Secure Time Synchronization Protocol for Wireless Sensor Network Based on uTESLA Broadcasting Protocol

Xiaogang Wang[1,2], Weiren Shi[1]

(Corresponding author: Xiaogang Wang)

College of Automation, Chongqing University[1]
No. 174, Shapingba Road, Shapingba District, Chongqing 400044, China
School of Automation and Information Engineering, Sichuan University of Science and Engineering[2]
No. 108, XueYuan Road, Ziliujing District, Zigong City, Sichuan 643000, China
(Email: wxg_zf@163.com)

(Received July 21, 2016; revised and accepted Feb. 2, 2017)

Abstract

A secure time synchronization protocol (STSP) for wireless sensor network based on uTESLA protocol is proposed according to the issues that DCS algorithm in wireless sensor network time synchronization is limited in the threats of compromised nodes, big communication cost and deficiency in coverage. Firstly, initializing the network topology based on LEACH protocol to balance the network node load and prolong network life cycle. Secondly, excluding the malicious nodes based on uTESLA protocol for making security condition check positively before the network time synchronization. Thirdly, making time synchronization quickly based on TPSN protocol for base station to cluster head node. Fourthly, making time synchronization based on HRTS protocol for cluster head nodes to their own common nodes. Lastly, making reverse authentication based on uTESLA protocol for checking and excluding the compromise nodes after the network time synchronization. The security analysis and simulation show that two times authentication ensure the absolute security of the time synchronization work in STSP, and STSP is much better than DCS in synchronization cycle, precision, ratio and cost.

Keywords: Broadcast Authentication; Time Synchronization; Wireless Sensor Network

1 Introduction

Wireless sensor network (WSN) is a new distributed system in which the nodes are independent and communicate wirelessly [2]. In particular, each node maintains a local clock and the timing signal of each node clock is generally maintained by an inexpensive crystal oscillator, and because of the limitation of crystal oscillator manufacturing process, it is easy to be influenced by the external factors in the process of operation, which leads to the deviation of the time ratio of the node, it's also known as the time out of step [4, 7, 11, 12, 20, 21, 30]. So it is necessary to regularly carry out network time synchronization for maintaining the consistency of the local clock nodes.

Time synchronization is the process of providing a unified time scale for a distributed system by doing some operations on the local clock. Network time protocol (NTP) is the standard of the time synchronization protocol on the internet [23], which is used to synchronize the computer time with universal time coordinated (UTC) and obtain a high precision time by the external connection of a time receiver (such as WWVB, GPS, *etc.*). However, NTP, GPS and other similar traditional time synchronization technology can't be directly applied to WSN because of the following three differences [27]:

1) The sensor nodes in WSN are limited in volume, power supply, computing power, storage space, which causes the NTP protocol can't be run on sensor nodes.

2) There are great differences in bandwidth, anti-interference ability, and the ability to resist weak between them, because WSN uses wireless transmission mode, and the tradition internet mainly uses a reliable cable transmission mode.

3) WSN applications are highly localized or local optimum, while the tradition internet emphasizes the overall optimality.

There have been a number of protocols for WSN time synchronization proposed about the research topic of time synchronization for WSN in recent years. Such as: group authentication and group key distribution for

Ad-Hoc networks(GAGKD) [29],an accurate on-demand time synchronization protocol (AODTSP) [16] and long term and large scale time synchronization (LTLSTS) [17] proposed by Huang Ge, timing-sync protocol for sensor networks (TPSN) [10], improved time synchronization in ML-MAC [19], analysis of quantities influencing the performance of time synchronization(AQIPTS) [3], tiny-sync/mini-sync(TS/MS) [5], hierarchy referencing time synchronization protocol (HRTS) [1], *etc.* Although these time synchronization protocols have achieved good performance from the perspective of each highlighted, they are only applicable to the benign environment without any malicious nodes. While WSN is usually used in some military and commercial areas, it's inevitably that there will be a variety of malicious nodes attacks [13, 15, 22, 26, 31], such as that the malicious nodes can forge the time synchronization message, send the synchronization message containing the error time information, delay sending the synchronization message or not, and destroy the WSN normal time synchronization process [6, 8, 9, 18, 24, 28]. Therefore, the security issues of WSN time synchronization are particularly important.

At present, there are only a few protocols on the security aspects for WSN time synchronization, where diffusion-based clock synchronization (DCS) is the most typical one, which ensures the safety of time synchronization based on receiving 2s+1 synchronization messages from last layer nodes, but it caused a lot of difficulty and communication costs from beginning. In this paper, a secure time synchronization protocol (STSP) for wireless sensor network based on uTESLA protocol is proposed according to the issues that DCS algorithm is limited in the threats of compromised nodes, big communication cost and deficiency in coverage. Firstly, initializing the network topology based on LEACH protocol to balance the network node load and prolong network life cycle. Secondly, excluding the malicious nodes based on uTESLA protocol for making security condition check positively before network time synchronization. Thirdly, making time synchronization quickly based on TPSN protocol for base station to cluster head node. Fourthly, making time synchronization based on HRTS protocol for cluster head nodes to their own common nodes. Lastly, making reverse authentication based on uTESLA protocol for checking and excluding the compromise nodes after network time synchronization. The security analysis and simulation show that two times authentication ensure the absolute security of the time synchronization work in STSP, and STSP is much better than DCS in synchronization cycle, precision, ratio and cost.

This paper is organized as follows. In Section 2, analyze the related work, such as DCS algorithm principle and its issues. In Section 3, discuss the specific principle of STSP, including network model assumptions, initialization and STSP steps. In Section 4, analyze the security of STSP, and make a simulation compared with DCS. Lastly, make a summary in Section 5.

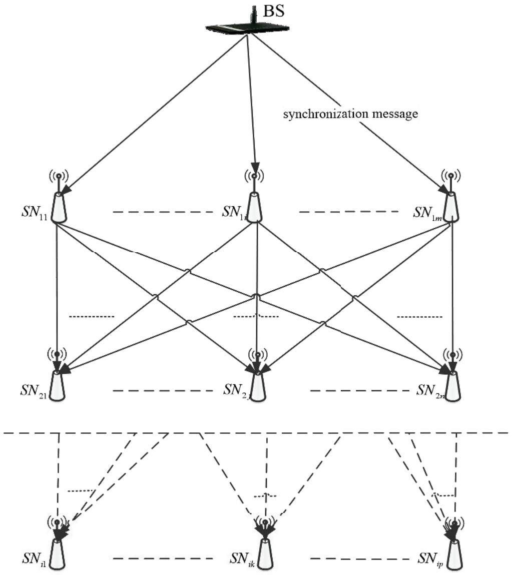

Figure 1: DCS algorithm

2 The Related Work

2.1 DCS Algorithm

DCS algorithm is a cyclical secure time synchronization algorithm for WSN: on the one hand, all nodes in the network can be synchronized to the reference node through the multi-hop manner; on the other hand, providing redundant paths for each receiving node in the multi-hop process, which can offset and tolerate the attacks of malicious nodes on some paths, and achieve the goal of secure time synchronization.

The execution process of DCS algorithm is shown in Figure 1.

- Firstly, the reference node BS(base station) sends the synchronization message to its neighbor nodes SN_{1i},(1...m).
- Secondly, SN_{1i}sends the synchronization message to its neighbor nodesSN_{2j},(1...n) when SN_{1i} completes the calibration of the clock.
- Thirdly, for tolerating the attacks of malicious nodes, the node SN_{ij} $(i > 1)$ needs to receive more than 2s+1 synchronization messages from last layer nodes$SN_{(i-1)j}$, and takes the average of these 2s+1 time messages to calibrate their own local clock.
- Lastly, the synchronous message is passed in turn until the whole network nodes can be synchronized.

2.2 DCS Issues

1) Can not guarantee that all nodes SN_{1i},(1...m) are non-malicious nodes

The nodes SN_{1i},(1...m) do not calibrate the local time by receiving 2S+1 synchronization messages from last layer node BS, so there may be some malicious or compromised nodes in SN_{1i},(1...m), and it caused a lot of difficulty and communication costs from beginning.

2) Serious error accumulation

   It's obviously shown in Figure 1 that the more far away from BS, the more error accumulation. We know that the calibrated time of node SN_{ij} $(i > 1)$ is an estimated value which is not precise, because the node SN_{ij} $(i > 1)$ needs to take the average of 2s+1 time messages receiving from last layer nodes $SN_{(i-1)j}$ to calibrate their own local clock, while the time message from $SN_{(i-1)j}$ is also an estimated value because of the same principle. So the error will be accumulated layer by layer.

3) Communication cost

   It is bound to increase the redundancy of the message in the network and lead to the increase of the communication cost, because that all nodes need to receive more than 2S+1 synchronization messages from last layer nodes.

4) The condition of 2S+1 is difficult to meet

   It is not all the nodes in the network can receive more than 2S+1 synchronization messages from last layer. For instance, the neighbor nodes are less than 2S+1, or the neighbor nodes are malicious nodes. So the DCS algorithm coverage is not good.

5) Compromised node threat

   Although the authentication method can be used to defend against the attacks from external malicious nodes, the attacker can still attack the time synchronization process by compromised nodes. Especially for the multi-hop time synchronization process, if the intermediate node is compromised node, this effect is fatal.

3 STSP

3.1 Network Model Assumptions

In order to facilitate the description of STSP, the network is assumed as follows:

1) Assume that the network is isomorphic and static, each sensor node has been uniformly deployed in the target area and has same configure in software and hardware, where the network size is N, including 3 types of nodes: base station BS, cluster head node CH, common sensor node SN, planning network topology by LEACH protocol [14], as shown in Figure 2.

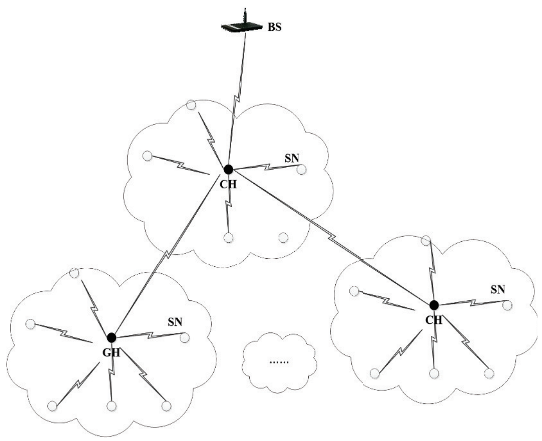

Figure 2: The network topology based on LEACH

2) Assume that base station BS is the time reference node for the network and equipped with abundant software and hardware resources, it is responsible for storing the basic information of all the nodes and has the ability to detect compromised or captured nodes.

3) Assume that common sensor node SN is responsible for collecting environmental data. The ability to process data of sensor node is limited by storage space, energy reserves, and communication distance. The messages between communication nodes which are not in the communication radius should be transferred by their neighbor node.

4) Assume that cluster head node CH is selected from the common nodes based on LEACH and responsible for the data transmission between SN and BS.

The main symbols in the text are shown in Table 1.

Table 1: Explanation of symbols

Symbol	Implication
BS	base station
CH	cluster head
SN	sensor node
h(x)	hash function
K	authentication key
ID_{SN_i}	identity symbol of node SN_i
D	time slice length
δ	key delay time
P	plaintext
L(i)	authentication message of time slice i

3.2 STSP Principle

3.2.1 Initialization

1) Initialization network topology based on LEACH

In order to ensure that each node can obtain the synchronization message from the reference node, synchronous data packet switching as an important process of WSN time synchronization is usually based on the specific network topology path. There is no essential difference between synchronization topology and routing topology except the types of data packet on transmission path, so it can be effectively combined with synchronous topology and routing topology for reducing the energy consumption of the network.

Compared with flooding type network topology of DCS algorithm, the network topology structure based on clustering hierarchical is more suitable for WSN applications (as shown in Figure 2), such as the application of broadcast authentication, reducing the amount of network data traffic, and prolonging the survival time of the network. The most classical clustering protocol LEACH in WSN is chosen to initialize the network topology in this paper, and the selection of cluster head node is the key to LEACH protocol.

Assume that each common sensor node generates a random number between 0 and 1, and it will be the cluster head if some random number is less than a certain threshold value $T(n)$,and set Equation (1):

$$T(n) = \left\{ \begin{array}{ll} 1-p[rmod(1/p)] & n \in G \\ 0 & else \end{array} \right\} \quad (1)$$

Where, p is the percentage of desired cluster head nodes, r is the round, G is the common sensor nodes set in last 1/p round.

2) BS presets the initial parameters of each node SN_i, including the last key $K^0_{SN_i}$ of the key chain, the key delay time δ,the time slice length D, the beginning time T_0, the node identity ID_{SN_i}.

3) SN_i presets the initial parameters of BS, including the last key K^0_{BS} of the key chain, the key delay time δ, time slice length D, the beginning time T_0.

3.2.2 STSP Steps

Step 1. Making security condition check positively based on uTESLA broadcasting protocol.

WSN internal safety hazards are generally caused by the malicious nodes and the compromised nodes, but the malicious nodes cannot access the network without getting the network authentication key because of the local broadcast authentication protocol such as uTESLA, and it can't cause any bad effect for the network time synchronization process.

In uTESLA [25], the asymmetric characteristic of broadcast authentication is realized by using the symmetric encryption mechanism in condition of the loose time synchronization of sending nodes and receiving nodes. The key points of uTESLA protocol are using hash key chain and publishing key delayed, as showed in Figure 3, a one-way function key chain is established by the sending node, where the length of key chain is n+1, and the first key K_n of the key chain is generated randomly by the sending node, but the next keys are all generated by the one-way function acting on the last key repeatedly, such as $K_j = H(K_{j+1})$. The sending node divides the communication time into equal time slices, where the length of each time slice is D, and each time slice is assigned a key in order, but the order of the assigned keys is the opposite order of the key chain, and each message P_i of time slice j is encrypted by K_j, such as $MAC_{k_j}(P_i)$. The sending node determines the key delay time δ based on the time slice length, and the key K_j on time slice will be published after δ, such as $\delta = 2$ in Figure 3.

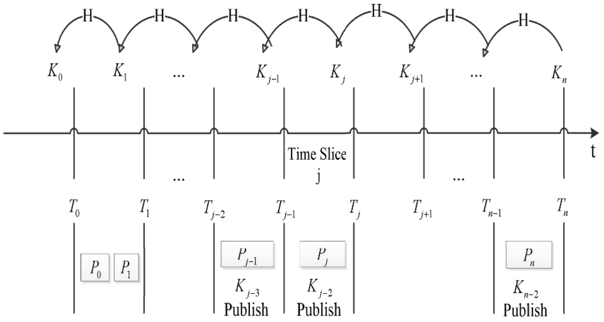

Figure 3: uTESLA protocol

To avoid the additional communication cost, the published key is sent to the receiving nodes by being attached with the data packet. If there is no data packet on some time slice, the key attached with the data packet won't be published, and this key can be calculated by the next keys in one-way function hash. More importantly, the initial parameters K_0, δ, D and starting time T_0 should be sent to receiving nodes before authentication.

The specific steps for making security condition check based on the network topology (as shown in Figure 2) are as follows:

Firstly, BS builds the broadcast authentication information $L(j)$, and assume that $L(j)$ is the broadcast authentication information of the time $j(t)$ on time slice j, where t is a certain time on time slice j, and set Equation (2):

$$L(j) = \left\{ \begin{array}{l} h^j_{BS}(P_{j(t)}, T_{BS}(j(t)), ID_{BS})|| \\ \\ P_{j(t)}||h^{j-2}_{BS}||ID_{BS}||T_{BS}(j(t)) \end{array} \right\} \quad (2)$$

Where, $P_{j(t)}$ is the plaintext message of time $j(t)$, $T_{BS}(j(t))$ is the standard reference time from BS on time $j(t)$, h^{j-2}_{BS} is the published key by BS on time $j(t)$.

Secondly, rebuilding the broadcast authentication information $L_{CH_{1i}}(j)$ when the neighbor cluster head nodes

CH_{1i} obtain $L(j)$, and set Equation (3):

$$L_{CH_{1i}}(j) = \left\{ \begin{array}{l} h^j_{BS}(P_{j(t)}, T_{BS}(j(t)), ID_{BS}) \\ ||h^{j-2}_{BS}||ID_{CH_{1i}}||T_{BS}(j(t)) \\ ||T_{CH_{1i}}||P_{j(t)}||ID_{BS} \end{array} \right\} \quad (3)$$

And then CH_{1i} send $L_{CH_{1i}}(j)$ to their member nodes and neighbor cluster nodes CH_{2i}. Where $T_{CH_{1i}}$ is the local time of CH_{1i}.

Lastly, each node can get the information $L(j)$ based on receiving the broadcast authentication information from their neighbor nodes one by one, and waiting for getting the published key h^j_{BS} by BS after δ to certificate the correctness of $h^j_{BS}(P_{j(t)}, T_{BS}(j(t))$ which can illustrate the identity of BS and the correctness of source attestation. In this time, each node can verify whether the key h^j_{BS} has been published based on $h^{j-2}_{BS} = H^2(h^j_{BS})$: if h^j_{BS} has been published by BS, that each node which has obtained the key h^j_{BS} can forge or tamper with the information $L(j)$, so the information $L(j)$ from these nodes will be judged to be unsafe, and abandon it; if h^j_{BS} has not been published by BS, that each $L(j)$ will be cached until that h^j_{BS} is published.

In this step, on the one hand, the external malicious nodes couldn't get the authentication key to participate in the authentication work, and the external malicious nodes are excluded; on the other hand, if the compromised nodes forge or tamper with the information $L(j)$, that the wrong information $L(j)$ can be detected by the verification step whether the key h^j_{BS} is published based on $h^{j-2}_{BS} = H^2(h^j_{BS})$, and the compromised nodes are detected and excluded.

Therefore, Step 1 is called to be the security condition check because of the excluding of malicious nodes and compromised nodes.

Step 2. The time synchronization of the cluster head nodes.

After the completion of the security condition check in Step 1, it is the time to make time synchronization for the network. In this paper, the first time synchronization work is to realize the time synchronization of the cluster head nodes. Because the proportion of the cluster head nodes in the network is very small, TPSN is the most suitable synchronization protocol, which can avoid large amounts of computation in flooding network topology and keep high precision.

The realization of TPSN is divided into 2 stages: layer discovery and synchronization.

Stage 1. Layer discovery

Assume that BS is the first layer of network called Layer 0, and the cluster head nodes are divided into different layers by receiving the hierarchical data packet from their neighbor cluster nodes. Assume that the hierarchical data packet DP_i includes ID_{BS} and the layer grade L_i, set $DP_i = (ID_{BS}||L_i)$, such as that cluster nodes $CH_{1j}(j > 1)$ receive the packet $DP_0 = (ID_{BS}||L_0)$ from BS, they all needs to reset the layer grade L_1, and broadcast the new packet $DP_1 = (ID_{BS}||L_1)$ to the next neighbor nodes $CH_{2j}(j > 1)$. The layer discovery rule is that each cluster head just receives the first hierarchical data packet from the neighbor cluster heads.

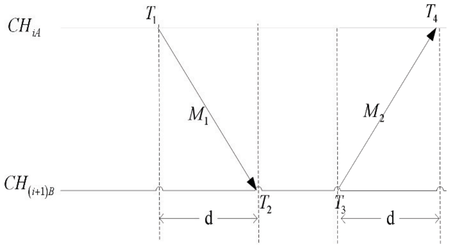

Figure 4: TPSN protocol

Stage 2. Synchronization

After layer discovery phase, BS starts the synchronization work by broadcasting the time synchronization data packet. As shown in Figure 4, there is a mutual information exchange between two neighbor cluster heads CH_{iA} and $CH_{(i+1)B}$, CH_{iA} sends the synchronization information packet at its' local time T_1, $CH_{(i+1)B}$ receives the packet at its' local time T_2, where $T_2 = (T_1 + d + \Delta)$, Δ is the time deviation between CH_{iA} and $CH_{(i+1)B}$, d is the signal propagation delays, $CH_{(i+1)B}$ returns the confirmation packet at its' local time T_3, and CH_{iA} receives confirmation packet at its' local time T_4, where $T_4 = (T_3 + d + \Delta)$, so we can get d and Δ as shown in following Equations (4) and (5):

$$d = \{ [(T_1 - T_2) + (T_4 - T_3)]/2 \} \quad (4)$$

$$\Delta = \{ [(T_1 - T_2) - (T_3 - T_4)]/2 \} \quad (5)$$

So $CH_{(i+1)B}$ can make its local time consistent with last layer node CH_{iA} based on d, and it shows that each cluster node can get the precise time same as time reference node BS.

Step 3. The time synchronization of common sensor nodes.

In Step 2, the cluster head nodes have all completed the synchronization work and become the reference nodes for their own cluster members.

As shown in Figure 2, there are many common sensor nodes belong to the one cluster head based on LEACH, so each common sensor node needs to make time synchronization work consistent with their cluster head based on TPSN, which will cause a large amount of computation and error accumulation. But the all cluster members can achieve the time synchronization work by three times data communication based on HRTS protocol (as shown in Figure 5), which can reduce the computation greatly and maintain a high precision.

In the first time data communication, the reference node CH_{iA} broadcasts a synchronous request packet F_1 and records the sending time t_1, SN_i is the response node chosen randomly by CH_{iA} . All member nodes record the receiving time, and only SN_i needs to reply the response packet F_2, where SN_j records the receiving time t_{2j}, SN_i records the receiving time t_2, t_3 is the sending time of F_2 recorded by SN_i.

In the second time data communication, F_2 is sent to CH_{iA} by SN_i, where t_2 and t_3 are part of F_2, t_4 is the receiving time of F_2 recorded by CH_{iA}, so CH_{iA} can get t_1 to t_4 after receiving F_2 .

Assume that $\Delta^{'}$ is the time deviation between CH_{iA} and SN_i, $d^{'}$ is the signal propagation delays, T_r is the local time of CH_{iA}, T_p is the local time of SN_i, T_j is the local time of SN_j, so we can getT_r, t_2, t_4 in Equations (6) (7) (8):

$$T_r = \{ \; T_p - \Delta^{'} \; \} \tag{6}$$

$$t_2 = \{ \; t_1 + d^{'} + \Delta^{'} \; \} \tag{7}$$

$$t_4 = \{ \; t_3 + d^{'} - \Delta^{'} \; \} \tag{8}$$

And get $d^{'}$ and $\Delta^{'}$ in Equations (9) and (10):

$$d^{'} = \{ \; [(t_2 - t_1) + (t_4 - t_3)]/2 \; \} \tag{9}$$

$$\Delta^{'} = \{ \; [(t_2 - t_1) + (t_3 - t_4)]/2 \; \} \tag{10}$$

In the third time data communication, CH_{iA} broadcasts a new synchronous packet F_3 which includes t_2 and $\Delta^{'}$, and SN_j can correct its local time based on the following relationship in Equations (11)and (12):

$$T_p - T_j = \{ \; t_2 - t_{2j} \; \} \tag{11}$$

$$T_r = \{ \; T_j + t_2 - t_{2j} - \Delta^{'} \; \} \tag{12}$$

As the same way of SN_j, each cluster member node can correct its local time consistent with the reference node CH_{iA} and BS.

Step 4. Reverse authentication.

Although the synchronization work has been completed after Step 2 and Step 3, it's not sure whether all nodes in the network are synchronized, because there are some latent compromised nodes which can make some damage in the time synchronization process. So a reverse authentication method based on uTESLA is proposed for further detecting, and the specific steps are as follows:

Firstly, the cluster member node SN_i builds the reverse authentication information $L(j)^{'}$ (Equation (13)) and sends it to the cluster head CH_{iA} . Assume that $L(j)^{'}$ is the reverse authentication information of the time $j(t)$on time slice j, where t is a certain time on time slice j.

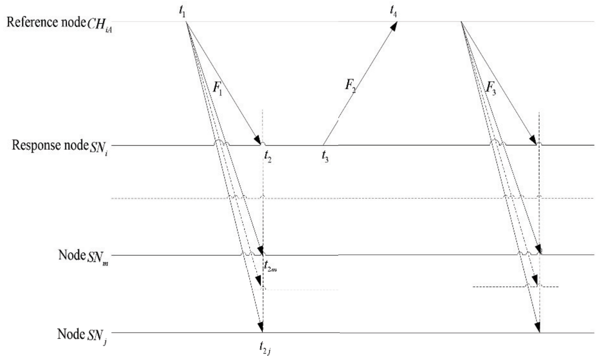

Figure 5: HRTS protocol

$$L(j)^{'} = \left\{ \begin{array}{c} h^{j}_{SN_i}(T_{SN_i}(j(t)))||h^{j-2}_{SN_i} \\ ||ID_{SN_i}||T_{SN_i}(j(t)) \end{array} \right\} \tag{13}$$

Where, $T_{SN_i}(j(t))$ is the local time of SN_i, $h^{j-2}_{SN_i}$ is the published key by SN_i at $T_{SN_i}(j(t))$.

Secondly, CH_{iA} rebuilds the reverse authentication information $L_{CH_{iA}}(j)^{'}$ (Equation (14)) and sends it to last layer cluster nodes $CH_{(i-1)B}$ when obtains $L(j)^{'}$.

$$L_{CH_{iA}}(j)^{'} = \left\{ \begin{array}{c} h^{j}_{SN_i}(T_{SN_i}(j(t)))||ID_{SN_i} \\ h^{n}_{CH_{iA}}(h^{j}_{SN_i}(T_{SN_i}(j(t))), \\ T_{SN_{iA}})||h^{j-2}_{SN_i}||h^{n-2}_{SN_{iA}} \\ ||ID_{CH_{iA}}||T_{SN_i}(j(t))||T_{SN_{iA}} \end{array} \right\} \tag{14}$$

Where, $T_{CH_{iA}}$ is the local time of CH_{iA} when broadcasting $L_{CH_{iA}}(j)^{'}$, $h^{n-2}_{SN_{iA}}$ is the published key by CH_{iA} at $T_{CH_{iA}}$.

Lastly, BS can get the information $L(j)^{'}$ based on receiving the reverse authentication information from the cluster nodes one by one, and waiting for getting the published key $h^{j}_{SN_i}$ by SN_i after δ to certificate the correctness of $h^{j}_{SN_i}(T_{SN_i}(j(t)))$ which can illustrate the identity of SN_i and the correctness of source attestation. In this time, each node can verify whether the key $h^{j}_{SN_i}$ is published based on $h^{j-2}_{SN_i} = H^2(h^{j}_{SN_i})$: if $h^{j}_{SN_i}$ has been published by SN_i, that each node which has obtained $h^{j}_{SN_i}$ can forge or tamper with the information $L(j)^{'}$, so the information $L(j)^{'}$ from these nodes will be judged to be unsafe, and abandon it, if $h^{j}_{SN_i}$ has not been published by SN_i, that each $L(j)^{'}$ will be cached until that $h^{j}_{SN_i}$ is published.

Firstly, BS can make a security condition check again by the reverse authentication in Step 4. Secondly, if $j(t)$ is much different with other cluster members after $h^{j}_{SN_i}$ published, it can be judged that SN_i is the compromised node, and abandon it directly by BS. Thirdly, if the local time of most of the cluster members in CH_{iA} is much different with other cluster heads, it can be judged that CH_{iA} is the compromised node, and all the nodes associated with CH_{iA} are dangerous, so it needs to rebuild the network topology based on LEACH after abandoning CH_{iA}.

Therefore, the latent compromised nodes are detected and excluded in this step, which make the time synchronization work more secure.

The flow chart of STSP is showed in Figure 6.

4 Security Analysis and Simulation

4.1 Security Analysis

The main security problem of WSN time synchronization algorithms is the attack from the malicious nodes and the compromised nodes, the malicious nodes can be excluded by security condition check (such as Step 1 of STSP), but the compromised nodes can be latent down to waiting for an opportunity to attack, such as the attacker can send the cached legitimate data repeatedly to BS, which can cause a large amount of energy consumption. In addition, the false data forged by the compromised nodes in different geographic areas cant be detected and filtered.

The security characteristics of STSP proposed in this paper are analyzed as follows:

1) Excluding the malicious nodes

   Because of the one-way property of the hash key chain, the malicious nodes can't get the unpublished key, such as h^j_{BS} in $L(j) = (P_{j(t)}||h^j_{BS}(P_{j(t)}, T_{BS}(j(t)), ID_{BS})||h^{j-2}_{BS}||ID_{BS}||T_{BS}(j(t)))$ where $h^{j-2}_{BS} = H^2(h^j_{BS})$, so that the external malicious nodes couldn't get the authentication key h^j_{BS} to participate in the authentication work and make any bad effect. Lastly, the malicious nodes will be detected and excluded by BS.

2) Making security condition check positively based on uTESLA

   Before time synchronization of network, each node can get the information $L(j)$ based on receiving the broadcast authentication information from their neighbor nodes one by one, and waiting for getting the published key h^j_{BS} by BS after δ to certificate the correctness of $h^j_{BS}(P_{j(t)}, T_{BS}(j(t)))$ which can illustrate the identity of BS and the correctness of source attestation. In addition, Each node can verify whether the key h^j_{BS} has been published based on$h^{j-2}_{BS} = H^2(h^j_{BS})$: if h^j_{BS} has been published by BS, that each node which has obtained the key h^j_{BS} can forge or tamper with the information $L(j)$, so the information $L(j)$ from these nodes will be judged to be unsafe, and abandon it; if h^j_{BS} has not been published by BS, that each $L(j)$ will be cached until that h^j_{BS} is published.

3) Detecting and excluding the latent compromised nodes based on uTESLA

   It's not sure whether all the nodes in the network are synchronized after the time synchronization work, because there are some latent compromised nodes which can make some damage in the time synchronization process, a reverse authentication method based on uTESLA (Step 4 of STSP) is proposed for further detecting: if the local time $j(t)$ of SN_i in CH_{iA} is much different with other cluster members after $h^j_{SN_i}$ been published, it can be judged that SN_i is the compromised node, and abandon it directly by BS, if the local time of most of the cluster members in CH_{iA} is much different with other cluster heads, it can be judged that CH_{iA} is the compromised node, and all the nodes associated with CH_{iA} are dangerous, so it needs to rebuild the network topology based on LEACH after abandoning CH_{iA}. Therefore, the latent compromised nodes can be detected and excluded by Step 4.

4) Small network load

   In DCS, it is bound to increase the redundancy of the message in the network and lead to the increase of the communication cost based on the flooding type network topology, because all nodes need to receive more than 2s+1 synchronization messages from last layer nodes.

   In STSP, the most classical clustering protocol LEACH in WSN is chosen to initialize the network topology in STSP, and the network topology structure based on clustering hierarchical is more suitable for WSN applications (as shown in Figure 2), which can avoid the over energy consumption of cluster head nodes, reduce the communication traffic effectively, and extend the life cycle of the network by 15%.

4.2 Simulation

In order to test the validity of STSP, a simulation work is made in the software platform of MATLAB R2014a to compare the difference in synchronization cycle, synchronization precision, synchronization ratio and synchronization cost between STSP and DCS.

The main simulation parameters are shown in Table 2:

Time synchronization cycle is the period that BS initiates the network time synchronization work to the end of the synchronization work, and the shorter the synchronization cycle, the better the convergence of the synchronization algorithm. In STSP, the synchronization cycle is the running time of Step 2 and Step 3. As shown in Figure 7, take the average value of 30 simulation data, it's

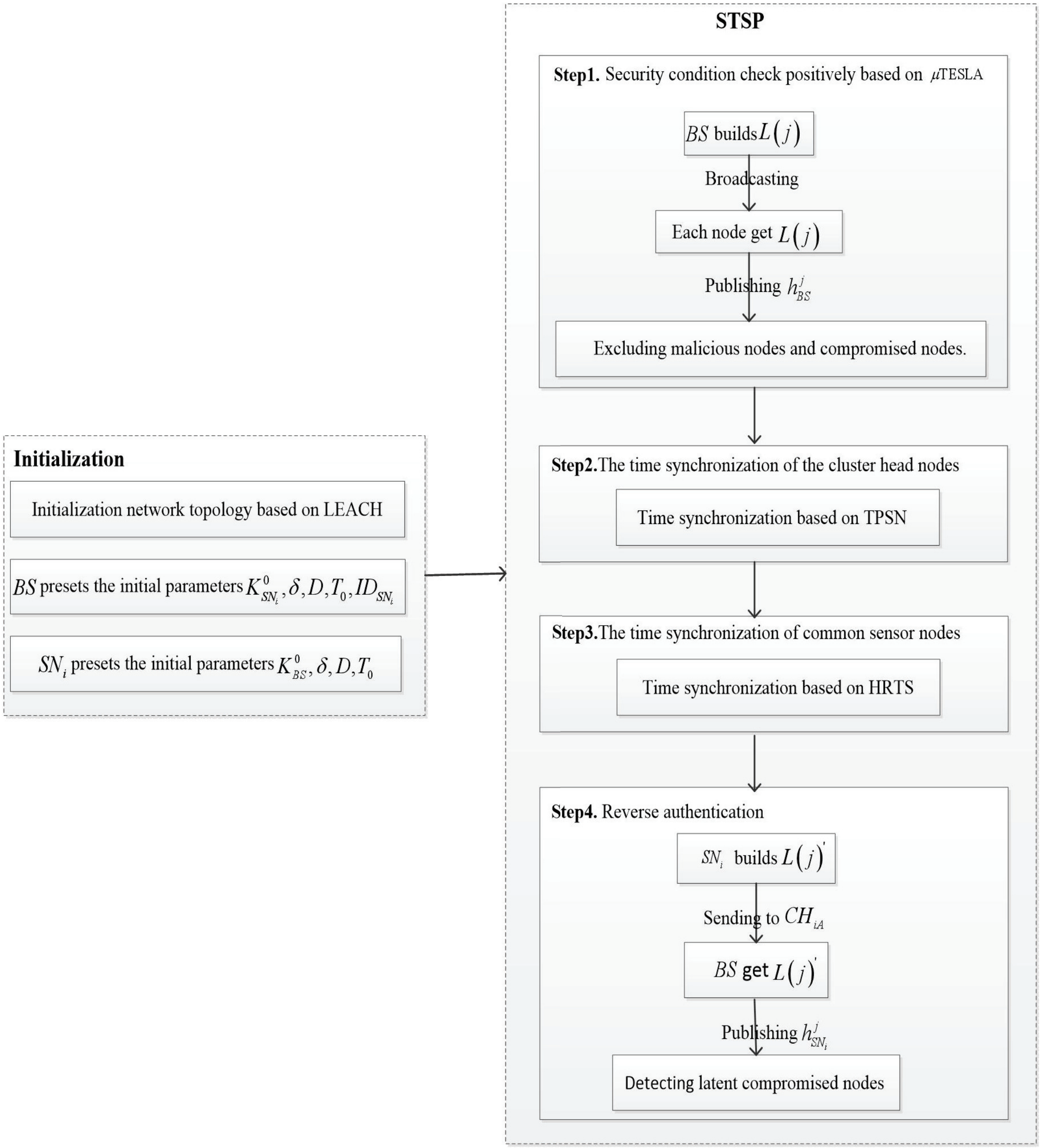
Initialization
Initialization network topology based on LEACH
BS presets the initial parameters $K^0_{SN_i}, \delta, D, T_0, ID_{SN_i}$
SN_i presets the initial parameters K^0_{BS}, δ, D, T_0
STSP
Step1. Security condition check positively based on μTESLA
BS builds $L(j)$
Broadcasting
Each node get $L(j)$
Publishing h^j_{BS}
Excluding malicious nodes and compromised nodes.
Step2.The time synchronization of the cluster head nodes
Time synchronization based on TPSN
Step3.The time synchronization of common sensor nodes
Time synchronization based on HRTS
Step4. Reverse authentication
SN_i builds $L(j)'$
Sending to CH_{iA}
BS get $L(j)'$
Publishing $h^j_{SN_i}$
Detecting latent compromised nodes

Figure 6: STSP flow chart

Table 2: Simulation parameters

Base station	1
Network size N	[50,100,150,......,450,500]
Crystal oscillator	32MHZ
Area	1000m*1000m
Radius	500m
Modular	CC2430
Protocol	IEEE802.15.4
Rate	250kb/s
Power	20dBm
Malicious nodes MN	[0, 3, 5]

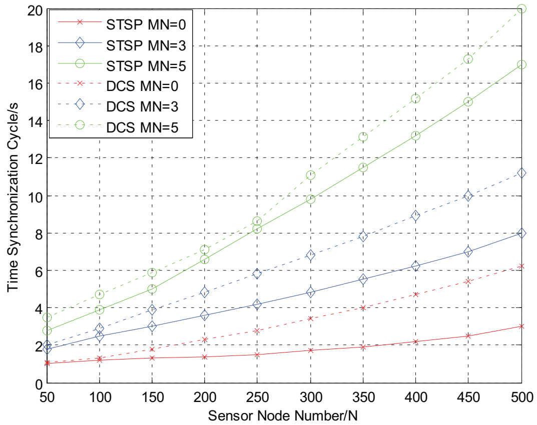

Figure 7: Time synchronization cycle

indicated that the synchronization cycle of these two algorithms will be extended with the increase in the number of network nodes, and the increase in the number of malicious nodes will extend the synchronization cycle too. In addition, the synchronization cycle of STSP is much better than DCS.

Synchronization error is the main characteristic of time synchronization precision, and the synchronization error is the time error between the network nodes and the base station. As shown in Figure 8, take the average value of 30 simulation data, it's indicated that the synchronization error will be increased with the increase in the number of network nodes, and the more the malicious nodes, the worse the synchronization precision. In addition, the synchronization precision of STSP is much better than DCS.

Synchronization ratio is the ratio between the synchronized nodes and the total network nodes, which embodies the security of time synchronization algorithms. As shown in Figure 9, take the average value of 30 simulation data, it's indicated that the synchronization ratio of STSP is much better than DCS. The reason is that not all the nodes in the network can receive more than 2s+1 synchronization messages from last layer in DCS, and it's easy to cause an attack from compromised nodes, but the malicious nodes and the compromised nodes will be detected and excluded by uTESLA in STSP.

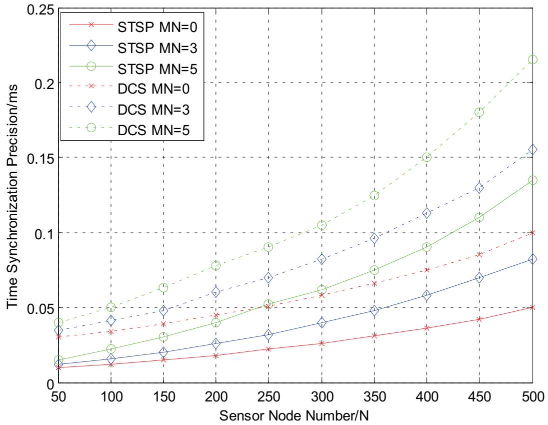

Figure 8: Time synchronization precision

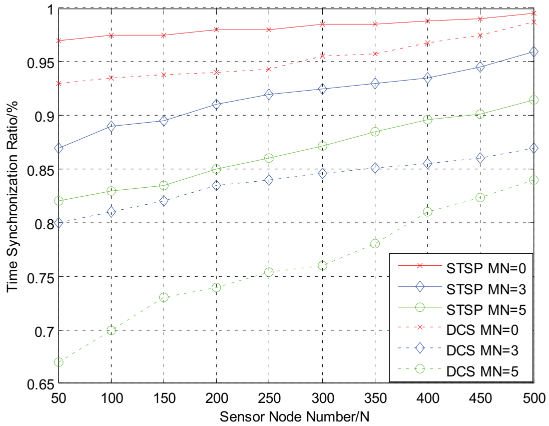

Figure 9: Time synchronization ratio

Synchronization cost is the number of packets transmitted in once synchronization process. As shown in Figure 10, take the average value of 30 simulation data, it's indicated that the synchronization cost of STSP is much better than DCS. The reason is that it is bound to increase the redundancy of the message in the network and lead to the increase of the communication cost in DCS, that all nodes need to receive more than 2s+1 synchronization messages from last layer nodes, but the synchronization work in STSP only needs three times data communication.

5 Conclusions

A secure time synchronization protocol (STSP) for wireless sensor network based on uTESLA protocol is proposed according to the issues that DCS algorithm in wireless sensor network time synchronization is limited in the threats of compromised nodes, big communication cost and deficiency in coverage. Firstly, initializing the net-

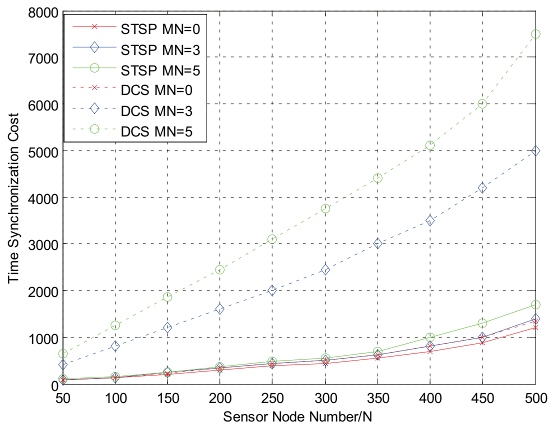

Figure 10: Time synchronization cost

work topology based on LEACH protocol to balance the network node load and prolong network life cycle. Secondly, excluding the malicious nodes based on uTESLA protocol for making security condition check positively before the network time synchronization. Thirdly, making time synchronization quickly based on TPSN protocol for base station to cluster head node. Fourthly, making time synchronization based on HRTS protocol for cluster head nodes to their own common nodes. Lastly, making reverse authentication based on uTESLA protocol for checking and excluding the compromise nodes after the network time synchronization. The security analysis and simulation show that two times authentication ensure the absolute security of the time synchronization work in STSP, and STSP is much better than DCS in synchronization cycle, precision, ratio and cost.

In addition, STSP in this paper still has much room for improvement based on the following reasons:

- Computation cost

  The uTESLA protocol has higher authentication efficiency in the case of sending data packets frequently, but it has a very low sending frequency in some applications, such as fire alarm and other event-driven applications, where the transmission interval of the adjacent data packets may be far greater than the time slice D of uTESLA, and causes lot of keys not used for the data packets authentication, the distance between adjacent keys on the key chain is also increased, and causes a large computation cost and authentication delay.

  Increasing D can alleviate this problem, but it also causes a lot of authentication delay, and the receiving nodes also need more memory space for buffering packets.

- Delay

  In uTESLA, the time interval of sending message $(MAC_{k_i}(P_i)||k_{i-2}||P_i||i(t))$ will be increased gradually, and the time for buffering data packets is also increased because of the authentication delay, which also makes the protocol more vulnerable to be attacked by DoS. Therefore, the authentication mechanism of uTESLA is not suitable for the situation of large sending time interval.

Acknowledgments

This work is funded by the National Science and Technology Planning of China (2015BAG10B00). The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] S. Ahmed, F. Xiao, and T. Chen, "Asynchronous consensus-based time synchronisation in wireless sensor networks using unreliable communication links," *IET Control Theory and Applications*, vol. 8, no. 12, pp. 1083–1090, 2014.

[2] I. F. Akyildiz, W. L. Su, and Y. Sankarasubramaniam, "Wireless sensor networks: A survey," *Computer Networks*, vol. 38, no. 1, pp. 393–422, 2002.

[3] D. Capriglione, D. Casinelli, and L. Ferrigno, "Analysis of quantities influencing the performance of time synchronization based on linear regression in low cost WSNs," *Measurement (02632241)*, vol. 77, no. 1, pp. 105–116, 2016.

[4] Z. Chen, D. Li, and Y. Huang, "Vent-triggered communication for time synchronization in WSNs," *Neurocomputing*, vol. 177, no. 2, pp. 416–426, 2016.

[5] H. Dai and R. Han, "Tsync: A lightweight bidirectional time synchronization service for wireless sensor networks," *ACM Mobile Computing and Communications Review*, vol. 1, no. 8, pp. 125–139, 2004.

[6] I. Davut, B. Kemal, and T. Bulent, "Evaluating energy cost of route diversity for security in wireless sensor networks," *Computer Standards and Interfaces*, vol. 39, no. 3, pp. 44–57, 2015.

[7] L. Deng, H. Huang, and Y. Qu, "Identity based proxy signature from rsa without pairings," *International Journal of Network Security*, vol. 19, no. 2, pp. 229–235, 2017.

[8] A. Diaz and P. Sanchez, "Simulation of attacks for security in wireless sensor network," *Sensors(14248220)*, vol. 16, no. 11, pp. 1–27, 2016.

[9] N. S. Fayed and E. M. Daydamoniand A. Atwan, "Efficient combined security system for wireless sensor network," *Egyptian Informatics Journal*, vol. 13, no. 3, pp. 185–190, 2012.

[10] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks," in *Proceeding of the 1st ACM Conference on Embedded Networked Sensor Systems (SenSys'03)*, Los Angeles, California, USA, 2003.

[11] Y. Gao, P. Zeng, K. K. R Choo, and F. Song, "An improved online/offline identity-based signature scheme

for WSNs," *International Journal of Network Security*, vol. 18, no. 6, pp. 1143–1151, 2016.

[12] J. He, J. Chen, and P. Cheng, "Secure time synchronization in wireless sensor networks: A maximum consensus-based approach," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 4, pp. 1055–1065, 2014.

[13] J. P. He, P. Cheng, and L. Hi, "Time synchronization in wsns: A maximum-value-based consensus approach," *IEEE Transactions on Automatic Control*, vol. 59, no. 3, pp. 660–675, 2014.

[14] W. R. Heinzelman, A. Chandrakasanand, and H. Balakrishnan, "Energy-efficient communication protocol for wireless micro-sensor networks," in *Proceeding of the 33rd Annual Hawaii International Conference on System Sciences(AHICSS'00)*, pp. 3005–3014, Maul: IEEE Computer Society, Jan 2000.

[15] W. K. Hu and J. C. Lin, "Ratio-based time synchronization protocol in wireless sensor networks," *Telecommunication Systems*, vol. 39, no. 1, pp. 25–35, 2008.

[16] G. Huang, A. Y. Zomayaand, and F. C. Delicato, "An cccurate on-demand time synchronization protocol for wireless sensor networks," *Journal of Parallel and Distributed Computing*, vol. 72, no. 10, pp. 1332–1346, 2012.

[17] G. Huang, A. Y. Zomayaand, and F. C. Delicato, "Long term and large scale time synchronization in wireless sensor networks," *Computer Communications*, vol. 37, no. 1, pp. 77–91, 2014.

[18] M. Jef, M. Sam, and H. Danny, "A comprehensive security middleware architecture for shared wireless sensor networks," *Ad Hoc Networks*, vol. 25, no. 2, pp. 141–169, 2015.

[19] M. Khurana, R. Thalore, and V. Raina, "Improved time synchronization in ML-MAC for WSN using relay nodes," *International Journal of Electronics and Communications*, vol. 69, no. 11, pp. 1622–1626, 2015.

[20] C. Lan, H. Li, S. Yin, and L. Teng, "A new security cloud storage data encryption scheme based on identity proxy re-encryption," *International Journal of Network Security*, vol. 19, no. 5, pp. 804–810, 2017.

[21] C. H. Ling, C. C. Lee, C. C. Yang, and M. S. Hwang, "A secure and efficient one-time password authentication scheme for WSN," *International Journal of Network Security*, vol. 19, no. 2, pp. 177–181, 2017.

[22] J. Liu, Z. Zhou, and Z. Peng, "Mobi-sync: Efficient time synchronization for mobile underwater sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 2, pp. 406–416, 2013.

[23] D. L. Mills, "Adaptive hybrid clock discipline algorithm for the network time protocol," *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 505–514, 1998.

[24] K. Pardeep and H. J. Lee, "Security issues in healthcare applications using wireless medical sensor networks: A survey," *Sensors (14248220)*, vol. 12, no. 1, pp. 55–91, 2012.

[25] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and E. C. David, "Spins: Security protocols for sensor networks," *Wireless Networks*, vol. 5, no. 8, pp. 521–534, 2002.

[26] U. A. Selcuk, R. A. Beyah, and J. A. Copeland, "Secure source-based loose synchronization (sobas) for wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 4, pp. 803–813, 2013.

[27] F. Sivrikaya and B. Yener, "Time synchronization in sensor networks:a survey," *IEEE Network*, vol. 4, no. 18, pp. 45–50, 2004.

[28] K. Sun, P. Ning, and C. Wang, "Secure and resilient clock synchronization in wireless sensor network," *IEEE Journal on Selected Areas in Communications*, vol. 2, no. 24, pp. 395–408, 2006.

[29] F. Wang, C. C. Chang, and Y. C. Chou, "Group authentication and group key distribution for ad hoc networks," *International Journal of Network Security*, vol. 17, no. 2, pp. 199–207, 2015.

[30] F. Wang, C. Yu, and X. Wu, "Dual time synchronisation method for wireless sensor networks," *Electronics Letters*, vol. 51, no. 2, pp. 1–2, 2015.

[31] B. S. Yosra and O. Alexis, "A lightweight threat detection system for industrial wireless sensor networks," *International Journal of Network Security*, vol. 18, no. 15, pp. 842–854, 2016.

Biography

Xiaogang Wang: College of Automation, Chongqing University, China. Major in wireless sensor network and security. Room 2508 in the main teaching building of Chongqing University, Shaping Ba distract, Chongqing city, China. (400044). Artificial Intelligence Key Laboratory of Sichuan Province, School of Automation and Information Engineering, Sichuan University of Science and Engineering, Sichuan 643000, China.

Weiren Shi: Prof. College of Automation, Chongqing University, China. Major in wireless sensor network and applications, information control and intelligent systems, embedded systems, pervasive computing, *etc.*