secure manner. We note that xa is not known to the user, as it is generated and stored in user’s smart card securely by the GW-node.

B. Authentication Phase

The authentication phase is invoked when Ui wants
to perform some query to or access data from the network. The phase is further divided into Login and Verification phases.

1) Login Phase: Ui inserts her/his smart card to a terminal, and keys IDi and PWi. The smart card validates IDi and PWi with the stored ones in it. If the entered IDi and PWi are correct, the smart card performs the following operations:
Step-L1**)** Compute DIDi = h(IDi || PWi)⊕ h(xa || T), where T is the current timestamp of Ui’s system.
Step-L2) Compute Ci = h(Ni || xa || T). Then send <DIDi, Ci, T > to the GW-node.

2) Verification Phase: Upon receiving the login request < DIDi, Ci, T > at time T ∗, the GW-node authenticates Ui by the following steps:
Step-V1) Validate T. If (T ∗ − T) ≤ ΔT then the GW node proceeds to next step, else abort, where ΔT denotes the expected time interval for the transmission delay.
Step-V2) Compute h(IDi_PWi)* = DIDi ⊕ h(xa_T) and Ci * = h((h(IDi || PWi)* || h(K)) || xa || T)
Step-V3) If Ci *= Ci, the GW-node accepts the login request; else rejects it.
Step-V4) GW-node now sends a message < DIDi,Ai, T ‘> to some nearest sensor node, say, Sn, over a public channelto respond the query/data what Ui is looking for, where Ai = h(DIDi || Sn || xa || T’) and T ‘ is the current timestamp of GW-node’s system. Here, Ai is used to ensure the sensor node that the message < DIDi,Ai, T ‘ > has come from the legitimate GW-node, as Ai is generated with secret parameter xa which is known to both sensor and GW nodes.
Step-V5) Sn first validates T ‘ in similar line of Step-V1. Then Sn computes h(DIDi || Sn || xa || T’) and checks whether it is equal to Ai. If these two checks pass correctly then Sn responds to Ui’s query.

IV.ANALYSIS OF THE PROTOCOL

This section shows our protocol’s strength in terms of security and efficiency.

A. Security Analysis

we assume that an intruder can physically capture a node, but cannot able to extract data from the node.

With these assumptions, the proposed protocol resists the following attacks:

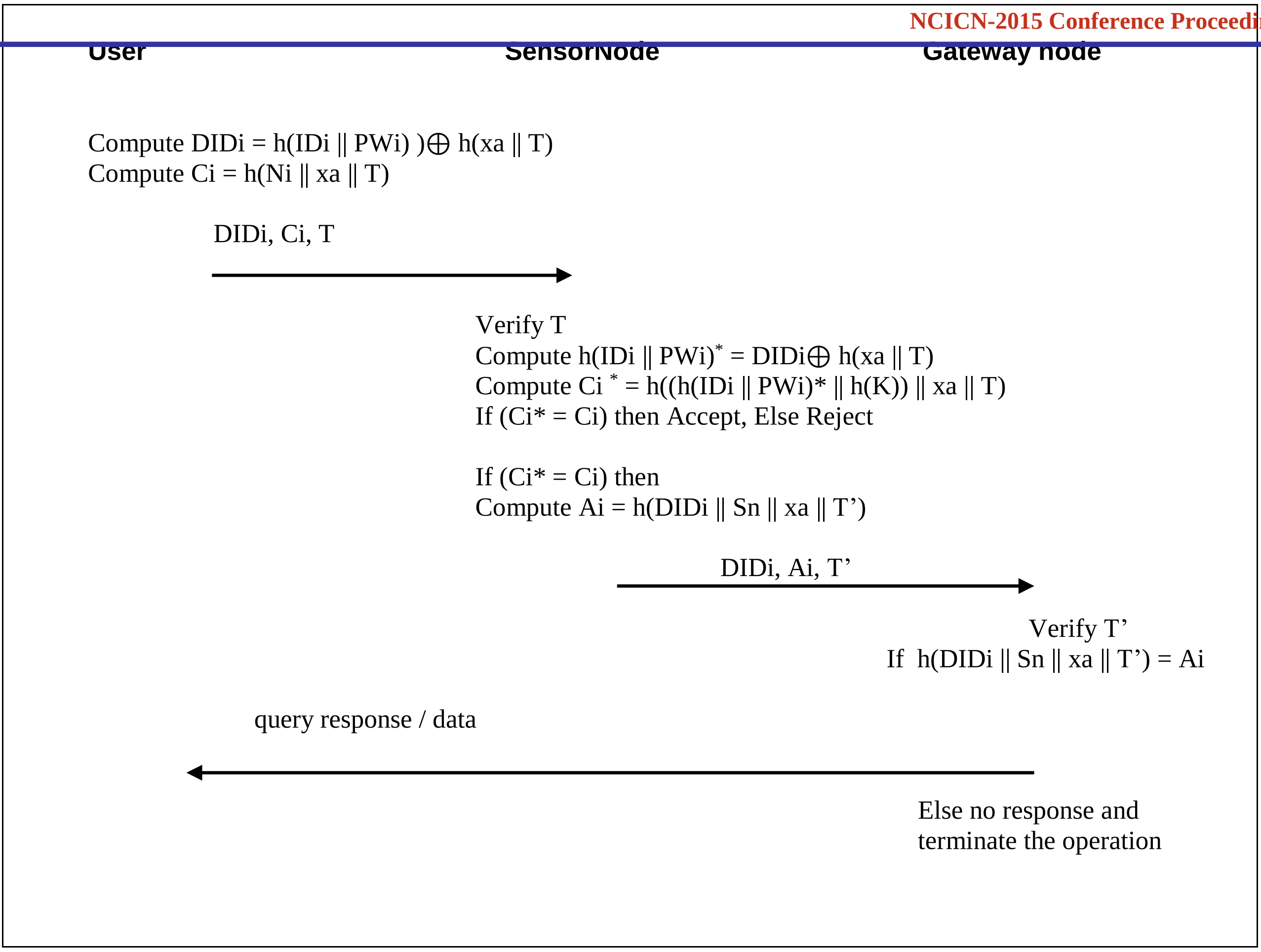

User
SensorNode
Gateway node
Compute DIDi = h(IDi || PWi))⊕ h(xa || T)
Compute Ci = h(Ni || xa || T)
DIDi, Ci, T
Verify T
Compute h(IDi || PWi)* = DIDi⊕ h(xa || T)
Compute Ci * = h((h(IDi || PWi)* || h(K)) || xa || T)
If (Ci* = Ci) then Accept, Else Reject
If (Ci* = Ci) then
Compute Ai = h(DIDi || Sn || xa || T’)
DIDi, Ai, T’
Verify T’
If h(DIDi || Sn || xa || T’) = Ai
query response / data
Else no response and terminate the operation

Fig 4. Authentication Phase

- *Replay Attack:* A replay attack (replaying an intercepted message) cannot work in our protocol. Suppose the intruder intercepts a valid login request < DIDi, Ci, Ti > and tries to login to the GW-node by replaying the same. The verification of this login request fails because of the interval $(Ti' - Ti) > \Delta T$, where Ti’ is the GW-node’s system time while receiving the replayed message.
- *Impersonation Attack:* On intercepting a valid login request < DIDi, Ci, Ti >, the intruder will have DIDi, but, to login again,

  DIDi needs to be recomputed with a new timestamp, say Tnew, to avoid the replay attack, which is not possible without knowing PWi and xa, as $DIDi = h(IDi\_PWi) \oplus h(xa||T)$. It is practically infeasible to obtain PWi and/or xa from the intercepted parameters, because of the one-way property of h(•). Therefore, the intruder cannot impersonate auser. It should be noted that no one (including a valid user) can forge GW-node or others’ login request. A valid user, say, Ui knows PWi, but obtaining xa from DIDi or smart card is again a hard problem, as a valid login request requires both PWi and xa. Ui may also try to obtain h(K) and if s/he succeeds over it then s/he can personalize as many as registered users without GW-node’s knowledge. But, s/he cannot succeed to get h(K), because to get h(K) s/he has to have Ni which is stored in her/his smart card and the smart card uses it for on-card computation to generate login request. Consequently, impersonating user or GW-node is prevented in our protocol.
- *Stolen-verifier Attack:* One of the interesting characteristics of our protocol is that it is free from password/verifier table, which prevents our protocol from stolen-verifier attack. The insider of the network cannot get/steal user’s password, as the GW/sensor node does not need to maintain any password/ verifier table to validate user’s login request. Although the user submits her/his PWi to the GW-node during registration process, the GW-node (a trusted entity in the network) should delete user’s password record once the user registration process is over. As a consequence, stolen-verifier attack is prevented in our protocol.
- *Guessing Attack:* Guessing attack is a crucial concern in any password-based system. We note that our scheme is free from password/verifier table, and user password is not transmitted simply hash of the password. Instead, we let password to be transmitted as a digest of some other secret

components. Although the intruder will have DIDi which contains user password and secret parameter xa, the intruder cannot guess either user's password or xa from DIDi, as the security is based on the one-way property of the hash function.

- *Denial-of-Service Attack:* Denial-of-service attack is a potential attack in every system, where communication channel is public (insecure). A value-added customer will be deprived from the service due to network/service provider's rival or intruder. Our protocol does not provide any protection against this attack. This is due to the fact that it is a one-way protocol in which the GW-node sends message without expecting any acknowledgement. If the adversary blocks the message from reaching the nodes, neither the GW-node nor the sensor node will know about it.

  Up to a certain limit we can control the **denial of service attack** by sending an acknowledgement message from the gateway node to the user after the verification process. If the acknowledgement message receives from gateway to the user then the user can confirm that his request is accepted and can wait for the data up to some time. After sometimes if the requested data is not received, then the user should confirm that his request is being hacked. If the acknowledgement message doesn't receive, then the user may confirm that his request is being hacked at the initial stage and there is no need to wait for the data.

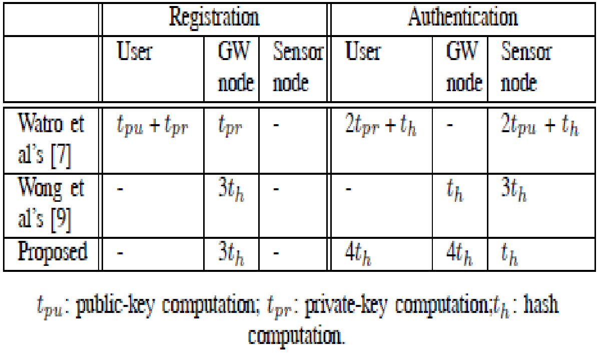
- *Node Compromise Attack*: Typically, WSN are deployed in an unattended and hostile environment. One could easily capture a node and try to collect some secret information from it about the networks. Implementation of one-time sensors can prevent this attack, but it is limited to some applications such as fire alarm, where confidentiality of the transmitted data is not required/important. When confidentiality of data is a concern, it is a difficult task to prevent this attack if sensor nodes are not tamper-proof and the environment is unattended. The GW-node, however, can monitor periodically whether any node is captured or not. If user authentication and data access from node are allowed to the user directly (i.e., without GW node's notice) then the impact of "node compromise" attack is very high, which occurs in Watro et al. protocol. Whereas, in our protocol, the user's request first gets authenticated by the GW-node and then the instruction is sent to the node for responding to the user query. We note that none of the three protocols (Watro et al., Wong et al.and Proposed one) provides an inherent method to detect a compromise node. This opens a prominent future scope of this work to mitigate the node compromise attack. Additionally, the proposed protocol successfully prevents the many logged in users with the same login-id threat. Most of the password-based systems which maintain the verifier table to validate user login suffer from this threat. However, our protocol resists this threat without maintaining any verifier table at the GW/sensor node, as one has to have a valid <ID,PW > and a smart card corresponding to < ID,PW > to login to the network. The proposed protocol requires oncard computation for login to the network and once the smart card is removed from the user system, the login session will be terminated.

B. Efficiency

In order to analyze the efficiency of our protocol, we compare the protocol with Watro et al. and Wong et al.. The Table-2 shows efficiency of our protocol.

1.Computational cost:

The computational cost for user registration is a one-time job for certain period of time. But, the computational cost for user authentication is of prime concern, as this is required as and when a user wants to login to the WSN. From Table-2, it is easy to see that the computational cost of our protocol is well-suited to the resource-constrained sensor node, as the sensor node requires only 1 hash operation, whereas the sensor node in the Wong et al.'s protocol requires 3 hash operation. The computational cost of Watro et al.'s protocol is high in comparison with our and Wong et al.'s protocols, as Watro et al.'s protocol requires modular exponentiation which is computationally expensive. In our protocol, the GW-node handles major computational burden, as it computes 4 hash operation for a successful user authentication, and the user also requires to compute 4 hash operation. We assume GW-node and user's device will haveenough computational resource to compute these operations.We believe this is a reasonable assumption, as the GW-node needs to collect huge information in a form of query-response from all sensor nodes. Our goal is to minimize computational overhead on sensor nodes, and in this context our protocol achieves efficiency in comparison with other protocols.



	Registration			Authentication		
	User	GW node	Sensor node	User	GW node	Sensor node
Watro et al's [7]	$t_{pu} + t_{pr}$	t_{pr}	-	$2t_{pr} + t_h$	-	$2t_{pu} + t_h$
Wong et al's [9]	-	$3t_h$	-	-	t_h	$3t_h$
Proposed	-	$3t_h$	-	$4t_h$	$4t_h$	t_h

t_{pu}: public-key computation; t_{pr}: private-key computation;t_h: hash computation.

Table 2. performance of the protocol

2. Communication cost:

From Figure-4, it is easy to visualize that a successful user authentication in our protocol requires three message exchanges, whereas Wong et al.'s and Watro et al.'s protocol requires four and two exchanges, respectively. Although Watro et al.'s protocol requires less number of message exchanges, their protocol is computationally expensive for the resource-constrained environment. Moreover, the message size (i.e., the number of sub-messages and their sizes) of our protocol is lesser than the Wong et al.'s and Watro et al.'s protocols. We note that the actual number of message exchanges could vary if the message transmission between the GW-node and sensor node requires multi-hop.

3. Node Energy cost:

Node energy cost combines both computational and communication costs. The sensor node of Watro et al.'s protocol consumes battery for nonce validation, checksum generation and verification and two public key operations, then responds to the user query. InWong et al.'s protocol, the sensor node consumes battery for a lookup table query and three hash operations for parameters generation and then wait time for GW-node's response before responding to the user query. In contrast, the sensor node in our protocol requires timestamp validation and one hash operation for parameter generation and then responds to the user query. Consequently, sensor node's energy consumption in our protocol is significantly lesser than other two protocols. Considering computational, communication and node energy costs, it is clear that our protocol is efficient compared to Wong et al.'s and Watro et al.'s protocols.

V.CONCLUSION

The letter proposed a two-factor user authentication protocol for WSN using only hash function. The proposed protocol avoids many logged in users with the same login-id and stolen-verifier attacks, which are prominent threats for a password-based system if it maintains verifier table at the GW-node or sensor node. In addition, the proposed protocol resists other attacks in WSN such as Denial of service attack up to a certain limit by sending an acknowledgement message from gateway node to user after the verification process. We have showed the efficiency of the proposed protocol in comparisons with the related ones. However, a simulated/experimental result would have been a better picture to show the feasibility of the proposed protocol and the work can be further extended with an experimental result along with the counter measure against the node compromise security threats.

REFERENCE

[1] E. H. (Jr) Callaway. *Wireless Sensor Networks, Architectures and Protocols*. Auerbach Publications, 2003.

[2] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar, "SPINS: security protocols for sensor networks," *Wireless Networks*, vol. 8, no. 5, pp. 521-534, 2002.

[3] N. Sastry and D. Wagner, "Security considerations for IEEE 802.15.4 networks," in *Proc. ACM Workshop Wireless Security, ACM Press*, pp. 32-42, 2004.

[4] C. Karlof, N. Sastry, and D. Wagner. "TinySec: a link layer security architecture for wireless sensor networks," in *Proc. International Conf. Embedded Networked Sensor Syst.*, ACM Press, pp. 162-175, 2004.

[5] "RSA SecureID, "Secure identity." [Online] Available: http://www.rsa.com/node.aspx?id=1156.

[6] M. L. Das, A. Saxena, and V. P. Gulati. "A dynamic ID-based remoteuser authentication scheme," *IEEE Trans. Consumer Electron.*, vol. 50, no. 2, pp. 629-631, 2004.

[7] R. Watro, D. Kong, S. Cuti, C. Gardiner, C. Lynn, and P. Kruus, "TinyPK: securing sensor networks with public key technology," in *Proc. ACM Workshop Security of Ad Hoc Sensor Networks*, pp. 59-64, 2004.

[8] Z. Benenson, F. Gartner, and D. Kesdogan. "User authentication in sensor networks," in *Proc. Workshop Sensor Networks, Lecture Notes Informatics Proceedings Informatik*, 2004.

[9] K. Wong, Y. Zheng, J. Cao, and S. Wang. "A dynamic user authentication scheme for wireless sensor networks," in *Proc. IEEE International Conf. Sensor Networks, Ubiquitous, Trustworthy Computing, IEEE Computer Society*, pp. 244-251, 2006.

[10] IEEE Standards for 802.15.4; Part 15.4: "Wireless medium access control and physical layer specifications for low-rate wireless personal area networks," 2003.

[11] B. Schneier. *Applied Cryptography*. John Wiley & Sons, 1996.

[12] P. C. Kocher, J. Jaffe, and B. Jun. "Differential power analysis," in *Proc. Advances Cryptology*, Springer-Verlag, LNCS 1666, pp. 388–397, 1999.

[13] D. Dolev and A. C. Yao. "On the security of public-key protocols," *IEEE Trans. Inform. Theory*, vol. 29, no. 2, pp. 198-208, 1983.

[14] K. Bicakci, C. Gamage, B. Crispo, and A. S. Tanenbaum. "One-time sensors: a novel concept to mitigate node-capture attacks," in *Proc. Workshop Security Privacy in Ad-hoc Sensor Networks, LNCS 3813*, pp. 80-90, 2005.

[15] Y. W. Law, S. Etalle, and P. H. Hartel. "Assessing security in energy-efficient.