Time Synchronization Protocol in Wireless Sensor Network based on Hash Code

Sachin Umrao
Department of Computer Application
Krishna Institute of Engineering and Technology, Ghaziabad, India

Arun Kumar Tripathi
Associate Professor
Krishna Institute of Engineering and Technology, Ghaziabad, India

ABSTRACT
In current time peoples are switching from wired network to wireless network. This is great achievement for technology. Peoples are using wireless networks, but there is main problem which arises in wireless network is security. Many researches are involved in this field. Wireless sensor networks (WSN) have achieved a lot of consideration recently due to wide range of research applications such as target tracking, environment monitoring, and scientific exploration in dodgy environments. Each sensor node in WSN contains a local clock, required for time synchronization. Time synchronization is a significant module of sensor networks to grant a common clock time in sensor nodes. Some sensor nodes may be harmful, which can disturb the normal function of a sensor network. In this paper, main focus is to uncover malicious nodes and propose time synchronization protocol based on hash code.

Keywords
Sensor Networks, Security, Time Synchronization, Malicious nodes, .

1. INTRODUCTION
Wireless Sensor Network (WSN) [1] consists of hundreds or more of micro sensor nodes. These nodes are combined together to form a sensor network. Each sensor node monitors the environment parameters such as temperature, pressure, and wind speed etc. individually and sends it to server to achieve a common objective. Each sensor node has its own local clocks to measure the time. The clocks of all sensor nodes in WSN must exhibit same time. To achieve this time synchronization among clocks of sensor nodes is required. Time synchronization aims to provide a common time for local clocks in WSN. A sensor network may suffer from attack of intruders. An intruder may capture the synchronization packet and replay it after the modification. Intruders have main objective to somehow induce some nodes to show false time [2] than actual one. There exist two types of attackers [3]:

I. external attackers and
II. Internal attackers

External attackers may be defined as those in which an external invader manipulates the communication between trusted node and the node that is going to synchronized and results the nodes to desynchronize, or to remain unsynchronized even after a successful execution of the synchronization protocol. One example of external attack is Pulse delay attack. Internal attacks may be defined as those in which internal invader (group members) report false clock references to their adjacent nodes.

The paper is organized as follows: In Section 2 consists of analysis of the existing time synchronization protocols [4]. In Section 3 way to find the location of sensor node is given. In section 4 proposed protocol is given. In section 5 and section 6 conclusion and future work is discussed.

2. RELATED WORKS
Researchers have proposed many protocols for time synchronization [5] like sender-receiver [6, 8]. For discussion secure pair-wise synchronization (SPS) [14] protocol is considered as sender-receiver based protocol.

In sender-receiver based synchronization [7] protocol, the sender node episodically sends a message with its local time as a timestamp to the receiver node. Then the receiver synchronizes with the sender using the timestamp which is received from the sender. The message delay [8] between the sender and receiver is intended by measuring the total time taken, from the time a receiver requests a timestamp to receiving a response.

2.1 Sender-Receiver Synchronization
In sender-receiver approach all receiver nodes should be synchronized with the sender. This approach mainly includes three steps.

I. The sender node at regular intervals sends a message with its local time as a timestamp to the receiver.
II. The receiver then synchronizes with the sender using the timestamp which is received from the sender.
III. The delay in message between the sender and receiver is intended by measuring the total time from the time a receiver requests a timestamp to the time it really receives a response.

Table1. Pseudo code for sender-receiver synchronization

Sender-receiver Synchronization
1) $P_i(T_i) \longrightarrow (T_j)\ P_j : P_i, P_j$, sync
/**P_i is sender node and P_j is receiver node, and T_i & T_j is time. Sender P_i sends request to Receiver P_j. Packet includes synchronization message time stamp with node-id of node P_i and P_j.**/

2) $P_j(T'_j) \longrightarrow (T'_i)\ P_i : P_j, P_i, T_j, T'_j$, ack

/** Node P_j at time T'_j sends response packet to P_i at time T'_i. The response packet includes node-id of nodes P_i and P_j with (receiving time of synchronization packet)T_j, (sending time of response packet) T'_j and acknowledgement. **/

3) Pi calculates offset between the nodes P_i and P_j.

The pseudo code used in sender-receiver synchronization [14] is given in Table 1. Here, T_i, T'_i symbolizes the time measured by the local clock of node P_i. Similarly T_j, T'_j represents the time measured at node P_j. At time T_i, P_i sends synchronization pulse packet to P_j. Node P_j receives this packet at time T_j, where $T_j = T_i + d + \delta$. Here, δ and d symbolize the offset between the two nodes and end-to-end delay respectively. At time T'_j, T_j sends back an acknowledgement packet. This packet contains the values of T_j and T'_j. Node P_i receives the packet at T'_i. Similarly, T'_i is related to T'_j as $T'_i = T'_j + d - \delta$. Node P_i can compute the clock offset [14] and the end-to-end delay [14] as:

Offset (δ) = $((T_j – T_i) – (T'_i – T'_j))/2$ (1)

Delay (d) = $((T_j – T_i) + (T'_i – T'_j))/2$ (2)

Sender-receiver synchronization suffers from pulse delay attack. The pulse-delay attack [10], [11] is performed by blocking the initial pulse, storing it in memory and then replaying it later at an arbitrary time. Fig. 1 represents the idea behind pulse-delay attack.

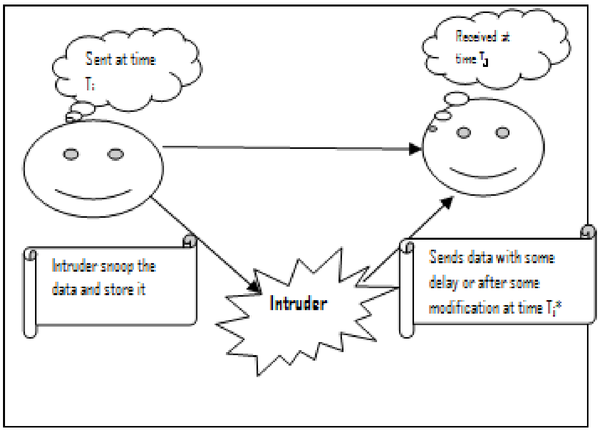

Fig.1. Pulse delay attack

Without any pulse delay [17] attack the $T_j = T_i + \delta + d$ and $T'_i = T'_j - \delta + d$. If an intruder performs pulse-delay attack (e.g., on the initial sync packet), the T_j and T'_i will change to: $T_j^* = T_i + \delta + d + \Delta$ and $T'^*_i = T'_j - \delta + d + \Delta$respectively. Here Δ is the pulse-delay introduced by the intruder [3]. In existence of pulse delay clock offset and the end-to-end delay will be

Offset (δ) = $((T_j – T_i) – (T'_i – T'_j) + \Delta)/2$ (3)

Delay (d) = $((T_j – T_i) + (T'_i – T'_j) + \Delta)/2$ (4)

Secure pair-wise synchronization (SPS) is a sender-receiver based approach. In Sender-receiver synchronization approach security mechanism is incorporated to make it flexible to adversarial attacks from intruders [13]. In this protocol, message integrity and authenticity [15] are implemented through the use of Message Authentication Codes (MAC) and a key K_{ij} [18, 19, 20] which is shared between P_i and P_j. This prevents external intruders from altering any values in the synchronization pulse or in the acknowledgement packet. Furthermore, the intruders cannot guess an identity of node P_j as it does not contain the secret key K_{ij}. An intruder can hear the packet over the wireless channel and can use the MAC in future to produce authenticated packets. Using a random nonce, Np, during the handshake safeguards the protocol against such replay attacks.

In SPS, pulse delay attacks are uncovered through a comparison of the computed message end-to-end delay, d, with the maximal expected message delay d*. Note that the computation of the end-to-end delay, d. If the calculated delay is greater than the maximal expected delay, we identify that there is replay on packet. The pseudo code for SPS protocol is given in Table 2.

Table 2. Pseudo code for secure pair-wise synchronization

Secure Pair-wise Synchronization (SPS)

1) $P_i\ (T_i) \longrightarrow (T_j)\ P_j : P_i, P_j, N_p$, sync

/** node P_i sends a synchronization packet at Time T_i which receives node P_j at time T_j. Packet includes synchronization message time stamp, nonce N_p (pseudo-random number which is used in an authentication protocol to guarantee that old communications cannot be reused in replay attacks) along with node-id of node P_i and P_j.* */

2) $P_j\ (T'_j) \longrightarrow (T'_i)P_i : P_j, P_i, N_p, T_j, T'_j$, ack, MAC $\{K_{ij}\}[P_j, P_i, N_p, T_j, T'_j$, ack]

/** In response to synchronization packet node P_j sends response packet at time T'_j is received by node P_i at time T'_i. The response packet includes node-id of nodes P_i and P_j, nonce N_p, T_j: receiving time of synchronization packet, T'_j: sending time of response packet, and acknowledgement along with all above contains encrypted by shared key K_{ij} and then protected by MAC. **/

3) Node P_i calculates end-to-end delay

$d = \{(T_j – T_i) + (T'_i – T'_j)\}/2$

if $d \leq d^*$

then $\delta = \{(T_j – T_i) - (T'_i – T'_j)\}/2$,

else

abort

3. Finding Location of Malicious node

There are two condition arises

1. Location of malicious node is known.
2. Location of malicious node is unknown.

Sometimes malicious node may behave like a trusted node by steeling identity of any trusted node. So the location of non-malicious nodes should be known to each trusted node. In this paper it is assumed that the location of each trusted node is fixed, but if the sensor node is mobile then there should be procedure to measure the location of nodes and to identify the malicious node. In [21] the procedure for finding the location with the help of angle of arrival is discussed.

To find out exact location of malicious node it must be known angle of arrival (AOA) with respect to some reference direction. Here it is assumed the four directions north, east, south and west are fixed. Here AOA is measured in between north direction and incident ray.

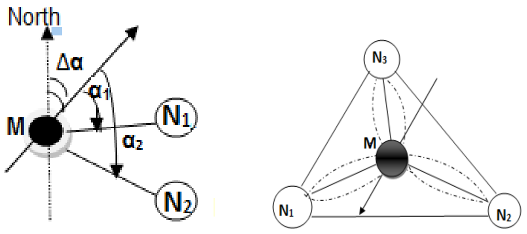

Fig 2. Triangulation (a) Localization with known orientation of malicious node. (b) Localization with unknown orientation of malicious node.

In fig.2 (a), M is unknown (Malicious) and its orientation is $\Delta\alpha$. α_1 and α_2 are the relative angle of arrival (AOA) of signal sent from two trusted nodes N_1 and N_2. Absolute AOAs can be calculated by $(\alpha i + \Delta\alpha)(\text{mod } 2\pi)$, i = {1,2}. Whereas in fig 2(b). Orientation of unknown node (malicious node) is unknown so here at least three trusted nodes are needed and angles $\angle N_1MN_2$, $\angle N_1MN_3$ and $\angle N_2MN_3$ can be calculated by using the relative AOAs. In this fig chord N_2 N_3 and angle$\angle$ N_2M N_3 and arc N_2MN_3 restricts the position of malicious node M. In proposed protocol it is assumed that the position of sensor nodes (trusted + malicious) is known.

4. Proposed Protocol

The proposed a protocol is useful in order to implement security in WSN. The proposed protocol finds malicious node as well as guarantee to send the secure message in the network. The proposed protocol will find the malicious node in the pair, which wants to be synchronized. Basically in pair wise synchronization a pair of sensor nodes wants to be synchronized and for this they should check whether there local clock timing is same or not. If clock timing is same then they can be synchronized otherwise they have to match their clock timings.

In this protocol hash function has been implemented, which will calculate the hash code for sender's message and append the hash code with the message, and then it will be send to receiver node. Each sensor node must reside in the power range of trusted node. Sender node will monitor that the time of receiving and time of response is equal or not at receiver node. Here P_i is sender node and P_j is receiver node which is to be synchronized.

Node P_i sends packet at time T_i (time measured by node P_i) and node P_j receives packet at time T_j (already sent by node P_i). These times are determined by two different clocks. T_i is determined in the local clock of node P_i (i.e. C_i) whereas T_j is determined by the local clock of node P_j (i.e. C_j). The offset (or the variation of the local clocks) of paired nodes is represented by δ_{ij} (calculated by node P_j with respect to node P_i). The hold-up for the packet transfer from P_i to P_j is represented by dij. In proposed protocol node P_j is treated as malicious node, if it does not report the exact time of receiving and sending. In this paper it is assumed that malicious node [19] does not report the exact time at which it receives the packet.

4.1 Steps in Proposed Protocol

This protocol is consists of 4 steps:

1. In this step sender P_i sends a synchronization packet at time T_i to the receiver node P_j Receiver node receives the synchronization packet at time T_j. The synchronization packet contains a random number called nonce N_p ,which is issued by a authentication protocol to make sure that old communication cannot be used again in replay attack, synchronization message time stamp, along with node id of both sender and receiver nodes.
2. In this step receiver (P_j) sends response packet to sender (P_i) at time T_j which is by P_i at time $T`_i$. The response packet includes the node id of both sender and receiver node , time stamps $T_i$ , $T_j$ ,$T`_j$, $T`_I$, nonce N_p and acknowledgement with hash function H(n) and then protected by hash value.
3. In this step Sender node will calculate end to end delay if the delay (d) is less than the maximal delay (d*) then sender node P_i calculates the offset (δ_{ij}) for P_j and start message transmission.
4. Otherwise it will abort the synchronization.

Table 3. Pseudo code for proposed protocol

Proposed Protocol for Time Synchronization

1. P_i (T_i) —> (T_j) P_j : P_i, P_j, Np, sync
2. P_j ($T`_j$) —> ($T`_i$)P_i : P_j, P_i, N_p, T_j, $T`_j$, ack, H(n){h}[ $P_j$, $P_i$, $N_p$,$T_j$, $T`_j$, ack]
3. $d = \{(T_j – T_i) + (T`_i – T`_j)\}/2$

   if $d \le d^*$

   then $\delta = \{(T_j – T_i) - (T`_i – T`_j)\}/2$,

   Start Message transmission.
4. else

   Abort the synchronization process.

The proposed protocol is suitable to provide security from external attacks and it is capable to synchronize non-malicious nodes.

There are two proposed theorems which are as follows.

Theorem 1: Show that a group of non-malicious nodes can be synchronized to a trusted node using pair wise synchronization.

Proof: Let There is a pair of sensor nodes P. P_i and P_j are two sensor nodes where P_i is a trusted node and we don't know about the P_j that it is trusted or malicious node. So this theorem will prove whether it is trusted pair or not.

Assume a pair P= {P_i, P_j} of length one is formed by nodes P_i and P_j.

The offset of node P_j with respect to node P_i,

$\delta_{ij} =[(T_j-T_i)–(T'_i-T'_j)]/2$

Similarly, offset of node P_i with respect to node P_j,

The offset of node P_i with respect to node P_j,

$\delta_{ji} =[(T_i-T_j)–(T'_j-T'_i)]/2$

For safe synchronization

$\delta_{ij} = \delta_{ji}$

=> δ_{ji}- δ_{ji}=0

It proves that this pair of nodes is non-malicious.

This shows that the node can be synchronized to trusted node.

Hence Proved.

Theorem 2: show that if any node is malicious in the pair of sensor nodes i.e. $P\{P_i , P_j\}$; nodes (P_i , P_j) cannot be synchronized to the clock of trusted node.

Proof: A Malicious node may be defined as a node which does not report the exact time at which it receives or sends the packet. Here, it is considered that the malicious node does not report the exact time at which it receives the packet.
Here sensor node P_j is considered as malicious node. Here, it is considered that malicious node do not report the exact time of packet receiving. Therefore, instead of T_j, node P_j will send receiving time of challenge packet as time T''_j in response packet. In non-malicious environment, sending time and receiving time of the packet must be equal (since nodes are directly linked to each other in pair).

$|T_j-T_i|=|T'_i-T'_j|$

Now, since node P_j sends receiving time of packet

T''_j instead of T_j.

$T_j \neq T''_j$

Therefore, P_i will determine

$|T''_j-T_i|\neq|T'_i-T'_j|$

It shows that P_j is malicious node.

P_i will recognize node p_j as malicious node, and, therefore P_i and P_j cannot be synchronized to the clock of trusted node.

Hence Proved.

5. Conclusions

In existing solutions of time synchronization in WSN are not very much reliable. Still there are lots of problems in existing solutions. External intruders can take advantage of these weaknesses n harm to our network. Pulse delay attack is still feasible is also the reason of worry .The external attacks can be resolved with the help of MAC (i.e message authentication code) by using a shared private key. But the main problem is of internal attacks in pair wise synchronization and another problem arises is intruders

So in proposed protocol all these problems got the attention. This protocol has implemented hash code instead of MAC to make it reliable and make it safe from external attackers. Because in MAC there are chances of steeling private key but in hash code there is no chances. Here main point of discussion was about Pair wise synchronization in which if the receiver is not synchronized (i.e. local clock timing not matches) then sender will send their clock timing to receiver and then update the clock timing of receiver in order to get synchronized.

6. Future Works

Synchronization in WSN can be faster and secured and can consume less energy. In further main focus will be on WSN in order to make it secure and reduce power consumption.

7. REFERENCES

[1] Mukherjee, B., Ghosal, D., Yick, J.: Wireless sensor network survey. Computer Network 52(12), 2292–2330 (2008)

[2] Kshemkalyani, A.D., Sundararaman, B., Buy, U.: Clock synchronization for wireless sensor networks. A Survey on Ad-hoc Networks, 281–323 (2005)

[3] Capkunl, S., Ganeriwal, S., Han, S., Srivastava, M.: Securing Timing Synchronization in Sensor Networks. In: Proceedings of, pp. 369–390. Springer, New York (2006)

[4] Cayirci, E., Akyildiz, I.F., Su, W., Sankarasubramaniam, Y.: A Survey on Sensor Networks. IEEE Communications Magazine, 102–114 (2002)

[5] Kopetz, H., Ochsenreiter, W.: Clock Synchronization in Distributed Real-Time Systems. IEEE Transactions on Computers 36(8), 933–940 (1987)

[6] Li, H., Chen, K., Wen, M., Zheng, Y.: A Secure Time Synchronization Protocol for Sensor Network. In: Washio, T., Zhou, Z.-H., Huang, J.Z., Hu, X., Li, J., Xie, C., He, J., Zou, D., Li, K.-C., Freire, M.M. (eds.) PAKDD 2007. LNCS (LNAI), vol. 4819, pp. 515–526. Springer, Heidelberg (2007)

[7] Wang, C., Ning, P., Sun, K.: Secure and resilient clock synchronization in wireless sensor networks. IEEE Journal on Selected Areas in Communications 24(2), 395–408 (2006)

[8] Song, H., Zhu, G.C.S.: Attack-resilient time synchronization for wireless sensor networks. In: IEEE International Conference on Mobile Adhoc and Sensor Systems Conference, p. 772 (2005)

[9] Estrin, D., Elson, J., Girod, L.: Fine-grained network time synchronization using reference broadcasts. In: Proceedings of the 5th Symposium on Operating Systems Design and Implementation Special Issue, Boston, pp. 147–163 (2002)

[10] Trappe, W., Xu., W., Zhang, Y., Wood, T.: The feasibility of launching and detecting jamming attacks in wireless networks. In: Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing, Urbana Champaign, IL, USA, pp. 46–57 (2005)

[11] Ping, S.: Delay Measurement Time Synchronization for Wireless Sensor Networks, Intel Corporation, Intel Research, Berkeley (2002)

[12] Srivastava, M.B., Kumar, R., Ganeriwal, S.: Timing-sync protocol for sensor Networks. In: Proceedings of the First ACM Conference on Embedded Networked Sensor Systems, Los Angeles, CA, pp. 138–149 (2003)

[13] Manzo, M., Roosta, T., Sastry, S.: Time synchronization attacks in sensor networks. In: Proceedings of the 3rd ACM Workshop on Security of Ad Hoc and Sensor Networks, pp. 107–116 (2005)

[14] Ganeriwal, S., Popper, C., Capkun, S., Srivastava, M.B.: Secure Time Synchronization in Sensor Networks. ACM Transactions on Information and System Security, Article No: 23, 11(4) (2008)

[15] Jajodia, S., Setia, S., Zhu, S.: LEAP: Efficient security mechanisms for large-scale distributed sensor networks. In: Proceedings of the 10th ACM Conference on Computer and Communications Security, Washington DC., USA, pp. 62–72 (2003)

[16] Simon, G., Kusy, B., Ledeczi Maroti, M.: A Clock synchronization for wireless sensor networks: A Survey. In: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, pp. 30–49 (2004)

[17] Hu, H., Atakli, I.M., Chen, Y., Ku, W.-S., Su, Z.: Malicious Node Detection in Wireless Sensor Networks. In: The Symposium on Simulation of Systems Security, pp. 836–843 (2008)

[18] Eschenauer, L., Gligor, V.D.: A key-management scheme for distributed sensor networks. In: Proceedings of the 9th ACM Conference on Computer and Communications Security, Washington, DC, USA, pp. 41–47 (2002)

[19] Chan, H., Perrig, A., Song, D.: Random key predistribution scheme for sensor networks. In: Proceedings of the 2003 IEEE Symposium on Security and Privacy, p. 197 (2003)

[20] Hwang, J., Kim, Y.: Revisiting random key pre-distribution schemes for wireless sensor networks. In: Proceedings of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks, Washington DC, USA, pp. 43–52 (2004)

[21] Rong Peng : Angle of Arrival Localization for Wireless Sensor Networks : Sensor and Ad Hoc Communications and Networks, 2006. SECON '06. 2006 3rd Annual IEEE Communications Society on 28 Sept. 2006 p.p.- 374 - 382