Improved User Authentication in Wireless Sensor Networks

K.S.Arikumar,
Department of CSE
Mepco Schlenk Engineering College
Sivakasi, India
ksarikumar@gmail.com

K.Thirumoorthy,
Department of CSE
Mepco Schlenk Engineering College
Sivakasi, India
kthirumoorthy@mepcoeng.ac.in

***Abstract*— The primary requirements of a successful Wireless Sensor Network (WSN) security architecture are confidentiality, integrity and authentication. User authentication is essential for customized services and privileged access control in wireless sensor network. These sensors will have limited processing power, storage, bandwidth and limited communication capabilities. Two-factor user authentication concept is a most widely used technique in WSN. Exiting technique takes more energy consumption and shortening the sensor node lifetime. In this paper, we present a two-factor user authentication protocol for WSN, which provides strong authentication, session key establishment. This scheme is well-designed for sensor nodes which typically have limited resources in the sense that its authentication procedure requires no public key operations but it uses only one-way hash functions and smart cards and can be implemented efficiently. Our scheme allows the users to choose and change their passwords freely, and do not maintain any verifier table. We compare our proposal with other exiting technique through simulation and we show that it achieves high efficiency.**

***Keywords*— Authentication, sensor networks, wireless security, hash function, smart card.**

I. INTRODUCTION

A wireless sensor network (WSN) is a network made of many small sensor nodes and one or more base stations which centralizes the data gathered by sensor nodes. Sensor nodes are small computers, extremely basic in terms of their interfaces and their components. They usually only consist of a processing unit with limited computational power and limited memory, sensors, a communication device, and a power supply [1]. In general, most of the queries in WSN applications are issued at the points of base stations or Gateway (GW) nodes of the network. However, one can foresee that there should have great needs to access the real-time data inside WSN, where 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.

When information is particularly sensitive or vulnerable, using a password alone may not be enough protection. A stronger means of authentication, something that's harder to compromise is necessary. Two-factor authentication is also called strong authentication. It is defined as two out of the following three proofs: Something known, like a password, something possessed, like ATM card, something unique about appearance of person, like a fingerprint.

In this paper, presents 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, and 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 smart card) is required to authenticate the communicating party. Authentication procedure requires no public key operations but it utilizes only cryptographic hash function. By using the one way cryptographic hash function the protocol Achieves efficiency of the protocol.

The remainder of the paper is organized as follows. Section 2 reviews the related work in WSN. Section 3 presents our protocol and Section 4 emulation and performance analyzes of the protocol. Section 5 concludes the paper.

II. RELATED WORK

There have been significant progress in WSN for link layer security [3], [4] and network layer security [2].However, the application layer security in WSN has not been addressed effectively. Benenson et al. [7] proposed a protocol for WSN, where user can successfully authenticate with any subset of sensors out of a set of *n* sensors. Subsequently, Watro et al. [5] proposed a user authentication protocol, named TinyPK, using the RSA and Diffie-Hellman algorithms [6]. 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. [8] proposed an efficient user authentication protocol for WSN using only hash function, 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. Recently, M.L. Das [9] proposed a two-factor user authentication scheme in WSNs. M.L. Das also identified that Wong *et al.*'s protocol is vulnerable to many logged-in users with the same login-id threat, that is, who has a valid user's password can easily login to the sensor network [9]. He also identified that Wong *et al.*'s protocol is susceptible to stolen-verifier attack, because the GW-node and login-node maintain the lookup table of all the registered users' credentials. Consequently, M.L. Das proposed his protocol to overcome the security flaws of Wong *et al.*'s scheme. His protocol uses the two factor authentication concept based on password and smart card and resists many logged-in users with the same login identity, stolen-verifier, guessing, replay, and impersonation attacks. However, in this paper, we identify that the M.L. Das-scheme is still not secure and vulnerable to several critical security attacks. We show that the M.L. Das-scheme is defenseless against GW-node by-passing attack, does not provide mutual authentication between GW-node and sensor nodes, has the security threat of insider attack, and does not have provision for changing or updating passwords of registered users. To fix the aforementioned weaknesses of the M.L. Das-scheme, we propose security improvements in our paper. The proposed security improvements can easily be incorporated into the M.L. Das-scheme to take the benefit of more secure and robust two-factor user authentication in WSN.

III. THE PROPOSED PROTOCOL

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 three phases namely, registration phase, authentication phase and password change phase. The registration phase is performed only once, and the authentication phase is executed every time the user logs into the system. The notation used throughout the paper is shown in Table-1.

TABLE I
NOTATION USED IN THE PROTOCOL

Notation	Meaning
U_i	User *i*
ID_i	identity of user *i*
PW_i	password of user *i*
x_a	secret parameter
$h(.)$	cryptographic hash function
S_n	identity of sensor node *n*
$\oplus$	exclusive OR
$\|\|$	concatenation
x_s	secret parameter
K	symmetric key of the gateway node
DID_i	dynamic login identity of user *i*

A. Registration Phase

This phase is invoked when a user U_i, wants to register with the WSN. U_i Submits his/her identity (ID_i) and password (PW_i) to the GW-node in a secure manner. Upon receiving the registration request, the GW-node computes $N_i = h(ID_i \| PW_i \| x_a) \oplus h(K)$ 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 $h(.)$, ID_i, N_i, $h(PW_i)$ and x_a, where $h(.)$ is a cryptographically secure hash function. Here, x_a is a secret parameter generated securely by the GW-node and stored in some designated sensor nodes before deploying the nodes in the field, which are responsible to exchange data with users (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 U_i in a secure manner. We note that x_a 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 U_i 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: U_i inserts her/his smart card to a terminal, and keys ID_i and PW_i. The smart card validates ID_i and PW_i with the stored ones in it. If the entered ID_i and PW_i are correct, the smart card performs the following operations:

- **i)** Compute $DID_i = h(ID_i \| PW_i \| x_a) \oplus h(x_a \| T)$, Where T is the current timestamp of is U_i 's system.
- **ii)** Compute $C_i = h(N_i \| x_a \| T)$. Then send $< DID_i, C_i, T >$ to the GW-node.

2) Verification Phase: Upon receiving the login request $< DID_i, C_i, T >$ at time T_1, the GW-node authenticates U_i by the following steps:

- **i)** Validate T. If $(T_1 - T) \le \Delta T$ then the GW node proceeds to next step, else abort, where ΔT denotes the expected time interval for the transmission delay.
- **ii)** Compute $h(ID_i \| PW_i \| x_a)^* = DID_i \oplus h(x_a \| T)$ and $C_i^* = h((h(ID_i \| PW_i \| x_a)^* \oplus h(K)) \| x_a \| T)$.
- **iii)** If $C_i^* = C_i$, the GW-node accepts the login request; else rejects it.
- **iv)** GW-node now sends a message $< DID_i, A_i, T_2 >$ to some nearest sensor node, say, S_n, over a public channel to respond the query/data what U_i is looking for, where $A_i = h(DID_i \| S_n \| x_s \| T_2)$, and T_2 is the current timestamp of GW-node's system. Where x_s should be another secret parameter, which

should only be known to the GW-node and sensor nodes, and can be stored in sensor nodes before their deployment in the field. Here, A_i is used to ensure the sensor node that the message $< DID_i, A_i, T_2 >$ has come from the legitimate GW-node, as A_i is generated with secret parameter x_a which is known to both sensor and GW nodes.

v) S_n first validates T_2 in similar line of Step-**i**. Then S_n computes $h(DID_i \| S_n \| x_s \| T_2)$ and checks whether it is equal to A_i. If these two checks pass correctly then S_n responds to U_i 's query.

vi) To provide mutual authentication between GW-node and sensor node, S_n now computes $M_i = h(S_n \| x_s \| T_4)$. Here T_4 is the current timestamp of sensor node's system and sends back mutual authentication message $< M_i, T_4 >$to the GW-node.

vii) After receiving the mutual authentication message $< M_i, T_4 >$, the GW-node first checks the validity of time-stamp. If $(T_5 - T_4) \le \Delta T$, then GW node performs the further operations, otherwise the mutual authentication phase is terminated. Here, ΔT shows the expected time interval for the transmission delay and T_5 is the current timestamp of GW-node.

viii) GW-node now computes $M_i^* = h(S_n \| x_s \| T_4)$ and checks whether $M_i^* = M_i$ or not. If it is true, then GW-node establishes trust on sensor node, otherwise, GW-node intimates U_i about the possibility of malicious sensor node in the network and sends a process-termination message.

C. Password Change Phase

We introduce the password-change/update phase in the M.L. Das-scheme. In the password change phase, when a user wants to change his password PW_i to a new password PW_i^* , he/she inserts his/her smart card into the terminal and enters his ID and password. Smart card validates his ID_i and PW_i with the stored values and if the entered ID_i and PW_i are correct, then the smart card performs the following operations without interacting with GW-node:

i) Computes $N_i^* = N_i \oplus h(ID_i \| PW_i \| x_a) \oplus h(ID_i \| PW_i^* \| x_a)$, where the value of N_i is already stored on smart card.

ii) Smart card replaces the old value of N_i with the new values N_i^* and $h(PW_i^*)$. Now, the new password is successfully changed and this phase is terminated.

IV. ANALYSIS OF THE PROTOCOL

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

A. Security Analysis

Before analyzing the protocol, first assume that replication or extraction of parameter from the private space of the smart card is quite difficult as per the present literature. Although it happens by the side channel attacks, the experimental cost is much higher than the value of the intended parameter. Further, some of the smart card manufacturers consider the risk of side channel attacks [10], and provide countermeasure to deter the reverse engineering attempt. The Dolev-Yao [11] threat model often used to formally analyze crypto protocols in communication networks, where the model assumes that two communicating parties communicate over an insecure channel. WSN could adopt the similar threat model where the channel is insecure and the end-points (user, sensor node) cannot in general be trusted. To limit node capturing threat, assume that each sensor node is equipped with a tamper-resistant component for storing sensitive data. Although tamper-resistant nodes demand more cost and many applications do not require this costly feature, some applications, such as, healthcare, border security, require tamperproof storage devices for preventing the network from data leakage. This protocol pertains to such types of applications and assumes 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:

Replay Attack: A replay attack (replaying an intercepted message) cannot work in this protocol. Suppose the intruder intercepts a valid login request $< DID_i, C_i, T >$ and tries to login to the GW-node by replaying the same. The verification of this login request fails because of the interval $(T_1^* - T) \le \Delta T$, where T_1 is the GW-node's system time while receiving the replayed message.

Impersonation Attack: On intercepting a valid login request $< DID_i, C_i, T >$, the intruder will have DID_i , but, to login again, DID_i needs to be recomputed with a new timestamp, say T_{new}, to avoid the replay attack, which is not possible without knowing PW_i and x_a , as $DID_i = h(ID_i \| PW_i \| x_a) \oplus h(x_a \| T)$. It is practically infeasible to obtain PW_i and/or x_a from the intercepted parameters, because of the one-way property of $h(.)$. Therefore, the intruder cannot impersonate a user. It should be noted that no one (including a valid user) can forge GW-node or others' login request. A valid user, say, U_i knows PW_i but obtaining x_a from DID_i or smart card is again a hard problem, as a valid login request requires both PW_i and x_a . U_i 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 N_i 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 this protocol.

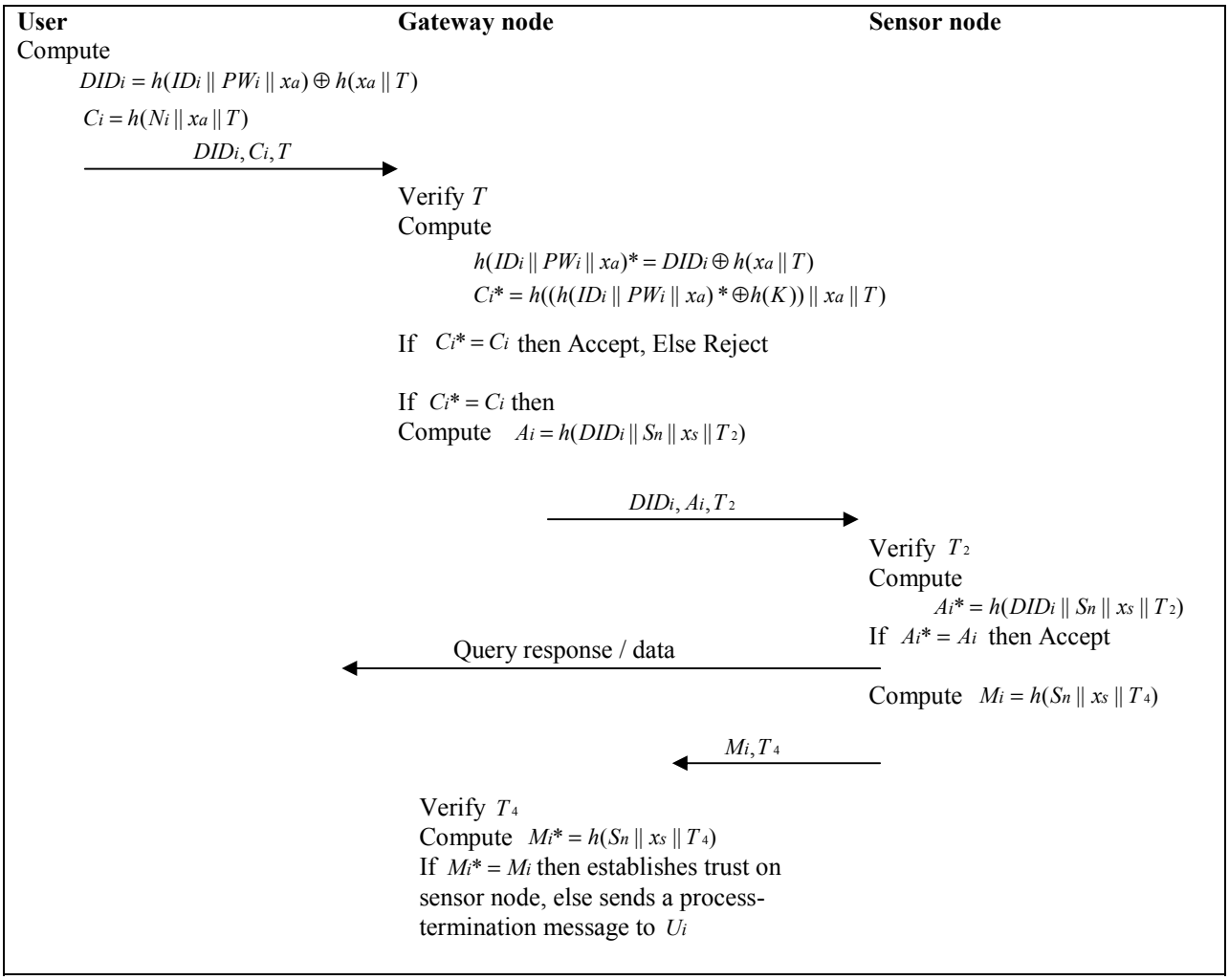

Fig. 1 Authentication phase.

Stolen-verifier Attack: One of the interesting characteristics of this protocol is that it is free from password/verifier table, which prevents this 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 PW_i 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 this protocol.

Guessing Attack: Guessing attack is a crucial concern in any password-based system. Here note this 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 DID_i which contains user password and secret parameter x_a, the intruder cannot guess either user's password or x_a from DID_i, as the security is based on the one-way property of the hash function.

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 this 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.

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, this protocol resists this threat without maintaining any verifier table at the GW/sensor node, as one has to have a valid $< ID_i , PW >$ and a smart card corresponding to $< ID_i , PW >$ to login to the network. The proposed protocol requires on-card 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 Wong et al. and M.L. Das scheme. We describe in this section some performance measurements of the proposed mechanism. We emulate our protocol in a Sun Java Wireless Toolkit for CLDC, which is a state-of-the-art toolbox for developing wireless applications that are based on Java ME's Connected Limited Device Configuration (CLDC) and Mobile Information Device Profile (MIDP), and designed to run on cell phones, mainstream personal digital assistants, and other small mobile devices. The toolkit includes the emulation environments, performance optimization and tuning features. We can run a MIDlet suite directly in the emulator or install it using a process that resembles application installation on a real device. A memory monitor, network monitor, and method profiler are provided to analyze the operation of our MIDlets.

Computational cost: The computational cost for user registration is a one-time job for certain period of time. But, the computational cost or 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 2 hash operations, whereas the sensor node in the M.L.Das's protocol requires 1 hash operations. But M.L.Das's protocol doesn't provide mutual authentication between the sensor node and gateway node.

TABLE II
PERFORMANCE OF THE PROTOCOLS

	Registration			Authentication		
	User	GW node	Sensor node	User	GW node	Sensor node
Wong et al's	-	3 t_h	-	-	t_h	3 t_h
M.L.Das's protocol	-	3 t_h	-	4 t_h	4 t_h	t_h
Proposed Protocol	-	3 t_h		4 t_h	5 t_h	2 t_h

t_h : hash computation.

From Figure-2, we can say proposed protocol consumes less memory compare to Wong et al. Protocol. But consumes more memory compare to M.L.Das's protocol. But the proposed scheme much more secure against M.L.Das's protocol. The goal is to minimize computational overhead on sensor nodes, and in this context this protocol achieves efficiency in comparison with other protocols. Compare with other exiting scheme's proposed protocol is computational cost is slightly higher than M.L.Das's protocol.

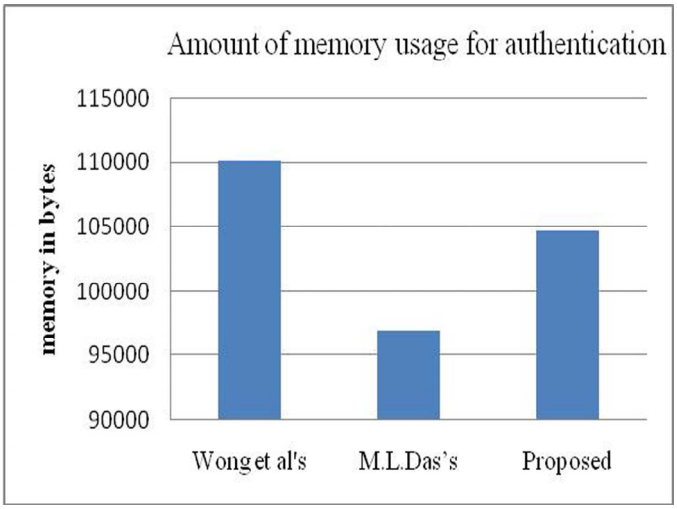

Fig. 2 Amount of memory usage for authentication

Communication cost: From Figure-1, it is easy to visualize that a successful user authentication in this protocol requires four message exchanges, whereas Wong et al.'s and M.L.Das's protocol requires four and three exchanges, respectively. Although Wong et al.'s protocol requires equal number of message exchanges; their protocol is computationally expensive for the resource-constrained environment.

Considering computational and communication costs, it is clear that our protocol is efficient compared to Wong et al.'s and slightly higher than M.L.Das's protocols.

V. CONCLUSION

In this paper, we have shown that a recently proposed two-factor user authentication scheme in WSN environment is insecure against different kinds of attack and should not be implemented in real applications. This paper 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 except the denial-of-service and node compromise attacks. This protocol allows the users to choose and change their passwords freely. This scheme is well-designed for sensor nodes which typically have limited resources in the sense that its authentication procedure requires no public key operations but it uses only one-way hash functions and smart cards and can be implemented efficiently. This proposed model compare with other exiting technique through emulation and show that it achieves high efficiency.

REFERENCES

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

[2] 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] 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.

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

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

[8] 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.

[9] Das, M.L. "Two-Factor User Authentication in Wireless Sensor Networks" in *IEEE Trans. Wireless Comm.* 2009, *8*, 1086-1090.

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

[11] 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.