DCOSS ´09

International Conference on Distributed Computing in Sensor Systems

June 8 – 10, 2009

Marina Del Rey, California

Adjunct workshop proceedings

LOCALGOS, IWSNE, WITS

RWSN, DWKDSS, RWI

Preface

This Proceedings Volume contains the papers of the following Workshops, that were held together with the 5th IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS '09), which took place at Marina Del Rey, California, USA, during Monday, June 8 - Wednesday, June 10, 2009:

- Third International Workshop on Localized Algorithms and Protocols for Wireless Sensor Networks (LOCALGOS)
Chairs: WenZhan Song, Washington State University, Vancouver (PC Co-Chair)
Hongyi Wu, University of Louisiana at Lafayette (PC Co-Chair)
Ivan Stojmenovic, University of Ottawa, Canada (SC Chair)

- Second International Workshop on Sensor Network Engineering (IWSNE)
Co-Chairs: Stefan Fischer, University of Luebeck, Germany
Dennis Pfisterer, University of Luebeck, Germany

- Third International Workshop on Information Theory for Sensor Networks (WITS)
Chairs: Joao Barros, Universidade do Porto, Portugal
Sandeep Pradhan, University of Michigan, USA

- International Workshop on Robotic Wireless Sensor Networks
Chairs: Andreas Terzis, Johns Hopkins University, USA
Volkan Isler, University of Minnesota, USA

- First International Workshop on Data Warehousing and Knowledge Discovery from Sensors and Streams
Chair: Alfredo Cuzzocrea, University of Calabria, Italy

Sotiris Nikoletseas
DCOSS '09 Workshops Chair

Two Tier User Authentication Scheme for Heterogeneous Sensor Networks

Le Xuan Hung
Dept. of Computer Engineering
Kyung Hee University
Korea, 446-701
Email: lxhung@oslab.khu.ac.kr

Sungyoung Lee
Dept. of Computer Engineering
Kyung Hee University
Korea, 446-701
Email: sylee@oslab.khu.ac.kr

Young-Koo Lee
Dept. of Computer Engineering
Kyung Hee University
Korea, 446-701
Email: yklee@khu.ac.kr

***Abstract*—For many sensor network applications such as military or homeland security, sensed data is critical that only legitimate users should be allowed to access. Therefore, a user authentication scheme is essential. Existing solutions for computers and ad hoc wireless networks are not suitable because they are too heavy to deploy on resource-constraint sensor nodes. In this paper, we propose a Two Tier User Authentication scheme (TTUA) for heterogeneous sensor networks. Our analysis and simulation results have shown that TTUA is more secure and energy-efficient than existing approaches.**

***Index Terms*—heterogeneous sensor networks, security, user authentication, energy efficient, distributed.**

I. Introduction

Sensor networks have many applications, such as military, homeland security, environment surveillance, manufacturing. Basically, sensed data is collected from the sensor network and presented to users either upon inquiries or upon event detection. For many sensor network applications, the sensed data is critical so that it should be presented only to 'legitimate' users. Therefore, an user authentication is very essential.

Due to limitations in power, communication, and computation capacities, applying conventional public key approaches such as RSA is not feasible because they are too heavy for sensor nodes. Symmetric key-based schemes are more suitable. The simplest way is to authenticate the user at the base station (BS). However, this is not an efficient way because it requires considerable communication cost and is more time consuming. Recently, several attempts have been made to provide user authentication schemes for sensor networks [5]-[8][14]. However, they are not energy-efficient and have some drawbacks. For example, the public key approach based on Eclipse Curve Cryptography (ECC) [5] consumes much more energy and computational time than symmetric key-based schemes, approximately 80-fold in energy consumption and 143-fold in computational time; the *n-authentication* [8] requires many nodes to involve in authentication and communication, leading to high energy consumption and high communication overhead. Those limitations will be discussed in more details in Section II. Most existing work on sensor networks considers homogeneous sensor networks where all sensors have the same capacities in communication, computation, memory storage, power supply, etc. However, it has been shown that homogeneous sensor networks have poor performance and many limitations [9][10]. Recently, Heterogeneous Sensor Networks (HSNs) have become popular, particularly in real deployments because of their potential to increase network lifetime and reliability without significantly increasing the cost [15]. For example, Intel has deployed a pilot application of sensor networks to monitor the health of mechanical equipment in its fabrication plants [15]. In general, a HSN is composed of a large number of homogeneous sensors (e.g. MICA2) along with a small number of additional powerful Personal Digital Assistants (PDAs). Several security studies have explored heterogeneous sensor networks [11][12] to achieve better security and efficiency.

This paper presents our proposed scheme, Two-Tier User Authentication (TTUA), for HSNs. Our contributions are two-fold. First, TTUA is a distributed user authentication scheme. It does not require the involvement of a BS or any center during authenticating process. Second, an careful analysis and simulation are given along with comparison with an existing approach. we analyze and simulate our scheme and compared with existing approaches. Both analysis and simulation results show that TTUA is more secure and significantly reduces energy and time.

The remainder of the paper is organized as following. In Section II we briefly review related work and discuss about their limitations. The system model and assumptions are described in Section III. In Section IV, we present our proposed scheme. The advantages of our scheme are shown through mathematical analysis (Section V) and performance evaluation (Section VI). Finally, Section VII concludes the paper and outlines our future works.

II. Related work

Recently, several studies on user authentication for sensor networks have been proposed. In general, they can be classified

[1] This research was supported by the MKE (Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center)support program supervised by the IITA (Institute of Information Technology Advancement)(IITA-2009-(C1090-0902- 0002)). Also, this work was supported by the Korea Science and Engineering Foundation(KOSEF) grant funded by the Korea government(MOST) (No. 2008-1342), and partially supported by the Brain Korea 21 projects.

Dr. Sungyoung Lee is the corresponding author.

[31] two types: one is based on public key cryptography [12][14], and the other is based on symmetric key cryptography [6][7].

In public key-based approaches, Haodong et al. proposed ECC-based access control protocol [5]. Benenson et al. proposed an algorithmic framework for robust authentication and access control in sensor networks [8] which can withstand capture of up to t nodes. Benenson. et al realized robust user authentication [14] which let the sensors in the communication range of the user serve as interpreters between the "public key crypto world" of the sensor network. Though public key cryptography has been proofed feasible on sensors, it is much more resource consuming than symmetric key-based approaches. For example, the ECC-based scheme [5] consumes much energy and computational time than symmetric key-based schemes, approximately 80-fold in energy consumption and 143-fold in computational time.

Few works based on symmetric key cryptography have been proposed. Wong et al. proposed a dynamic user authentication protocol for sensor networks which imposes very light computational load and requires simple operations [6]. However, it is vulnerable from replay and forgery attacks. Also, passwords could be revealed by any of the sensor nodes. Therefore, H.R.Tseng, R.H.Jan, and W.Yang [7] improved Wong et al.'s scheme which not only retains all the advantages in the original scheme but also enhances its security. We name it TJY. However, the scheme relies on a centralized gateway-node to perform authenticating process which introduces high communication overhead and prolongs response time. On the other hand, the message ACC_LOGIN between the gateway-node and the login-node is not protected. As a consequence, an adversary could easily modify this message to fool the login-node that she is an authenticated one.

III. System and adversarial model

We consider a large-scale HSN with two types of nodes: a large number of ordinary sensors (e.g. MICA2), and a small number of powerful sensors (called cluster heads - *CHs*). Ordinary sensors (for short, we call *sensors*) have a short transmission range, and a small power supply. Due to resource limitation, sensors are not equipped with any tamper-resistant hardware and they are susceptible to node capture attacks. In contrast, *CHs* have more energy supply, longer transmission range, higher data rate than ordinary sensors. Especially, *CHs* are equipped with tamper-resistant hardware. This assumption is reasonable because the number of cluster heads in an HSN is relatively small (e.g., 20 cluster heads for 1,000 sensors) so the cost of tamper-resistant hardware is small [11]. We assume that both sensors and *CHs* are uniformly distributed in the sensor area. After deployment, sensors are grouped in clusters. Each cluster maintains one *CH*. The *CHs* form a backbone in the network so that sensed data after being collected will be transmitted through *CHs* towards the requesting users (see Fig.1). A user may user a powerful computing device, such as a PDA, mobile phone, or laptop, to interact with the sensor network.

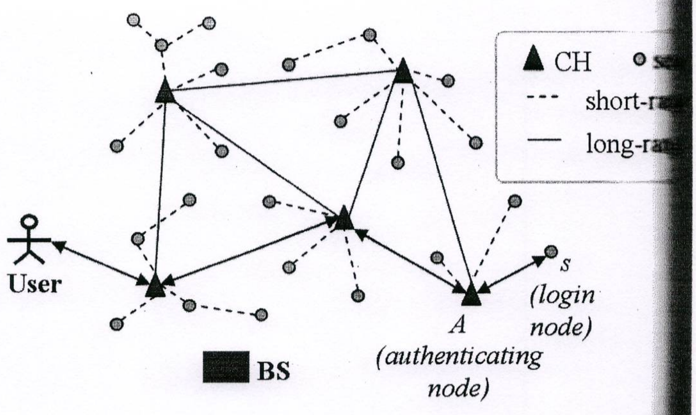

Fig. 1. Our system model

The sensor network is managed by a base station (BS), which is responsible for generating all security primitives. For the sake of simplicity, we assume that the BS also acts as a key distribution center. A key pre-distribution operation is carried out by the BS right after the network deployment stage. We can assume that a key pre-distribution scheme [13] is used to distribute symmetric keys to all nodes. The BS maintains a list of pair-wise keys K_i with each CH_i. Also, each CH_i shares a pair-wise key with its member sensor.

An adversary may make use of all possible means to authenticate as a legitimate user. She can capture a small amount of sensor nodes, read out their memory contents to get user authentication information. Through these compromised sensors, the adversary could not only carry out eavesdropping to extract transmitted information, by also could replay some messages to destroy the regular authentication process of a legitimate user. Even more, an authorized user can collude with the compromised sensors to cheat the entire sensor network that he is an authorized one. Also, the adversary can use a powerful device to perform *Denial-of-Service* (DoS) attacks by sending a large number of authentication requests to a particular node or the whole sensor network.

IV. Two Tier User Authentication

The Two Tier User Authentication scheme (TTUA) allows a user to register once and authenticate to the network many times. He also can change the password at will. The sequence diagram of TTUA is illustrated in Fig.2. TTUA includes three phases: user registration, user authentication, and password change. In the user registration phase, a user comes to the BS and applies for relevant secret information. In user authentication phase, he uses this information to authenticate a sensor. The user may change his password to protect his authentication account. We discuss each step in TTUA as follows.

A. User registration

Initially, the user comes to the BS to register his ID and password and applies for relevant secret information to his devices via a secure channel. He sends his ID (*UID*) and password (*PW*) to the BS. To make password confidential

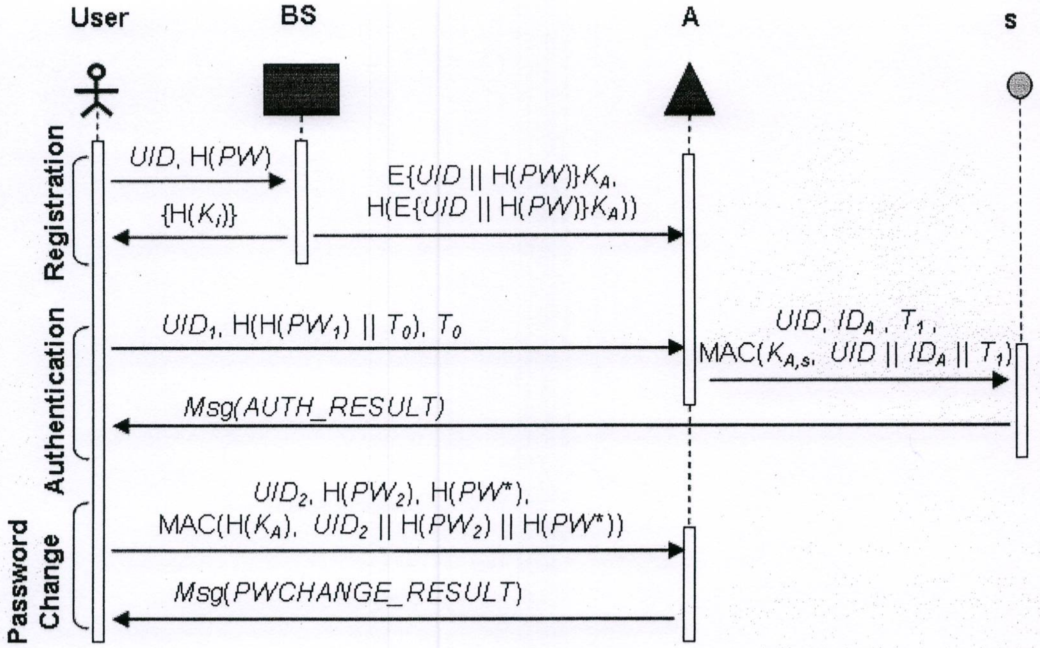

Fig. 2. The proposed scheme

it is needed to hash the password using some hash function such as SHA-1 [16] before sending to the BS. Doing so helps to protect the password from being revealed to anyone, even the BS's administrator. The BS sends a list of pair-wise keys with each CH_i to the user's devices K_i. These pair-wise keys are derived from the pair-wise keys shared between BS and eachCH_i by using a hash function.

$user \rightarrow BS : UID, H(PW)$

$BS \rightarrow user : list\{H(K_i)\}$

The BS then broadcasts *UID* and H(*PW*) to all *CHs* in an encrypted form.

$BS \rightarrow CH_i : E\{UID, H(PW)\}K_i, H(E\{UID\|H(PW)\}K_i$

where E{M}K is an symmetric encryption using a secret key K, and || means concatenation.

After verifying the message using its own share pair-wise key K_i, each CH_i stores UID and H(PW) in its database.

B. User authentication

Notice that *CHs* form a backbone so that all communications within the network are relayed through it. Suppose a user wants to access data stored at a sensor *s* (assume *A* is the cluster head of *s*). The authentication process includes the following steps:

Step 1)At first, the user sends his ID (UID_1) and his hashed password $H(PW_1)$ to A as follows:
$user \rightarrow A : UID_1, H(H(PW_1) \oplus T_0), T_0$
where T_0 is a current timestamp used to defend against replay attacks, and ⊕ means XOR operation.

At first, the user sends his ID (UID_1) and his hashed password H(PW_1) to A. A first checks if the timestamp T_0 is valid ($T_0 > T_{now}$, where T_{now} is current time) and UID_1 exists in its storage. If yes, then it verifies the password H(PW_1) by using H(PW) stored in its database.

A checks:
(1) $T_0 > T_{now}$, $UID_1 = UID$
(2) H(H(PW_1) $\oplus T_0$) = H(H(PW) $\oplus T_0$)

Step2) If the verification is successful, A sends *UID*, its ID (ID_A), and new timestamp (T_1) along with a MAC using its share pair-wise key ($K_{A,s}$) with the sensor *s*.

$A \rightarrow s : UID, ID_A, T_1,$
MAC($K_{A,s}, UID\|ID_A\|T_1$)

Upon receiving the message, *s* first checks if T_1 is valid. If yes, it verifies UID and ID_A by generating a MAC with the shared pair-wise key with A ($K_{A,s}$) and comparing with received MAC. If all of these are successful, then the user is authentic.

After successful authentication, sensor *s* is ready to send data to the user. *s* may send a short message to inform user that he is authenticated.

C. Password change

TTUA allows users to change their password at will. He can make it locally via a nearby CH. He can make it done at any location within the network. To do this, he broadcasts his current ID (UID_2) and hashed password H(PW_2), along with a new hashed password H(PW^*) to all CHs.

$user \rightarrow \{CH_i\}$: $broadcast\{UID_2$, H(PW_2) , H(PW^*) , MAC(H(K_i),$UID_2\|$ H(PW_2)$\|$ H(PW^*))}

Each receiving CH_i computes MAC value and ensures that UID_2, H(PW_2), H(PW^*) are not modified. After that, it verifies if the current password is valid by comparing H(PW_2) with the current hashed password H(PW) in its database. If matched, then it replaces the current hashed password with the new one. After that, the node which receives directly the message from the user may send a message to the user to inform the password change process is successful.

V. Security analysis

In this section, we analyze security of TTUA. TTUA can defend against typical attack on sensor network authentication as presented in [1]-[4] including node compromised attacks, replay attacks, impersonate attacks, and DoS. In the following, we discuss how TTUA can defend against these attacks.

- **Secure against node compromise**: In our scheme, *CHs* are equipped with tamper-resistant hardware, so they cannot be compromised. The keys and user information stored on CHs cannot be disclosed to anyone. Therefore, the adversary cannot make use of the CH to convince the sensor *s* that she is a legitimate user. If the adversary compromise *s*, what she can get is a pair-wise shared key $K_{A,s}$ with the *CH*. However, it is not possible for her to use $K_{A,s}$ to authenticate herself to the sensor network because the key is shared between *CH* and *s* only.
- **Secure against replay attacks**: In TTUA, an adversary cannot use the previous message to login successfully. In the authentication phase, the password and a timestamp are hashed before being sent to the *CH*. If the adversary intercepts the message $(UID_1, H(H(PW_1) \oplus T_0), T_0)$ and reuses it somewhere else, the CH can detect by checking the timestamp T_0. If $T_0 < T_{now}$, the *CH* knows that the message has already been used. Thus, replay attacks are not possible.
- **Secure against impersonate attacks**: The proposed scheme can resist against impersonate attacks. That is he cannot impersonate legitimate user even he intercepts any message on communication. There are three cases as follows:
  - The adversary intercepts the message (UID_1, $H(H(PW_1)$ $\oplus T_0)$, T_0) between the user and the *CH*, and attempts to change UID_1 with her own UID^*. The *CH* can easily detects because UID^* does not exist in its database.
  - With the above message, the adversary attempts to get the password from $H(H(PW_1)$ $\oplus T_0)$. However, it is considered practically impossible for her to derive the password from the hashed value.
  - The adversary intercepts the message UID, ID_A, T_1, $MAC(K_{A,s}$, $UID\|$ $ID_A\|T_1)$ between the *CH* and the sensor *s*. She then attempts to modify UID in the message by her own UID^*. However, *s* can detect this by building the MAC from UID^*, ID_A, T_1) and comparing with the received one. It is obvious MAC($K_{A,s}, UID\|ID_A\|T_1$) $\neq$ MAC($K_{A,s}, UID^*\|$ $ID_A\|T_1$). Therefore, it is not possible for the adversary to impersonate a legitimate user.
- **Secure against *Denial-of-Service* (DoS) attacks**: There are two cases for an adversary to attempt DoS attacks to the sensor network:
  - She can use a powerful device such as a laptop to send many authentication requests to the sensor network. TTUA can mitigate the attacks by first check if user ID exists in the *CH*'s database. If no, then it will not compute the hashing value of the password.
  - If she intercepts the broadcasting message between the BS and the *CH* $(UID, H(PW)$, MAC(K_i, $UID\|H(PW)$), she may get user id and the hashed password. She then instantly creates UID,$H(H(PW)\|T_0)$,T_0 and requests authentication to *CHs*. In this case, the CH has to instantly compute hash values to verify the request. If the CH cannot handle such request storms, DoS is possible. To solve this problem, we set a time-out period for requests of each user ID. If the adversary sends a large amount number of requests with the same user ID within a short period of time, the *CH* will not accept and process those requests. Therefore, DoS is not possible at the CH. On the other hand, since the request is not passed at the *CH*, the *CH* would not build a MAC to send to the sensor *s*. As a consequence, DoS is not possible at the sensor *s*.

VI. Performance evaluation

To the best of our knowledge, TTUA is the first user authentication scheme for HSNs. Therefore, it is not possible to compare it with another scheme for HSNs. Instead, we compare TTUA with an existing user authentication scheme for homogeneous sensor networks in order to show the advantages of TTUA and the benefit of using HSNs. We select TJY [7] as it is one of the most efficient schemes that we have surveyed. We start with an analysis-based evaluation, and then simulation-based evaluation.

A. Analysis

We analyze performance of TTUA in terms of the storage requirement, communication overhead, and computation cost.

- **Storage**: In TTUA, the cluster head CH has to store user IDs and hashed password values. Supposed there are n user, user ID size is 8 bytes, hashed password value is 20-byte long. Thus, CH has to store 28×n bytes. In addition, each CH has to store secret keys $\{K_{A,s}\}$. The probability of CHs in a certain area must be sufficient so that the whole area is covered. This problem was well-investigated in [11] based on *Vapnik-Chervonenkis* (VC) theorem. So

TABLE I
COMPARISON OF MEMORY STORAGE(FOR 10 USERS)

Node type	TTUA	TJY
CH	400 bytes	0
s	0	100 bytes

TABLE II
COMPARISON OF COMPUTATIONAL COST

Phase	Node	TTUA	TJY
Registration	CH	$1T_E + T_H$	0
	s	0	$1T_H$
Authentication	CH	$1T_H$ + $3T_{MAC}$	0
	s	$1T_{MAC}$	$2T_H$ + $2T_{XOR}$
Total	CH	$2T_H$ + $3T_{MAC} + 1T_E$	0
	s	$1T_{MAC}$	$3T_H$ + $2T_{XOR}$

there are approximately 15 sensors $\{s\}$ connecting to one CH. Each secret key is 64 bits (8 bytes) long. Therefore, each CH stores 15×8=120 bytes of secret keys. Totally, each CH must store 28×n + 120 bytes. In TJY, the user does not need to store any secret information, but each sensor has to store 8 bytes for a user ID and 2 bytes for a timestamp. Thus it shall cost 10×n bytes of the storage. Assume n=10, the memory required is shown in Table I.

The CH such as iPAQ has 64MB memory storage. If we assume the storage is reserved for maintaining user IDs and passwords only, then TTUA can maintains approximately 2,340 users. Meanwhile, the sensor (e.g. TelosB) has 1Mb memory storage. Thus TJY scheme can maintain 102 users only. This means that if the memory storage is only used for storing user's information, TTUA can maintain 2,340 users, while TJY. can only maintain 102 users (Fig. 3).

- **Computation**: We define T_{MAC} , T_H , T_E, and T_{XOR} are computation cost of performing message authentication code (CBC-MAC), hash function (SHA-1), symmetric encryption, and XOR operations, respectively. To make the comparison with TJY. scheme fairly, we consider the computation cost on both CH and login-node s in TTUA; for TJY, we only take into account of the login-node, regardless the gateway-node because it is a very powerful node such as a PC. The computation cost are summarized in Table II

  We can see that our scheme requires an additional computation cost on CH ($2T_H + 3T_{MAC}+T_E$). However this computation cost is very small compared with that of sensors because CH is a powerful sensor node. The total computational cost for the sensor s in our scheme is smaller (1 T_{MAC}) than TJY ($3T_H$ + $2T_{XOR}$). According to practical results on real sensor devices [16], one MAC costs 7.1 ms, while a SHA-1 costs 3.5 ms. We summarize the computation cost through an numerical comparison in Fig. 4. The figure shows that our computation cost on sensors is less than TJY scheme.

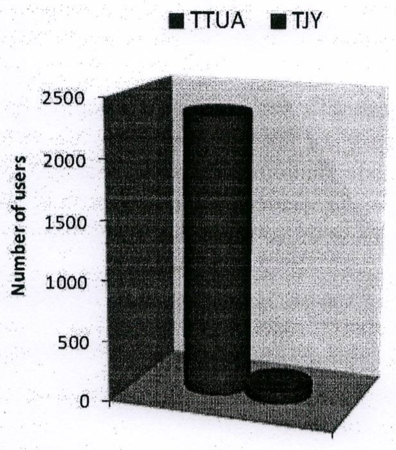

Fig. 3. Comparison of the number of users

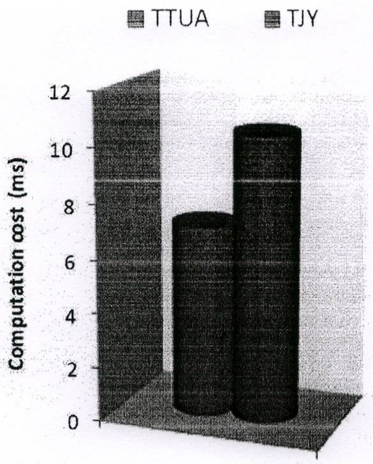

Fig. 4. Comparison of computational cost

- **Communication**: To calculate computation cost, we define a number of notations as follows (all of these are in number of hops).
  - $C_{broadcast}$: Communication cost for broadcasting user ID and password to all *CHs*
  - C_{U-A}: Communication cost between the user and the cluster head *A*
  - C_{U-s}: Communication cost between the user and the sensor *s*
  - C_{A-s}: Communication cost between *A* and *s*
  - C_{s-GW}: Communication cost between *s* and the gate-way node (in TJY)
  - C_{U-GW}: Communication cost between the user and the gate-way node (in TJY)

Note that C_{U-s} = C_{U-A} + C_{A-s} because any message sent to *s* must be relayed through its cluster head *A*. The communication cost is summarized in Table III.

In TTUA, it requires an extra communication cost for broadcasting user IDs and passwords to all *CHs*. This is insignificant cost for the powerful *CHs*. In authentication phase, we can see that TTUA only requires commu-

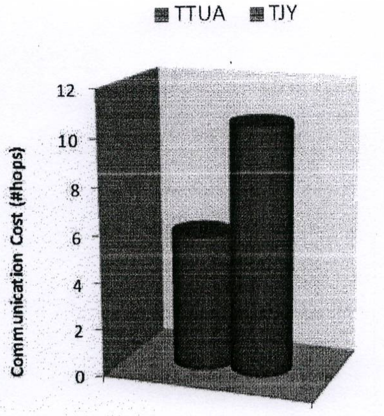

Fig. 5. Comparison of communicational cost

TABLE III
COMPARISON OF COMMUNICATION COST

Phase	TTUA	TJY
Registration	$C_{broadcast}$	$2C_{U-GW}$ + C_{s-GW}
Authentication	$2C_{U-s}$	$2C_{U-s}$ + $2C_{s-GW}$

nication between the user and the sensor s ($2C_{U-s}$). Meanwhile, in TJY, every authentication process requires extra communication with the gate-way node ($2C_{U-s}$ + $2C_{s-GW}$). Practically, how much the communication costs depending on physical distance between the login-node and the gate-way. The further distance is, the more it costs.

To illustrate this in a straightforward manner, we perform a number of experiments (see Section VI-B) to measure the communication cost. The user's location and the login-sensor are randomly changed. The gate-way node is located at the center of the sensor field. The result is shown in Fig. 5. It shows that the communication cost of TTUA is about twice less than that of TJY.

B. Simulation

This section describes our simulation result and a comparison with TJY. The simulation result shows the average energy consumption and delay time of different network topologies. For each network topology, user's location and the login-node are randomly changed within the sensor field.

a) Simulation model

We simulated the TTUA on SENSE simulator (*Sensor Network Simulator and Emulator*) [20]. For comparison, we also simulated TJY with the same network topologies and authentication scenarios.

The network deployment is similar to [11]. The default simulation testbed has 1 base station and 300 sensors randomly distributed in a 300m×300m area. There are additional 20 CHs in the sensor field [11]. The transmission range of a sensor s and a CH is 60m and 150m, respectively. Sensors and CHs are formed in clusters. Each cluster has one CH. Sensors in the same cluster are connected with its CH via one or more hops. We use the same energy model used in *ns-2.1b8a* [21] that requires 0.66W, 0.359W, and 0.035W for transmitting, receiving, and idling, respectively. We set the power consumption rate according to [11][19] for SHA-1 and CBC-MAC calculation 0.48W. As analyzed in [17][18], we set the time consumption for computing a CBC-MAC and a SHA-1 is 7.1ms and 3.5ms, respectively. The simulation uses MAC802.11 Distributed Coordination Function (DCF). Two-ray ground [22] is used as the radio propagation model.

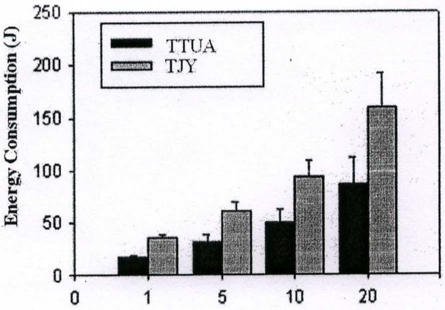

Fig. 6. Comparison of energy consumption

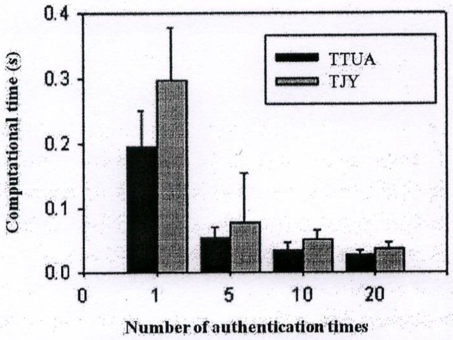

Fig. 7. Comparison of delay time

For routing in both TTUA and TJY, we applied *Ad hoc On-Demand Distance Vector* (AODV) protocol. User ID length is 8 bytes, SHA-1 value is 20 bytes. As discussed in [17], the choice of 4- bytes MAC is not detrimental in the context of sensor networks. So we apply 4-byte CBC-MAC for every message.

We run five different network topologies. For each topology, five scenarios are applied, in which user's location and the login-node are randomly selected. For TJY, we set the gate-way node in the center of the sensor field. We then average the results from those scenarios.

b) Results

Our simulation result is shown in Fig. 6 and Fig 7. For our registration, the user authenticates 1, 5, 10, and 20 times.

The energy consumption of computation and communication in Fig. 6 shows that the energy consumption of TTUA is about twice less than that of TJY. This is consistent with our analysis result in Section VI-A. This is because our computation cost is less than TJY. scheme and TTUA does not require any extra communication with the gate-way node during authentication process. In Fig.7, the delay time of TTUA is also much less than that of TJY. It means that TTUA responses to user authentication request more quickly than TJY. For both schemes, the delay time for the first authentication is relatively higher than later ones because the AODV routing protocol takes time to request a routing-path establishment.

Both performance analysis and simulation results have shown that TTUA met the requirements and is dominant over the existing scheme.

VII. Conclusion and Future Work

This paper presents Two Tier User Authentication scheme (TTUA) for heterogeneous sensor networks. TTUA takes advantages of powerful sensor nodes to increase security and efficiency. Any user can use user ID and password to access any node in the sensor network in a real-time manner. Through security analysis, we show that TTUA is secure against node compromise, reply attacks, and forgery attacks. Both performance analysis and simulation results have shown that TTUA achieves better energy-efficiency compared with the existing scheme such as TJY. The proposed scheme is more secure and energy-efficient than existing approaches.

For our future work, we are going to implement the scheme on our real sensor devices. More importantly, we will provide a mutual authentication, in which sensors must authenticate to users.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, A Survey on Sensor Networks, IEEE Communications, Vol.40(8), pp.102-114, 2002.

[2] H. Wen, C. Lin and T. Hwang, Provably Secure Authenticated Key Exchange Protocols for Low Power Computing Clients, Computers and Security, Vol.25(2), pp. 106-113, 2006.

[3] C. Karlof, D. Wagner. Secure routing in sensor networks: Attacks and countermeasures. Elsevier AdHoc Networks, May 2003.

[4] Adrian Perrig , John Stankovic , David Wagner, Security in wireless sensor networks, Communications of the ACM, Vol.47(6), June 2004.

[5] Haodong W., Bo S., Qun L.. Elliptic curve cryptography-based access control in sensor networks. Int. J. Security and Networks. December 2006, pp. 127-137

[6] Kirk H.M. Wong, Yuan Z., Jiannong C., Shengwei W., "A Dynamic User Authentication Scheme for Wireless Sensor Networks," IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing -Vol 1 (SUTC'06), pp. 244-251

[7] H.-R. Tseng, R-H Jan, and W. Yang (2007), "An Improved Dynamic User Authentication Scheme for Wireless Sensor Networks", IEEE Global Communications Conference (GLOBECOM 2007), USA, Nov. 2007. pp. 986-990.

[8] Z. Benenson, F. Freiling, D. Kesdogan. User authentication in sensor networks (extended abstract) Proceedings of Informatik 2004, Workshop on Sensor Networks, Lecture Notes in Informatics, Ulm, Germany

[9] P. Gupta and P. R. Kumar, "The capacity of wireless networks," IEEE Trans. Inform. Theory, vol. 46, no. 2, pp. 388-404, Mar. 2000.

[10] M. Yarvis, N. Kushalnagar, H. Singh, et al., "Exploiting heterogeneity in sensor networks," in Proc. IEEE INFOCOM, Mar. 2005, pp. 878-890.

[11] X. Du; Mohsen G.; Yang X.o; Hsiao-Hwa C.; "Two Tier Secure Routing Protocol for Heterogeneous Sensor Networks". IEEE Transactions on Wireless Communications, Vol. 6(9), September 2007 p.p:3395 - 3401

[12] X.Du, Y. Xiao, M. Guizani, and H. H. Chen, "An effective key management scheme for heterogeneous sensor networks," Ad Hoc Networks, Vol5(1), pp. 24-34, 2007.

[13] Secure Hash Standard. FIPS PUB 180-1, April 1995. Available at http://www.itl.nist.gov/ pspubs/p80-1.htm

[14] Z.Benenson, N. et al. "Realizing Robust User Authentication in Sensor Networks", Workshop on Real- World Wireless Sensor Networks (REALWSN), Stockholm Sweden, June 2005. PP:135-142

[15] Lakshman K., "Sensor Networks: Promise and Reality," invited presentation, Sensors Expo and Conference, Detroit, MI, June 9, 2004

[16] N. Gura, A. Patel, A. Wander, H. Eberle, and S.C. Shantz. Comparing elliptic curve cryptography and rsa on 8-bit cpus. In CHES, Boston, Aug. 2004.

[17] Karlof, C., Naveen S., and David W. TinySec: A Link Layer Security Architecture for Wireless Sensor Networks. Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys'04)

[18] H. Lee, Y. Choi, H. Kim. Implementation of TinyHash based on Hash Algorithm for Sensor Network. Proceedings of World Academy of Science, Engineering and Technology vol.10 December 2005.

[19] Q. Xue and A. Ganz, Runtime security composition for sensor networks" in Proc. IEEE Veh.Technol. Conf., Oct. 2003, pp. 105-111

[20] SENSE - Sensor Network Simulator and Emulator, http://www.cs.rpi.edu/čheng3/sense/

[21] NS-2, http://www.isi.edu/nsnam/ns

[22] T. S. Rappaport. Wireless communications, principles and practice.Prentice Hall, 1996.