A Lightweight User Authentication Scheme for Wireless Sensor Networks

Omar Cheikhrouhou[1,2], Anis Koubâa[3,4], Manel Boujelben[1], Mohamed Abid[1]

[1]CES Research Unit, National School of Engineers of Sfax-Tunisia

[2]Higher Institute of Technological Studies, ISET Sfax-Tunisia

[3]IPP-HURRAY Research Group, CISTER/ISEP, Polytechnic Institute of Porto, Porto, Portugal

[4]Al-Imam Muhammad ibn Saud University, Computer Science Dept., Riyadh, Saudi Arabia

Omar.cheikhrouhou@isetsf.rnu.tn, aska@isep.ipp.pt, boujelben_manel@yahoo.fr, mohamed.abid@enis.rnu.tn

***Abstract*—User authentication in classical networks is deeply addressed, but few results are related to Wireless Sensor Networks (WSNs). In addition, the proposed schemes do not provide mutual authentication or session-key agreement between the server and the user. Therefore, we present in this paper a lightweight user authentication scheme adapted to WSNs that provides mutual authentication and session-key agreement. The proposed scheme allows a user equipped with mobile device (typically PDA) to authenticate himself before gaining access to the WSN. The scheme is executed at two sides; the client side which controls the user's mobile device and the server side represented by the coordinator of the WSN. A security analysis of the scheme is presented and it proves its resilience against classical types of attacks. The scheme is also implemented on real platform of sensor nodes. This implementation proves that our scheme is lightweight and rapid as it requires approximately only 1s to be fully executed. In addition, we have made a comparison between our scheme and the existing ones based on their security properties, and shown that our proposed scheme outperforms the existing ones in terms of confidentiality, integrity, mutual authentication and session key generation with a lightweight computation overhead.**

***Index Terms*—user authentication, wireless sensor networks, WSNs, mutual authentication, session key agreement.**

I. Introduction

Securing Wireless Sensor Networks (WSNs) is a challenging task as it presents a hard environment with constrained resources [1]. In fact, the radio communication opens the door to attackers to intercept messages, insert false data or impersonate users. Thus, robust security mechanisms must be deployed in order to prevent illegal access of unauthorized parties. However, the limited size of sensor nodes implies other constraints such as limited energy and computation capabilities. Therefore, security mechanisms designed for WSNs must be lightweight and efficient [2].

One of the major important risks that faced WSNs is the illegal access of attackers to the data. Therefore, a robust user authentication scheme must be set up in order to prevent malicious entities from accessing the WSNs.

In this paper, we propose a lightweight user authentication scheme that provides mutual authentication and session-key agreement. The scheme is executed on both sides; the WSN's coordinator side playing the role of the server, and the user's device side acting as a client.

We have validated our scheme by two means; a security analysis in order to prove the robustness of our approach against classical types of attacks, and an implementation on a real WSN platform to validate the integration of such a scheme in resource-constrained sensor devices. We have demonstrated that our scheme is lightweight and has a moderate storage and computation overhead. The remainder of this paper is organized as follows. First, in Section II, we discuss related works on user authentication scheme in WSNs. Then, in Section III, we present the network and intruder model. In Section IV, we describe our proposed user authentication scheme. Then, we give a security analysis and a performance evaluation in Section V and VI, respectively. Section VII gives a comparison between our scheme and other proposed user authentication schemes. Finally, we conclude and provide future works.

II. Related work

Although user authentication in E-Commerce and M-Commerce applications has been deeply addressed, the problem of user authentication in WSNs was firstly identified, only in 2004, by Benenson *et al.* [3]. The several proposed mechanisms of user authentication in classical insecure networks cannot be used in WSNs as this type of networks presents different properties and new constraints. The limited power energy and computation capabilities render classical user authentication schemes impractical in WSNs. In addition, WSNs are generally deployed in a distributed environment, which makes it vulnerable to node compromise attacks (the attacker gains physical access to node, and therefore extracts all data and security parameters). Therefore, the verifier role (the entity that verifies the legality of users) must not be confined to a simple single node, as in classical user authentication schemes. As a result, the majority of proposed user authentication schemes in WSNs use the notion of threshold authentication: they divided the role of verifier to t (the threshold) sensors. In [4], Benenson *et al.* propose a user authentication scheme based on public key cryptography, which addresses the problem of node capture attacks. The scheme prevents unauthorized users from accessing data collected by sensor nodes even in the presence of node capture attacks. The

scheme is *t-out-n*, i.e. as the number of compromised node is less than *t* (where $t < n$, and n is the number of nodes in the communication range of the user) it remains secure. The process of authentication is as follows. First, the user broadcasts his identity and his certificate as a request. Then, each node in user's proximity sends a nonce to the user. This latter signs the nonce and sends it back to the former, which verifies the validity of the signed hash using user's certificate and the public key of the certification authority. User must be authenticated by *m* nodes in order to be allowed to post queries in the network. However, this scheme presents some drawbacks. First, it requires that each pair of node shares a secret key, which leads to high storage space and hence does not scale well. Second, the scheme allows querying only one node of the WSN. This node must be identified by nodes in user's proximity. The way to identify the target node is not presented in Benenson *et al.*'s solution, and this necessarily requires that each node has knowledge of the entire network. Third, the scheme does not address the case where the node responsible for processing the query is compromised and thus can send false information. Banerjee *et al.* [5] proposed a symmetric key-based user authentication scheme. Contrary to Benenson *et al.*'s scheme, in their solution a set of nodes replies to the user's query. The scheme is based on Blundo *et al.*'s techniques [6] for sharing pair-wise key. Sensor nodes involved in user's query generate a nonce and then the user must compute valid Message Authentication Codes (MACs) of this generated nonce using pair-wise keys shared with these sensor nodes. Each sensor node receiving a valid MAC replies to the user, otherwise it drops the request. However, Banerjee *et al.* did not mention how to determine the sensor nodes involved in the user's query. In addition, the scheme is vulnerable to node compromise and it does not provide mutual authentication. Jiang *et al.* [7], also proposed a distributed user authentication scheme based on the Self-Certified Keys cryptosystem (SCK), which they modified to use Elliptic Curve Cryptography (ECC). In their scheme, they assume the presence of a Key Distribution Center (KDC), which is responsible for generating a private/public key for each sensor node in the network and for users. When a user wishes to gain access, he first broadcasts his identity and the parameter *R* (used to compute the public key of user). Then, each node receiving this access request computes the pairwise key, shared with the user, using ECC and then send an encrypted nonce to the user. This latter must decrypt k nonces (where k is the threshold) in order to gain access to network. Tseng *et al.* [8] proposed an improvement of the dynamic user authentication scheme proposed by Wong *et al.* [9]. The improved scheme withstands the security weakness such as the replay attacks and the forgery attacks. In addition, it allows user to freely change his password. In Tseng *et al.*'s scheme, a user can login from any sensor node in the network. This sensor node will forward user's authentication message to a gateway node, which will verify the user authenticity. The registration phase is also made in the gateway node. However, the scheme cannot resist the node compromise attacks and it requires time synchronization between sensor nodes. Noting that time synchronization is a difficult task to achieve in WSNs. Chai *et al.* [10] proposed a threshold password authentication scheme, which meets both availability and strong security requirements in the mobile Ad-hoc networks. In their scheme, the secret is divided between *n* servers, and users must be authenticated with at least *t* servers in order to gain access to the network. This permits to avoid node compromise attacks.

In this work, we focus on the efficiency of the user authentication scheme and we do not address the node compromise attack. This is true for some specific applications. For example, in a biomedical context, sensor nodes are attached to patient and thus, the node compromise attack is no more possible. In addition, our proposed scheme provides mutual authentication and key establishment, which permits to maintain confidentiality and data integrity.

III. Network and intruder models

In this section, we present the network model and the intruder model.

A. Network model

We consider a WSN organized in a star topology, and is managed by a special sensor node called the coordinator. Sensor nodes can use, for example, the ZigBee/ 802.15.4 standard as a communication protocol. The coordinator is a sensor node that will play the role of relay between the user and the rest of the WSN: the user receives data through the coordinator and sends commands also through that coordinator. Thus, the coordinator represents the point of access to the WSN. Therefore, the access control process is implemented at the coordinator and also the authentication of users is made by the coordinator.

This kind of small sensor networks can be found, for instance, in health-care system. In fact, health-care systems use a body sensor network which consists of a set of sensor nodes attached to the body of the patient. These sensor nodes serve to monitor the physiological parameters of the patient such as heart rate, ECG, etc.

Roughly speaking, our WSN consists of two types of sensor node: the coordinator and the end devices. The coordinator is responsible for maintaining the network security, managing the network, responding to user queries and sending commands to end devices. However, end devices does not have the ability to interact with users or to treat queries. Their role consists in collecting data, sending them to the coordinator and executing commands received from this latter.

We assume also that the user is equipped with a mobile device (such as PDA, mobile phone, etc.) that allows him to wirelessly communicate (using the IEEE 802.15.4 standard for example) with the WSN coordinator.

Users can be present near the WSN (Figure 1) or can also remotely access to the WSN (Figure 2) using an infrastructure network such as (Wi-Fi, Internet, etc.). In the former case, the user's mobile device communicates directly with the coordinator. However, in the latter case the communication between

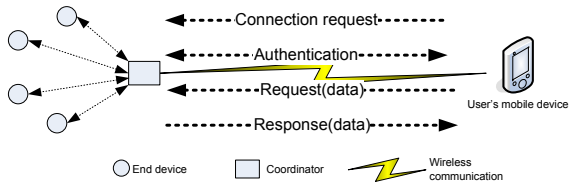

Figure 1. Direct access to WSN

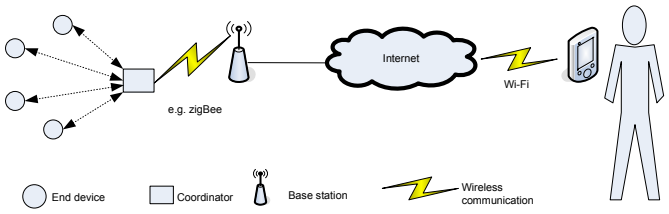

Figure 2. Remote access to WSN

the user's mobile device and the coordinator is remotely made through an infrastructure network.

Without loss of generality, our network can be modeled as follows (Figure 3):

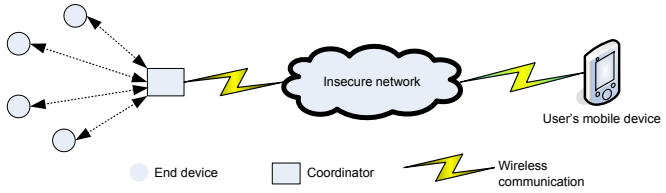

Figure 3. Network model

In both cases, the scenario of communication is as follows. First, the mobile device requests to establish connection with the WSN through the coordinator. Then, a set of authentication message exchanges will take place between the mobile device and the coordinator. If the authentication phase terminates successfully, a key is established between the user and the WSN and a secure communication can begin.

B. Intruder model

As information is exchanged over an insecure network, adversary can eavesdrop messages, replay old intercepted messages, forge messages, send false data to user or send false commands to the WSN. Also, the adversary can steal user's mobile device and/or any login material (e.g. smart card). The attacker can also possess a more computational material that can communicate with the WSN. However, we can consider that the coordinator is physically secure and accordingly the compromise attack, which is addressed in [3], [9] and [10], is not addressed in our case. We also assume that a memorized password is not revealed to a third party. We attempt to provide

Table I
NOTATION

Symbol	Meaning
x	The secret key of the system
$\|\|$	Concatenation
$\oplus$	exclusive-or (xor) operation
Nu	Nonce value of the user
Ns	Nonce value of the coordinator
$H()$	A one way hash function
$Enc(N,k)$	Encryption of the value N using the secret key k

authentication, confidentiality, and integrity. Authentication makes each party trust each other, and therefore we can control access to the WSN. Confidentiality assures that critical (confidential) data is not revealed to an unauthorized party. Integrity assures that data transmitted through the network is not modified. In this paper, we address the authentication problem and in what follows we present a user authentication scheme that allows to provide mutual authentication and key establishment. The established key can then be used to encrypt data (to maintain confidentiality) and to calculate a Message Integrity Code (MIC)(to maintain integrity).

IV. THE PROPOSED USER AUTHENTICATION SCHEME

In this section, we present the proposed user authentication scheme. The scheme consists of two phases. The registration phase, whereby we register legal users in the system for future access to the WSN. The login and authentication phase, which is executed by the WSN coordinator and the user, whenever this latter wishes to access to the WSN. For reader's convenience, we list the notations used in our scheme in Table I.

A. Security initialization

We assume that there is an administrator, which is responsible for loading necessary secret keys in the WSN and for registration of users.

First, the administrator chooses a secret key *x* and then loads the system server and the coordinator with this secret key *x*. The system server uses this secret key for registration of users. The coordinator uses this secret key in order to verify the authenticity of users.

B. Registration phase

When a new user wishes to register, he interacts with the system server. In order to avoid that the system administrator impersonates a user, we recommend that the user interacts directly with the server and does not reveal his password to the administrator. Thus, the role of the administrator in the registration phase is just allowing legal user to interact with the server. In order to register to the system, the user proceeds as depicted in Figure 4:

1) The user chooses an identity (*ID*) and a password (*Pw*), and then inputs them to the server.
2) The server computes $S = h(ID||x)$ and $A = Pw \oplus S$, and then registers A in the user's mobile device, where

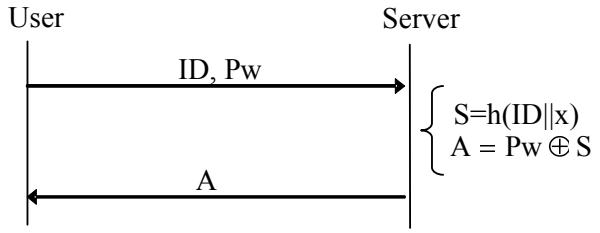

Figure 4. Registration Phase

h is a hash function, || denotes concatenation and $\oplus$ denotes exclusive-or (xor) operation.

C. Login and authentication process

Whenever the user wishes to communicate with the WSN, he must execute the authentication process, which is depicted in Figure 5. The authentication process is as follows:

1) The user inputs his memorized password in the login interface of his mobile device. Then, the user device computes the secret value *S,* using this introduced password and the saved parameters *A*, as $S = Pw \oplus A$. Then, the user device generates a nonce (random number used once) *Nu* , and then sends to the coordinator the message M1, containing the user identity and an encrypted value of the nonce *Nu,* as $M1 = \{ID, Enc(Nu, S)\}$. Where *Enc* is a symmetric encryption function, such as AES (Advanced Encryption Standard), and *Dec* is its associated decryption function. Note that the AES algorithm [11] is used in low-rate and low-power networks [12], [13], [14].
2) When receiving the message *M1*, the coordinator computes the secret value S as $S = h(ID||x)$ and then decrypts the message *M1* and extracts *Nu* as $Nu = Dec(Enc(Nu, S), S)$. Then, the coordinator generates *a nonce Ns* and sends $M2 = Enc(Nu||Ns, S)$ to the user device.
3) Upon receiving the message *M2*, the user device decrypts it and then extracts *Nu* and *Ns*. Then, the user device verifies that the received *Nu* is equal to the sent *Nu*. If the equality holds, the user computes $K = Nu \oplus Ns$ and then sends $M3 = Enc(Ns, K)$.
4) Upon receiving the message *M3*, the coordinator decrypts it and then extracts *Ns*. Then, the coordinator verifies that *Ns* received is equal to *Ns* sent. If the equality holds, the server trusts the user and allows him to communicate with the network.

The key *K* established between the user device and the WSN coordinator can be used as a master key.

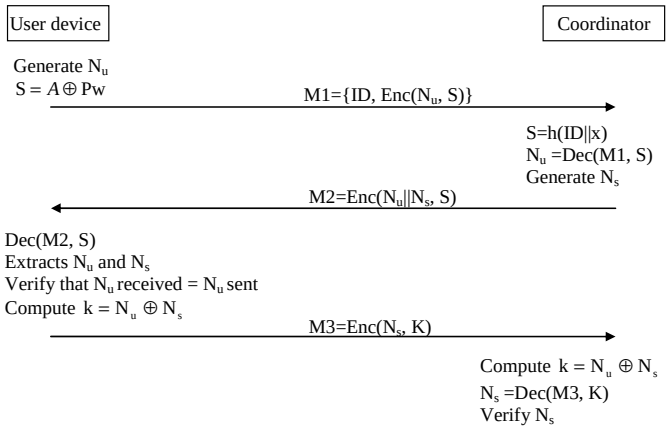

Figure 5. Login and authentication process

V. Security analysis

The security of the proposed scheme relies on the security of the secret key x. That is, the secret key x must be kept secret and not revealed to a third party, even legal users. In addition, the secret key x must be chosen appropriately to avoid guessing attack.

Before presenting the security properties of our scheme, we begin by describing the features of the used cryptographic tools. The proposed scheme use a one way hash function in order to compute the secret value *S*. A one way hash function has the following properties [15], [16]:

P1: It is infeasible to derive the value v from a given $H(v)$.

P2: It is also infeasible to find different value v and v' such that $H(v) = H(v')$.

Based on these properties, the proposed scheme is secure on the registration phase. In fact, the user can know the secret value S as $S = h(ID||x)$, however he cannot computes the secret key x.

Moreover, the encryption function has the following property:

P3: it is infeasible to decrypt an encrypted value $Enc(v, k)$ without knowing the secret key k.

That is, the proposed scheme is secure on the login and authentication phase. In fact, an intruder can intercept the exchanged messages, however, he cannot reveals the nonce value neither the used keys.

In addition, our proposed user authentication scheme is robust against the following attacks:

- Impersonation attack: an attacker cannot impersonate a legal user and he will be blocked in message *M3*. As the attacker does not know the secret *S*, he cannot extract *Ns* from *M2* and as a result he cannot compute an acceptable message *M3*.
- Replay attack: suppose an attacker intercepts a previous message exchange of a legal user and he tries to replay it in order to impersonate the user or the coordinator, the attacker cannot succeed in impersonating the user (respectively the coordinator) as he cannot extract the new value of *Ns* (respectively *Nu*).
- User's mobile device stolen attack: when an attacker gains

access to a user's mobile device, he can reveal the value of *A*. However, as he does not know the value of the password, he cannot compute *S* and therefore cannot compute a correct message.
- Guessing attack: guessing attack on *S* or *K* is not possible in our scheme as the value of the plaintext of the encrypted quantity (i.e *Nu*, *Ns*) is not transmitted. The attacker has access only to the encrypted value and as a result he cannot launch a guessing attack.

In addition, to the robustness against the bellow attacks, the proposed scheme presents the following advantages:

- Mutual authentication: in our scheme not only users authenticate themselves before accessing data, but also the coordinator proves its authenticity. This allows to avoid impersonating the coordinator as an attacker can impersonate the coordinator in order to send false data to users. Therefore, users are sure about the authenticity of the received data.
- Session key agreement: in our scheme, at the end of a successful authentication the user and the coordinator establish a secret key. This key can be used as a session key in order to secure the communication between the two entities (the user and the coordinator).
- Synchronization independence: Some proposed scheme (such as [9], [8]) avoid replay attack by adding a time-stamp to every sent message. This time-stamp guarantees that the message is fresh and therefore is not an old replayed message. However, one of the disadvantages of using a time-stamp is that it requires synchronization between entities. In our scheme we provides this security service (freshness of message) by using the concept of nonce and therefore synchronization between entities is not required.

VI. Performance evaluation

In order to evaluate our user authentication scheme, we have implemented it in real world using TelosB motes [17]. TelosB mote has a 8 MHz microcontroller, 10 Kbytes of RAM memory, and 48 Kbytes of ROM memory.

We have used two motes: one containing the code of the user device and the other containing the code of the coordinator. The program was written in nesC language [18] in order to be supported by TinyOS operating system [19]. Performance of the proposed user authentication scheme is shown in Table III.

Table III
Performance evaluation of the proposed scheme

	The user device	The coordinator
ROM consumption (in bytes)	26274	25832
RAM consumption (in bytes)	2870	2852
Time of execution of authentication process (in ms)	1250	1006

We choose the AES algorithm [11] as a cipher function. The length of nonce used is 16 bytes. The average Time of execution of Cipher is 237 ms. We note that the authentication process is rapid as it requires approximately 1s to execute. In terms of memory consumption, our scheme needs about 26 KB of ROM, which represents about 50% of available ROM memory, and less than 3 KB of RAM memory, which represents about 30% of available RAM memory.

VII. Comparison to other schemes

In Table II we have made a comparison between our scheme and those proposed in the literature (Section II). Our scheme provides mutual authentication (the user as well as the WSN is authenticated: each one to the other). However, Benenson et al. [4], Banerjee et al. [5], Jiang et al. [7], and Tseng et al. [8] schemes do not provide mutual authentication. This can lead to sending false data to users or intercepting confidential data of users by compromised nodes. Our scheme also permits to establish session key between the user and the WSN. However, other ones do not permit to establish a session key. Moreover, in our scheme data integrity and confidentiality are maintained in contrast to other schemes. In terms of infrastructure, our scheme does not require any infrastructure in contrast to Benenson et al.'s [4] and Jiang et al.'s [7] which require a Public Key Infrastructure (PKI) in order to provide a private/public key for each node. As a consequence, our scheme is scalable. Moreover, Benenson et al. [4] is vulnerable to DoS attacks by broadcasting several bogus certificates. Also Jiang et al.'s scheme [7] is vulnerable to node capture attacks and it also requires synchronization between nodes.

In terms of efficiency, we have made a comparison between our scheme and Benenson et al.'s one. This choice is made because it is the only implemented scheme. Moreover, it is implemented in the same language as our scheme. Benenson et al.'s scheme requires three rounds.Whereas, our scheme requires only two rounds. In addition, in Benenson et al.'s scheme, the user communication overhead is high (1+m messages, where m is the number of user's node neighbors) compared to our scheme (2 messages). Also, the computation overhead in Benenson et al.'s scheme is high as the user must calculate m signatures and each sensor two signatures. Whereas, our scheme necessitates only three AES executions. Finally, the execution time of our scheme is only 1s, however Benenson et al.'s scheme requires 440s.

VIII. Conclusion

In this paper, we proposed a user authentication scheme adapted to resource constrained WSNs. The security of the scheme is based on a password memorized by the user and a secret key saved in his device. Thus, the proposed scheme does not require any infrastructure and it is also lightweight and rapid in execution as demonstrated in Section VI. As shown in Table II, our scheme outperforms security properties of existing solution as it maintains confidentiality and integrity. In addition, our scheme allows to establish a session key. The efficiency comparison with Benenson *et al.*'s scheme (Table IV) proves that our solution is lightweight and consumes less computation and communication overhead. In the future, we plan to deploy the proposed scheme in a real context using medical WSN and PDA as a user device. Furthermore, we

Table II
COMPARISON OF SECURITY PROPERTIES

	Benenson et al.[4]	Banerjee et al. [5]	Jiang et al. [7]	Tseng et al [8]	Chai et al. [10]	Our Scheme
Authentication	Unilateral	Unilateral	Unilateral	Unilateral	Mutual	Mutual
Session key agreement	No	No	No	No	No	Yes
Data integrity	Not maintained	Not maintained	Not maintained	Not maintained	Not maintained	Maintained
Confidentiality	Not maintained	Not maintained	Not maintained	Not maintained	Not maintained	Maintained
Cryptographic techniques	PKI with ECC	Symmetric cryptography based on Blundo et al.'s techniques	Based on the self-certified keys cryptosystem (SCK) and ECC	Hash and XOR	Threshold cryptography (Shamir)	Encryption and XOR
Infrastructure	PKI The CA could be the BS	No	KDC for providing private/public key	No	No	No
Scalability	Yes	No (due to Blundo)	Yes	Yes	Yes	Yes
Target of the query	Single sensor	Set of sensors	Set of sensors	Single sensor	Set of sensors	The coordinator
Vulnerabilities	Possibility of DoS attacks by broadcasting several bogus certificates	Computation and communication overhead	Computation and communication overhead	Require synchronization between nodes	Require synchronization between nodes	-
Robustness	Avoids node capture attack	Avoids node capture attack	Avoids node capture attack	Efficiency	Avoids node capture attack	Efficiency

plan to optimize the implementation of our code, especially, the implementation of the AES algorithm, which is actually implemented with C language and we will convert it to nesC.

REFERENCES

[1] N. Li, N. Zhang, S. K. Das, and B. Thuraisingham, "Privacy preservation in wireless sensor networks: A state-of-the-art survey," *Ad Hoc Networks*, vol. 7, no. 8, pp. 1501 – 1514, 2009.

[2] M. Sharifi, S. S. Kashi, and S. P. Ardakani, "Lap: A lightweight authentication protocol for smart dust wireless sensor networks," *International Symposium on Collaborative Technologies and Systems*, pp. 258–265, 2009.

[3] Z. Benenson, F. Gartner, and D. Kesdogan, "User authentication in sensor networks, (extended abstract)," *In Informatik 2004, Workshop on Sensor Networks*, 2004.

[4] Z. Benenson, N. Gedicke, and O. Raivio, "Realizing robust user authentication in sensor networks," *Workshop on Real-World Wireless Sensor Networks (REALWSN 2005)*, Jun 2005.

[5] S. Banerjee and D. Mukhopadhyay, "Symmetric key based authentication querying in wireless sensor networks," in *in Proceedings of the First International Conference on Integrated Internet Ad Hoc and Sensor Networks*, 2006.

[6] C. Blundo, A. D. Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly-secure key distribution for dynamic conferences," in *in Advances in Cryptology CRYPTO 92*, ser. LNCS 740, 1993, pp. 471–486.

[7] C. Jiang, B. Li, and H. Xu, "An efficient scheme for user authentication in wireless sensor networks," in *21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07)*, 2007.

[8] H.-R. Tseng, R.-H. Jan, and W. Yang, "An improved dynamic user authentication scheme for wireless sensor networks."

[9] Y. Z. K. H. M., J. Cao, and S. Wang, "A dynamic user authentication scheme for wireless sensor networks," in *In Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC '06)*, 2006.

[10] Z. Chai and a. R. L. Zhenfu Cao, "Threshold password authentication against guessing attacks in ad hoc networks," *Ad Hoc Networks*, vol. 5, 2007.

[11] "National institute of standards and technology (nist), advanced encryption standard (aes)." Federal Information Processing Standards Publications (FIPS PUBS) 197, 2001.

[12] S. Didla, A. Ault, and S. Bagchi, "Optimizing aes for embedded devices and wireless sensor networks," in *TridentCom '08: Proceedings of the 4th International Conference on Testbeds and research infrastructures for the development of networks & communities*. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, pp. 1–10.

[13] A. J. Elbirt, "Accelerated aes implementations via generalized instruction set extensions," *J. Comput. Secur.*, vol. 16, no. 3, pp. 265–288, 2008.

[14] L. Huai, X. Zou, Z. Liu, and Y. Han, "An energy-efficient aes-ccm implementation for ieee802.15.4 wireless sensor networks," in *NSWCTC '09: Proceedings of the 2009 International Conference on Networks Security, Wireless Communications and Trusted Computing*. Washington, DC, USA: IEEE Computer Society, 2009, pp. 394–397.

[15] C. Cid, "Recent developments in cryptographic hash functions: Security implications and future directions," *Information Security Technical Report*, vol. 11, no. 2, pp. 100 – 107, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/B6VJC-4K4H47T-7/2/501a7fb3b6cf98383eddb6b3dc1ecf05

[16] Y. Zheng, T. Matsumoto, and H. Imai, "Structural properties of one-way hash functions," in *CRYPTO '90: Proceedings of the 10th Annual International Cryptology Conference on Advances in Cryptology*. London, UK: Springer-Verlag, 1991, pp. 285–302.

[17] Telos Mote: www.ece.osu.edu/ bibyk/ee582/telosMote.pdf.

[18] NesC: A Programming Language for Deeply Networked Systems. http://nescc.sourceforge.net/, 2009.

[19] TinyOS: http://www.tinyos.net/, 2009.

Table IV
EFFICIENCY COMPARISON

	Nb of rounds	Nb of messages	Communication overhead	Computation overhead	Storage requirement	Code size	Execution time
Benenson *et al.*[4]	3	User:1+m Sensor: 1	User: 4*163-bits + m messages of size 2* 160 bits Sensor: Size of nonce	User: m signature calculation Sensor: two signature verification	User: certU 163 bits is the size of a private key. Public keys consist of two 163-bit numbers Sensor: the public key of the CA	45,5 KB of ROM and 2,0 KB of RAM.	440 s
Our scheme	2	User: 2 Sensor: 1	User: 2 * 32 B Sensor: 1*32 B	3 AES execution (Both user and sensor)	Secret key	26 KB of ROM and 2,8 KB of RAM.	1s