Graduate Research in Engineering and Technology (GRET)

Volume 1 | Issue 2 Article 5

October 2013

MOBILE DATA COLLECTOR FOR SECURE TIME SYNCHRONIZATION IN CLUSTERED WIRELESS SENSOR NETWORK

RAMYA M DR.
Department of Computer Science and Engineering, Siddaganga Institute of Technology, Tumkur, Karnataka, India., ramyam1290@gmail.com

A.S. POORNIMA
Department of Computer Science and Engineering, Siddaganga Institute of Technology, Tumkur, Karnataka, India., aspoornima@sit.ac.in

Recommended Citation
M, RAMYA DR. and POORNIMA, A.S. (2013) "MOBILE DATA COLLECTOR FOR SECURE TIME SYNCHRONIZATION IN CLUSTERED WIRELESS SENSOR NETWORK," *Graduate Research in Engineering and Technology (GRET)*: Vol. 1 : Iss. 2 , Article 5.
DOI: 10.47893/GRET.2013.1023
Available at: https://www.interscience.in/gret/vol1/iss2/5

MOBILE DATA COLLECTOR FOR SECURE TIME SYNCHRONIZATION IN CLUSTERED WIRELESS SENSOR NETWORK

RAMYA M, DR. A.S.POORNIMA

Department of Computer Science and Engineering, Siddaganga Institute of Technology, Tumkur, Karnataka, India.
Email: ramyam1290@gmail.com, aspoornima@sit.ac.in

Abstract—Secure time synchronization is a key requirement for many sophisticated application running on these networks. Most of the existing secure time synchronization protocols incur high communication and storage costs and are subject to a few known security attacks. In wireless sensor network (WSN), lifetime of the network is determined by the amount of energy consumption by the nodes. To improve the lifetime of the network, nodes are organized into clusters, in which the cluster head (CH) collects and aggregates the data. A special node called mobile data collector (MDC) is used to collect the data from the CH and transfer it to the base station (BS) By using proposed method MDC authenticated to CH by computing shared secret keys on the fly. Once the MDC and CH are authenticated, all the sensor nodes in the cluster are synchronized, time synchronization reduce the communication and storage requirements of each CH. Security analysis of this proposed system shows that it is highly robust against different attacks namely compromised CH, reply attack, message manipulation attack as well as pulse delay attack.

keywords— *Cluster Head, Mobile Data Collector, Message Authentication Code, Secure Data Collection, Time Synchronization.*

I. INTRODUCTION

Wireless sensor Networks (WSN) comprised mainly of small sensor nodes with limited resources and a base station. The nodes in a network are deployed over a geographic area to sense and gather various types of data that includes temperature, humidity, intrusion detection, vehicular motion and so on [1]. Time synchronization is critical to sensor networks at many layers of its design. It enables better duty-cycling of the radio, accurate and secure localization, beam forming and other collaborative signal processing.

Main goal of time synchronization is to provide a common notion of time for a group of nodes in the network. Different applications pose different accuracy requirements, but in any case synchronization should be performed with as small overhead as possible to achieve the required accuracy. Synchronization in wireless sensor network is vital aspect of successful and efficient network operations in any business settings, particularly in military and medical applications as they latter rely on the data accuracy to make rapid and sound decision. Imagine the detrimental affect on the functionality of all these application if a malicious adversary is able to abuse the underlying time synchronization protocol. Nodes will have faulty estimates about the location of the other cluster nodes. This can further trigger unnecessary packet retransmission if MAC layer acknowledgements are enabled. It will trivial for adversaries to perform reply attacks.

The time synchronization problem has been thoroughly studied in sensor networks [2] and there are several prototype implementations, such as RBS [3], TPSN [4], FTSP [5] that can achieve microseconds accuracy, none of the existing protocols are resilient to malicious attacks. These protocols have not been built with security in mind. Realizing the inadequacy of existing time synchronization solutions, we develop schemes for achieving secure time synchronization using mobile data collectors in clustered wireless sensor network.

In majority of the applications sensed data is aggregated by sensors called cluster heads and sent to base station for analysis. Sending data directly to base station will waste energy at intermediate sensors and it increase delay so to overcome from these problem notion of mobile nodes is introduced in WSN[6]. The approach uses mobile data collection agents [6]. The mobile agent, called a Mobile Data Collector (MDC) traverse in the network and collects the data from the nodes and dumps the data back at BS. These MDC's can also help in data aggregation. The use of MDC for data collection in clustered WSN is depicted in Fig1.

In this paper we are proposed a secure time synchronization protocol using MDC. By using proposed method MDC authenticated to CH by computing shared secret keys on the fly. Once the MDC and CH are authenticated, all the sensor nodes in the cluster are synchronized, after time synchronization CH transfer aggregated data to MDC. Time synchronization prevents external attackers from successfully modifying any values in synchronization pulse or in acknowledgement packet, finally show that this protocol is resilient to attacks from external attacker as well as to attacks from a compromised MDC.

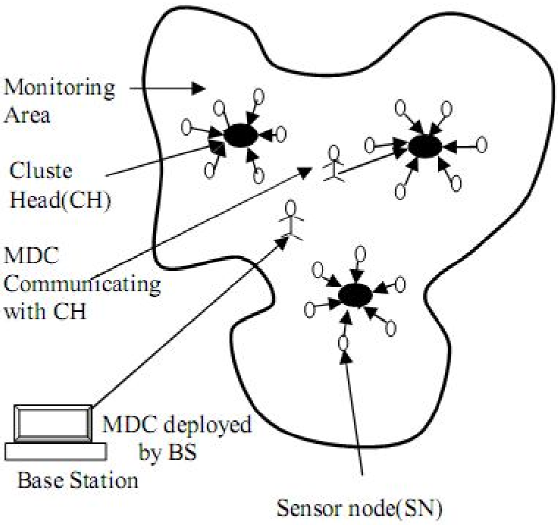


Fig.1 MDC collecting the aggregated data from CHs in a clustered sensor network.

The rest of the paper is organised as follows: in Section II we elaborate on related work in detail. We explain cluster formation and the time synchronization scheme used in designing the protocols in section Ⅲ. In Section IV , we explain Time Synchronization Protocol using MDC is explained in detail in Section. A complete security analysis of the proposed protocols is discussed in Section V. Performance analysis of the protocols is explained in Section VI. Simulation results are explained in Section VII.We conclude in Section VIII.

II. RELATED WORK

We classify the related work into two parts first, we discuss about the concept of secure data collection using MDC and the second is the concept of secure time synchronization.

First concept is secure data collection using MDC, The MDC-based data collection is studied thoroughly in the literature in the context of various mobility models. However, the security aspect in MDC-based data collection is not studied in detail. In [3] key management for secure communication and data collection in distributed WSN is discussed. The scheme ensures only confidentiality of the collected data. Identifying malicious MDC and attacks caused by malicious MDC are not considered. In [8], mobile sink is used for secure data collection. Here a fixed path is used by the mobile sink and only the nodes in this path will be able to communicate with the mobile sink and transfer data. The nodes in the path are overloaded with data transfer function every time a mobile sink visits the nodes for data collection. Also, deterministic path used by MDC leads to various attacks.

Second concept is Secure time synchronization, The primary functionality of wireless sensor networks is to sense the environment and transmit the acquired information to base stations for further processing with secure time information. Several recent contributions to the literature have addressed security and privacy issues in sensor networks [9]-[11] for routing, but it is difficult to implement along with existing time synchronization approaches because they require lots of computations for routing. Thus, the existing time synchronization schemes in sensor networks were not designed with security in mind, thus leaving them vulnerable to security attacks in addition to establishing the need of secure time synchronization for individual sensor nodes.

Security in RBS[5] Scheme is vulnerable and it is prone to many attacks [10]. Adding a sequence number to beacon messages or other messages will prevent message replay attacks. RBS works by receiver to receiver synchronization in which both nodes receive the reference beacon and then calculate their offset with one another. An attack would be as simple as compromising one of the nodes with an incorrect time. The non compromised node will then calculate an incorrect offset during the exchange period.

TPSN[6] is a sender to receiver tree based protocol with two phases, the level discovery phase and the synchronization phase. Both of the phases are initiated by the root node. In the synchronization phase the level number and the time are both sent through the tree. An attack would simply be to compromise a non-root node with the incorrect time. This will propagate through the tree and the closer the compromised node is to the root node, the more incorrect synchronization will occur. Also a node could lie about its level. That would cause other nodes to request synchronization information in which it could give inaccurate information. That node also could refuse to participate in the level discovery phase, which could eliminate its children from the network. Since none of the protocols were designed with security in mind, attacks on the synchronization are easily executed by following the rules of the protocol.

III. SYSTEM MODEL

In this section first we explain cluster formation and time synchronization.

A.Cluster Formation

The information preloaded to CH-sensors and SN-sensors before deployment are :

• Each node is preloaded with secret key k_i that it shares with the BS for confidential communication and secret key k_i shared between the node and its CH.

• Each CH-sensor is preloaded with private key k_i of all the SN-sensors in the network.

Following are some of the notations used in the paper :

BS → Base Station
CH → Cluster Head

CCHK → Common Cluster Head Key
S_i →i^{th} Sensor node
CK →Cluster Key
k_i →Private key of the i^{th} sensor node
SK_i → Session key assigned to Mobile agent for the i^{th}
round.
TS_i → Time stamp assigned to mobile agent for the i^{th} round.
MAC→ Message Authentication Code.

After deployment, clusters are formed using the preloaded information. We assume that after deployment, all the nodes are localized and know their respective positions in the network. Cluster formation is explained with the help of following steps.

1) CH-sensors broadcast *hello* message :
$CH \rightarrow U : ID_{CH} \,||\, POS_{CH} \,||\, hello$

2) Upon receiving *hello* message from CH's, each SN-sensor decides which CH to select based on POS_{CH}.

3) Let jth CH is nearer to node S_i.

4) Now node S_i sends the join request to CH_j:
$S_i \rightarrow CH_j : ID_{Si} \,||\, h(k_i) \,||\, k_i \oplus Nonce \,||\, join.$

5) CH-sensors verifies the join request using k_i and if the node S_i is authorized, accepts it as one of the cluster member.

6) CH sends the *confirm* message to node S_i
$CH \rightarrow S_i : ID_{CH} \,||\, h(Nonce) \,||\, confirm.$

7) After receiving *join* request from nearest SN-sensors, CH sends the *confirm* message. Now, CH retains the information of SN-sensors in its cluster and erases the rest from its memory.

B. Time synchronization

This protocol executed in three steps as follows
1. $A(T_1) \rightarrow (T_2)B$: *A, B, sync*
i.e., We use the following notation throughout this paper.
Node-id (Send time) → (Receive time) Node-id : Packet contents.
2. $B(T_3) \rightarrow (T_4)A$: *B, A*, T_2, T_3, *ack*
3. *A calculates offset between the nodes.*

Here, T_1, T_4 represent the time measured by the local clock of node *A*, *CA*. Similarly T_2, T_3 represent the time measured by *CB*. At time *T*1, *A* sends a synchronization pulse packet to *B*. Node *B* receives this packet at T_2, where T_2 is equal to $T_1+\delta+d$. Here, δ and *d* represent the offset between the two nodes and end-to-end delay respectively. At time T_3, *B* sends back an acknowledgement packet. This packet contains the values of T_2 and T_3. Node *A* receives the packet at T_4. Similarly, T_4 is related to T_3 as $T_4 = T_3 - \delta + d$. Node *A* can now calculate the clock offset and the end-to-end delay as:
$\delta = ((T_2 - T_1) - (T_4 - T_3))/2$; $d = ((T_2 - T_1) + (T_4 - T_3))/2$.

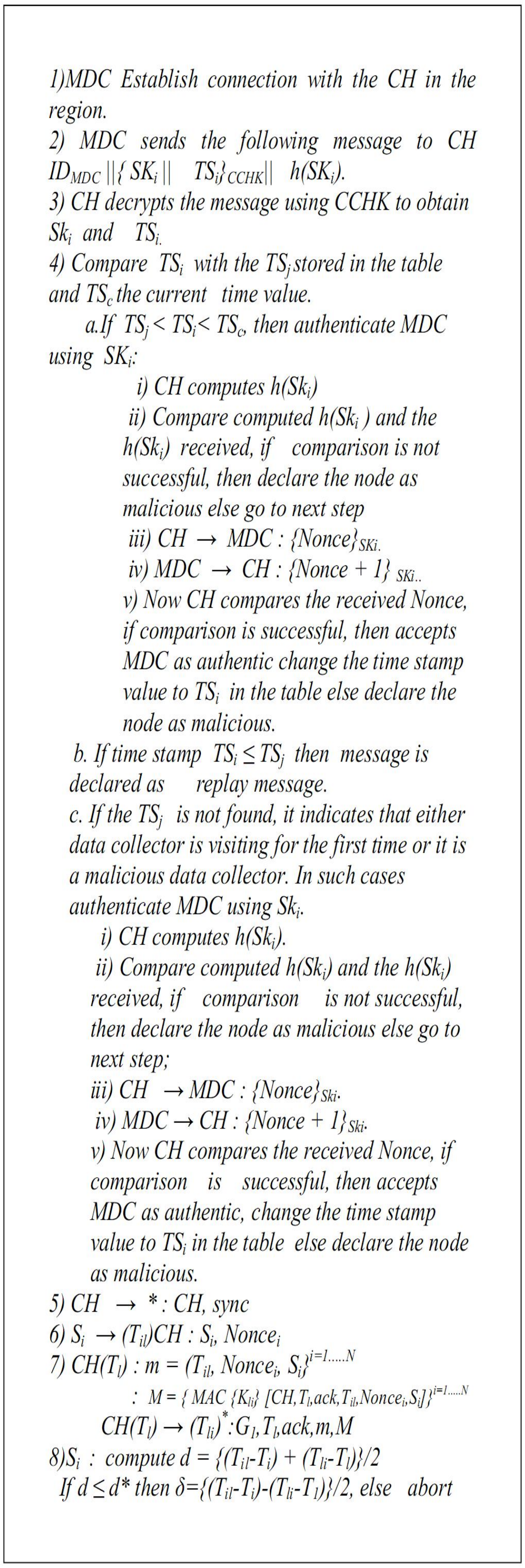
IV. PROPOSED SYSTEM

We are proposed a concept of secure time synchronization using MDC in clustered WSN, By using proposed method MDC authenticated to CH by computing shared secret keys on the fly. Once the MDC and CH are authenticated, all the sensor nodes in the cluster are synchronized, after time synchronization CH transfer aggregated data to MDC. Time synchronization prevents external attackers from successfully modifying any values in synchronization pulse or in acknowledgement packet, finally show that this protocol is resilient to attacks from external attacker as well as to attacks from a compromised MDC.

A.Time synchronization using Mobile Data Collector in clustered wireless network

In this section, we discuss a simple time synchronization scheme to identify the malicious MDC and the tree based key management scheme[12]. Our scheme not only identifies the malicious MDC, reply attack as well as pulse delay attack. BS selects a session key Sk_i for the ith round of MDC and constructs a message { SK_i || TS_i}$_{CCHK}$|| h(SK_i) || $ID_{MDC.}$ Before deployment MDC is preloaded with session key and the message. Here CCHK is the common cluster head key which is shared by all the cluster heads and the base station, SK_i is the session key and TS_i is the time stamp assigned to MDC for the ith round. Here time stamp TS_i corresponds to current time, we assume that the clock value of all CH's and BS are synchronized. Also every CH maintains a table in which it stores information regarding the TS_i along with unique ID of mobile data collector ID_{MDC}.
Algorithm1.1. explains mechanism which identifies malicious MDC, reply as replay messages. The Common Cluster Head Key CCHK is known only to the CH's, therefore only an authorized CH is able to authenticate MDC. The time stamp associated with the message enables CH to identify the replay messages. Pulse delay attacks are detected through a comparison of the computed message end-to-end delay, d, with the maximal expected message delay d*. In this protocol, CH initiates the time synchronization (step 5) to which the rest of the cluster members reply with messages containing their ids and

Algorithm 1. Identification of malicious Mobile Data Collector Replay messages, pulse delay attack in time synchronization protocol.



```
1)MDC Establish connection with the CH in the region.
2) MDC sends the following message to CH
   $ID_{MDC} || \{ SK_i || TS_i\}_{CCHK} || h(SK_i)$.
3) CH decrypts the message using CCHK to obtain $Sk_i$ and $TS_i$.
4) Compare $TS_i$ with the $TS_j$ stored in the table and $TS_c$ the current time value.
     a.If $TS_j < TS_i < TS_c$, then authenticate MDC using $SK_i$:
          i) CH computes $h(Sk_i)$
          ii) Compare computed $h(Sk_i)$ and the $h(Sk_i)$ received, if comparison is not
              successful, then declare the node as malicious else go to next step
          iii) CH → MDC : $\{Nonce\}_{SKi.}$
          iv) MDC → CH : $\{Nonce + 1\}_{SKi..}$
          v) Now CH compares the received Nonce, if comparison is successful, then accepts
             MDC as authentic change the time stamp value to $TS_i$ in the table else declare the
             node as malicious.
     b. If time stamp $TS_i \leq TS_j$ then message is declared as replay message.
     c. If the $TS_j$ is not found, it indicates that either data collector is visiting for the first
        time or it is a malicious data collector. In such cases authenticate MDC using $Sk_i$.
          i) CH computes $h(Sk_i)$.
          ii) Compare computed $h(Sk_i)$ and the $h(Sk_i)$ received, if comparison is not successful,
              then declare the node as malicious else go to next step;
          iii) CH → MDC : $\{Nonce\}_{Ski}$.
          iv) MDC → CH : $\{Nonce + 1\}_{Ski}$.
          v) Now CH compares the received Nonce, if comparison is successful, then accepts
             MDC as authentic, change the time stamp value to $TS_i$ in the table else declare the node
             as malicious.
5) CH → * : CH, sync
6) $S_i \rightarrow (T_{il})CH : S_i, Nonce_i$
7) $CH(T_l) : m = (T_{il}, Nonce_i, S_i\}^{i=1.....N}$
          $: M = \{ MAC \{K_{li}\} [CH, T_l, ack, T_{il}, Nonce_i, S_i]\}^{i=1.....N}$
     $CH(T_l) \rightarrow (T_{li})^{*} : G_l, T_l, ack, m, M$
8)$S_i$ : compute $d = \{(T_{il}-T_i) + (T_{li}-T_l)\}/2$
  If $d \leq d^*$ then $\delta=\{(T_{il}-T_i)-(T_{li}-T_l)\}/2$, else abort
```

challenge nonce's (step 6). In step 7 of the protocol, CH replies with a single broadcast message to all sensor nodes, containing MACs of the challenges and node ids. Note that the last protocol message (step 7) contains N triples $\{T_{il}, N_i, S_i\}$, one for each S_i, containing the receipt time of the challenge packet from $S_i(T_{il})$, the nonce of S_i and the node -id of S_i respectively. It also contains N MACs, one for each (CH, S_i) pair, which enable each node S_i to authenticate the packet broadcast by CH. In the last protocol step $S_1,...,S_N$ synchronize to CH.

By using above algorithm MDC is authenticated with CH, all the sensor nodes in cluster and CH are synchronized, Once they synchronized CH transfer the aggregated data to MDC. Before transferring the data CH encrypts the data using its secret key P_{ki} such that only the base station is able to decrypt the data. If MDC is compromised the collected data is not exposed as it is in encrypted form. Using the same secret key to encrypt the data every time may result in cryptanalysis of the corresponding key. To overcome this problem secret key P_{ki} of the CH's are refreshed at regular interval. Every time *CCHK* is changed, P_{ki} is changed to P_{ki}` using simple transformation function which can be executed both at the node and the base station.The transformation used to change the private key is : $P_{ki}$` $\longleftarrow$ F (P_{ki},CCHK`), where F is a collision resistant one way function.

V. SECURITY ANALYSIS

In this section we analyze security of the proposed protocol.

Mainly we consider the following security issues for time synchronization.

- Identifying malicious nodes acting as MDC.
- Identifying replay messages.
- Node Compromise.
  - Compromise of MDC.
  - Compromise of CH.

  .
- Message manipulation attack.
- Reply attack.
- Pulse Delay attack.

A. Identifying malicious nodes acting as MDC

Malicious MDC is capable of launching various attacks, all the attacks launched by malicious MDC considered in this paper are passive attacks. The adversary try to collect the data from legitimate CH's using the recorded messages. By using Time stamp for identifying malicious MDC we can identify such malicious MDC to counter the attacks. Information a malicious MDC gets by eavesdropping is the beacon message $ID_{MDC} || \{ Sk_i || TS_i\}_{\text{CCHK}} || h(Sk_i)$. The message is encrypted by CCHK which is known only to CH and BS. Hence malicious MDC can not decrypt the message. By forwarding the message to CH, malicious MDC fails to authenticate itself as Sk_i used in authentication phase is not known to malicious MDC. Therefore, the authentication protocol presented in algorithm1. ensures that only legitimate MDC is able to complete the authentication phase.

B. Identifying replay messages

The adversary captures the beacon message and replay; the message to authenticate itself. The proposed protocol identify such replay messages, the time stamps associated with the beacon message and the previous time stamps stored in CH enables a CH to identify replay messages.

C. Node Compromise

Node compromise is one of the important attack to be considered in WSN. We consider node compromise with respect to CH-sensor and MDC as these two are the attractive targets for an adversary.

1) *Compromise of MDC :* We describe what is the impact of MDC compromise on the collected data with respect to the proposed protocols. The analysis is based on the information an adversary gets by compromising a MDC.

• *Collected Data* : The CH encrypts the aggregated data using secret key Pk_i before transferring the data to MDC. The secret key Pk_i is known only to CH and BS. Also the secret key Pk_i is changed at regular interval. Therefore the compromised MDC will not reveal any data to an adversary.

• *Beacon Message* : The beacon message consists of information required to authenticate MDC. The beacon message do not reveal any information about the secret key Pk_i which is used to encrypt the collected data.

• *Session key Sk_i* : Using the session key *Ski* the adversary may try to get data from a CH that has not yet transferred its data. As the data is encrypted using Pk_i it is not possible for an adversary to forge the data. Hence the protocol is resilient to compromise of MDC.

2) *Compromise of CH-sensor*

Now adversary is able to compromise more CH's. In this proposed protocol the beacon message used to authenticate MDC is encrypted using CCHK. Therefore compromise of CH-sensor enables an adversary to compromise MDC and collect data from other CH-sensors.

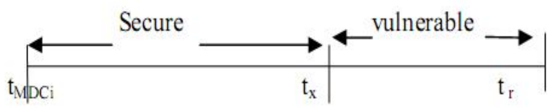


Fig.2.time line representing secure and vulnerable communication when a CH-sensor is compramised for Time stamp method.

D. Message manipulation attack

In this attack, an malicious MDC may drop, modify, or even forge the exchanged messages to interrupt the communication process. To perform this attack an malicious MDC needs to take part in the message communication. To this end, it is necessary to be a valid MDC in the network. In our protocol, an malicious MDC cannot forge the path or packet. Thus, this attack is not effective with this protocol.

E. Reply attack

In our proposed protocol, an malicious MDC cannot pass the authentication process. Even if a malicious MDC makes a copy of the previous TS_i, it still cannot forge the session as TS_i is updated every time the message is broadcasted.

F. Pulse Delay attack

After cluster formation in WSNs, nodes are authenticated by a hierarchy level corresponding to their cluster heads as well as to the base station. Thus, no malicious can disguise as a legitimate in the network. For successfully executing the delay attack on the network, an malicious node needs to capture the timing message, intentionally delay it, and later send it to the node. Hence, if an malicious node captures the timing message and intentionally delays it only to later pass it on to the node, it will be automatically discarded by the node as this timing message has already reached the node through a legitimate node. Furthermore, each timing message is different from one another by its Sequence# and TS_i. As a result, the delay attack is not effective against our proposed protocol.

VI. PERFORMANCE ANALYSIS

In this section we analyze the proposed protocols with respect to communication, computation, storage required for time synchronization.

A. Storage

Here we study the amount of storage required to store cryptographic secrets used for authentication and encryption of the data. The storage with respect to SN-sensor, CH-sensor and MDC are discussed for the proposed protocol.

• SN-Sensor : All the keys along the path of the tree for which the SN-sensor belongs to are stored in SN-sensors. The number of keys of the tree each SN-sensor stores are $\log_m n + 1$ for a cluster of size n and degree of the tree m. In addition to this each SN-sensor stores secret key P_{ki} used for confidential communication with the BS.

•CH-Sensor : The CH-sensor maintains m array tree consisting of all the SN-sensors in the cluster. Therefore storage required to maintain the tree is (m/m−1)n. All CH's form a m array tree maintained by BS. Each CH-sensor stores all the $\log_m n+1$ keys along the path of this tree. Also each CH-sensor stores its secret key P_{ki} for confidential communication with the BS.

- MDC : MDC stores the session key S_{ki} and the beacon message ID_{MDC}|| { S_{Ki} || TS_i}$_{\text{CCHK}}$ || $h(k_i)$.

B. Communication

The Communication cost is measured in terms of number of messages exchanged between a CH and the MDC to complete authentication In this protocol a total of six messages are exchanged between a CH and the MDC for authentication and time synchronization. After authentication, time synchronization message is transferred to all sensors in the cluster and are synchronized. After synchronization , single unicast message CH transfers the encrypted data and the same is received by the MDC.

C. Computation

The computation cost is measured in terms of various operations that are performed to authenticate MDC and to time synchronize nodes . The protocol performs only encryption/decryption operations and one way hash functions to authenticate MDC and then time synchronization packet is sent to CH after time synchronization. Here CH-sensor performs one encryption and one decryption operations and a single hash function to authenticate MDC. To transfer data CH performs single encryption operation. The computations at MDC are one encryption and one decryption for authentication. Time stamping the packets ensures that the transmission delay is symmetric for the data exchange between MDC-CH and prevent the reply attack, minimizes the synchronization precision and maximize the attacker impact.

VII. SIMULATION RESULTS

Simulations were conducted to study the end-to-end delay of the proposed protocol. The proposed protocol is simulated, where we recorded the end-to end delay of the nodes on round basis. The end-to-end delay shown in the simulations is for entire one round of the MDC including authentication, data transfer and time synchronization at each node.

For our experiments we considered 10% of the nodes as CHs,for different sized network a same percentage of CHs is considered. The delay analysis and synchronization error for the proposed protocol are shown in Fig.3(a),(b), respectively.

The graphs represents end- to-end delay analysis for the different rounds. If we observe the pattern of delay for varying cluster size, it reveals new things. Intensifies when there are more nodes in the network and these computations are higher. Thus additional security provided by proposed protocol requires higher delay. Therefore there is a trade-off between delay and level of security being provided.

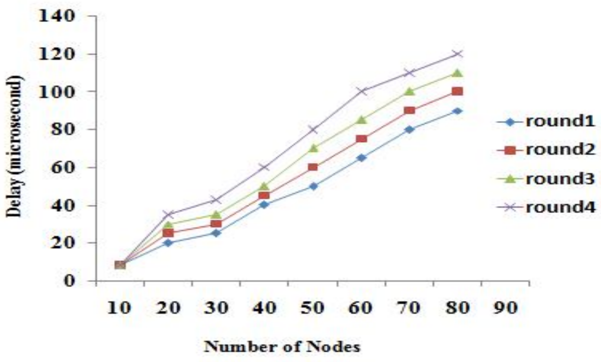


Fig3(a). End-to-End delay of CH for different rounds of MDC visits in Proposed protocol.

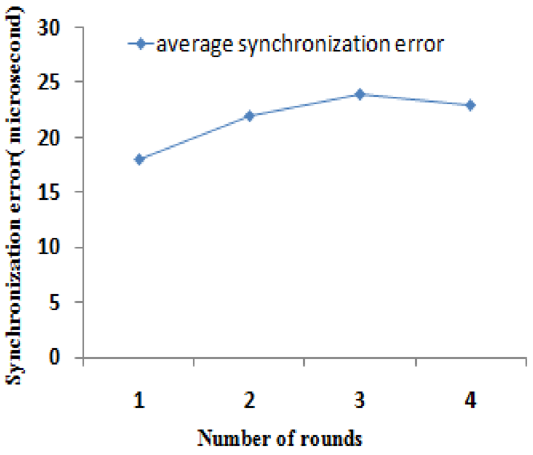


Fig.3(b) Synchronization error over different rounds of MDC.

VIII. CONCLUSION

The secure time synchronization and data collection in clustered WSN using MDC is not explored in detail in the literature. Almost all of the existing time synchronization protocols are vulnerable to different attacks and communication and computation costs are very high in terms of resource constrained WSN. By using proposed method MDC authenticated to CH by computing shared secret keys on the fly. Once the MDC and CH are authenticated, all the sensor nodes in the cluster are synchronized, after time synchronization CH transfer aggregated data to MDC. By using proposed method we can resist various attacks such as compromised CH, compromised MDC, reply attack, message manipulation attack and pulse delay attack. The analysis shows that proposed method provide varying level of security against node compromise attack, pulse delay attack by imposing addition computational overhead. We believe that we have just scratched the surface in the solution space of secure time synchronization and data collection in clustered WSN. Our future work includes investigation of various time synchronization schemes using MDC in clustered WSN, In parallel we are also developing better remedial actions against the malicious attacks.

REFERENCES

[1] I.F.Akyildiz, W.Su, Y.Sankarasubramanian, and E.Cayirci, A Survey on Sensor Networks", IEEE Communications Magazine,vol.40, no.8, pp 102-114, Aug. 2002.

[2] Sundararaman, B., Buy, U., Kshemkalyani, D. Clocksynchronization for wireless sensor networks: A Survey. Ad-hoc Networks, 3(3): 281-323, May 2005.

[3] Elson, J., Girod, L., Estrin D. Fine-grained network time synchronization using reference broadcasts. In Proceedings of the Fifth Symposium on Operating Systems Design and Implementation (OSDI), Boston, MA,December 2002.

[4] Ganeriwal, S., Kumar, R., Srivastava, M. B.. Timing-sync protocol for sensor networks. In Proceedings of the First ACM Conference on Embedded Networked Sensor Systems (SenSys), Los Angeles, CA, November 2003.

[5] Maroti, M., Kusy, B., Simon, G., Ledeczi, A.. The flooding time synchronization protocol. In Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys), November 2004.

[6] E.Ekili, Y.Gu, and D.Bozdag. Mobility based communication in wireless Sensor Networks, IEEE Communications Magazine, vol.44, no.7,pp.56-62, July 2006.

[7] Rasheed, A., Mahapatra, R.: ‘Secure data collection scheme in wireless sensor networks with mobile sink’. Proc. of Seventh IEEE Int. Symp. On Network Computing Applications, 2008.

[8] Poornima, A.S., Amberker, B.B.: ‘Agent based secure data collection in heterogeneous sensor networks’. Proc. Second Int. Conf. on Machine Learning and Computing (ICMLC 2010), Bangalore, India, 9–11 February 2010.

[9] C. Karlof and D. Wagner, “Secure Routing in Sensor Networks:Attacks and Countermeasures,” Proc. 1st IEEE Int’l.Wksp. Sensor Network Protocols and Apps., 2003.

[10] H. Chan and A. Perrig, “Security and Privacy in Sensor Networks,” Computer, vol. 36, no. 10, Oct. 2003, pp. 103–05. [11] H. Chan, A. Perrig, and D. Song, “Random Key Predistribution Schemes for Sensor Networks,” Proc. 2003 IEEE Symp. Sec. and Privacy, May 2003, pp. 197–213.

[11] D. Liu and P. Ning, “Establishing Pairwise Keys in Distributed Sensor Networks,” Proc. 10th ACM Conf. Comp. and Commun. Sec., 2003,pp. 52–61.

[12] A S Poornima, B.B.Amberker Tree-based Key Management Scheme for Heterogeneous Sensor Networks 14th IEEE International Conference on Networks, New Delhi, 2008.