Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 11, November 2014, pg.598– 604

RESEARCH ARTICLE

ATTACKS AND REQUIREMENTS OF TIME SYNCHRONIZATION

JYOTI YADAV
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GURGAON INSTITUTE OF TECHNOLOGY AND MANAGEMENT
GURGAON
Jyotiyadav034@gmail.com

RAJESH YADAV
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GURGAON INSTITUTE OF TECHNOLOGY AND MANAGENT
GURGAON
Rajeshyadav02@gmail.com

Abstract: ***Time synchronization is a basic requirement for various applications in wireless sensor network, e.g., event detection, speeds estimating, environment monitoring, data aggregation, target tracking, scheduling and sensor nodes cooperation. Time synchronization is also helpful to save energy in WSN because it provides the possibility to set nodes into the sleeping mode. In wireless sensor networks all of above applications need that all sensor nodes have a common time reference. However, most existing time synchronization protocols are likely to deteriorate or even be destroyed when the WSNs attack by malicious intruders. The recently developed maximum and minimum consensus based time synchronization protocol (MMTS) is a promising alternative as it does not depend on any reference node or network topology. But MMTS is vulnerable to message manipulation attacks. In this thesis, we focus on how to defend the MMTS protocol in wireless sensor networks under message manipulation attacks. We investigate the impact of message manipulation attacks over MMTS. Then, a novel Secured Maximum and Minimum Consensus based Time Synchronization (SMMTS) protocol is proposed to detect and invalidate message manipulation attacks.***

Keywords: wireless sensor network, time synchronization, maximum consensus, minimum consensus, message manipulation attack.

Introduction

1. ATTACKS OF SYNCHRONIZATION

There is a common problem among the three protocols presented. They were all developed to be energy efficient, precise, robust, and so on, but none of them were developed with security in mind. As in all computer protocols security is always an issue and attacks on protocols is inevitable.

In all synchronization attacks, the goal is to somehow convince nodes that their neighboring nodes are at a different time then they really are. Since global synchronization is the goal for some protocols and they rely on the neighboring nodes to pass the synchronization information on, compromising a node would disrupt the global synchronization.

1.1 ATTACKS ON RBS, TPSN, AND FTSP

For RBS, an attack on the synchronization can be executed easily. 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.

Remember TPSN 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.

The fundamental problem in FTSP is that it allows for any node to elect itself the root after a period of time of not receiving the synchronization information. A corrupt node could claim itself to be the root and now the other nodes will respond to its timing information instead of the

correct information from the real root node. The will of course propagate through the network until all nodes have incorrectly calculated their skew and offset.

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. In the sender to receiver synchronization, an attack will institute more damage because it will propagate through the network.

1.2 COUNTER MEASURES FOR ATTACKS

There are two major types of synchronization protocols. The single hop protocols, RBS, and the multi-hop protocols, TPSN and FTSP. In either case, the goal is to authenticate the synchronization messaging. Redundancy as well as nodes refusing to pass on bad information are other ways to combat synchronization attacks.

For single hop networks, the challenge for synchronization security is to make sure the sending node is not compromised to send out erroneous timing information. This can be accomplished by an authentication process. Either an authentication process or the use of a different private key between the sending node and each receiving node should be used for security.

In the multi-hop case an attack on a node close to the root could compromise a large portion of the network. The use of private keys in this case could also be used, but there are a few other idea. For FTSP, redundancy could be introduced so that it does not calculate its timing from just one neighbor, but from several. It could then determine if there is a corrupt node. If a node was suspicious that it was receiving bad synchronization data, it could cease retransmission of the data. This would stop the desynchronization from propagating throughout the network.

Once again, none of the protocols discussed were designed with security in mind. Therefore it is easy to compromise a node's timing and have the erroneous timing propagate through the network, especially on multi-hop networks. Authentication, redundancy, and refusal to transmit corrupt synchronization information are ways to combat attacks. The tradeoff being that these countermeasures require overhead and will induce more network traffic, but it may be a small price to pay to keep synchronization attacks from compromising the network.

2. REQUIREMENTS OF SYNCHRONIZATION

- **Energy Efficiency**: As with all of the protocols designed for sensor networks, synchronization schemes should take into account the limited energy resources contained in sensor nodes.
- **Scalability:** Most sensor network applications need deployment of a large number of sensor nodes. A synchronization scheme should scale well with increasing number of nodes and/or high density in the network.
- **Precision:** The need for precision, or accuracy, may vary significantly depending on the specific application and the purpose of synchronization. For some applications, even a simple ordering of events and messages may suffice whereas for some others, the requirement for synchronization accuracy may be on the order of a few secs.
- **Robustness:** A sensor network is typically left unattended for long times of operation in possibly hostile environments. In case of the failure of a few sensor nodes, the synchronization scheme should remain valid and functional for the rest of the network.
- **Lifetime:** The synchronized time among sensor nodes provided by a synchronization algorithm may be instantaneous, or may last as long as the operation time of the network.
- **Scope:** The synchronization scheme may provide a global time-base for all nodes in the network, or provide local synchronization only among spatially close nodes. Because of the scalability issues, global synchronization is difficult to achieve or too costly (considering energy and bandwidth usage) in large sensor networks. On the other hand, a common time-base for a large number of nodes might be needed for aggregating data collected from distant nodes, dictating a global synchronization.
- ***Cost and Size:*** Wireless sensor nodes are very small and inexpensive devices. Therefore, as noted earlier, attaching a relatively large or expensive hardware (such as a GPS receiver) on a small, cheap device is not a logical option for synchronizing sensor nodes. The synchronization method for sensor networks should be developed with limited cost and size issues in mind.

Immediacy: Some sensor network applications such as emergency detection (e.g. gas leak detection, intruder detection) require the occurring event to be communicated immediately to the sink node. In this kind of applications, the network cannot tolerate any kind of delay when such an emergency situation is detected. This is called the immediacy requirement, and might prevent the protocol designer from relying on excessive processing after such an event of interest occurs, which in turn requires that nodes be *pre*-synchronized at all times.

LITERATURE REVIEW

Many time synchronization protocols have been proposed in the past few years, e.g. RBS [28] [29] [30], TPSN [31], FTSP [32] [33] [34] [35] [36], etc. However, most of these protocols are root-based or tree-based time synchronization protocols, which are sensitive to the dynamic network topology. Thus, in order to enhance the robustness and scalability of the protocols, consensus concept, e.g., average consensus, has been introduced to solve the time synchronization problem in WSNs recently, which is called consensus-based time synchronization [20] [37] [38] [39] [40] [41] [42] [43]. Compared with the traditional root-based or tree-based time synchronization protocols, consensus-based time synchronization protocols are fully distributed without requiring any certain reference node. Meanwhile, the consensus-based time synchronization protocols are able to simultaneously compensate both the clock offset, i.e., instantaneous clock difference, and the clock skew, i.e., clock speed, which can prolong there synchronization period and thus reducing communication and energy costs. The existing consensus based time synchronization protocols can be divided into two categories, i.e., average consensus-based [37] [21] and maximum consensus-based [20].

In RBS [28] [29] [30], at first sender node broadcast reference message and then receiver node record their local time when they received a reference broadcast. After that, they exchange the recorded time with each other.

J. Elson et al., in [28] proposed RBS protocol in which sender nodes send reference signals to their neighbors utilizing physical-layer broadcasts. A reference broadcasts does not contain an express timestamp; rather, beneficiaries utilize its entry time as a perspective for looking at their clocks. They utilize estimations from two wireless used to show that expelling the sender's non determinism from the critical path in this way result in a dramatic improvement in synchronization over using NTP, their protocol permits time to be proliferated crosswise over

broadcast domains without losing the reference-broadcast property. Their protocol keeps up microsecond-level synchronization to an external timescale, for example, UTC. As NTP protocol is not suited for energy use, precision, cost, scope, and lifetime. Elson et al., in [29] proposed some configuration standard use numerous, tunable modes of synchronization; don't keep up a global timescale for the whole network; use post-facto synchronization; adjust to the application, and exploit domain knowledge.

F. Ren et al., proposed a new time synchronization protocol called Self-Correcting Time Synchronization (SCTS). This protocol converts the time synchronization problem into an online dynamic self-adjusting optimizing process. This conversion is done to make offset and drift compensation simultaneously. The SCTS protocol proposed by [30] completely misuses the inherent broadcast property of wireless channel, so the communication overhead is noticeably low. They also proposed equivalent digital PLL without a real voltage controlled oscillator to evade the additional hardware needed by a traditional PLL circuit.

3.CONCLUSION AND FUTURE WORK

This thesis investigates time synchronization under cyber physical attacks in WSNs. By theoretical analysis and simulation results it is clear that existing Maximum and Minimum consensus based Time Synchronization (MMTS) protocol is invalid under message manipulation attacks defined in this thesis. A Secured Maximum and Minimum consensus based Time Synchronization (SMMTS) protocol is proposed to defend against message manipulation attacks. Specifically, in SMMTS, by carefully designing the hardware clock and logical clock checking processes, it will be able to detect and invalidate the potential message manipulation attacks. Meanwhile, the maximum and minimum consensus based logical clock updating process guarantees faster convergence and compensates clock skew and offset simultaneously and logical clock does not deviate more from real clock. In future we can investigate more attack on Time Synchronization Algorithm and proposed proper solution for that attack

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, \Wireless sensor networks: a survey," Computer networks, vol. 38, no. 4, pp. 393{422, 2002.
[2] A. Chandrakasan, R. Amirtharajah, S. Cho, J. Goodman, G. Konduri, J. Kulik, W. Rabiner, and A. Wang, \Design considerations for distributed microsensor systems," in Custom Integrated Circuits, 1999. Proceedings of the IEEE 1999, pp. 279{286, IEEE, 1999.
[3] P. Bonnet, J. Gehrke, and P. Seshadri, \Querying the physical world," Personal Communications, IEEE, vol. 7, no. 5, pp. 10{15, 2000.
[4] M. Castillo-E_er, D. H. Quintela, W. Moreno, R. Jordan, and W. Westho Wireless sensor networks for ash flood alerting," in Devices, Circuits and Systems, 2004. Proceedings of the Fifth IEEE International Caracas Conference on, vol. 1, pp. 142{146, IEEE, 2004.

BIOGRAPHY

Jyoti Yadav passed B.Tech in Information Technology from GCEW, GURGAON, pursuing M.Tech in Computer Science and Engineering from GITM, GURGAON, The area of research is Time Synchronization In Wireless Sensor Network

