Cryptanalysis of a strong authentication scheme with user privacy for wireless sensor networks

Chengbo Xu
School of Mathematical Sciences, University of Jinan, Jinan 250022, Shandong Province, P. R. China
E-mail: cbqysy@163.com

Keywords: authentication; password; smart card; session key agreement; wireless sensor networks

Abstract. Authentication and key agreement scheme is an important mechanism for legal users to access the services of wireless sensor network. However, the design of authentication and key agreement schemes in WSNs is still quite a challenging problem. In this paper, we analyze a strong authentication scheme with user privacy for WSNs proposed by Kumar et al. in 2013, and point out the scheme can not resist known session key attack, impersonation attack, sensor node capture attack and suffer from forward security problem, anonymity and untraceability problem.

Introduction

Nowadays, wireless sensor networks (WSNs) are the first choices for a wide range of real-time monitoring applications, such as health care, environmental monitoring, traffic monitoring, etc. In WSNs, data collected by sensor nodes sometimes contain valuable and confidential information that only authorized users are allowed to access. As yet, the design of user authentication and key agreement scheme for resource deficient wireless sensor networks has been substantially addressed by various researchers.

In 2007, Das [1] proposed a two-factor authentication scheme using smart card in which users are authenticated by gateway nodes. The scheme became a center of attraction for many researchers [2-6] working in this field. Das claimed his scheme to be free from the security problems such as stolen-verifier, many logged-in-users with the same identity, guessing, impersonation and replay attacks. In 2010, He et al. [2] pointed out that Das’s scheme does not resist impersonation attack, privileged insider attack and lack of password update mechanism. During the same time, Khan and Alghathbar [3] showed that Das’s scheme susceptible to gateway node bypassing attack and privileged insider attack and proposed an improved scheme. Later on, the improved scheme was pointed out that it does not realize mutual authentication and user’s anonimity, and lacks a mechanism of establishing a session key. [7] Based on this, Yoo et al. proposed a new scheme in 2012. However, Kumar at al. [8] pointed out that Yoo et al.’s scheme does not resist impersonation attack and man-in-the-middle attack, and further proposed an improved scheme.

In this paper, we will point out that Kumar et al.’s scheme [8] does not resist known session key attack, impersonation attack, sensor node capture attack and suffer from forward security problem, anonymity and untraceability problem.

The rest of this paper is organized as follows: in section 2, we briefly review Kumar et al.'s scheme. Section 3 points out the weaknesses of Kumar et al.'s scheme. Finally, we draw our conclusion in section 4.

The notations used throughout this paper are summarized in Table 1.

Review of Kumar et al.'s scheme

In this section, we briefly review the Kumar et al.’s scheme [8]. Their scheme includes three phases: registration phase, authentication phase and password-change phase; and involves three entities: users, gate-way node (GW) and sensor nodes.
Table 1. Notations

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID_k, PW_k</td>
<td>The identity and password of user U_k</td>
</tr>
<tr>
<td>GW_{id}, Sn_{id}</td>
<td>The identities of the gate-way node and sensor node</td>
</tr>
<tr>
<td>J and X</td>
<td>GW secret numbers (that is, 256 bits)</td>
</tr>
<tr>
<td>b</td>
<td>User random number</td>
</tr>
<tr>
<td>E_r[], D_r[]</td>
<td>Symmetric encryption and decryption using key x</td>
</tr>
<tr>
<td>h()</td>
<td>A secure one-way hash function</td>
</tr>
<tr>
<td>Ω</td>
<td>The bitwise exclusive-or operation</td>
</tr>
<tr>
<td>II</td>
<td>Message concatenation operation</td>
</tr>
</tbody>
</table>

Registration phase: To begin with, User U_k selects his/her identity ID_k and password PW_k freely, and then generates a random number b. In addition, GW and sensor nodes are supposed to share a long-term secret key \( LT_{secret} = h(GW_{id} \| Sn_{id} \| h(Y)) \), where Y is a high entropy secret number generated and maintained by GW. The following steps are:

Step 1: User U_k passes his/her identity ID_k and \( h(b \oplus PW_k) \) to the gate-way node (GW).

Step 2: Upon receiving the message, GW computes \( A_k = E_j[ID_k \| GW_{id} \| h(X)] \) and \( B_k = h(ID_k) \| h(b \oplus PW_k) \oplus A_k \), where the secret number J and X are respectively with a lifetime (such as one year).

Step 3: GW stores \( A_k, B_k, h() \) and \( h(X) \) into a smart card and issues this card to U_k through an security channel.

Step 4: Upon receiving the card, U_k stores the random number b into it such that the number need not be remembered.

Eventually, smart card has the following: \( A_k, B_k, h(), h(X) \) and b.

Authentication Phase: In Kumar et al.’s scheme, authentication phase is further divided into two subphases: Login phase and verification phase.

Login Phase: When the user U_k wants to acquire relative data from some sensor node, he/she inserts his/her smart card into a card reader and then keys in his/her identity ID_k and password PW_k.

Step 1: The smart card computes \( B_k' = h(ID_k) \| h(b \oplus PW_k) \oplus A_k \) , and then checks whether \( B_k' \) and \( B_k \) are equal. If they are not equal, terminate the scheme; otherwise, conduct the following steps.

Step 2: Generates a random number \( C_k \) , computes a temporary key \( M = h(h(X) || ID_k || T') \) , and then use the temporary key to encrypt information \( h(ID_k) \| h(X) \| C_k \| T' \) , that is \( P_k = E_{st}(h(ID_k) \| h(X) \| C_k \| T') \) , where T’ is user’s current timestamp.

Step 3: The smart card sends the login message \(< P_k, A_k, T'\>) \) to gate-way node GW.

Verification Phase: Upon receiving the login request message \(< P_k, A_k, T'\>) \), GW will conduct the following steps to verify the validity of user U_k.

Step 1: Checks the inequality \((T'-\Delta T) > T'\) . If yes, terminates; otherwise, proceeds to the next step, where T’ is the current timestamp, \( \Delta T \) is an expected time interval for the message transmission delay.

Step 2: Decrypts \( A_k \) with the GW’s secret number J, and obtains \( ID_k', GW_{id}' \) and \( h(X)' \).

Step 3: Computes \( M = h(h(X) || ID_k' || T') \) , and then decrypts \( P_k \) and obtains \( h(ID_k)^*, h(X)^*, C_k \).

Step 4: Checks \( T'^* = T' \) . If they are not equal, terminates; otherwise, conducts the following steps.
Step 5: Computes $h(ID_\text{k}')$ and checks $h(ID_\text{k}) = h(ID_\text{k}')$, $GW_{id} = GW_{id}'$, $h(X) = h(X)'$. If the three equations are all correct, the validity of user $U_k$ is verified by GW; otherwise, terminate the scheme.

Step 6: Computes $S_{ID_k} = E_{LT_{key}}[h(ID_\text{k}) \parallel GW_{id} \parallel C_k \parallel h(X)' \parallel T'']$ , where $T''$ is the current timestamp. Then, GW sends message $<SID_k, T''>$ to the wireless sensor node $S_n$.

Step 7: Upon receiving the message $<SID_k, T''>$, the sensor node checks the inequality $(T'' - T) > \Delta T$. If yes, terminate the scheme; otherwise, conduct the following steps, where $T''$ is the current timestamp. $\Delta T$ is an expected time interval for the message transmission delay.

Step 8: The sensor node $S_n$ uses its long-term key $LT_{key}$ to decrypt message $SID_k$ and obtains $h(ID_\text{k})'$, $GW_{id}'$, $C_k'$, $h(X)'$, $S_n$' and $T''$.

Step 9: Checks $T'' = T'$, $GW_{id}' = GW_{id}$ and $S_n' = S_n$. If the three equations are all correct, the validity of user $U_k$ and $GW$ is verified by $S_n$; otherwise, terminate the scheme.

Step 10: The sensor node $S_n$ computes session key $S_{key} = h(h(ID_\text{k})') \parallel C_k' \parallel h(X)' \parallel S_n \parallel T'''$ , where $T'''$ is the current timestamp of the sensor node $S_n$.

Step 11: Computes $N_k = E_{LT_{key}}[S_n \parallel C_k \parallel h(X)' \parallel T''']$, and then sends message $<N_k, S_n, T'''>$ to $U_k$.

Step 12: Upon receiving the message $<N_k, S_n, T'''>$ , the sensor node checks the inequality $(T'' - T''') > \Delta T$. If yes, terminate the scheme; otherwise, conduct the following steps, where $T'$ is the current timestamp. $\Delta T$ is an expected time interval for the message transmission delay.

Step 13: User $U_k$ computes session key $S_{key} = h(h(ID_\text{k}) \parallel C_k \parallel h(X) \parallel S_n \parallel T''')$, and then uses the session key $S_{key}$ to decrypt message $N_k$ and obtains $S_n'$, $C_k'$, $h(X)'$ and $T'''$. Finally, $U_k$ checks the following equations $T''' = T''$, $S_n' = S_n$, $C_k' = C_k$ and $h(X)' = h(X)$. If the four equations are all correct, the validity of the sensor node $S_n$ is verified by $U_k$; otherwise, terminate the scheme.

**Weaknesses of Kumar et al.'s scheme**

In this section, we will show that Kumar et al.'s scheme [8] does not implement forward security and user’s untraceability, and is also vulnerable to known session key attack and node capture attack.

**Known session key attack** According to the process of Kumar et al.'s scheme, attacker $A$ can easily eavesdrop the login request message $<P_k, A_k, T'>$ which user $U_k$ sends to the gate-way node $GW$ and the verification message $<N_k, S_n, T'''>$ which the sensor node $S_n$ sends to user $U_k$. Suppose a session key $S_{key}$ is acquired by attacker $A$ in some way. Then, the attacker can use the session key $S_{key}$ to decrypt the data $N_k$ in the message $<N_k, S_n, T'''>$ and obtains $S_n$, $C_k$, $h(X)$ and $T''$. Based on these information, the attacker $A$ can off-line guess user $U_k$'s identity $ID_k$. The concrete steps are as follows:

a) Attacker $A$ guesses a possible identity $ID_k'$;

b) Computes $M' = h(h(X) \parallel ID_k' \parallel T')$, where $h(X)$ and $T'$ have been obtained.

c) Computes $P_k' = E_M[h(ID_k') \parallel h(X) \parallel C_k \parallel T']$;

d) Attacker $A$ checks whether $P_k'$ and $P_k$ are equal. If they are equal, it means the guessed $ID_k'$ is equal to the actual identity $ID_k$ and the attacker succeeds; otherwise, repeat the steps a)-d) until guess succeeds.

Once attacker obtains the identity $ID_k$, he/she can further implement the following two types of attackers.
Impersonation attack

a) Attacker A extracts the current timestamp $T_a$, and generates a random number $C_a'$. simultaneouly.

b) Computes $M = h(h(X) || ID_k || T_a)$, where $h(X)$ has been already obtained.

c) Computes $P_k = E_k[h(ID_k) || h(X) || C_k' || T_a']$, where $ID_k$ has been obtained from offline guess.

d) Attacker A sends the constructed login request message $<P_k, A_k, T_a'>$ to GW. Then, the login request message will pass the verification from GW and $S_n$, and a session key $S_{mn} = h(h(ID_k)||C_k'||h(X)||S_n||T_a'')$ will be agreed in the end, where $T_a''$ is the corresponded timestamp generated by the sensor node $S_n$.

Forward security problem Forward security problem means that once attacker obtains a session key in some way, he/she will restore some previous session keys using the known session key and the information intercepted or eavesdropped from the public communicational channel. In this way, the attacker can easily decrypt the data transmitted in previous sessions.

In Kumar et al.’s scheme, suppose attacker eavesdrops the mutual information $<P_{a-old}, A_{a-old}, T_a'>$ and $<N_{b-old}, S_{b-old}, T_{old}''$ in the authentication phase of some previous session, the attacker can restore the session key as follows:

a) Computes $M_{old} = h(h(X) || ID_{a-old} || T_{old}''$, where $h(X)$ and $ID_{a-old}$ have been obtained previously.

b) Uses $M_{old}$ to decrypt $P_{a-old}$ and obtains $C_{a-old}$.

c) Computes $S_{key-old} = h(h(ID_{a-old}) || C_{a-old} || h(X) || S_{b-old}||T_{old}''$, where $S_{b-old}$ and $T_{old}''$ are from the eavesdropped message $<N_{b-old}, S_{b-old}, T_{old}''>$. Obviously, the obtained $S_{key-old}$ is the actual key applied into that session.

Sensor node capture attack Generally, any identity authentication and session key agreement scheme for wireless sensor network would encounter sensor node capture attack. In this case, how to measure the degree of an authentication scheme’s resistance to sensor node capture attack? That is, how to compare the security performance of different schemes in the sense of sensor node which may be captured.

In 2012, Ashok Kumar Das et al. [9] proposed a method of measuring. In the assumption that there are $c$ nodes have been captured successfully, they define the ratio between the number of captured sessions and that of the total sessions as the toughness of an authentication scheme’s resistance to sensor node capture attack. For a specific scheme, the value of toughness is between 0 and 1, and as small as possible. A scheme would be unconditioanlional security to resist sensor node capture attack if you could prove the value of toughness is 0. The basic idea of this method is quantitative analysis that the $c$ captured nodes effects on all other nodes in this wireless sensor network. Though the value of toughness in this definition would be very difficult to calculate in practice, it took the first step after all in the problem that how to measure the degree of an authentication scheme’s resistance to sensor node capture attack.

In terms of Kumar et al.’s scheme, suppose that some sensor node $S_n$ has been captured, the attacker is able to extract data $LT_{key}$ stored in node $S_n$. Once $LT_{key}$ is known, attacker can obtain $h(ID_{a-old})$, $C_{a-old}$, $h(X)$ and $S_{n-old}$ from $SID_{a-old}$ stored in $<SID_{a-old}, T>'$ which was sended from gateway node to sensor node. Further, attacker can restore the session key $S_{mn} = h(h(ID_{a-old}) || C_{a-old} || h(X) || S_{n-old}||T''$) from the eavesdropped message $<N_{a-old}, S_{n-old}, T''>$. For sessions between user $U_k$ and other nodes (such as $S_{m-old}$), as long as eavesdrop the login request message $<P_k, A_k, T'>$ and the verification message $<N_k, S_{m-old}, T''>$ sended from $S_{m-old}$ to user $U_k$, an attacker is able to obtain the session key as follows:

1) Guess $ID_{a-old}$ offline using $h(ID_{a-old})$;

2) Compute $M = h(h(X) || ID_{a-old} || T''$);

3) Obtain $C_{a-old}$ through decrypting $P_k$ using $M$;
4) So far, all the information needed to compute the session key has been known, compute \( S_{\text{an}} = h(h(ID_k) || C_k || h(X) || S_k || T^*) \).

Through above analysis, it is easy to find that for Kumar et al. ’s scheme, once one sensor node is captured, the data transferred between user and any other sensor node will be obtained since attacker would compute the session key. According to the definition of toughness in [9], it is not difficult to know the value of toughness to resist to sensor node captured attack is 1. In other words, whole the sensor network would have no secure data communication as long as some sensor node has been captured.

**Anonymity and untraceability problem** In [8], Kumar et al. indicated that their scheme has realized user anonymity. The reason is that an attacker could not obtain the real identity \( ID_k \) of user \( U_k \) from the login request message \( \langle P_k , A_k , T' \rangle \) intercepted or eavesdropped, since \( A_k \) is encrypted by the advanced secret key \( J \) of gateway \( GW_k \). \( P_k \) is encrypted by \( M \) which is difficult to compute. If we don’t consider that attacker could be beyond the ability to control the communication channel, the analysis of Kumar et al. is correct. However, it is a common case in reality to capture sensor nodes or steal user’s smart card and extract the information stored. According to the former analysis, it is not difficult to know that several methods can be used to obtain user’s identity in this case.

When it comes to user’s untraceability, for Kumar et al.’s scheme [8], even if the ability beyond that to control the communication channel is not took into consideration, an attacker can also easily judge whether two users are the same from the login request message eavesdropped. The reason is that the data \( A_k \) in the login request message \( \langle P_k , A_k , T' \rangle \) is not changed along with the different of the session, which is decided by the identity of the user itself. Thus, when attacker intercepted or eavesdropped two or more login request messages \( \langle P_k , A_k , T' \rangle \), he/she can recognize whether the two or more users is the same by comparing the data of \( A_k \) in \( \langle P_k , A_k , T' \rangle \). The same \( A_k \) implies the same user, different \( A_k \) means different users.

**Conclusions**

In this paper, we analyze a strong authentication scheme with user privacy for WSNs proposed by Kumar et al. in 2013, and point out the scheme can not resist known session key attack, impersonation attack, sensor node capture attack and suffer from forward security problem, anonymity and untraceability problem.

**Acknowledgements**

This work was partially supported by the Doctoral Fund of University of Jinan (Granted No. XBS1455), and the project of Shandong Natural Science Foundation (Granted No. ZR2013FM009).

**References**


