Improvement of Das's Two-Factor Authentication Protocol in Wireless Sensor Networks

DaeHun Nyang and Mun-Kyu Lee

***Abstract*— User authentication is essential for customized services and privileged access control in wireless sensor network. In 2009, Das proposed a novel two-factor authentication scheme for wireless sensor network, where a user must prove the possession of both a password and a smart card. His 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 utilizes only cryptographic hash function. In this letter, we point out that Das's protocol is vulnerable to an off-line password guessing attack, and also show a countermeasure to overcome the vulnerability without sacrificing any efficiency and usability. Besides the patch, we suggest a method to protect query response messages from wireless a sensor node to a user, which is necessary in serving a user in a confidential and authentic way.**

***Index Terms*— Wireless sensor network, authentication, password, smart card**

I. Introduction

Wireless Sensor Networks (WSNs) have many promising applications including environmental monitoring, traffic monitoring, fire alarming, logistics, military sensing and tracking, and so on. In most of these cases, real-time transmission of sensed data is critical and it will be preferable for users to access directly the sensed data without involving the gateway node. For allowance of sensed data only to privileged users, user authentication and authorization are mandatory, which also provide each individual with customized services. Although there are already many well-known authentication mechanisms in the literature, two-factor authentication which requires the proof of possession of both a password and a smart card achieves effectively the purpose of authentic delivery of sensed data while minimizing the amount of user-specific authentication information in the gateway node. This is because the tamper-proof smart card that a user possesses plays the role of secure storage of authentication information instead of the gateway node. To this end, Das proposed an efficient two-factor user authentication scheme, which requires only a small number of applications of cryptographic hash functions [2]. In this letter, however, we point out that Das's protocol has a weakness against an off-line password guessing attack, and propose a countermeasure to this problem without any additional operations. We also present a method for protection of query responses from sensor nodes, which is necessary for a confidential and authenticated service.

This letter is organized as follows: Section II reviews Das's two-factor authentication protocol, and we analyze the security of Das's protocol in Section III. Section IV presents our security-enhanced protocol and in Section V we evaluate our protocol in terms of security and performance. Section VI concludes this letter.

The authors are with the School of Computer and Information Engineering, Inha University, Incheon, 402-751, Korea

II. Review of Das's Two-Factor User Authentication Protocol

In this section, we will briefly review Das's two-factor user authentication protocol which uses both a smart card and a password. The notations used throughout this letter are shown in Table I. The upper part of the table shows Das's original notations and added notations are given in the lower part.

Das's protocol is composed of two phases: the registration phase and the authentication phase. In the registration phase, the gateway node computes and stores $N_i = h(\mathrm{AUTH_i}) \oplus h(K)$, where the authentication information $\mathrm{AUTH_i}$ is defined as $\mathrm{AUTH_i} = ID_i || PW_i$. Then the user's tamper-proof smart card is customized with $ID_i, h(PW_i), N_i$ and x_a, where x_a is a secret key generated by the gateway node and is shared with sensor nodes that are responsible to exchange data with users. Here, x_a is not known to the user.

The authentication phase consists of the login stage and the verification stage. In the login stage, the user's smart card authenticates ID_i and PW_i provided by the user using $h(PW_i)$, and sends DID_i and C_i to the gateway node at time T as shown in Fig. 1. The login stage is followed by the verification stage which begins by checking if $T^* - T$ is within the predefined interval ΔT, where T^* is its current time. After successful verification of the authentication tag C_i, the gateway node transmits DID_i, T', A_i to the nearest sensor node S_n at time T'. Then S_n validates T' and verifies A_i which guarantees the request has come from the legitimate gateway node. If it is successful, S_n responds to U_i's query with sensed data.

TABLE I: Notation

notation	meaning
GW	identity of gateway node
ID_i	identity of user i
PW_i	password of user i
DID_i	dynamic login identity of user i
S_n	identity of sensor node n
K	symmetric key of the gateway node
x_a	secret parameter
$\oplus$	exclusive OR
$\|\|$	concatenation
$EK_{i,n}$	encryption key of user i and sensor node n
$MK_{i,n}$	MAC key of user i and sensor node n
x_n	symmetric key of sensor node n
$h(m)$	cryptographic hash of m
$h_0(m,k)$	MAC of m using key k
$h_1(s), h_2(s)$	key derivation function with seed s

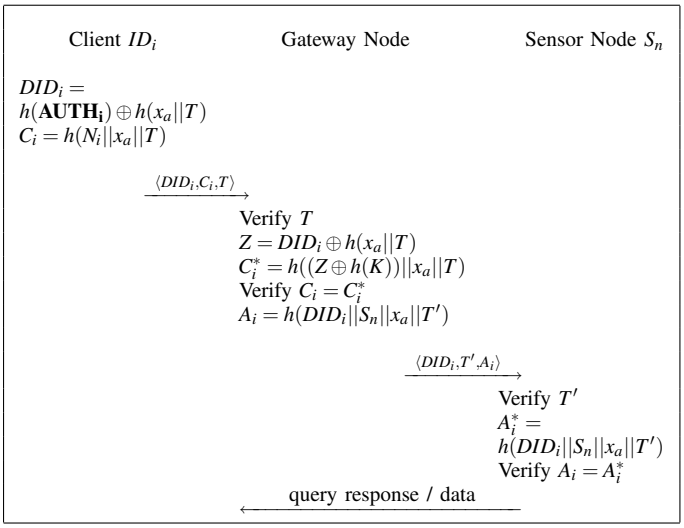

Fig. 1: Two Factor Authentication Protocol: $\mathbf{AUTH_i} = ID_i||PW_i$ for Das's protocol [2] and $\mathbf{AUTH_i} = ID_i||PW_i||x_a$ for the patched protocol

III. Security Consideration on Das's Protocol

A. Password Guessing Attack by Insiders

The author in [2] claimed that password guessing is not possible because a password is transmitted implicitly in DID_i as a digest of some secret components. However, we show that a password guessing attack is possible, that is, the password of the target user i may be guessed by another user who is an insider with his or her own password and smart card. Our attack is based on the observation that DID_i is not exactly a digest of PW_i and some secret components, but it is the XOR-ed value of a digest of PW_i and that of other secret components. The problem is that the secret components are commonly shared among all the users. Note that the timestamp T is necessarily given to a smart card by the user's system, because the smart card cannot synchronize its time without any external power. To be more precise, the password guessing attack can be implemented as follows:

1) The attacker (say, user j) eavesdrops the victim's (say, user i's) authentication session from which DID_i and T can be extracted.
2) The attacker computes his own DID_j with ID_j, PW_j, and T, where T is the timestamp from the captured session above.
3) $B = h(ID_j\|PW_j)$ can be prepared because ID_j and PW_j are the attacker's ID and password, respectively, and h is a publicly-known cryptographic hash function.
4) Now the attacker can compute $h(ID_i\|PW_i)$ by calculating $DID_i \oplus B \oplus DID_j$.
5) Now the attacker is ready to mount an off-line password guessing attack with $h(ID_i\|PW_i)$.

Usually ID's are open to public. In this case, the search space for the guessing attack is $|\mathbf{PW}|$, where $\mathbf{PW}$ is the set of possible passwords and $|\cdot|$ represents the cardinality of a set. Even though the ID is hidden, the search space is $|\mathbf{ID}| \times |\mathbf{PW}|$, where $\mathbf{ID}$ is the set of possible ID's. Note that generally $|\mathbf{ID}|$ is not so big unlike a space for cryptographic key.

The author's idea that a password must be transmitted as a digest of some secret components is correct and appropriate, but its implementation failed to accommodate the idea. We will present an improved implementation adopting the author's idea. Our patch to the original protocol is to use $h(ID_i\|PW_i\|x_a)$ instead of $h(ID_i\|PW_i)$, which prevents the attacker from completing step 3 because $h(ID_j\|PW_j\|x_a)$ requires knowledge of x_a hidden in the tamper-proof smart card. The revised protocol uses newly-defined $\mathrm{AUTH_i}$ and accordingly-modified $N_i = h(ID_i||PW_i||x_a) \oplus h(K)$, which is shown in Fig. 1.

B. Protection of Query Response

It is desirable that a user authentication protocol should be followed by some specific security services such as secure and authentic delivery of sensed data. However, the original protocol in [2] did not care about those security services and omitted the necessary steps such as encryption and authenticity verification of query response. Thus we will add mechanisms that provides those security services by making the gateway node play a role of key distribution center and providing a unique secure channel between a user and a sensor node.

These mechanisms can be realized by establishing a distinct session key for every session and for every pair of a user and a sensor node to frustrate inside attackers and node-capturing attackers. Using the information shared with user i, the gateway node computes those keys, $EK_{i,n}$ and $MK_{i,n}$, which are an encryption key and a MAC key for user i and sensor node S_n, respectively. The keys will be computed as $EK_{i,n} = h_1(DID_i||S_n||(Z \oplus h(K))||x_a||T)$ and $MK_{i,n} = h_2(DID_i||S_n||(Z \oplus h(K))||x_a||T)$. User i can compute $EK_{i,n}$ and $MK_{i,n}$ by itself, but sensor node S_n cannot. So the gateway node must send these keys to S_n in a secure manner. To achieve this secure key transport, S_n and the gateway node are assumed to have a shared key x_n. Now the sensor node S_n can give user i its response in a secure and authentic way.

C. Node Compromise Attack

Das [2] claimed that a sensor node must be equipped with a tamper-proof module to be strong against various attacks caused by node compromise. We show that even with the tamper-proof module, careless implementation of Das's protocol may also cause other security problems such as password guessing attack by outsiders and impersonation of the gateway node. We exemplify a 'careless implementation' as follows:

- The tamper-proof module in a sensor node externally outputs A_i^* so that the sensor node may compare it with A_i.
- The tamper-proof module does not validate its input properly. For example, it does not check if DID_i and S_n are null or not. Also, it does not examine if the given S_n is the correct ID of the sensor node in which it is installed.

In this case, the following attacks are possible.

Password guessing attack: An attacker from outside who captures sensor node S_n can derive $A^* = h(x_a||T) = h(\text{null}||\text{null}||x_a||T)$ by giving null instead of the original values of DID_i and S_n to the tamper-proof module in S_n. Recall that the timestamp T is necessarily given to a smart card and thus, the attacker may control its value. Consequently, someone who does not have x_a can mount an off-line password guessing attack after computing $h(ID_i||PW_i) = DID_i \oplus h(x_a||T)$.

Impersonation of the gateway node: An attacker who captures a sensor node S_n or an insider who has a sensor node S_n can pretend to be the gateway node and then can gather sensed data from an arbitrary sensor node S_m.

1) First, the attacker composes a message using S_n as follows: (s)he sets DID_i as a random string R of a proper length and gives S_m instead of S_n to the tamper-proof module in S_n, the captured node. Note that the tamper-proof module cannot recognize these invalid inputs because we are assuming that it does not check the validity of input values. Then the tamper-proof module will return $A_i^* = h(R||S_m||x_a||T')$.
2) Now the attacker sends (R, T', A_i^*) to S_m. The target node S_m will accept this request and then reply to the attacker with sensed data.

As a result, an attacker can impersonate the gateway node to retrieve information from any node of its own choice. Even if the attacker repeats this attack using the same captured node, the source of this attack, i.e., S_n, cannot be identified or traced because a random string R is refreshed for every trial.

Therefore, we remark that care must be taken so that the authentication protocol may not have security flaws in its implementation.

IV. Security-Enhanced Protocol

We present a security-enhanced two-factor authentication protocol which has the following security properties:

- Prevention of the off-line password guessing attack using $h(ID_i||PW_i||x_a)$ instead of $h(ID_i||PW_i)$.
- Protection of query responses by establishing a unique secure channel between a user and a sensor node.

We remark that our protocol must be implemented carefully so that the node compromise attack explained in Section III-C may not be applicable.

Our authentication protocol is composed of two phases; the registration phase and the authentication phase. The registration phase is the same as that of Das's protocol except that N_i is computed as $N_i = h(ID_i||PW_i||x_a) \oplus h(K)$. The authentication phase begins by user's submitting ID_i and PW_i to his/her smart card, and the following steps are performed:

1) The user's smart card authenticates ID_i and PW_i.
2) The smart card computes $DID_i = h(ID_i||PW_i||x_a) \oplus h(x_a||T)$ and $C_i = h(N_i||x_a||T)$ to send DID_i, C_i, T to the gateway node.
3) Upon receiving the request from the user, the gateway node validates T and authenticates C_i by comparing it with $C_i^* = h((DID_i \oplus h(x_a||T) \oplus h(K))||x_a||T)$.
4) The gateway node computes an encryption key
$$EK_{i,n} = h_1(DID_i||S_n||(DID_i \oplus h(x_a||T) \oplus h(K))||x_a||T)$$
and a MAC key
$$MK_{i,n} = h_2(DID_i||S_n||(DID_i \oplus h(x_a||T) \oplus h(K))||x_a||T)$$
between user i and sensor node S_n to play the role of a key distribution center.
5) To provide a secure channel for $EK_{i,n}$ and $MK_{i,n}$ between S_n and itself, the gateway node computes the encryption key $EK_{GW,n} = h_1(GW||S_n||x_n||T')$ and the MAC key $MK_{GW,n} = h_2(GW||S_n||x_n||T')$, respectively, where x_n is a predistributed symmetric key between the gateway node and S_n and T' is the current time.
6) The gateway node encrypts $EK_{i,n}$ and $MK_{i,n}$ using the key $EK_{GW,n}$ computed in the previous step and produces $D_i = E_{EK_{GW,n}}(EK_{i,n}, MK_{i,n})$. It also computes a MAC $A_i = h_0(DID_i||S_n||D_i||T', MK_{GW,n})$ using the key $MK_{GW,n}$.
7) The gateway transmits DID_i, D_i, T', A_i to S_n.
8) When S_n receives these data, it first verifies T' and computes $A_i^* = h_0(DID_i||S_n||D_i||T', MK_{GW,n})$ using $MK_{GW,n}$. Then it checks if $A_i = A_i^*$.
9) S_n decrypts D_i with $EK_{GW,n}$ and recovers $EK_{i,n}$ and $MK_{i,n}$.
10) Sensed data to be transmitted is encrypted with $EK_{i,n}$ as $R = E_{EK_{i,n}}(Data)$ and a MAC is computed with $MK_{i,n}$ as $B_i = h_0(DID_i||S_n||R||T'', MK_{i,n})$, where T'' is the current time.
11) S_n, R, T'' and B_i are sent to user i.
12) The user verifies T'' and checks B_i by comparing it with $B_i^* = h_0(DID_i, S_n, R||T'', MK_{i,n})$, where $MK_{i,n} = h_2(DID_i||S_n||N_i||x_a||T)$.
13) If this verification is successful, the sensed data is recovered by decrypting R using $EK_{i,n} = h_1(DID_i||S_n||N_i||x_a||T)$.

V. Analysis of the Protocol

This section treats the security and performance aspects of our protocol.

A. Security Analysis

We follow the same threat model as that of Das's protocol, where all parties involved in the protocol communicate over an insecure channel and each party is equipped with a tamper-proof module [3]. To be precise, each user has a tamper-proof smart card which stores $ID_i, h(PW_i), N_i$ and x_a, and each sensor node stores x_a and x_n in the tamper-proof module. Because our protocol is an augmentation of Das's protocol, it enjoys the strength against replay attacks, impersonation attacks and stolen-verifier attacks, and also has the same drawbacks such as the vulnerability to denial-of-service attacks. We will explain in detail that the augmentation does not degrade the security related to those attacks, but it will enhance the security of communication channel between a user and a sensor node and the immunity on password guessing attacks. Note that we augmented Das's protocol in two aspects.

- We redefined N_i as $N_i = h(ID_i||PW_i||x_a) \oplus h(K)$ instead of $N_i = h(ID_i||PW_i) \oplus h(K)$. Also, DID_i and C_i are changed accordingly.

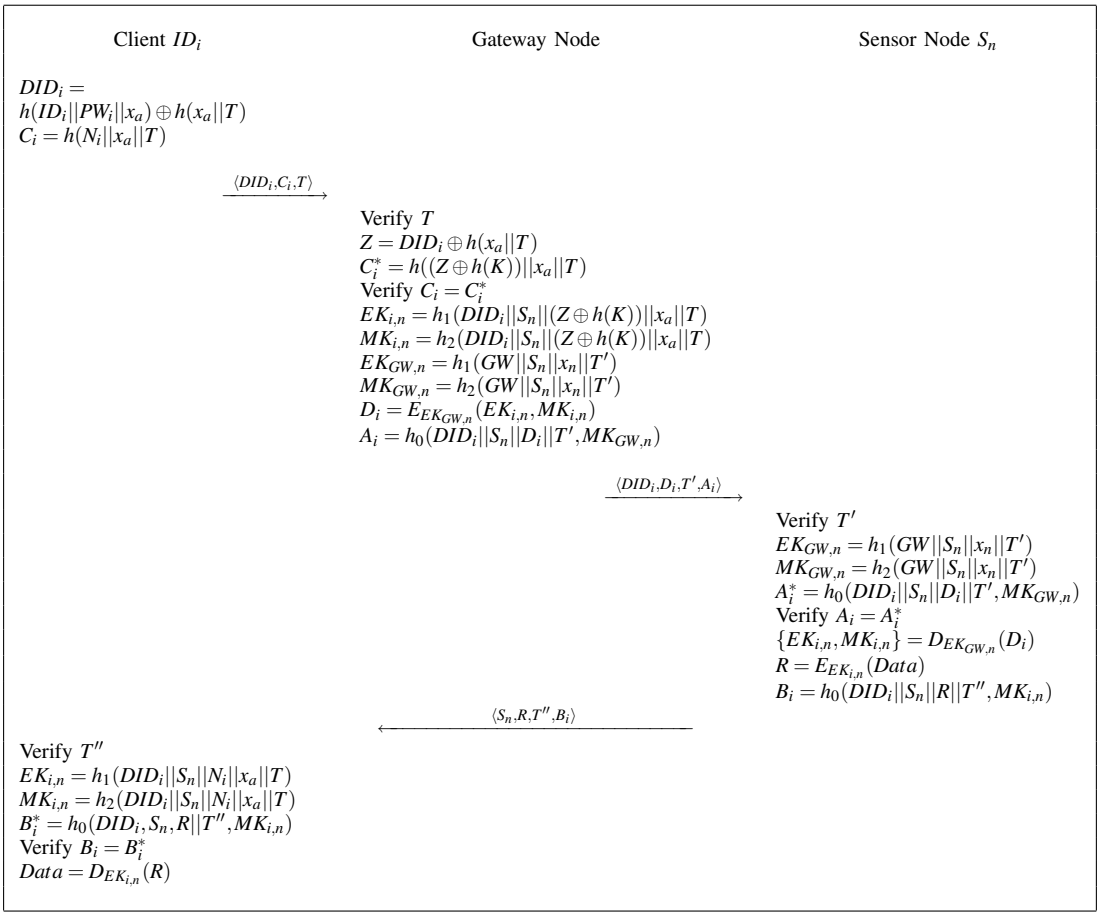

Fig. 2: Security-enhanced Two Factor Authentication

- We introduced a secure channel between a user and a sensor node by defining x_n, a symmetric key shared between the gateway node and sensor node S_n.

First, we consider the effect of redefinition of N_i. Because $h(x_a||T)$ is a random sequence, it plays the role of a key for a one-time pad. Even if $h(ID_i||PW_i)$ is replaced by $h(ID_i||PW_i||x_a)$, the resulting DID_i remains random. C_i also remains random because it is the output of a cryptographic hash function even if its input is modified. Therefore, a passive attacker who does not have any valid smart card cannot distinguish the message DID_i, C_i, T in our protocol from the original one in Das's protocol, which means the level of security of both protocols are the same under the existence of a passive attacker. On the other hand, our protocol is much stronger than the original one with respect to a password gussing attacker who is an insider with a valid smart card, as shown in Section III-A. It is also easy to see that an active attacker does not have any advantage due to the one-wayness of cryptographic hash functions. We remark that most threats regarding a node compromise attack can be frustrated with a careful implementation explained in Section III-C.

Now we consider the effect of introduction of new keys x_n. Our method to protect messages from a sensor node to a user using x_n can be viewed as a variation of traditional approaches used in many key distribution systems such as Kerberos [6]. Therefore the aim to protect the messages is achieved by making the gateway node play a role of key distribution center.

B. Performance Analysis

To defeat the password guessing attack by insiders in section III-A, it is enough to change the definition of $\mathrm{AUTH_i}$ in DID_i and in N_i, so the number of hash calculations and the amount of communication remain the same as those of Das's protocol.

When the protection of query responses is not a concern, the above correction is sufficient. For the protection of query responses, however, additional messages and computations are required. In our proposal, the added operations are computation of message authentication code (MAC), evaluation of key derivation functions (KDF), and encryption and decryption. More specifically, the gateway node is required to evaluate four additional KDFs and one encryption. Two KDF evaluations, decryption of two keys, one MAC computation and encryption of bulk data are additionally asked of a sensor node. A user must perform two KDF evaluations, one MAC computation, and decryption of bulk data. Note that all the above operations can be efficiently implemented using standard block ciphers using appropriate mode of operation. That is, the MAC can be implemented using any block cipher with CBC-MAC or CMAC [5] and the KDF also can be defined by MAC [1]. Con-

sequently, all the operations added to protect query responses can be implemented using a block cipher such as AES [7]. Also, note that most of the recently developed sensor nodes, e.g., TMote, TelosB and Micaz, already have a built-in AES module and thus, no additional hardware is required [4].

On the other hand, additional messages for securing query responses are only one encrypted value of keys (D_i) from the gateway node to the sensor node, and one MAC (B_i) from the sensor node to the user, assuming that the sensor node's identity and a timestamp are already included in the original message.

VI. Conclusion

In this letter, we pointed out that Das's two-factor user authentication protocol is weak against the off-line password guessing attack by insiders, and showed that a simple patch that appends x_a to the authentication information can eliminate this weakness without sacrificing any efficiency and usability. Also, to protect query responses from wireless sensor nodes to a user, we proposed an efficient method which can be easily implemented using a built-in AES function in sensor nodes. Finally, we gave a guideline for secure implementation of authentication protocols which prevents the outsider who captures a sensor node from mounting password guessing attack and from impersonating the gateway node.

References

[1] Lily Chen. NIST Special Publication 800-108, Recommendation for Key Derivation Using Pseudorandom Functions, 2009.

[2] Manik Lal Das. Two-factor user authentication in wireless sensor networks. *IEEE Trans. Wireless. Comm.*, 8(3):1086–1090, 2009.

[3] D. Dolev and A.C. Yao. On the security of public key protocols. *IEEE Trans. Inform. Theory*, 29:198–208, 1983.

[4] Texas Instruments. CC2420: 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver, 2006.

[5] J. Lee JH. Song and T. Iwata. The AES-CMAC algorithm, IETF Request for Comments RFC-4493, 2006.

[6] John Kohl and B. Clifford Neuman. The Kerberos Network Authentication Service (Version 5). IETF Request for Comments RFC-1510, 1993.

[7] National Institute of Standards and Technology. Advanced Encryption Standard(AES), Federal Information Processing Standards Publication 197, 2001.