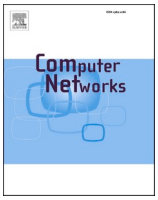
Computer Networks 221 (2023) 109529

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet



A secure model against mobile sink replication attacks in unattended sensor networks

Boqing Zhou [a], Sujun Li [a,*], Jianxin Wang [b,*], Yun Cheng [c,*], Jie Wu [d]

[a] The School of Information Engineering, Shaoguan University, Shaoguan 512005, China
[b] The School of Information Science and Engineering, Central South University, Changsha 410083, China
[c] The Department of Information, Hunan University of Humanities, Science and Technology, Loudi 417000, China
[d] The Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122 USA

ARTICLE INFO

Keywords:
Unattended sensor network
Secure model
Mobile sinks replication attacks
False data injection attacks

ABSTRACT

Unattended wireless sensor networks (UWSNs), in which mobile sinks (MSs) are responsible for the data collection, and which are vulnerable to multiple attacks. For example, the adversaries can initiate MS replication attack after compromising a large number of sensor nodes (SNs). To resist such attacks, some scholars have proposed some schemes. In these schemes, it is assumed that MSs are equipped with tamper resistance hardware, so they are pre-distributed most of keys of a key pool for achieving authentication with SNs. However, in outdoor and even sensitive areas, tamper-resistant hardware is not always absolutely safe. Once MSs are compromised, networks become insecure. In this paper, we propose a secure model. The model contains three types of nodes, namely SNs, MSs, and a base station (BS). MSs are responsible for collecting the data encrypted by SNs and forwarding it to BS; BS is responsible for decrypting and analyzing the collected data. During the data collection process, an MS can collect an SN's data only after being authenticated by the SN using the pre-distribution key information. But it cannot decrypt the collected data. In this model, the adversaries can induce a new type of false data injection attack. In the attack, the replicated MSs impersonate uncompromised SNs to send false data to BS by using the compromised key information. If BS accepts a large amount of false data, it will make a wrong judgment. Analysis and simulation show that the proposed model has excellent resistance against MS replication attacks and false data injection attacks by setting appropriate parameter values.

1. Introduction

In UWSNs, there is no fixed BS, and the data of SNs can only be collected by MSs. Such networks have a wide range of applications [1,2]. Here are 2 examples. One is a monitoring system deployed in a natural park to detect poaching activities that would require an MS to collect data periodically because of the lack of regular access routes and the size of the surveillance area. Another example is a monitoring system deployed along an international border to record illegal crossings. The size of the surveillance area would require an MS to collect data periodically. In addition, if a fixed BS is used, wireless sensor networks are vulnerable to energy holes, where SNs close to the fixed BS are fast drained of their energy. Using MSs can conquer this predicament and extend SNs' lifetime [3–5].

When UWSNs are applied to non-commercial and non-hostile areas, their security issues are not very important. However, in the era of big data, if the raw data is not processed before being shared, new security issues may be raised in data mining applications [6,7]. Especially in commercial and hostile environments, their security issues are particularly important. In such environments, UWSNs are vulnerable to various attacks [8–12], such as MS replication attacks. In this attack, the adversaries can replicate MSs by using a large number of keys obtained by capturing SNs, and then they can collect data from networks through these replicated MSs. To resist this attacks, authentication and pairwise key establishment between nodes become extremely important. However, the resource constraints of SNs and their nature of communication over a wireless medium make it a nontrivial task. Due to the high computation and storage overhead of asymmetric key schemes, symmetric key schemes are still very attractive [13–31].

* Corresponding authors.
E-mail addresses: lsj_paper@163.com (S. Li), jxwang@mail.csu.edu.cn (J. Wang), yuncheng@huhst.edu.cn (Y. Cheng).

https://doi.org/10.1016/j.comnet.2022.109529
Received 25 June 2022; Received in revised form 26 November 2022; Accepted 13 December 2022
Available online 14 December 2022

1.1. Motivation

However, the problem of authentication and pairwise key establishment between nodes is still not solved in the face of MS replication attacks. In [23], Li et al. proposed an EQ method against H-sensors replication attacks in heterogeneous sensor networks. In the scheme, a shared key between an H-sensor and an SN is determined by the SN randomly selecting EQ keys from its pre-distribution keys. The security analysis and simulation indicate that the EQ method can provide a very good resilience against H-sensors replication attacks if there are a few SNs compromised in the bootstrap phase. However, in the EQ method, an authentication between an SN and an H-sensor requires the cooperation of multiple H-sensors to complete. In UWSNs, the data of a region is generally collected by an MS. If the method is directly applied in UWSNs, a large number of SNs will be wasted because they cannot authenticate the MS. In [33], a three-layer security model integrated with q-scheme [14] and key space scheme [15] is presented. In this model, pre-distribution keys of MSs and SNs come from two completely different key pools, and the communication between them can only be achieved by establishing shared keys with static access nodes which pre-distribute keys from the above two key pools. As a result, replication MSs can collect data from UWSNs by impersonating static access nodes when many keys coming from the key pool of SNs are compromised. To improve UWSNs' resilience against MS replication attacks, Li et al. [34] proposed a joint authentication scheme based on key space scheme [15]. However, in this scheme, authentication between an SN and an MS requires the participation of multiple neighbors of the SN. As a result, the communication overhead of the scheme is large, and DoS attacks may occur when there are many SNs compromised and these compromised nodes refuse to send correct authentication information to their neighbors. In addition, from the analysis of the key space scheme [15], it can be known that there is a safe threshold for such schemes. If the number of captured nodes is less than this threshold, the security performance of the network is very good; otherwise, its security performance decreases significantly with the increase of the number of compromised nodes. In order to improve network performance, the use of deployment knowledge is a simple and efficient method [18, 26–31]. Using deployment knowledge, Zhou et al. proposed a scheme for direct authentication between an SN and an MS [35]. The analysis and simulation show that compared with the scheme [34], its energy consumption is reduced, and the resistance against replication MS attacks is more stable. Table 1

In the above four schemes [32–35], it is assumed that H-sensors or MSs are equipped with tamper resistance hardware, that is, the adversaries cannot obtain their pre-distribution key information even if they are captured. However, tamper resistant hardware is not completely secure [36]. If H-sensors or MSs are compromised, the replicated H-sensors or MSs can collect data from SNs in networks freely. Therefore, in UWSNs, the mechanism for improving the resistance against MS replication attacks still needs to be further studied.

1.2. Main contribution of our scheme

In this paper, we propose a new secure model for UWSNs. Our main contributions are as follows.

1) In this model, authentication between MSs and SNs is strengthened. The pre-distribution keys of an MS can be used to complete the authentication with an SN, but cannot be used to decrypt the encrypted information of the SN. When an MS in a certain region is compromised, a new MS can be deployed into the region. After the new MS is authenticated by SNs in the region, these SNs will automatically disconnect with the previously deployed MSs. That is, the model does not rely on the assumption that MSs are absolutely safe. The adversaries can replicate MSs by using the key information obtained from these compromised MSs and SNs, and then they can collect data from SNs in the network through these replicated MSs. In this case, even if no new MSs are deployed, the probability of replicated MSs collecting data from SNs is low by setting appropriate parameter values. Besides, in the model, if an MS is not compromised, SNs can cooperate with their neighbors to authenticate the MS. If an MS cannot be authenticated by an SN, it cannot collect data from it. In short, by setting appropriate parameters, this model has excellent resistance against the above attacks.
2) In this model, the adversaries can conduct a new type of false data injection attacks, which use the replicated MSs to impersonate uncompromised SNs to send false data to BS by using the compromised key information. If a large amount of false information is accepted by BS, BS will make a wrong conclusion. To resist these attacks, in the secure model, the data of an SN can be accepted by BS only after it has been decrypted and authenticated by BS. During the authentication and encryption process, all pre-distribution keys of an SN are involved in. Therefore, an MS can successfully impersonate an uncompromised SN if and only if the pre-distribution keys of the SN have all been compromised. By setting proper parameters, this model has excellent resistance against false data injection attacks.
3) In this model, a new method for determining an MS's pre-assigned keys' ID based on its ID is proposed. In the new method, only by getting the ID of an MS, SNs can quickly calculate the IDs of the pre-distribution keys of the MS. During the share keys establishment between an MS and an SN, the MS only needs to send its ID to the SN, thereby reducing the energy consumption generated by the previous method of discovering the shared keys by exchanging pre-distribution keys' IDs with each other.

Table 1
Notations used.

Notation	Description
MS	Mobile sink
SN	Sensor node
R	Communication radius of an SN
N	The expected value of the number of neighbors of a sensor node in the network
ID_{MS}	The ID of an MS
ID_{SN}	The ID of an SN
G_{MS}	The number of deployment generation of an MS
AMG_{MS}^{SN}	The maximum number of deployment generation of an MS which has been authenticated by an SN
ID_K	The ID of the key K
KP	The key pool
M	The size of the key pool
$\|\|$	Concatenation operation
$H()$	A one-way hash function
$H_K()$	A one-way hash function with the key K
$E_K()$	A encryption function with the key K
$t1$	The number of keys pre-distributed to an MS
$t2$	The number of keys pre-distributed to an SN
%	Represents the residual operator
(i_1,i_2)	Represents the greatest common divisor of the integers i_1 and i_2
n	The number of neighbors of an SN involved in the authentication between an MS and the SN
Q	The minimum number of correct authentication messages should be provided by an MS when it wants to pass the authentication of an SN
RN_{MS}	A random number generated by an MS
$\oplus$	XOR operation

1.3. Organization

The rest of this paper is organized as follows. At first, the background of our scheme is presented in Section 2. Subsequently, the proposed scheme will be presented in Section 3. The theoretical and experimental results will be described in Section 4. At last, the conclusion will be made in Section 5.

2. Related work

To improve networks' resilience against H-sensors replication

attacks in heterogeneous sensor networks, an EQ method is proposed [32]. In the method, an SN randomly selects EQ keys from its pre-distribution keys as authentication keys with its cluster head. And by flooding, a cluster head can request other cluster heads to send these keys selected by SNs in its cluster which it doesn't have. This scheme can provide good resilience against H-sensor replication attacks. However, the probability that a cluster head has the EQ keys selected by sensor nodes in its cluster is low. If this scheme is directly applied to UWSNs, without the help of other MSs, an MS cannot be authenticated by many SNs in the network. To improve networks' resilience against MS replication attacks, a three-tier authentication scheme is proposed [33]. In the scheme, MSs and static access nodes, static access nodes and SNs share the dynamic key pool and static key pool respectively. As a result, an authentication between an MS and an SN is accomplished with the help of static access nodes, which makes an adversary easily know a great deal of key information of the static polynomial pool by compromising a small fraction of SNs. As a result, the adversaries can collect data from SNs by launching static access node replication attacks. Li. et al. proposed a (*M, m*) authentication scheme [34]. In the scheme, an SN randomly chooses *M* neighbors to form its candidate authentication set. During the authentication process, if the number of key spaces shared between an MS and an SN cannot be less than *q* and the number of correct authentication information received by the MS from the candidate authentication set of the SN is not less than *m*, the MS can pass the authentication of the SN. The (*M,m*) authentication scheme can significantly improve the resilience against MS replication attacks of two schemes [23] and [24]. However, DoS attacks occur when these compromised SNs jointly send forged authentication information to an MS. As a result, the number of correct authentication messages collected by an MS is less than *m*. At the same time, the energy consumption of joint certification is relatively high. Zhou et al. proposed a new method where the number of keys pre-distributed to an MS is independent of the size of the deployment area by using three-dimension backward key-chain, and proposed a scheme that can provide direct authentication between an MS and an SN by using deployment knowledge [35]. Compared with the scheme in [34], its energy consumption is significantly reduced; its resistance against MS replication attacks decreases slowly, especially when the number of compromised nodes exceeds the threshold, its resistance against MS replication attacks is significantly better than that of the scheme in [34].

3. Our scheme

Here, we introduce our scheme from the following two aspects: one is the security model, including the network model, the threat model and a key management model. The other is the authentication method between nodes.

3.1. Notations

In this paper, we use the following notations for the description convenience.

3.2. Models

3.2.1. Network model

In our scheme, a USWN includes BS, MSs and a large number of SNs. Each SN has a unique ID, and stores the data that it collects by itself before an MS collects it. Each MS has a unique ID, and its' communication radius can be dynamically adjusted.

After an MS is successfully authenticated by an SN, the SN sends the encrypted data and the authentication information to the MS. The MS forwards the collected data to BS, but cannot decrypt it. BS is responsible for decrypting, authenticating and processing the collected data.

3.2.2. Threat model

In this article, the security of sensor networks deployment environment is not guaranteed. Therefore, both SNs and MSs can be compromised. That is, if an attacker captures an SN or an MS, all key information it holds will also be compromised. Moreover, the adversary may pool the keying materials from multiple compromised nodes to break the security of the network or to launch advanced attacks. Such as eavesdropping, replication attacks, fake data injection attacks, etc. In this article, an attack that attempts to illegally collect data from SNs in a network is referred to as an MS replication attack. Therefore, adversaries can launch such attacks in two ways: one is by copying compromised MSs, the other is by forging MSs based on compromised key information. Similarly, in this article, an attack that attempts to provide false data to BS is referred to as a false data injection attack. Adversaries can launch such attacks in the following two ways: one is by injecting fake data into BS through compromised SNs, the other is by injecting fake data into BS through replicated MSs. This article only discusses the second type of fake data injection attacks, because the first fake data attacks can be implemented by identifying compromised sensor nodes [37,38].

In addition, since BS is located far from the sensor network deployment environment and is well protected, therefore, we assume that BS is secure.

3.2.3. Key management model

BS generates a key pool KP and IDs of all keys. The size of *KP* is *M*. Each SN selects t2 keys from *KP* without repetition, and IDs of the keys are calculated by the following method (for details, see Algorithm 1):

In the above formula, $KID_0 = H(ID_{SN})\%M$, $H(H(ID_{SN})) = H^2(ID_{SN})$, …, $H(H^l(ID_{SN})) = H^2(H^{l\text{-}1}(ID_{SN})) = \ldots = H^{l+1}(ID_{SN})$. If KID_i has been assigned to itself, it needs to be recalculated until the calculated KID_i is not pre-assigned to itself.

An MS selects *t*1 keys from *KP* without repetition according to the Algorithm 2. The process of selection is as follows:

1. Divide the key pool *KP* into multiple disjoint sub-key pools. The number of sub-key pools is the greatest common divisor of *t*1 and *M*, which is denoted as $(M, t1)$, so the size of each sub-key pool can be calculated as: $\frac{M}{(M,t1)}$.
2. The ID of the *j*-th pre-distribution key selected from the *i* th subkey pool is calculated as follows:

$$KID_i^j = i \times \frac{M}{(M,t1)} + H^l(ID_{SN} \| \ i)\%(M/(M,t1)) \tag{2}$$

Algorithm 1
keys pre-distributed to an SN.

```
void pre_keys_to_SN(int ID, Key_List KList_SN[])
  //ID represents the ID_SN
  {int flag[M],i,l,KID;
  for(i = 0; i<M; i++)
  flag[i]=0;
  l = 1;
  for(i = 0; i<t2; i++)
  {while (1)
  {KID=H^l(ID)% M;
  l++;
  if (flag[KID]==0)
  {flag[KID]=1;
  //KP[KID] indicates the KID-th key in the key pool KP
  KList_SN[i]={KP[KID], KID};
  break;
  }
  }
  }
  }
```

Algorithm 2
pre-distribute keys to an MS.

```
void pre_keys_to_MS(int ID, int GCD, Key_List KList_MS[])
  //ID indicates the ID_MS, GCD=(M,t1)
  {int i,j,k,flag[M/GCD],l, index, KID;
  for(i = 0; i<GCD; i++)
  {for(j = 0; j<M/GCD; j++)
  flag[j]=0;
  l = 1;
  for(j = 0; j<t1/GCD; j++)
  {while (1)
  {index = H^l(ID || i)%(M/GCD);
  KID=i × M/GCD + index;
  l++;
  k = i × t1/GCD + j;
  if (flag[index]==0)
  {flag[index]=1;
  HKP[k] = H_KP|KID|(ID);
  KList_MS[k] = {HKP[k], KID};
  break;
  }
  }
  }
  }
  }
```

Algorithm 3
Shared Keys discovering Algorithm.

```
int search_share_key_in_MS (int ID0, int ID1)
  //ID0=KID_i, ID1= ID_MS
  {int i, l, KID, flag[M/(M,t1)],index,ID2;
  ID2 = ⌊KID_i/(M,t1)⌋;
  for(i = 0; i<M/(M,t1); i++)
  flag[i]=0;
  l = 1;
  for(i = 0; i<t1/(M,t1); i++)
  {while (1)
  {index = H^l(ID1 || ID2)%(M/(M,t1));
  KID=ID2 × M/(M,t1) + index;
  l++;
  if(KID==ID0)
  return 1;
  if(flag[index]==0)
  {flag[index]=1;
  break;
  }
  }
  }
  return 0;
  }
```

where $0 \le i < (M,t1)$, $0 \le j < \frac{t1}{(M,t1)}, l \ge 1$. The calculation process is similar to the formula (1), and will not be described in detail here.

These selected keys must be processed as follows before they are pre-distributed to an MS:

1. These keys must be hashed. For example, a key K, it can be calculated as follows:

$$HKP[ID_K] = H_K(ID_{MS}) \tag{3}$$

2. These keys are sorted in ascending order by their IDs.

3.3. Authentication method between nodes

In the scheme, authentication between nodes includes: 1. Authentication between an SN and an MS; 2. Authentication between BS and SNs. In this scheme, when an MS is located at a point, the data of all sensor nodes in the area with the point as its center and R as the radius are collected. In order to improve the security of authentication, the MS generates a random number for each area for authentication with the sensor node. The detailed process is described as follows:

3.3.1. Authentication between an SN and an MS

Step 1. The MS, which is responsible for the i th collection of data, broadcasts the following authentication request message with the communication radius of R:

$$Req^{Au}_{MS\_} = \{ID_{MS}, RN_{MS}\} \tag{4}$$

Step 2. When an SN receive the$Req^{Au}_{MS_}$, if $G_{MS} < AMG^{SN}_{MS}$, the SN ends the authentication process with the MS. Otherwise, the SN turns to the step 3 to establish a shared key between the SN and the MS;

Step 3. According to the Algorithm 3, the SN discovers the IDs of these keys shared with the MS. The search process is described as follows:

1. Using the formula (5), the SN calculates the index number of the sub-key pool according to each ID of the pre-distribution keys, i.e. $KID_i (1 \le i \le t2)$:

$$ID_{SKP} = \left\lfloor \frac{KID_i}{(M,t1)} \right\rfloor \tag{5}$$

2. According to the formula (2), the SN can calculate the ID of the j-th key from the sub-key pool ID_{SKP}, which is pre-distributed to the MS, that is, KID^j_t. If $KID^j_t = KID_i$, it returns a matching success message, and adds this ID to the list of common keys. If $j \ge \frac{t1}{(M,t1)}$, then a failed matching message is returned.

Assuming that the IDs of the common keys discovered are: KID_1,…, KID_q, then the key shared with the MS can be calculated as:

$$HSK_{SN\_MS} = HKP[KID_1] \oplus \cdots \oplus HKP[KID_q] \tag{6}$$

Step 4. If $q<1$, the SN ends the authentication process with the MS. Otherwise, the SN performs the following operations:

1. The SN uses Algorithm 1 and Algorithm 3 to find out its neighbor nodes that have at least one common key ID with the MS to form its candidate authentication set, namely CS^{Au}_{SN}.
2. The SN randomly selects n nodes from CS^{Au}_{SN} to form its joint authentication set, namely JS^{Au}_{SN}.

$$JS^{Au}_{SN} = \{ID_{SN_1}, \cdots, ID_{SN_n}\} \tag{7}$$

3. The SN sends the following reply information to the MS:

$$Rep^{Au}_{SN\_MS} = \{ID_{MS}, ID_{SN}, JS^{Au}_{SN}, Au_{SN-MS}\} \tag{8}$$

where, Au_{SN-MS} is the authentication code calculated by the following formula:

$$Au_{SN-MS} = H_{HSK_{SN\_MS}}(ID_{MS} \| ID_{SN} \| ID_{SN_1} \| \cdots \| D_{SN_n} \| RN_{MS}) \tag{9}$$

Step 5. After the MS receives $Rep^{Au}_{SN\_MS}$, it performs the following operations:

1. The MS calculates these IDs of these keys pre-distributed to the SN according to the Algorithm 1 and finds the IDs of the common keys with the SN in its own keychain, and then calculates the shared key HSK'_{SN_MS} between them according to the formula (6);

2. The MS recalculates the authentication code Au'_{SN_MS} according to the formula (9). If $Au_{SN\_MS} \neq Au'_{SN\_MS}$, then the MS ends the authentication process with the SN. Otherwise, the MS calculates the following authentication information:

$$Au_{MS\_SN} = H_{HSK'_{MS\_NS}}(ID_{SN} \| ID_{MS} \| RN_{MS} + 1) \tag{10}$$

3. The MS can get these SNs' IDs from JS^{SN}_{Au}. According the aforementioned methods, it can calculate shared keys $HSK'_{SN_i\_MS}(1 \le i \le n)$ with these SNs respectively. Lastly, it can calculate $JAu^1_{SN_i\_MS}$ $(1 \le i \le n)$ according the formulas (11) and (12);

$$JAu^0_{SN_i\_MS} = H_{HSK'_{SN_i\_MS}}(ID_{MS} \| RN_{MS}) \tag{11}$$

$$JAu^1_{SN_i\_MS} = H\left(JAu^0_{SN_i\_MS}\right) \tag{12}$$

4. The MS sends the following reply information to the SN in the communication radius of R:

$$Rep^{Au}_{MS\_SN} = \left\{ID_{SN}, ID_{MS}, Au_{MS\_SN}, JAu^1_{SN_1\_MS}, \cdots, JAu^1_{SN_n\_MS}\right\} \tag{13}$$

5. After the MS replies with the authentication information of all SNs in this area, the MS broadcasts the following request for assistance in authentication, namely $Req^{AAu}_{MS\_}$, with a communication radius of 2R:

$$Req^{AAu}_{MS\_} = \{ID_{MS}, RN_{MS}\} \tag{14}$$

Step 6. If an SN receives $Rep^{Au}_{MS\_SN}$, then the SN authenticates the *MS* according to formula (10). If authentication fails, the SN ends the authentication process with the MS.

Step 7. If an SN receives $Req^{AAu}_{MS\_}$, the SN calculates the shared key with the MS according to algorithm 3, then calculates the authentication code $JAu^0_{SN\_MS}$ according to formula (11), and finally broadcasts the following authentication assistance message:

$$Rep^{AAu}_{SN\_} = \left\{ID_{MS}, ID_{SN}, JAu^0_{SN\_MS}\right\} \tag{15}$$

Step 8. If the SN that successfully authenticates the information $Rep^{Au}_{MS\_SN}$ and receives $Rep^{AAu}_{SN_i\_}$ $(ID_{SN_i} \in JS^{Au}_{SN})$, then the SN verifies $JAu^1_{SN_i\_MS}$ sent by the MS to its as follows: if $JAu^1_{SN_i} = H(JAu^0_{SN_i})$ indicates that the authentication message provided by the MS for the SN is correct. Let x represent the number of correct authentication messages provided by the MS for the SN. If $x < Q$, then the MS cannot be authenticated by the SN, the authentication process ends. Otherwise, the SN updates the value of variable AMG^{SN}_{MS} with G_{MS}, and sends the collected information inf_{SN} to the MS as follows:

$$Einf_{SN} = \{ID_{MS}, ID_{SN}, E_{K_{SN}}(inf_{SN}, Auinf_{SN})\} \tag{16}$$

where K_{SN} indicates the key being XORed all pre-distribution keys and $Auinf_{SN} = H(ID_{SN} \| inf_{SN})$.

Step 9. The MS receives and stores $Einf_{SN}$.

3.3.2. Authentication between MSs and SNs, and BS

After BS obtains the encrypted data $Einf_{SN}$ of an SN in the network through an MS, it does as follows:

1. Calculates IDs of the t2 keys pre-distributed to the SN according to its ID based on the Algorithm 1, and obtains K_{SN} by XORing the $t2$ keys;
2. Obtains inf'_{SN} and $Auinf'_{SN}$ by decrypting information $E_{K_{SN}}(inf_{SN}, Auinf_{SN})$. If $Auinf'_{SN} \neq H(ID_{SN} \| inf'_{SN})$, BS discards the information $Einf_{SN}$. Otherwise, the authentication succeeds and BS stores the received data.

4. Security evaluation

In this section, we analyze the security of our scheme, including the probability that an MS can be authenticated by an SN, the resistance against false data injection attacks and MS replication attacks.

In our analysis and simulations, we use the following setups:

1. The deployment area is 1000 m × 1000 m. We assume that the deployment area is flat, and that the nodes follow a uniform distribution within the area. The neighbor relationship between SNs is symmetrical. That is, if A is a neighbor of B, then B is also a neighbor of A.
2. The wireless communication range of an SN is 40 m.
3. The presented experimental data is an average of 50 replicates.

4.1. The probability that an MS can be authenticated by an SN

4.1.1. Energy consumption analysis of improved shared keys discovery method

In the scheme, many keys are pre-distributed to an MS. If the traditional method of exchanging IDs of pre-distribution keys [13,14,16–28,18,29–35] is used to determine the common key between an SN and an MS, a large amount of energy of the SN will be wasted. For example, when $M = 20{,}000$, the number of keys pre-distributed to an MS is 16,000 and the size of a key's ID is 3B, the size of all keys' IDs is 46.875 KB. The energy consumption of an SN to receive this information can be estimated using the energy model proposed in [39]. To receive a message, the radio expends:

$$ET_{elec}(l) = l \cdot E_{elec} \tag{17}$$

where $ET_{elec}(l)$ represents the electronics energy consumed by an SN for receiving 1 bit data. The communication energy parameter is set as: $E_{elec} = 50nJ/bit$.

The energy consumption of an SN to receive IDs of pre-distribution keys of an MS is about 57.6mJ by using the formula (17). Such large energy consumption is not only unbearable for SNs, but also it will lead to new attacks. In the attacks, an adversary only needs to pretend to be an MS to repeatedly broadcast IDs of its pre-distribution keys, resulting in that the energy consumption of an SN is quickly and it will die early. Therefore, when a large amount of key information is pre-distributed to an MS, it is infeasible to use the traditional method to let SNs know the pre-distribution key information of the MS.

In our scheme, the key pool is divided into $(M, t1)$sub-key pools, and the size of a sub-key pool is: $\frac{M}{(M,t1)}$. These IDs of pre-distribution keys of an MS can be calculated by its ID (see Algorithm 2). During the process of establishing a shared key between the MS and an SN, the MS only needs to broadcast its ID. According to Algorithm 3, an SN can find out IDs of their shared keys. In the process of establishing a shared key, the efficiency of the algorithm is related to the number of computations. To determine whether a key is a shared key between them, the SN performs at least one Hash calculation and at most $\frac{t1}{(m,t1)} + \Delta$ Hash calculations (where Δ represents the total number of all duplicate keys' ID during the execution of Algorithm 3). Its value is generally small. For example, during our simulations, its value did not exceed 10. Therefore, NC, the times of an SN performing Hash calculation, satisfies the following equation:

$$NC \in \left[t2, t2 \times \left(\frac{t1}{(m,t1)} + \Delta\right)\right] \tag{18}$$

When $M = 20{,}000$,t1=16,000,t2=7, NC$\le 7 \times (4 + \Delta)$. In addition, SNs

can efficiently complete the calculation of the one-way Hash function, for details, please refer to [16,17,19,23,30–32,35]. So, its energy consumption is negligible as compared to the traditional method of broadcasting IDs of keys.

4.1.2. The probability of an MS being authenticated by an SN

In our scheme, an MS must meet the following 2 conditions to be authenticated by an SN:

1. There are common keys' IDs between their pre-distribution keys.
2. The number of correct authentication messages provided by an MS to an SN is not less than Q.

In this scheme, IDs of these pre-distribution keys of an MS and an SN are calculated by a one-way Hash function. However, when the number of SNs is large, the number of pre-distribution for each key in the key pool is basically the same. Therefore, all probabilities can be estimated by using a random pre-distribution key model. The probability that an MS and an SN can establish a shared key, namelyPc_{M_C}, can be calculated as follows:

$$Pc_{M\_S} = \begin{cases} 1 - \dfrac{\binom{M-t2}{t1}}{\binom{M}{t1}}, & t1 + t2 \le M \\ 1 \quad , & t1 + t2 > M \end{cases} \tag{19}$$

From the comparison of theoretical analysis results and simulation results, it can be concluded that it is feasible to use the random pre-distribution key model to estimate thePc_{M_C}. From the formula (19), it can be concluded that Pc_{M_C} increases with the increase of t1 and t2 when M is constant. For example, when $M = 20{,}000$, $t1$=0.8 M and $t2$ increases from 3 to 7 step by step, $Pc_{M_C}$is approximately 0.9979, 0.9988, 0.9998, 0.9999 and 1, respectively; when $M = 20{,}000$, $t2$=6 and $t1$= 0.7 M, 0.75 M, 0.8 M, 0.85 M and 0.9 M, $Pc_{M_C}$is equal to 0.9964, 0.9994, 0.9999, 1 and 1, respectively. In addition, it can also be concluded from Fig. 1 that when the ratio of keys pre-distribution to an MS and the value of t2 are both fixed, the influence of M on Pc_{M_C} can be ignored. Based on this feature, we can set up multiple key pools, and the number of nodes that each key pool can accommodate is set based on security factors. When the number of keys compromised in a key pool exceeds a certain threshold, we cannot distribute keys of this key pool to an MS. As a result, these SNs pre-distributed keys from this key pool will be disconnected from the network because they cannot be authenticated by the MS.

In our scheme, these selected joint authentication SNs by an SN are able to establish a shared key with the MS. Assume that all compromised SNs provide wrong authentication messages to its neighbors. In this case, condition 2 is satisfied as long as there are Q uncompromised SNs among the n jointly authenticated nodes. Suppose $N^{'}$ represents the expected value that the neighbor nodes of an SN can establish a shared key with an MS, and m represents the number of SN's neighbors compromised. Then the probability that an MS can be authenticated by an SN can be estimated by the following formula:

$$Pa_{M\_S} = \begin{cases} 0 & , N^{'} - Q < m \\ 1 & , n - Q \ge m \\ \sum\limits_{j=\max(Q,n-m)}^{\min(N^{'}-m,n)} \dfrac{\binom{N^{'}-m}{j}\binom{m}{n-j}}{\binom{N^{'}}{n}}, & others \end{cases} \tag{20}$$

In the above formula, When N, the number of an SN's neighbors, is known, $N^{'} = N \times Pc_{M\_S}$.

From the formula (20), it can be concluded that Pa_{M_S} increases with the increase of $N^{'}$and n-Q, and decreases with the increase of m. Fig. 2 shows the influence of each parameter on Pa_{M_S} when SNs are uniformly captured. It can be seen from Fig. 2 that there is a gap between the theoretical results and the simulation results, because the number of neighbor nodes of the network edge nodes is much smaller than N. As a result, compared with the nodes with N neighbors, the probability of edge nodes being able to authenticate an MS is significantly reduced. At the same time, for the non-network edge nodes with more than N neighbor nodes, the probability of being able to authenticate an MS is not significantly increased. However, in the theoretical analysis, the number of neighbors of all SNs is calculated by the average. As a result, the theoretical value of Pa_{M_S} is higher than its simulated value. But when the value of n-Q remains unchanged, the gap between the theoretical value of Pa_{M_S} and its simulated value decreases with the increase of the value $\frac{N^{'}-m}{N^{'}}$. For example, in Fig. 2, when$N^{'} \ge 30$, n-$Q = 5$ (that is, $B + 4$ in the figure), and $\frac{N^{'}-m}{N^{'}} \ge \frac{33}{40}$, the theoretical analysis value and the simulation value of $Pa_{M_S}$are about 1.

4.2. Resilience against false data injection attacks

In this model, SNs directly store keys from the key pool, but keys stored in an MS are processed by a one-way Hash function. From the properties of the one-way hash function, it can be known that it is computationally infeasible to calculate the original key from the key after being hash processed. The information of an SN, which is sent to BS, is encrypted and authenticated by using pre-distribution keys of the SN. Therefore, it is not possible to successfully impersonate un-captured SNs to provide false data information to BS just by compromising MSs. However, the adversaries can obtain more key information by compromising SNs, and then can replicate MSs by using compromised key information, and these replicated MSs can successfully impersonate to be uncompromised SNs to provide false data to BS. If the BS receives a

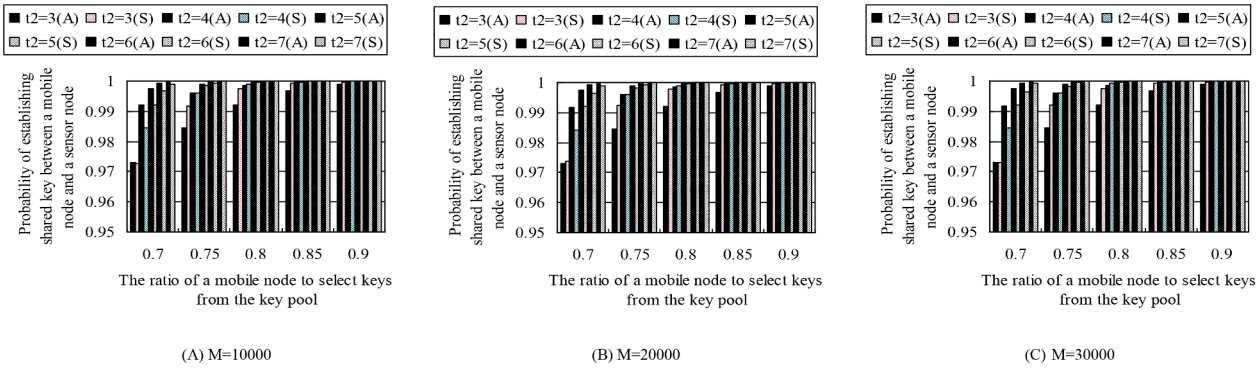


Fig. 1. The probability of establishing shared key between an MS and an SN as a function parameters $t1$, $t2$ and M. In the Fig., marks (A), (S) represent theoretical and simulated values, respectively.

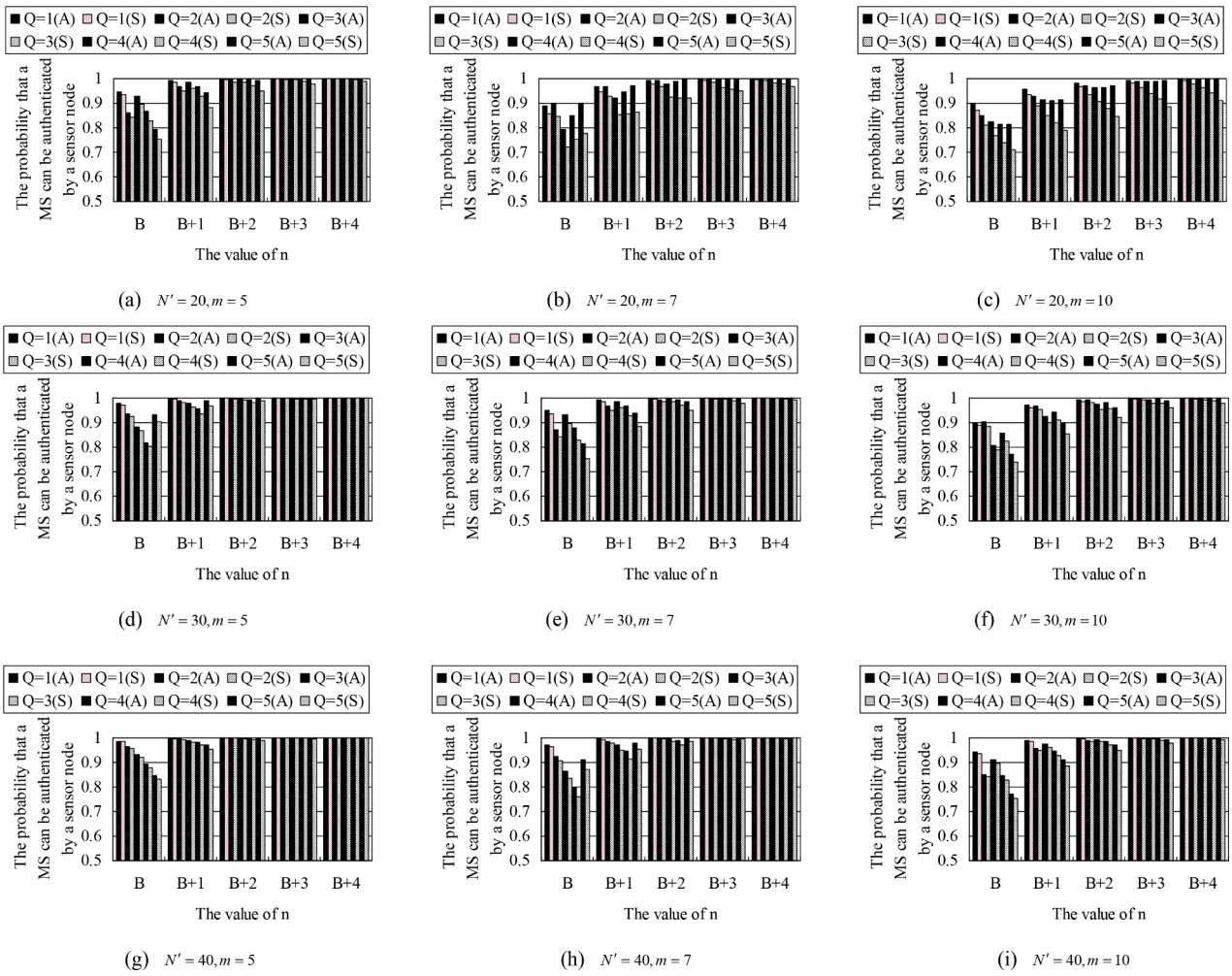


Fig. 2. The probability that an MS can be authenticated by an SN as a function parameters $N^{'}$, *m, n, Q*, where $B = \left\lceil \frac{Q}{1-m/N'} \right\rceil$.

large amount of false data, it may make a wrong judgment. So, this scheme must ensure that the probability of false data being received by BS is very low.

In our scheme, resilience against false data injection attacks namely Pr_f, is measured by the probability that a replication MS can impersonate a uncompromised SN to provide BS with false data that can be received by it when *x* SNs are compromised.

Suppose CM_xrepresents the expected value of the number of keys obtained by an adversary after compromising *x* SNs. Under the random pre-distribution key model, CM_x can be estimated by the following formula:

$$CM_x = \begin{cases} 0 & , x = 0 \\ t2 & , x = 1 \\ \sum_{j=0}^{t2-1} (t2-j) \times \frac{\binom{CM_{x-1}}{j}\binom{M-CM_{x-1}}{t2-j}}{\binom{M}{t2}} & , x > 1 \end{cases} \tag{21}$$

From the formula (21), it can be concluded that when *M* and t2 remain constant, CM_xincreases as *x* increases. As CM_x increases, the probability that pre-distribution keys of a newly captured SN are from the compromised key pool also increases, which leads to a decrease in the growth rate of CM_x as *x* increases. In Fig. 3, when $M = 10{,}000$, $t2{=}6$ and *x* increases from 1800 to 1900 and from 1900 to 2000, the increments of the compromised key pool are about 194 and 184 respectively. When *x* and *t*2 remain unchanged, as *M* increases, the times of keys being repeatedly distributed decreases, that is, the same number of SNs are compromised, and the probability of keys being repeatedly compromised decreases, which eventually leads CM_x to grow faster. In Fig. 3, when $t2{=}6$ and *M* increases from 10,000 to 30,000, CM_{2000} increases from about 6960 to 9800; when *M* and *x* are constant, CM_x increases significantly as *t*2 increases. As shown in Fig. 3, when $M = 20{,}000$, $x = 2000$ and t2 increases from 3 to 6, CM_{2000}increases from about 5250 to 9066.

Given $CM_x$is known, Pr_f can be estimated by the following formula:

$$\mathrm{Pr}_f = \frac{\binom{CM_x}{t2}}{\binom{M}{t2}} \tag{22}$$

From the formula (22), it can be concluded that when the number of compromised SNs and *t*2 are fixed, increasing *M*, Pr_f will decrease. For example, in Fig. 4, when $x = 2000$, $t2{=}7$, and *M* increases from 10,000 to 30,000, Pr_f decreases from about 0.14 to about 0.001. When the number of compromised SNs and *M* are fixed, there is no linear relationship between *t*2 and Pr_f. For example, in Fig. 4, when the number of compromised SNs is 2000, $M = 10{,}000$ and *t*2 increases from 3 to 7, the value of Pr_f is about 0.08, 0.09, 0.1, 0.12, 0.14, respectively. But when the number of compromised SNs is 2000, $M = 20{,}000$ or 30,000, Pr_f does not increase with the increase of *t*2 (see Fig. 4). In order to clarify the setting problem of parameters *t*2 and *M*, we set the value of Pr_f, and combine the formula (21) and (22) to obtain the relationship between the number of compromised SNs, *t*2 and *M*. Fig. 5 shows the maximum

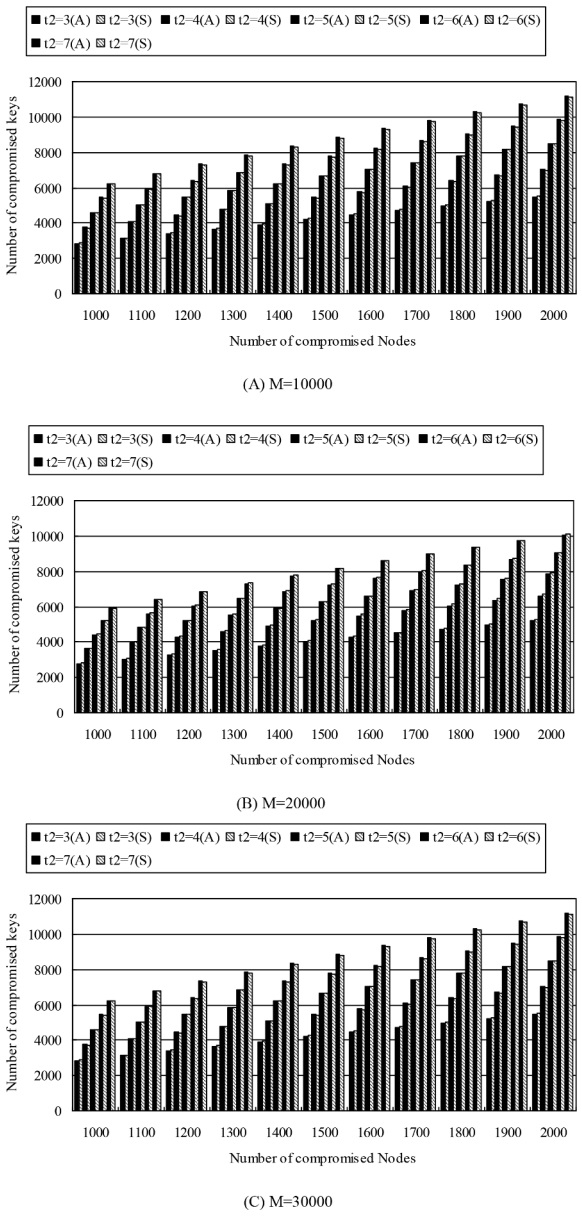


Fig. 3. Number of compromised keys as a function parameters $t2$ and M.

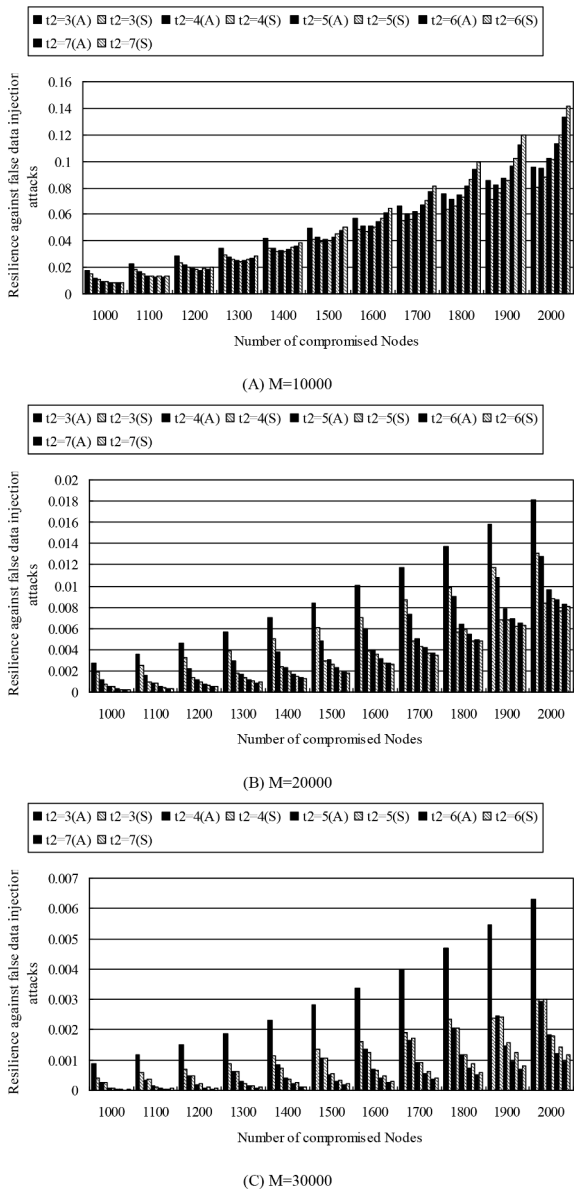


Fig. 4. Resilience against false data injection attacks as a function parameters $t2$ and M.

number of SNs allowed to be compromised as a function of $t2$ and M when Pr_fdoesn't exceed the set value. WhenPr_f is fixed, the larger the number of SNs are allowed to be compromised, the better the resistance is. From Fig. 5, we can draw the following two conclusions:

1. When$0.006 \leq Pr_f \leq 0.01$, t2 is best set to 7;

2. When $Pr_f$and t2 remain unchanged, M is approximately proportional to the maximum number of SNs allowed to be compromised. For example, when $Pr_f = 0.01$, t2=7 and $M = 10{,}000$, 20,000 and 3000, the maximum number of SNs allowed to be compromised is 1041, 2084, and 3126, respectively. That is to say, when M is expanded to 2 times and 3 times of the original, the maximum number of SNs allowed to be compromised is also expanded to about 2 times and 3 times of the original, respectively. According to this conclusion, we can set up multiple independent key pools, and the number of nodes supported by each key pool is determined by the maximum number of nodes allowed to be compromised when $Pr_f = 0.01$ and t2=7. This setting has the following benefits:

a) We can ensure that $Pr_f \leq 0.01$;
b) In this paper, when SNs, which are pre-distributed with keys from the same key pools, are essentially compromised, these key pools are called basically compromised key pools. In this case, we cannot pre-distribute keys from the basically compromised key pools to MSs, so that these nodes, which are pre-distributed keys from the compromised key pools, cannot be authenticated by MSs, and cannot send forged information to MSs;
c) Large networks can be divided into regions; the data of each region can be collected by an MS. Each region uses an independent key pool, which can reduce the overhead of MSs' storage, but also cannot affect the resistance of UWSNs.

4.3. Resistance against MS replication attacks

Resistance against MS replication attacks, namelyPr_r, is measured by the probability that original data of an uncompromised SN can be obtained by replication MSs when x SNs are compromised.

In our scheme, a replication MS can obtain the original data of an SN,

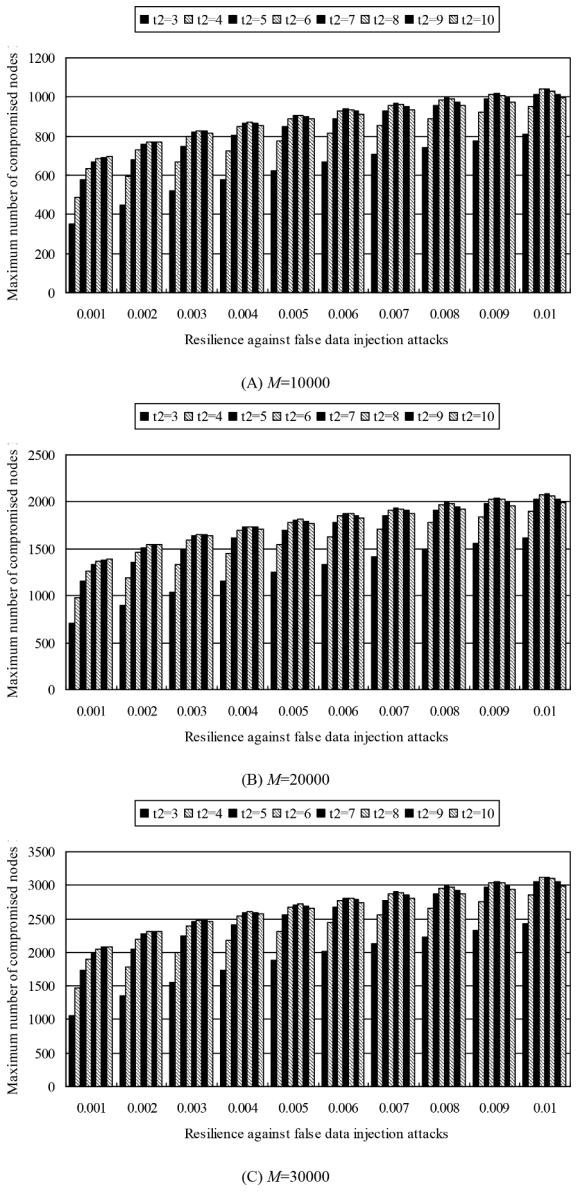


Fig. 5. Maximum number of compromised SNs as a function parameters Pr_s, $t2$ and M.

the following two conditions must be satisfied:

1. The replication MS can be authenticated by the SN. Because an SN sends its data to an MS if and only if the SN can authenticate the MS;
2. The replication MS has all pre-distribution keys stored by the SN. Only in this way, the replication MS can decrypt the data collected from the SN.

Here, the following two cases are discussed.

1. If the MS has been compromised or controlled, the MS only needs to satisfy clause 2 to collect data from the SN. In this case, the resistance against MS replication attacks is essentially the same as the resistance against false data injection attacks, and will not be described in detail here.
2. If the MS is not compromised or controlled, the MS needs to satisfy the above two conditions to obtain the original data of the SN. In this case, Pr_r can be estimated by the following formula:

$$Pr_r = Pa'_{M\_S} \times Pr_f \tag{23}$$

where Pa'_{M_S} indicates that the probability of the replication MS being authenticated by an uncompromised SN. During the authentication process, the SN requires the replication MS to provide at least Q correct authentication messages. Assuming that there exist $x1$ compromised SNs among the n neighbor nodes selected by the SN to participate in the authentication, when $x1 \geq Q$, the replication MS can be authenticated by the SN; otherwise, there exist at least Q-x SNs whose pre-distribution keys, which are shared with the replication MS, are compromised, the replication MS can be authenticated by the SN. Therefore, Pa'_{M_S} can be estimated by the following formula:

$$Pa'_{M\_S} = \sum_{j=\max(0,n-m)}^{\min(N'-m,n)} \frac{\binom{N'-m}{j}\binom{m}{n-j}}{\binom{N'}{n}} \times (\Pr)^{\max(Q+j-n,0)} \tag{24}$$

where Pr represents the probability of an SN's pre-distribution keys shared with an replication MS being compromised, so Pr can be estimated by the following formula:

$$\Pr = \frac{\sum_{l=1}^{t2}\binom{t1}{l}\binom{M-t1}{t2-l} \times \left(\frac{CM_x}{M}\right)^l}{\binom{M}{t2}} \tag{25}$$

Fig. 6 shows that Pr_r as a function parameters m, M, Q and N' when MSs are not compromised. In the figure, M is determined according to the rules of Section 4.2 and n is determined by Section 4.1.2, that is, $n = B + 4$. It can be seen from Fig. 6 that the theoretical value is basically consistent with the simulation result, but the theoretical value fluctuates, which is mainly related to the setting method of the parameter B. From Eq. (24), it can be concluded that Pa'_{M_S}increases as $\frac{m}{N'}$increases. And from Eq. (23), it can be concluded that Pr_r increases as Pa'_{M_S}increases. This can be confirmed from Fig. 6. For example, when $Q = 1$, the size of the key pool is $\frac{M}{2}$, $\frac{m}{N'}$ increases from $\frac{1}{4}$ to $\frac{1}{2}$, the simulated value of Pr_rincreases from about 0.0001 to 0.013. In addition, if the size of the key pool is determined according to the method introduced in this paper, our scheme can achieve excellent resistance against MS replication attacks. For example, when the key pool size is M, $Q = 1$, $m = 10$ and $N' = 20$, the simulated value of $Pr_r$is only about 0.0002.

5. Conclusion

This paper proposes a security model against MS replication attacks for UWSNs in which data is collected by MSs. In this model, MSs can complete the authentication with SNs, but cannot decrypt the data collected from SNs. Under the assumptions that BS is secure, both MSs and SNs can be compromised, and replication MSs can pretend to be uncompromised SNs to launch false data injection attacks, theoretical analysis and simulation show that when the number of SNs is about one-tenth of the size of a key pool, t1 is greater than or equal to three-quarters of the size of a key pool, and t2=7, the model can ensure that MSs and SNs can authenticate with each other with high probability and can have excellent resistance against MS replication attack and false data injection attack.

Author statement

All persons who have made substantial contributions to the work reported in the manuscript. We have reviewed the final version of the manuscript and approve it for publication. To the best of our knowledge and belief, this manuscript has not been published in whole or in part nor is it being considered for publication elsewhere. In addition, all authors declare no conflicts of interest.

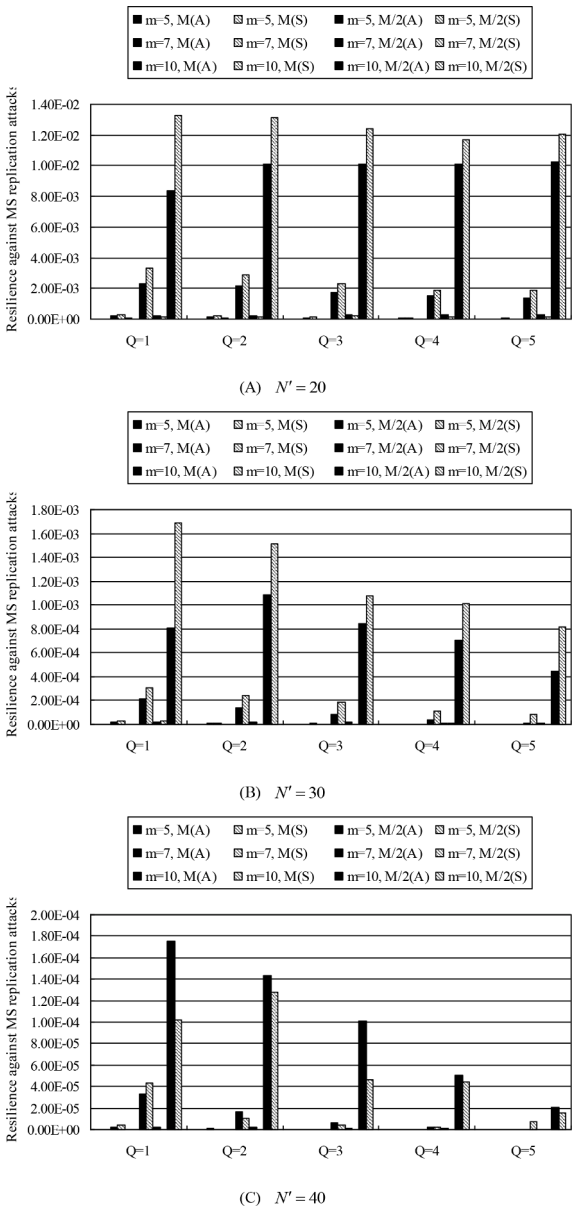


Fig. 6. Resilience against MS replication attacks as a function parameters *m, M, Q* and $N^{'}$.

Declaration of Competing Interest

The authors declared that they have no conflicts of interest to this work. We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported in part by the Natural Science Foundation of Guangdong Province, China, under Grant 2020A1515010923; in part by Guangdong Province Key Construction Discipline Scientific Research Ability Improvement Project, China, under Grant 2021ZDJS068; in part by the Talent Introduction Project of Shaoguan University, China, under Grant 99000618 and Grant 99000619; in part by the Construct Program of the Key Discipline in Hunan Province, China; in part by the Aid Program for Science and Technology Innovative Research Team in Higher Educational Institute of Hunan Province, China.

References

[1] D. Ma, C. Soriente, G. Tsudik, New adversary and new threats: security in unattended sensor networks, IEEE Netw 23 (2) (Mar. 2009) 43–48.
[2] R. Di Pietro, G. Oligeri, C. Soriente, G. Tsudik, United we stand: intrusion resilience in mobile unattended WSNs, IEEE Trans. Mobile Comput. 12 (7) (Jul. 2013) 1456–1468.
[3] Y. Wang, K. Chen, Efficient path planning for a mobile sink to reliably gather data from sensors with diverse sensing rates and limited buffers, IEEE Trans. Mobile Comput. 18 (7) (2019) 1527–1540.
[4] S. Redhu, R.M. Hegde, Cooperative network model for joint mobile sink scheduling and dynamic buffer management using Q-learning, IEEE Trans. Netw. Serv. Manag. 17 (3) (2020) 1853–1864.
[5] Z. Lin, H. Keh, R. Wu, et al., Joint data collection and fusion using mobile sink in heterogeneous wireless sensor networks, IEEE Sens. J. 21 (2) (2021) 2364–2376.
[6] Z. Guan, Z. Lv, X. Sun, et al., A differentially private big data nonparametric bayesian clustering algorithm in smart grid, IEEE Trans. Netw. Sci. Eng. 7 (4) (2020) 2631–2641.
[7] Lei Xu, Chunxiao Jiang, Yi Qian, et al., Privacy-accuracy trade-off in differentially-private distributed classification: a game theoretical approach, IEEE Trans. Big Data 7 (4) (2021) 770–783.
[8] Y. Ren, V.A. Oleshchuk, F.Y. Li, Optimized secure and reliable distributed data storage scheme and performance evaluation in unattended WSNs, Comput. Commun. 36 (9) (May 2013) 1067–1077.
[9] Y. Ren, V.I. Zadorozhny, V.A. Oleshchuk, F.Y. Li, A novel approach to trust management in unattended wireless sensor networks, IEEE Trans. Mobile Comput. 13 (7) (Jul. 2014) 1409–1423.
[10] Z. Zheng, A. Liu, L.X. Cai, et al., Energy and memory efficient clone detection in wireless sensor networks, IEEE Trans. Mobile Comput. 15 (5) (2016) 2031–2046.
[11] P.Y. Lee, C.M. Yu, T. Dargahi, et al., MDSClone: multidimensional Scaling Aided Clone Detection in Internet of Things, IEEE Trans. Inf Forens Secur 13 (8) (2018).
[12] S. Chen, Z. Pang, H. Wen, et al., Automated labeling and learning for physical layer authentication against clone node and sybil attacks in industrial wireless edge networks, IEEE Trans. Ind. Inf. 17 (3) (2021) 2041–2051.
[13] L. Eschenauer, V.D. Gligor, A key-management scheme for distributed sensor networks, in: Proc. ACM CCS, 2002, pp. 41–47.
[14] H. Chan, A. Perrig, D. Song, Random key predistribution schemes for sensor networks, in: Proc. IEEE Symp. Security Privacy, 2003, pp. 197–213.
[15] C. Blundo, et al., Perfectly-secure key distribution for dynamic conferences, Inf. Comput. 164 (1) (1998) 1–23.
[16] K. Ren, K. Zeng, W. Lou, A new approach for random key predistribution in large-scale wireless sensor networks, Wireless Commun. Mobile Comput. 6 (3) (2006) 307–318.
[17] S. Li, B. Zhou, J. Dai, X. Sun, A secure scheme of continuity based on two-dimensional backward hash key chains for sensor networks, IEEE Wireless Commun. Lett. 1 (5) (Oct. 2012) 416–419.
[18] B. Zhou, S. Li, Q. Li, X. Sun, X. Wang, An efficient and scalable pairwise key pre-distribution scheme for sensor networks using deployment knowledge, Comput. Commun. 32 (1) (2009) 124–133.
[19] W. Bechkit, Y. Challal, A. Bouabdallah, A new class of hash chain based key pre-distribution schemes for WSN, Comput. Commun. 36 (3) (2013) 243–255.
[20] F. Gandino, R. Ferrero, M. Rebaudengo, A key distribution scheme for mobile wireless sensor networks: q-s composite, IEEE Trans. Inf. Forens. Secur. 12 (1) (2017) 34–47.
[21] J. Zhao, Topological properties of secure wireless sensor networks under the q -Composite key predistribution scheme with unreliable links, IEEE/ACM Trans. Netw. 25 (3) (2017) 1789–1802.
[22] M. Sood, O. Yağan, On the minimum node degree and k-connectivity in inhomogeneous random K-out graphs, IEEE Trans. Inf. Theory 67 (10) (2021) 6868–6893.
[23] B. Zhou, J. Wang, S. Li, Y. Cheng, J. Wu, A continuous secure scheme in static heterogeneous sensor networks, IEEE Commun. Lett. 17 (9) (Sep. 2013) 1868–1871.
[24] A. Rasheed, R.N. Mahapatra, Key predistribution schemes for establishing pairwise keys with a mobile sink in sensor networks, IEEE Trans. Parallel Distrib. Syst 22 (1) (Jan. 2011) 176–184.
[25] R. Eletreby, O. Yağan, Connectivity of wireless sensor networks secured by heterogeneous key Predistribution under an On/Off channel model, IEEE Trans. Control Netw. Syst. 6 (1) (2019) 225–235.
[26] W. Du, J. Deng, Y.S. Han, P.K. Varshney, A key predistribution scheme for sensor networks using deployment knowledge, IEEE Trans. Depend. Secure Comput. 3 (1) (Mar. 2006) 62–77. Jan./.
[27] Z. Yu, Y. Guan, A key management scheme using deployment knowledge for wireless sensor networks, IEEE Trans. Parallel Distrib. Syst. 19 (10) (Oct. 2008) 1411–1425.
[28] A. Fanian, M. Berenjkoub, H. Saidi, T.A. Gulliver, A high performance and intrinsically secure key establishment protocol for wireless sensor networks, Comput. Netw. 55 (8) (2011) 1849–1863.

[29] B. Zhou, J. Wang, S. Li, W. Wang, A new key predistribution scheme for multiphase sensor networks using a new deployment mode, J. Sensors 2014 (May 2014) 10.
[30] B. Zhou, J. Wang, S. Li, W. Wang, Y. Cheng, J. Wu, A secure scheme based on layer model in multi-phase sensor networks, IEEE Commun. Lett. 20 (7) (Jul. 2016) 1421–1424.
[31] S. Li, B. Zhou, Q. Hu, et al., A secure scheme based on one-way associated key management model in wireless sensor networks, IEEE Internet Thing. J. 8 (4) (2021) 2920–2930.
[32] S. Li, W. Wang, B. Zhou, J. Wang, Y. Cheng, J. Wu, A secure scheme for heterogeneous sensor networks, IEEE Wireless Commun. Lett. 6 (2) (Apr. 2017) 182–185.
[33] A. Rasheed, R.N. Mahapatra, The three-tier security scheme in wireless sensor networks with mobile sinks, IEEE Trans. Parallel Distrib. Syst. 23 (5) (May 2012) 958–965.
[34] S. Li, W. Wang, B. Zhou, J. Wang, Y. Cheng, J. Wu, A (M,m) authentication scheme against mobile sink replication attacks in unattended sensor networks, IEEE Wireless Commun. Lett. 7 (2) (Apr. 2018) 250–253.
[35] B. Zhou, S. Li, W. Wang, et al., An efficient authentication scheme based on deployment knowledge against mobile sink replication attacks in UWSNs, IEEE IoT J. 6 (6) (2019) 9738–9747.
[36] R. Anderson, M. Kuhn, Tamper resistance-a cautionary note, in: Proc.1996 Usenix Workshop Electronic Commerce, 1996, pp. 1–11.
[37] Q. Zhang, T. Yu, P. Ning, A framework for identifying compromised nodes in wireless sensor networks, ACM Trans. Inf. Syst. Secur. 11 (3) (2008) 1–37.
[38] A. Yessembayev, D. Sarkar, F. Sikder, Detection of good and bad sensor nodes in the presence of malicious attacks and its application to data aggregation, IEEE Trans. Signal Inf. Process. Netw. 4 (3) (2018) 549–563.
[39] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, An application specific protocol architecture for wireless microsensor networks, IEEE Trans. Wirel. Commun. 1 (4) (2002) 660–670.

Boqing Zhou received his pH. D. degree in computer science from Hunan University, China, in 2011. His-current research interests include sensor networks and information security.

Sujun Li received his pH. D. degree in computer science from Central South University, China, in 2018. Her current research interests include sensor networks and information security.

Jianxin Wang received his PhD degree in computer science from Central South University, China, in 2006. Currently, he is working as a professor at the department of Information Science and Engineering, Central South University, Changsha, Hunan, PR China. His-current research interests include algorithm analysis and optimization, computer network and bioinformatics. He has published more than 100 papers in various International journals and refereed conferences. He is a senior member of the IEEE.

Yun Cheng received his PhD degree in computer science from National University of Defense Technology, China, in 2006. Currently, he is working as a professor at the school of information, Hunan University of Humanities, Science and Technology, Loudi, Hunan, PR China. His current research interests include algorithm analysis and optimization, computer network and optical communication technology.

Jie Wu is the Director of the Center for Networked Computing and Laura H. Carnell professor at Temple University. He also serves as the Director of International Affairs at the College of Science and Technology. He served as Chair of the Department of Computer and Information Sciences from the summer of 2009 to the summer of 2016 and Asscoaite Vice Provost for International Affairs from the fall of 2015 to the summer of 2017. Prior to joining Temple University, he was a program director at the National Science Foundation and was a distinguished professor at Florida Atlantic University. His current research interests include mobile computing and wireless networks, routing protocols, network trust and security, distributed algorithms, applied machine learning, and cloud computing. Dr. Wu regularly published in scholarly journals, conference proceedings, and books. He serves on several editorial boards, including IEEE Transactions on Mobile Computing, IEEE Transactions on Service Computing, Journal of Parallel and Distributed Computing, and Journal of Computer Science and Technology. Dr. Wu is/was general chair/co-chair for IEEE DCOSS'09, IEEE ICDCS'13, ICPP'16, IEEE CNS'16, WiOpt'21, ICDCN'22, IEEE IPDPS'23, and ACM MobiHoc 2023 as well as program chair/cochair for IEEE MASS'04, IEEE INFO COM'11, CCF CNCC'13, and ICCCN'20. He was an IEEE Computer Society Distinguished Visitor, ACM Distinguished Speaker, and chair for the IEEE Technical Committee on Distributed Processing (TCDP). Dr. Wu is a Fellow of the AAAS and a Fellow of the IEEE. He is the recipient of the 2011 China Computer Federation (CCF) Overseas Outstanding Achievement Award.