Clock Skew Based Node Identification in Wireless Sensor Networks

Ding-Jie Huang*, Wei-Chung Teng*, Chih-Yuan Wang*, Hsuan-Yu Huang* and Joseph M. Hellerstein†
* Dept. of Computer Science and Information Engineering
National Taiwan University of Science and Technology, Taipei, Taiwan
Email: {D9515002, weichung, M9615029, M9615052}@mail.ntust.edu.tw
† Dept. of Electrical Engineering & Computer Sciences, University of California, Berkeley, CA
Email: hellerstein@cs.berkeley.edu

***Abstract*—Node identification is one of the most important issues to wireless sensor network security. Current approaches use cryptographic authentication and certification tools to ensure the node identification, while this paper introduces an intuitive method to identify a node by measuring its clock skew. This method is based on our observation that every sensor node has a unique clock skew value that is different from any other node. We adopt Flooding Time Synchronization Protocol (FTSP) as the measuring tools, and the experimental data show that almost all measured clock skews of one sensor node vary inside a tiny bound. For any two nodes that their clock skews are very close to each other, a classifying function is proposed to check the line continuity of contiguous measured clock skews. The proposed method has successfully identified every node in our experiments, and its applications like Sybil attack detection is also discussed.**

I. Introduction

Security in wireless sensor networks (WSNs) has become a popular research field in recent years [1], and node identification is considered as one of the most important issues in this field. In WSNs, the mechanism to create and manage node identities is usually naive and is not well protected. Thus many attack techniques, such as Sybil attacks [2] and replication attacks [3], are used to exploit this vulnerability.

Since the node identities are easy to create and change, a reliable node identification mechanism is needed in sensor networks. Currently several authentication and certification methods have been proposed to ensure the node identification [4][5][6]. However, these approaches use cryptographic techniques, and thus inevitably increase computing overhead of sensor nodes. This paper introduces a simple but effective method to identify a node only by measuring its clock skew.

Recently, Kohno et al. [7] revealed the possibility to fingerprint every computer in general networks by their clock skews. Murdoch's research also used clock skew as a main method to detect the identities behind the Tor network [8]. However, there are few studies evaluating the characteristics of clock skew in WSNs. In this research, we use the Flooding Time Synchronization Protocol (FTSP) to measure the time information of each mote [9], and successfully observe that every sensor mote does have constant and unique clock skew. An algorithm to group and identify clock skews of large amount of motes is proposed, and its applications like Sybil attack detection are also discussed.

II. Node identification by clock skew

A. Clock skews of sensor motes

In WSNs every sensor mote has its own clock, and the resonant frequency of the quartz crystal in every mote is slightly different. The difference between two clocks is called the *offset*. Generally, the *offset* between two clocks gradually increases over time, and the relative speed of the *offset* is called the *skew* error. However, the frequency of the clocks may vary slightly over time due to aging or varying environmental conditions such as the temperature, and this fluctuation is called *drift* [10]. In fact, clock skews of the same mote measured in our experiments demonstrated the same characteristics with what observed in general networks [7].

B. Exploring the Flooding Time Synchronization Protocol

FTSP is currently the most advanced time synchronization protocol of WSNs [9][11][12][13] and is often used as a service for TinyOS 1.x applications. FTSP implements a dynamic hierarchical time synchronization topology. Every node, when synchronized with its parent node, retains the clock skew and *globalTime* from the root node. The reasons we adopt FTSP as the measuring tool are as follow:

1) Availability: Since current sensor motes do not have TCP/IP stacks implemented, there is no standard tool to fetch the timestamp as in general network devices [7]. FTSP provides the required clock skew measuring function, and current FTSP implementation is a mature one with its source code still maintained [14].

2) Accuracy: According to the past researches, the fluctuation of measured clock skews is around 1 to 2 parts per million (ppm) [7][8]. Therefore, the measuring tool must be able to provide high precision. FTSP uses MAC-layer time-stamping with jitter reducing techniques to achieve high precision. The average error of FTSP for a single-hop time synchronization is only 1.48μs. For multi-hop time synchronization, the average error is 11.7μs in a 7-hop network [9].

III. Analysis to empirical data

A. Experimental platform

The hardware we use is the Taroko mote, a Tmote Sky compatible product, running TinyOS 1.x [14]. Each sensor

978-1-4244-2324-8/08/$25.00 © 2008 IEEE.
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2008 proceedings.

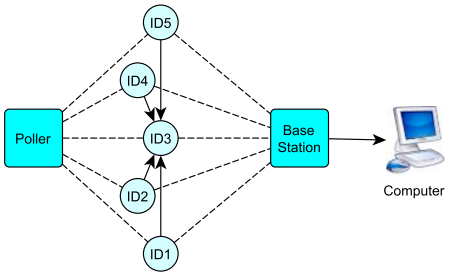

Fig. 1. Single-hop synchronization model.

mote has an 8MHz Texas Instruments MSP430 microcontroller with 10KB RAM and 48KB external Flash memory. By running TinyOS 1.x, we can simply apply FTSP on the top of it.

B. The two time synchronization scenarios

Since FTSP provides multi-hop time synchronization, we designed two kinds of experiments to find out if there exists any difference between single-hop and multi-hop synchronization when applied to node identification.

1) Single-hop synchronization: As mentioned above, each mote computes the clock skew between its local time and the root node's *globalTime*. In the preliminary experiment, we use a *Base Station* that is connected to a PC and a *Poller* to request the data. At the beginning 5 motes are used to measure the clock skews. Fig. 1 shows the scenario for gathering the clock skew from every node. Node ID2 to ID5 are normal nodes and they synchronize their time with the root node ID1 through single-hop. On the other hand, the *Poller* periodically requests data from each mote and forwards the collected data to the PC through the *Base Station*. Finally, the data is parsed and logged by a TinyOS java utility we developed.

After running for 7 hours, we collected more than 13,000 packets from each of the 5 motes and extracted the key information from a bunch of packets. In FTSP, each mote updates its own clock skew after increased its sequence number. Therefore, we record only one clock skew per sequence number.

Fig. 2 demonstrates how the clock skew of each mote differs. The x-axis is the sequence number of the collected packets, which indicates the times that each mote synchronized its time with its parent node. The y-axis is the clock skew between each node and the root node. Since node ID1 is the root node, the clock skew of node ID1 is always 0. Furthermore, the clock skew average of each mote is different from each other, and none of these clock skew lines overlapped. We can easily find that the clock skew between each mote is different and stable, as shown in Table I.

Fig. 3 shows another experiment with 27 motes running for two hours under the same configuration of the previous experiment. All clock skews of the same mote still form in straight lines parallel to x-axis such that we can easily distinguish most of the 27 lines.

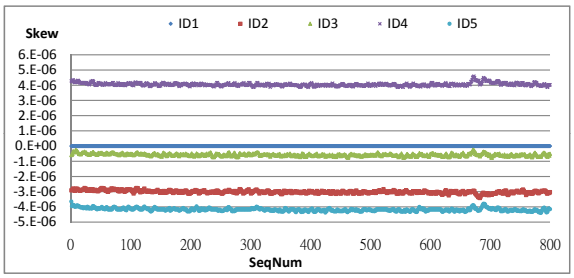

Fig. 2. Clock skews between node ID1 and the other nodes.

TABLE I
PROPERTIES OF THE CLOCK SKEWS IN FIG. 2

	Node1	Node2	Node3	Node4	Node5
Max.	0ppm	-2.75ppm	-0.248ppm	4.60ppm	-3.65ppm
Ave.	0ppm	-3.01ppm	-0.573ppm	4.05ppm	-4.18ppm
Min.	0ppm	-3.42ppm	-0.769ppm	3.86ppm	-4.41ppm
Max.-Min.	0ppm	0.67ppm	0.52ppm	0.74ppm	0.76ppm

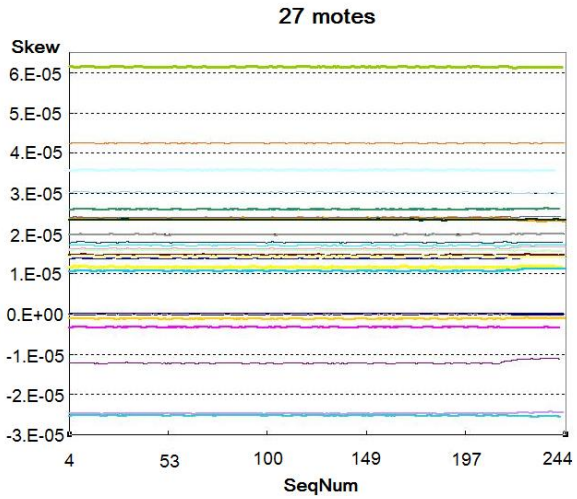

Fig. 3. Distribution of the clock skews of 27 motes.

2) Multi-hop synchronization: The scenario of multi-hop synchronization is a little bit different from the single-hop one. As Fig. 4 shows, there are 5 motes, named ID1 to ID5, in this scenario. ID1 is the root and ID2 calculates its clock skew by synchronizing its time to ID1. However, ID3 can not connect to ID1 directly because of the limited communication distance, so ID3 can only perform time synchronization with its parent node ID2. ID4 and ID5, like ID3, can only synchronize with their parent ID3 and ID4 respectively. The functions of the *Poller* and the *Base Station* are the same as in the single-hop scenario.

The experimental results are shown in Fig. 5. In the beginning 20 *SeqNum*, the skews of ID2 to ID5 fluctuate abnormally. This may be the side effect caused by initialization. After then, the values of the skews become stable, and each mote's skew is different from each other. The clock skews of ID2 fluctuate

978-1-4244-2324-8/08/$25.00 © 2008 IEEE.
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2008 proceedings.

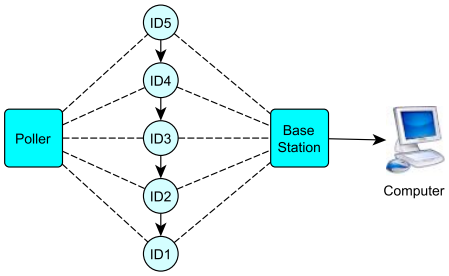

Fig. 4. Multi-hop synchronization with 5 motes.

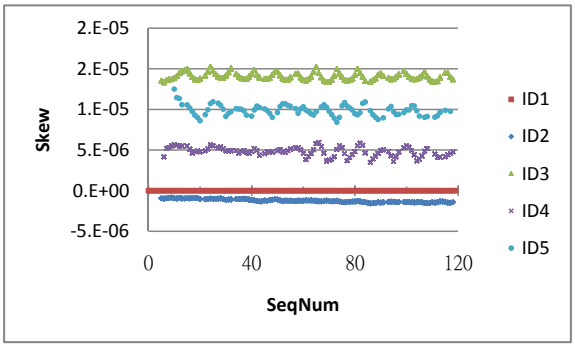

Fig. 5. Distribution of the clock skews of 5 motes in multi-hop synchronization.

less then 1ppm. However, the skews of ID3, ID4 and ID5 fluctuate around 2ppm, which may be caused by the delay of multi-hop synchronization. Although the results show that the fluctuation of skews in multi-hop scenario is larger than in single-hop one, we can still conclude that clock skews provide enough hints to identify every mote in multi-hop networks.

C. ID classification

ID classification is achieved if we can develop a method to successfully separate all measured clock skews into lines such that every line is distinguishable to other lines and represents the identity of one node. According to the categories of applications, we propose two kinds of approaches below:

1) In-network: For a node in WSNs to identify all its neighbor nodes, an approach with little calculation and memory is necessary. An intuitive way is to maintain the maximum skew, the minimum skew, and the skew average of each neighbor node, as shown in Table I. For any new incoming packet, the difference between its clock skew information and the current skew average is calculated. If the difference does not fall inside the threshold, an alarm of false ID is arisen. The value of threshold may be proportional to the difference between the maximum skew and the minimum skew. Because most of clock skews fluctuate no more than 1ppm, an appropriate threshold value should not exceed 0.5ppm. However, some skew averages are too close, like the lines shown in Fig. 3, such that we may not be able to distinguish every node unless the threshold is smaller than 0.15ppm. A threshold of 0.15ppm will generate a lot of false alarms and is unacceptable.

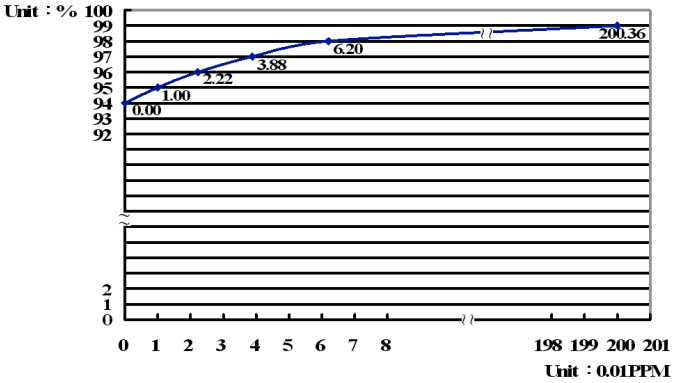

Fig. 6. Statistics of variance between contiguous clock skews of a sample mote.

To solve the above problem, we developed another method to confirm the continuity of each line. At first, we record the skew variations of a certain node by calculating the difference between two contiguous points. Fig. 6 shows the distribution of the skew variations of the node with the most rugged line. The y-axis is the cumulative percentage of the points with equal or less variation, and the x-axis is the variation in unit of 0.01ppm. Here we use 0.01ppm as our threshold because it concludes 95% of points whose variations are less than or equal to the threshold. Thus, for each neighbor node, the last three clock skews are stored. When a new packet arrives, the three contiguous variations, denoted as d_1, d_2, and d_3, are calculated and the following function is used to determine if this incoming packet is sent by a fake node:

$$f(d_1, d_2, d_3) = \begin{cases} true, & \text{if } d_1,d_2,d_3 > \text{threshold} \\ false, & \text{otherwise} \end{cases} \quad (1)$$

In this function, the true value means that the last three clock skews may not belong to the mote claimed by their packets because they failed on keeping the continuity of contiguous clock skews. If we assume that the variance between contiguous clock skews is independent to each other, then the probability to raise a false alarm will equal to 0.000125 or 0.05^3. In summary, we try to identify all neighbor nodes by calculating the clock skew average of each node. For those nodes with very close average, we in further use the last variations of each node to check the continuity of their lines. By applying these two methods, node identities can be confirmed with little computation and little memory space.

2) End-to-end: Since we can use *Base Station* to collect the clock skews from each mote, a more complex approach can be applied on the computer off line. Any two sets of skews may have very close average values but different pattern of skews. There are many analyzing tools to divide the data into groups, such like K-means clustering and QT (quality threshold) clustering. We leave this as a future work.

IV. APPLICATIONS

As introduced earlier, Sybil attacks and replication attacks are easy to implement because false identities are easy to

978-1-4244-2324-8/08/$25.00 © 2008 IEEE.
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2008 proceedings.

TABLE II
CONFUSION MATRIX FOR CLOCK SKEW AND ID

	Same line	**Different line**
Same ID	Normal	Replication attack
Different ID	Sybil attack	Normal

generate in sensor networks. By analyzing the clock skew of each mote, we can divide the status into four parts. If the same node ID is in the same line, it is a normal condition; if different node IDs are in different lines, it is also a normal condition. However, if the same node ID has different lines, the network may be under replication attacks; if different node IDs are in the same line, the network may be under Sybil attacks, as shown in table II.

Here we take Sybil attack as an example application. Karlof et al. had pointed out the importance to defend Sybil attack while designing routing protocol in [15]. Also in [16], James et al. had proposed several approaches to analyze and defend against Sybil attacks. They use radio resource testing and random key predistribution to defend against the Sybil attacks in WSNs. Also a survey of defenses against the Sybil attacks is done in [17] and they separate the approaches into five categories, including trusted certification, resource testing, recurring costs, trusted devices, and observation. However, there are few attention on the detection for Sybil attacks. Therefore, we want to utilize our node identification to detect Sybil attack. Since we can distinguish the identification by using clock skew, the mechanism is described as following:

A. Sybil attack detection

An attacker, or a malicious node, starts Sybil attack by pretending to have multiple identities. A scenario of Sybil attack to FTSP is shown in Fig. 7. Node ID1 to ID3 are normal nodes and the Base Station is the gateway used to collect data from each node. Sybil stands for the Sybil node that is fabricating identities ID4 to ID7. In the beginning, the Sybil node claims itself as ID4 and after a while it changes its identities to ID5, ID6, and ID7 in sequence.

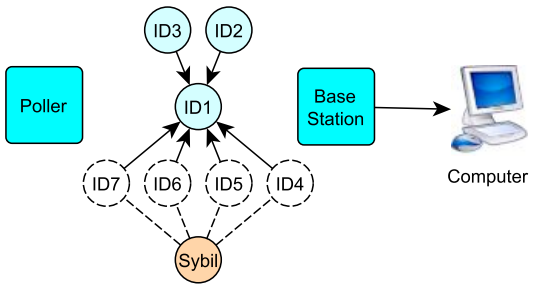

Fig. 7. The scenario of Sybil attack in our experiment.

After setting up the Sybil attack model, we use FTSP to calculate the clock skew between each node and gather the data from the base station. As shown in Fig. 8, ID1, ID2, and ID3 are the normal nodes with different average clock skews. ID4, ID5, ID6, and ID7 have the same pattern of clock skew and their average clock skews are all around 3.6ppm. After that, in order to check the continuity of the three nodes, three random variations from every 100 points are chosen. Because all the variations are less than 0.01ppm, we can easily find out that the whole network is under Sybil attack. In this case, we can easily apply our node identification method to detect Sybil attack.

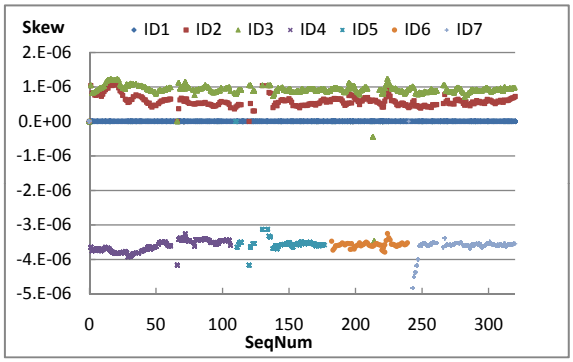

Fig. 8. Clock skew analysis for Sybil attack.

In addition, we believe that every node has few neighbors in the wireless sensor networks, so our node identification can be used to detect in-network Sybil attack. Thus, we propose an in-network detection model to detect the Sybil attack. As shown in Fig. 9, the Detection model is divided into two parts. The first part is data analysis. In this part, a mote collects the time information from its neighbor nodes and then computes all these clock skews. The average clock skews between each mote is calculated after gathering enough data (more than 100 samples). The second part is data grouping. The average clock skews are separated into groups according to the threshold in this part. As stated above, the threshold would be 0.15ppm. If there is only one average clock skew in the group, we state it as normal node. Otherwise, if there are more than two skews in the same group, we mark them as suspicious nodes. By confirming the continuity of the clock skews with the threshold 0.01ppm, these suspicious nodes can be further separated into two groups. If they are not in a continuous line, there must be several normal nodes in this group. Otherwise, we mark them as Sybil nodes. By this approach, we may simply identify every mote and detect in-network Sybil attack.

B. Replication attack detection

Replication attack happens when a malicious node tries to pretend some other nodes identity. As shown in table II, we can use the same method to detect the replication attack by first checking the continuity of the clock skew. If the line is not continuous, it means that the network may be under the replication attack.

C. Other possible attacks and the countermeasure

This paper mainly focuses on developing techniques to certify each mote in WSNs by utilizing the clock skew as

978-1-4244-2324-8/08/$25.00 © 2008 IEEE.
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2008 proceedings.

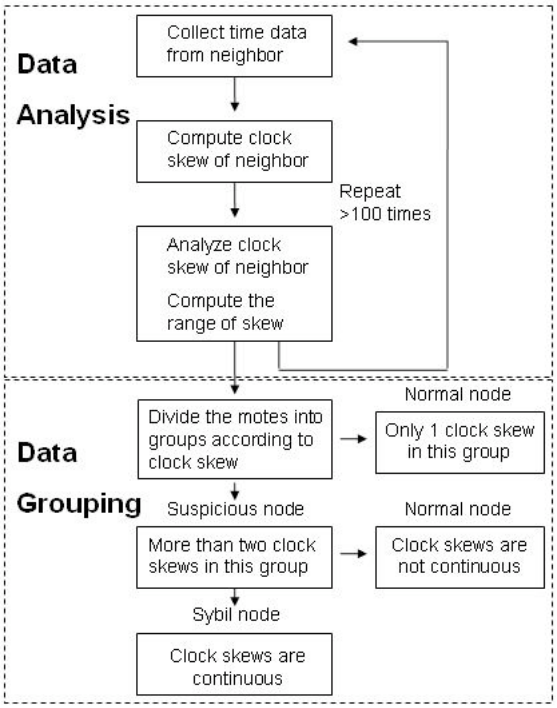

Fig. 9. In-network detection algorithm for Sybil attack.

a fingerprint. Although the approaches we described above are useful to identify each mote's identity, we suspect that some possible attacks may assault our node identification. For example, an adversary might try to alter the timestamps before the packet is sent. In this way, the adversary can change its own clock skew by faking timestamps and generating another identity. Actually Gehani et al. had proposed an approach, which is called PAST, to defend this attack [18]. They also show that PAST is low storage and power consumption overhead. Hence, in order to defend this vulnerability, we may apply the PAST to enhance our node identification as future work.

V. Conclusion

Node identification is vital to the use of wireless sensor networks in many applications. In this paper, we proposed a new node identification technique by utilizing clock skew as a fingerprint. By exploiting the FTSP, we confirmed the stability and the uniqueness of clock skew still holds in WSNs. For sensor nodes with very close clock skews, we developed a simple but effective method to distinguish their difference by utilizing the characteristics of the continuity between contiguous measured clock skews from the same mote. This method has the advantage that very little computing and little memory space required, especially comparing to other cryptographic approaches. We also realized the application of our node identification to detect Sybil attacks and other related attacks.

Acknowledgment

This work was supported in part by the International Collaboration for Advancing Security Technology (iCAST) and Taiwan Information Security Center (TWISC) projects under the National Science Council Grants NSC 96-3114-P-001-002-Y and NSC96-2219-E-009-013 respectively.

References

[1] T. Roosta, S. P. Shieh, and S. Sastry, "Taxonomy of security attacks in sensor networks and countermeasures," in *The First IEEE International Conference on System Integration and Reliability Improvements*, Hanoi, Vietnam, December 2006.

[2] J. R. Douceur, "The sybil attack," in *IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*, London, UK, 2002, pp. 251–260.

[3] H. Fu, K. Satoshi, M. Zhang, and L. Zhang, "Replication attack on random key pre-distribution schemes for wireless sensor networks," in *Information Assurance Workshop, 2005. IAW '05. Proceedings from the Sixth Annual IEEE SMC*, June 2005, pp. 134–141.

[4] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz, and A. Khalili, "A pairwise key predistribution scheme for wireless sensor networks," in *ACM Transactions on Information and System Security*, vol. 8, no. 2, New York, NY, USA, 2005, pp. 228–258.

[5] A. Perrig, R. Szewczyk, V. Wen, D. E. Culler, and J. D. Tygar, "Spins: security protocols for sensor netowrks," in *Mobile Computing and Networking*, Rome, Italy, 2001, pp. 189–199.

[6] R. Watro, D. Kong, S. fen Cuti, C. Gardiner, C. Lynn, and P. Kruus, "Tinypk: securing sensor networks with public key technology," in *SASN '04: Proceedings of the Second ACM Workshop on Security of Ad hoc and Sensor Networks*, 2004, pp. 59–64.

[7] T. Kohno, A. Broido, and K. C. Claffy, "Remote physical device fingerprinting," in *SP '05: Proceedings of the 2005 IEEE Symposium on Security and Privacy*, May 2005.

[8] S. J. Murdoch, "Hot or not: revealing hidden services by their clock skew," in *CCS '06: Proceedings of the 13th ACM Conference on Computer and Communications Security*, New York, NY, USA, 2006, pp. 27–36.

[9] M. Maróti, B. Kusy, G. Simon, and Ákos Lédeczi, "The flooding time synchronization protocol," in *SenSys '04: Proceedings of the Second International Conference on Embedded Networked Sensor Systems*. ACM, 2004, pp. 39–49.

[10] S. Ganeriwal, S. Capkun, S. Han, and M. B. Srivastava, "Secure time synchronization service for sensor networks," in *the ACM Workshop on Wireless Security (WiSe)*, October 2005.

[11] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," in *OSDI '02: Proceedings of the Fifth Symposium on Operating Systems Design and Implementation*, vol. 36, 2002, pp. 147–163.

[12] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks," in *SenSys '03: Proceedings of the First International Conference on Embedded Networked Sensor Systems*, Los Angeles, CA, USA, September 2003, pp. 138–149.

[13] T. Roosta and S. Sastry, "Securing flooding time synchronization protocol in sensor networks," in *First International Workshop on Embedded Systems Security*, 2006.

[14] TinyOS, http://www.tinyos.net/.

[15] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," in *First IEEE International Workshop on Sensor Network Protocols and Applications*, May 2003, pp. 113–127.

[16] J. Newsome, E. Shi, D. Song, and A. Perrig, "The sybil attack in sensor networks: analysis & defenses," in *IPSN'04: Proceedings of the Third International Symposium on Information Processing in Sensor Networks*, 2004, pp. 259–268.

[17] B. N. Levine, C. Shields, and N. B. Margolin, "A survey of solutions to the sybil attack," University of Massachusetts Amherst, Tech report 2006-052, October 2006.

[18] A. Gehani and S. Chandra, "Past: probabilistic authentication of sensor timestamps," in *Computer Security Applications Conference, 2006. ACSAC '06. 22nd Annual*, Miami Beach, Florida, USA, December 2006, pp. 439–448.

978-1-4244-2324-8/08/$25.00 © 2008 IEEE.
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2008 proceedings.