Performance Evaluation of Key Disclosure Delay-Based Schemes in Wireless Sensor Networks

Wafa Ben Jaballah*, Mohamed Mosbah*, Habib Youssef†
*Univ. Bordeaux, LaBRI, UMR 5800, F-33400 Talence, France
CNRS, LaBRI, UMR 5800, F-33400 Talence, France
Email: wafa.benjaballah@labri.fr, mosbah@labri.fr
†University of Sousse, Prince Research Unit, GP1, 4011, Hammam Sousse, Tunisia
Email: habib.youssef@fsm.rnu.tn

***Abstract*—Broadcast source authentication is a critical security service in wireless sensor networks which is still in its infancy. This service allows senders to broadcast messages to multiple receivers in a secure way. This paper evaluates the integration of staggered authentication in multi-level μTesla source authentication protocol called staggered multi-level μTesla. These two protocols are evaluated in terms of authentication delay, authentication probability, number of forged packets in the receiver's buffer, delay of forged packets in the receiver's buffer, memory, and energy consumption overhead. Simulation results show that these two protocols introduce negligible overhead without impeding the system performance. Moreover, staggered multi-level μTesla achieves better performance compared to multi-level μTesla, when reducing the average number and the delay of forged packets in the receiver's buffer.**

***Keywords*-Source Authentication; Key Disclosure Delay; TOSSIM Simulator; Performance Evaluation**

I. Introduction

In recent years wireless sensor networks (WSNs) have been attracting increased attention from the research and industrial community, motivated by applications like border protection, healthcare, civil applications, etc. These networks are a collection of sensors with constrained resources that are deployed in holistic environments. These sensor nodes gather data about the changes in their surroundings (in civilian as well as in military applications), and report these changes to a data sink. These networks are prone to many kinds of attacks. A malicious node could inject some bogus information in the network and let the honest nodes believe that it is an authentic participant in the network, thereby acquiring all the information traversing the network. This process refers to a source authentication misleading. Several attacks could be launched in the network, when there is no message source authentication [1], [2], [3], [4]. For instance, in healthcare application, a non desirable participant could inject false information, which could lead to a disaster. Since the malicious node has legitimate information, it may participate in the network operations; hence nodes can launch several kinds of attacks. Thus, providing source authentication of the messages transferred through the networks, is very crucial.

Authentication protocols presented in the literature [1], [2], [3], [4] have described a variety of ways in which the authentication function may be carried out. In fact, source authentication protocols could be classified into three categories. The first category is the signature based schemes [5], [6], [7], [8], [9], which requires the use of pairing operations. This first category involves asymmetric cryptography which is a heavy burden both at the sender and the receiver sides. Therefore, this category is not suitable for wireless sensor networks. The second category deals with asymmetric information (where each node is assigned a share in secret i.e. secret keys) [10]. In asymmetric information, a receiver can authenticate the source of a message without being able to generate the received MACs. This helps prevent the impersonation of data sources. Each node is assigned a set of keys using a key distribution scheme. A source concatenates the MACs related to the different keys, however a receiver could only verify the authenticity of the received MACs, without forging the MACs of other nodes. The problem of this category is related to the challenge between collision resilience and performance impact. Therefore, symmetric key based schemes [2], [3], [11], and especially time asymmetry are a good alternative to cope with the source authentication of messages in constrained networks such as WSN. One of the known and efficient time asymmetry schemes, based on key disclosure in WSNs, is μTesla [3]. The principle of time asymmetry is that a MAC is only valid on a time interval, thus forged packets traversing the network can be easily detected. For instance, a key remains secret until the expiration of a certain delay (few time intervals), and it will be disclosed after some time intervals. This process refers to temporal asymmetry based source authentication schemes.

In this paper, we focus on the evaluation of multi-level μTesla and a staggered multi-level μTesla protocols [12]

[1]This work is partially supported by the cluster of excellence CPU, University of Bordeaux.

within the TinyOS operating system. These protocols provide very good source authentication service while respecting the WSN constraints. These protocol implementations are evaluated and validated in terms of authentication delay, authentication probability, resilience against DoS attacks, memory and energy consumption overhead. The paper is organised as follows. Section II gives a brief overview of multi-level μTesla and staggered multi-level μTesla. In Section III we evaluate the performances of the proposed schemes, and we study their complexity as well. We conclude in Section IV.

II. Related Work

Each sensor node (MicaZ, TelosB) uses an open-source operating system called TinyOS, designed for embedded wireless sensor networks. In the following, we present an overview of two key disclosure based source authentication schemes (multi-level μTesla, and staggered multi-level μTesla).

A. Overview of Multi-level μTesla

Multi-level μTesla [11] is a source authentication protocol, based on multi-level key chains, which are used to enhance scalability with respect to the number of receivers. The authentication keys are derived from a one way hash function. This set of nodes form a one way hash chain. Initially, a source node picks a random value which is the first key of the chain. The different keys are generated recursively by applying a public one way hash function. Each key in the chain will be used to generate the MAC for the data packets. After verifying the authenticity of the disclosed key, the receiver is able to authenticate a buffered message. The receiver could generate an old key based on the received key and the hash function, without being able to guess the future key.

We can take the example of two-level key chains. The two-level key chains consist of a high-level key chain and multiple low-level key chains. The low-level key chains are used for authenticating broadcast messages, while the high-level key chain is used to authenticate commitments (or first key) of the low-level key chains. The low-level key chains have short enough intervals so that the delay between the reception and the verification of the messages is tolerable.

Multi-level μTesla offers scalability for large sensor networks, low overhead, tolerance of message loss, and resistance to replay attacks. There are two types of packets in multi-level μTesla: Commitment Distribution Message (CDM) packets and data packets. In order to use a low level key chain $< k_{i,0} >$ during the time interval T_i, sensor nodes must authenticate the commitment $K_{i,0}$ before T_i. To achieve this, the sender broadcasts a commitment distribution message (CDM_i) during each time interval T_i. We have: $CDM_i = i|k_{i+2}|MAC(k_i^{'}|i|k_{i+2})|k_{i-1}$.(1) where the | symbol denotes message concatenation, and k_i is derived from key k_{i-2} with a pseudo random function. The key k_i is generated in time interval T_i. The sender broadcasts a data packet generated in time interval T_i according to: $P_{i,j} = level\_number|index|M_{i,j}|MAC(k_i, M_{i,j})|k_{i-d}$ (2) where $level\_number$ represents the level of the hash chain, k_{i-d} represents the disclosure key in time interval T_i that was generated in T_{i-d} , and d represents the key disclosure delay. The following Figure 1 represents an example of the key disclosure mechanism between the sender and the receiver. The key k_i will be secret in Time T_i, and it will be disclosed in Time T_{i+d}.

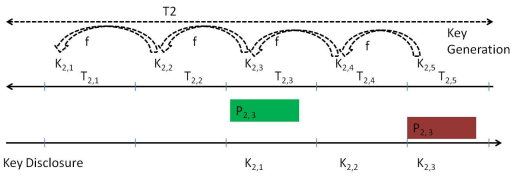

Figure 1. Multi-level μTesla: An example of execution

B. Overview of Staggered Multi-level μTesla

Staggered multi-level μTesla is an extended version of multi-level μTesla, that is proposed in [12]. The goal of this protocol is to reduce the delay of forged packets in the receiver's buffer. The basic idea is to split the time into equal length intervals. Thus, for each time interval corresponds an authentication key to all packets that are generated in this time period. All the keys are derived using a publicly one-way function. Staggered multi-level μTesla uses different MACs from successive multi-level μTesla keys. Thus, many malicious nodes are not able to forge all the generated MACs.

The additional computation and communication requirements introduced by the extra MACs will not cause significant performance degradation. When it receives a packet, the receiver puts the packet at the head of the queue, and degrades the packet to lower layers as additional keys arrive and the corresponding MACs are verified. If the verification fails, the packet is dropped from the queue. When the final key involved arrives and the corresponding MAC is verified, then complete authentication is achieved.

Thus, the receiver does not have to wait for d time intervals in order to start authenticating packets. Instead, the receiver can use any received keys to begin the authentication process and can thus promptly remove bogus packets. Hence, the number of forged packets in the receiver's buffer is decreased and the scheme is more resistant to DoS attacks. In staggered multi-level μTesla, the j^{th} data packet generated in T_i is constructed as follows:

$$P_{i,j} = M_{i,j}|MAC(k_i, M_{i,j})|MAC(k_{i-1}, M_{i,j})| MAC(k_{i-2}, M_{i,j})|...|MAC(k_{i-d-1}, M_{i,j})|k_{i-d}$$

Figure 2 represents an example of a staggered multi-level μTesla between the sender and the receiver. In fact, the receiver does not wait for three time intervals (in this example, d=3) to start authenticating packets (packet $P_{2,3}$ in Figure 2). The receiver can use a received key in a time interval to start the authentication process, and can thus promptly remove $P_{2,3}$ if it is a bogus packet.

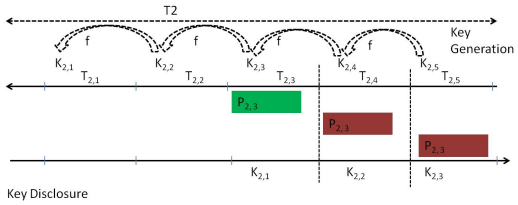

Figure 2. Staggered multi-level μTesla: An example of execution

III. Performance Evaluation

In this section we show that staggered multi-level μTesla satisfies the following requirements: low computation overhead, low communication overhead, low energy consumption, low authentication delay, and security robustness. Further, we compare staggered multi-level μTesla and multi-level μTesla, and show that staggered multi-level μTesla outperforms multi-level μTesla in several ways.

A. Simulation Environment

This section presents the implementation of two source authentication schemes in the TinyOS operating system and assuming TelosB sensors. The TelosB motes used in our simulations, run TinyOS operating system version 2.1 and support NesC as a programming language [13]. We used the meyer-heavy noise model which is a noise trace taken from Meyer Library at Stanford University. Furthermore, TinyOS is written in NesC language, that supports the concurrency model and TinyOS component. In this operating system, each component in our application corresponds to a hardware element (timer, ADC, led). Moreover, each application in TinyOS englobes different components. Each component should define its events, tasks, and commands.

Figure 3 shows the different interfaces as well as the interfaces of our schemes. According to the implementation we obtained, we use several Timer interfaces for handling message sending and reception. TimerMilliC is the standard millisecond timer abstraction. In order to enable the transmission of messages, our application uses the AMSenderC, AMReceiverC, MMTESLAReceiverC, and MMTESLASenderC components that provide interfaces Packet, AMPacket, RadioSend, RadioReceive, Primitive and Buffer. Moreover, we define less variables to save memory requirements in RAM and ROM. AMSender is a virtualized abstraction and can deal with a single packet. In our implementation, we used a component ActiveMessageC, which our implemented packet components (AMSenderC, AMReceiverC, etc.) wire to. Moreover, ActiveMessageC is a configuration that renames a particular radio chips active message layer. Furthermore, we use a RandomLFSR component to generate pseudo-random numbers required by key chains. We choose the RC5 block cipher component to authenticate and encrypt messages, since it has been tested in sensor platform. During testing, we used the LedsC component that turns the LEDs on and off. As LedsC consumes a significant energy, we replaced it with NoLedsC in the deployment phase.

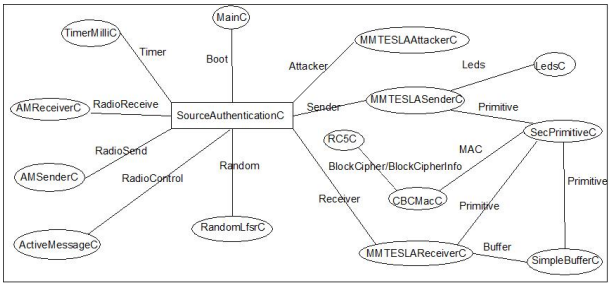

Figure 3. Our application architecture

However, in implementing the two schemes, we should allow a small available payload size of packets, since the standard packet payload is limited to 29 bytes under TinyOS. In multi-level μTesla scheme, it is easy to send all the data in fragments of 29 bytes. In staggered multi-level μTesla scheme, we could not send all the amount of data using fragements of 29 bytes. Hence, we modified the application Makefile using "$TOSH\_DATA\_LENGTH = 100$" in order to have a packet with maximum 100 bytes, since in 802.15.4, the packet should not exceed 128 bytes of length.

We notice that TinyOS has memory constraints. In fact, it allows for static memory allocation, in order to deal with the severe hardware constraints of sensor nodes. This makes it very space and time efficient. However, at compile time we should specify the used variables and their size. Moreover, if the application requires more memory than the available RAM, an EEPROM (also called flash) might be used. To validate our simulations with TOSSIM, we experimented with TelosB motes. However, the TelosB motes need synchronization since the implemented protocols require strict node's time synchronization. In order to cope with this constraint, we use the flooding time synchronisation protocol [14] because it is very efficient compared to other protocols and it had been tested also in TelosB motes.

B. Performance Results

In this section, we present performance results in terms of authentication probability, authentication delay, number and delay of forged packets in the receiver's buffer, and memory overhead. Simulations were conducted using TOSSIM [13]. Moreover, we evaluate the node energy cost needed to authenticate the source of a transmitted packet. This is

achieved using the PowerTOSSIM simulator [15]. We focus on the evaluation of the broadcast data packets. In our simulations, we use a sender, an attacker, and a receiver component, with the following setting.

1) Parameter Setting: The multi-level μTesla key disclosure delay is 3 time intervals, and each multi-level μTesla time interval is 100 ms. The key chain in multi-level μTesla consists of 600 keys. We assume also there are 200 multi-level μTesla instances, which spread to 200 minutes in time, as presented in [11].

The payload of each CDM and data message in multi-level μTesla is 29 bytes. We assume initially that the data packet rate from the source node is 100 data packets per minute, and at each sensor node we allocate 3 buffers for data packets. To evaluate the authentication probability under DOS attacks and communication failures, we consider the scenario when the attacker sends forged packets spanning from 100 to 900. We also assume a channel loss rate of 0.5. To enhance the accuracy of our results, 50 simulations have been run and their outcomes averaged to produce charts presented in the following subsections.

2) Authentication Probability: The authentication probability is the fraction of the received authenticated packets divided by the number of received packets. Figure 4 reports the authentication probability, for the two protocols. The x-axis of Figure 4 indicates the buffer capacity of the receiver while the y-axis shows the corresponding authentication probability of the messages. In fact, the authentication probability slightly increases when the buffer capacity is increased. When we assume the same buffer capacity, while varying the attack rate, the authentication probability decreases when the attack rate increases. It is interesting to note that for multi-level μTesla, the authentication probability increases when we decrease the attack rate. Moreover it also increases when increasing the buffer capacity. The two schemes always perform a higher authentication rate when the buffer capacity increases. The argue is that, once a sensor node receives a later disclosed key, it will be able to authenticate buffered packets. In multi-level μTesla scheme, when a key is lost due to channel losses or DOS attacks, a receiver node has to wait for a long time to recover the authentication key. Thus, several buffered packets are already dropped during this waiting time. It is straightforward to prove that in order to achieve a given authentication probability multi-level μTesla has to allocate a large buffer, especially when are severe DOS attacks. However, the staggered multi-level μTesla is more efficient since it does not need an additional buffer to achieve a higher authentication probability. Figure 4 also shows that the behaviour of staggered multi-level μTesla corresponds to that of an ideal protocol. The reason is that in this scheme, a sensor node can verify a data packet immediately and when receiving any disclosed key on the later time interval, however in the multi-level μTesla scheme, a receiver has to wait for the key disclosure delay before

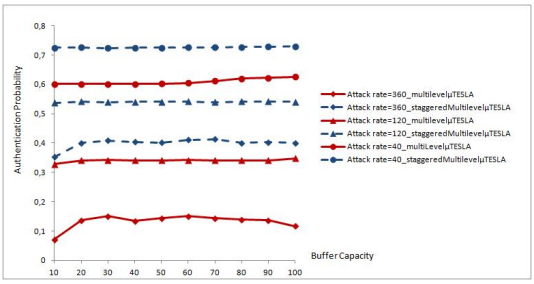

Figure 4. Authentication Probability

authenticating data messages.

Figure 5 shows the behaviour of multi-level μTesla when implemented on TelosB motes. The x-axis indicates the duration of an attack, while the y-axis shows the corresponding authentication probability of the messages, while varying the attack rate. The authentication probability increases slowly when the duration of the attack is low.

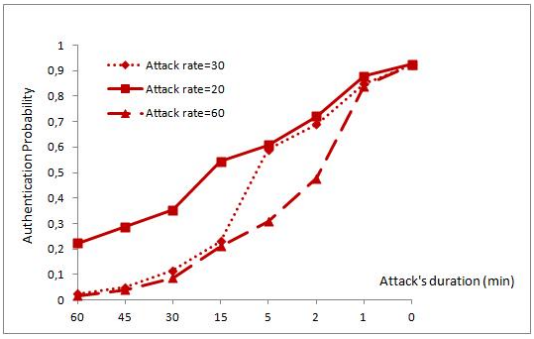

Figure 5. Authentication Probability versus attack rate and duration using TelosB motes

3) Authentication delay: To measure the time for computing the authentication delay, a SysTime component is used. The time to compute the average authentication delay (the delay of packet authentication minus the delay of received packet) is nearly the same. This metric varies when varying the loss rate. As it is shown in Figure 6, the authentication delay increases when the loss rate increases too. For example, when the loss rate= 20%, the authentication delay is 500 ms. However, for loss rate= 80%, the authentication delay is 1500 ms.

4) Forged Packets: Figure 7 reports the average number of forged packets with varying values of the attack rate using experiments on TelosB motes. We could see that the average number of forged packets increases when the capabilities of the malicious node become stronger. When the attack duration = 60 min, then the average number of forged packets is superior to 1200 (when attack rate= 60), however it is less than half that value when attack rate = 20.

Figure 8 reports the average delay of forged packets with

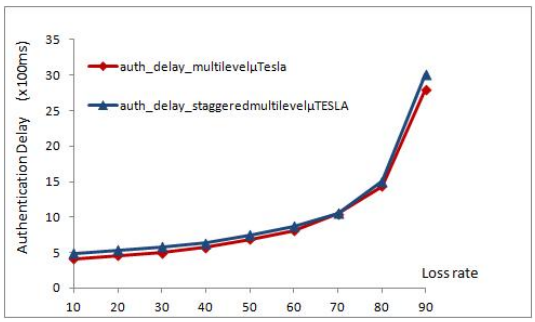

Figure 6. Authentication delay

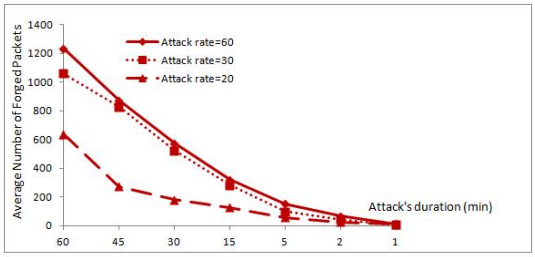

Figure 7. Average Number of Forged Packets in multi-Level μTesla using TelosB motes

varying loss rates. We could see that the delay of forged packets is much lower than the authentication delay.

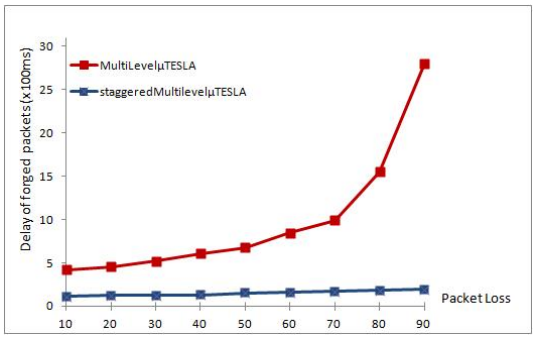

Figure 8. Delay of Forged Packets in the Buffer

In multi-level μTesla, the delay of forged packets is roughly the same as the authentication delay. This could be explained by the fact that in staggered multi-level μTesla, forged packets are quickly dropped and deleted, since received packets are only authenticated after d time intervals (d=3). Figure 9 represents the delay of forged and authenticated packets in staggered multi-level μTesla. The delay of forged packets in the receiver's buffer varies with the loss rate. Furthermore, we remark that staggered multi-level μTesla scheme allows a small delay of forged packets since these packets do not wait for the disclosure of one key. In this scheme, the delay of forged packets is between $(1.5 \times 100ms)$ and $(2 \times 100ms)$. This also demonstrates that forged packets in the staggered multi-level μTesla are quickly removed from the receiver's buffer. Forged packets remain a small time in the receiver's buffer, while the authenticated packets are only authenticated after receiving the three authentication keys, since the disclosure authenticated key is three time intervals.

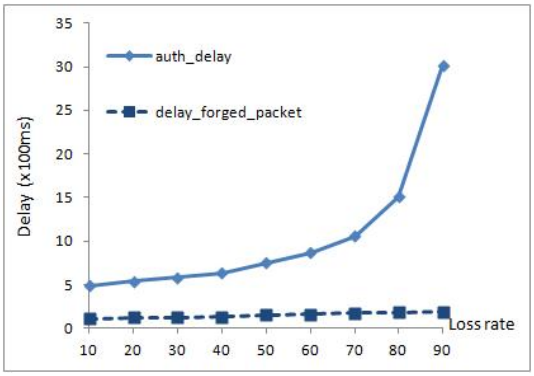

Figure 9. Delay of Forged and Authenticated Packets in Staggered multi-level μTesla

Figure 10 represents multi-level μTesla when varying the delay of forged and authenticated packets. In fact, these two delays are almost the same because forged and authenticated packets have to wait for receiving the three keys. For a loss rate = 10%, the delay is less than $(5 * 100ms)$ in multi-level μTesla.

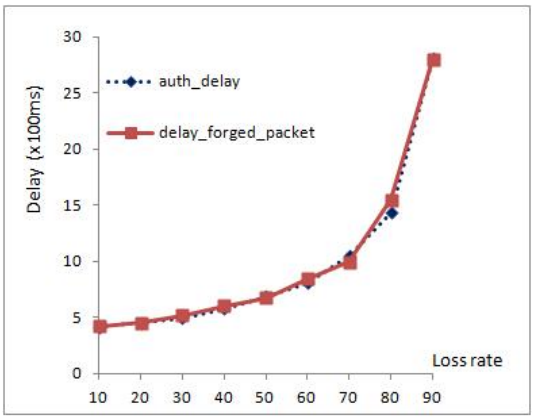

Figure 10. Delay of Forged and Authenticated Packets in multi-level μTesla

5) Memory requirements: Our implementation of multi-level μTesla in TelosB occupied approximately 2666 (bytes) in RAM and 24786 (bytes) in ROM, which represents 25% of the ROM and 50% of the RAM. Staggered multi-level μTesla occupied approximately 3318 (bytes) in RAM and 25962 (bytes) in ROM. The increase of the RAM in the

second scheme is due to the staggered mechanism, and the access to the receiver's buffer for each time interval.

6) Energy Overhead: We evaluate the energy consumption of the two schemes using powerTOSSIM-Z [15]. In fact, powerTOSSIM-Z is a realistic energy model in TOSSIM for the micaZ mote. PowerTOSSIM-Z uses the TinyOS and TOSSIM to track power consumption. Hardware components such as radio and Leds call to the PowerState module. MicaZ motes [16] require two AA batteries that provide an initial energy of 2850 mAh powered by a 3v voltage. A sensor node provides initially an energy of 30780 joule, according to the following formula.

$$Watt = Joules/sec = Volt * Ampere$$

The energy overhead introduced by these two applications is 248818 mj for multi-level μTesla and 248327 mj for Staggered multi-level μTesla, which is very small considering the energy of a mote ($< 1\%$). These values are computed using the total energy consumption by a mote for a data rate equal to 100 data packet per minute, and an attack rate=60 packets per minute. PowerTOSSIM-Z uses a non linear energy model. In fact, it includes a module named PowerCurses which uses interfaces to show the actual battery state of the mote. Due to the space limit, we didn't show the battery state of the different motes for the two schemes.

Energy is the scarcest resource. We demonstrate with our techniques that key disclosure based source authentication protocols can become an integral part of practical sensor networks.

IV. Conclusion

In this paper, we performed extensive simulations of multi-level μTesla and staggered multi-level μTesla, and estimated their authentication delay and authentication probability. Moreover, we have demonstrated that the overhead of such protocols should be small. Performance comparisons show that key disclosure based solutions exhibit better performance in terms of authentication probability, energy overhead, and resilience to DOS attacks.

References

[1] T. Kavitha and D. Sridharan, "Security Vulnerabilities In Wireless Sensor Networks: A Survey," *Journal of Information Assurance and Security*, vol. 5, pp. 31–44, 2010.

[2] O. D. Mohatara, A. F. Sabatera, and J. M. Sierrab, "A lightweight authentication scheme for wireless sensor networks," *Journal of Ad hoc networks*, vol. 9, pp. 727–735, Jul. 2011.

[3] A. Perrig, R. Szewczyk, D. C. V. Wen, and D. Tygar, "Spins: Security protocols for sensor networks," in *Proc. Seventh Annual International Conference on Mobile Computing and Networks*, 2002.

[4] I. Krontiris and T. Dimitriou, "Scatter – secure code authentication for efficient reprogramming in wireless sensor networks," *International Journal of Sensor Networks*, vol. 10, pp. 14–24, 2011.

[5] F. Hess, "Efficient identity based signature schemes based on pairings," in *Proc.SAC,St.John's*, Newfoundland,Canada, 2002.

[6] Z. Yu, Y. Wei, B. Ramkumar, and Y. Guan, "An Efficient Signature-Based Scheme for Securing Network Coding Against Pollution Attacks," in *Proc. the 27th Conference on Computer Communications INFOCOMM*, 2008, pp. 1409–1417.

[7] C. Zhang, R. Lu, X. Lin, P. Ho, and X. Shen, "An Efficient Identity-Based Batch Verification Scheme for Vehicular Sensor Networks," in *Proc. the 27th Conference on Computer Communications INFOCOMM*, 2008, pp. 246–250.

[8] S. Y. Chang, Y. H. Lin, H. M. Sun, and M. E. Wu, "Practical RSA signature scheme based on periodical rekeying for wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 8, Mar. 2012.

[9] X. Fan and G. Gong, "Accelerating signature-based broadcast authentication for wireless sensor networks," *Journal of Ad Hoc networks*, vol. 10, pp. 723 – 736, 2012.

[10] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, and B. Pinkas, "Multicast security: a taxonomy and some efficient constructions," ser. IEEE INFOCOM'99, vol. 2, Mar. 1999, pp. 708–716.

[11] D. Liu and P. Ning, "Multi-Level TESLA: Broadcast Authentication for Distributed Sensor Networks," *ACM Transactions in Embedded Computing Systems (TECS)*, vol. 3, no. 4, pp. 800–836, Nov. 2004.

[12] W. B. Jaballah, A. Meddeb, and H. Youssef, "An efficient source authentication scheme in wireless sensor networks," in *ACS/IEEE International Conference on Computer Systems and Applications- AICCSA 2010*, May 2010.

[13] P. Levis, N. Lee, M. Welsh, and D. Culler, "TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications," in *First ACM Conference on Embedded Networked Sensor Systems Sensys*, Nov. 2003.

[14] M. Maroti, B. Kusy, G. Simon, and A. Ledeczi, "The flooding time synchronisation protocol," in *2nd international conference on Embedded networked sensor systems*, Nov. 2004, pp. 39–49.

[15] E. Perla, R. S. Carbajo, M. Huggard, and C. M.Goldrick, "PowerTossim z: realistic energy modeliing for wireless sensor network envrionments," in *3nd ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks*, Oct. 2008, pp. 35–42.

[16] "Crossbow technology inc., www.xbow.com."