Efficient User and Broadcast Authentication Scheme for WSNs

Amerah Alabrah
Department of Electrical Engineering and Computer Science
University of Central Florida, Orlando, Florida
College of Computer and Information Sciences
King Saud University, Riyadh, Saudi Arabia
amerah@knights.ucf.edu

Mostafa Bassiouni
Department of Electrical Engineering and Computer Science
University of Central Florida, Orlando, Florida
USA
bassi@cs.ucf.edu

Abstract— Wireless sensor networks are collections of tiny sensors working collaboratively to gather information. Two major shortcomings of today's WSNs are scarce energy and memory resources making it difficult to devise robust and effective security measures. Keyed-hash chains have been proposed as an efficient solution. Nevertheless, the basic configuration of one-way hash chains, where a shared key is hashed with no limits on the length of the chain causes an unwanted computational overhead and a potential security breach. In this paper, we propose a mini one-way hash chain protocol for user and broadcast authentication that is as effective, while at the same time significantly reducing the computational overhead. We avoid the potential security breach by using easily computed one time authentication tokens to secure communication. We also demonstrate that our protocol is energy preserving, light and efficient. Our simulation and evaluation tests reflect its benefit over straightforwardly configured one-way hash chains.

Key Words: WSN, one-way hash chain, SHA-1, MOHC.

I. INTRODUCTION

A wireless sensor network (WSN) is a collection of tiny sensor nodes that work collaboratively to gather information. Typical shortcomings of wireless sensor nodes include their limited storage and power resources. Versatile as they are, WSNs have gained acceptance and are being widely used in various arenas such as military sensing and tracking, traffic monitoring, industrial quality control, medical monitoring, etc. With the fast rate of acceptance of such networks, they are expected to grow both in capacities and capabilities.

One of the important issues slowing down this growth in deployment lies in their security vulnerability. Attacks on WSNs can be as easy as intercepting messages, falsifying data and impersonating users to gain access. Such attacks can only be avoided by exercising robust and effective security measures that prevent unlawful access of unauthorized entities. The fact that sensor nodes in WSNs are not well-endowed in terms of memory and energy makes it necessary for solutions addressing such attacks to be lightweight, efficient and memory/energy preserving.

Solutions proposed to address the problem of security issues in WSNs are still work in progress. Specifically, authenticating the data broadcast and users over a WSN is very crucial. Researchers such as [1], [2], [3] suggest using public-key based solution. However, with limited energy and memory capabilities, such solutions can pose difficulties in deployment due to high cost of calculating public keys. Alternatively, solutions based on the keyed-hash chains have been proposed [4], [5]. However, if used in the basic configuration where a shared key is hashed with no limits on the length of keyed-hash chain, two major problems arise. First, the shared key can be compromised, and, hence, data might be at risk putting the whole WSN in danger. Second, the length of the keyed-hashed chain can grow exponentially leading to an unjustified overhead that cannot be handled by a sensor node with limited memory and energy. Therefore, in this paper, we address these two problems and present a broadcast and user authentication scheme for WSNs that is both lightweight and efficient with the capability of mutual authentication. Our solution is based on the idea of using a shared key only once, and, then, substituting this shared key with an authentication token in the subsequent communication iterations. The authentication tokens are derived by applying a hash function (e.g. SHA-1, SHA-2 etc.) to a shared key in the initial communication. In the subsequent transactions, easily computed authentication tokens are utilized instead of the shared key. Our solution can be easily applied in either a WSNs deployed in harsh environments or lenient environments. The simulation of the proposed solution and the evaluative experiments demonstrate its effectiveness and computational economy. To evaluate the proposed solution's effectiveness, we compare our results with one-way hash chain protocols straightforwardly configured.

The remainder of the paper is as follows: Section II surveys some of the related literature. Section III introduces the network model. Section IV introduces the proposed scheme and highlight its main features. The simulation and evaluation experiments are presented in section V, and in section VI, we present the performance evaluation results. Finally, section VII is a conclusion.

II. RELATED WORK

Different aspects of WSN security have been addressed in the literature. Among the problems are user authentication [1], [2], [4], [5], [16], and broadcast authentication [3], [6]. WSN security solutions have a general goal of improving WSN standards in terms of *authenticity, confidentiality* and *integrity*. Conventional user and broadcast authentication mechanisms are not as practical in WSNs due to the scarcity of memory and energy capacities of sensor nodes.

Conversely, one of the best solutions for user and broadcast authentication is the use of public-keys to secure

978-1-4799-3083-8/14/$31.00 ©2014IEEE

communication. For example, Benenson et al. proposed a solution in which user authentication is achieved by adopting public-key cryptography [1]. They basically use a certificate/signature generated by a base station. What renders this technique impractical for WSN is the high computation cost and large signature size. In addition, they make WSN susceptible to DOS attacks draining energy resources. To remedy this problem, shortened public-key based solutions designed for WSNs have been suggested. Addressing the problem of expensive user authentication if public-keys are used, Wang et al. suggested using a short public key whose lifetime is much shorter than regular public keys [3]. This solution aims at reducing computation, but it still suffers from vulnerability to DOS attacks.

An alternative and equally attractive solution is the use of one-way hash chains to achieve user and broadcast authentication. One of the influential papers to take this route is the μTESLA scheme introduced in [7] designed to achieve broadcast authentication. In μTESLA, the basic idea is to have a base station, assumed to be trusted all the time, that acts as a user of sensor nodes' services, i.e. the base station is where sensed information is collected and authenticated. Authentication is achieved by using one-way hash function $h()$ and using the hash preimages as keys in the Message Authentication Code (MAC). In the initialization stage, the base station calculates a hashed value of a secret x based on the number of nodes in the network and distributes it among relative sensor nodes in a unicast fashion. In the subsequent transactions, a preimage of the hashed value is used as a key in MAC. In the next stage, the sensors verify whether or not the hashed value and its preimage are consistent. The problem with this technique is low scalability and the high computational cost due the unicast nature of key distribution. To further improve μTESLA, Liu and Ning [8] suggested replacing unicast distribution of keys with a broadcast.

One-way hash based solutions have also been attractive in user authentication schemes of WSNs. One of the earliest implementations of user authentication employing one-way hash is introduced in [12]. The one-way hash operation is used in the initial stages to verify the user requesting access to the WSN is an authorized. Subsequent user authentication schemes which try to overcome the shortcoming of [12] include [9], [10] and [11]. Unlike the one-way hash chains used in broadcast authentication where computational overhead is an issue of concern; user authentication schemes do not suffer from such overhead since the one-way hash operations is only needed in the login stage.

In this paper, we propose a user/broadcast authentication scheme for WSNs based on one-way hash chain cryptography. Our scheme basically addresses the problem of computational overhead and potential key compromise suffered by most of the schemes surveyed earlier. Our scheme, inspired by the idea of one-time credentials [13] used to protect internet cookies, utilizes the concept of authentication tokens used only once to protect the data stream in the WSN traffic. Utilizing memory based solutions such as [17] is not optimal in this context of WSN due to memory constraints. We modify the traditional one-way hash chain constructions to achieve broadcast authentication in an efficient and secure way.

One-way Hash Chains (OHC)

Since we are using the *one-way hash protection scheme* as the backbone for our solution, it is worth illuminating its main aspects and how its hashing operation is carried out to protect communication. In the OHC scheme, a one-way hash chain of length N is used to protect a stream of N transactions of a web session (in WSNs, we replace a web session with a WSN session). During the initial steps, the base station and the sensor node(s) exchange a shared secret value S_0, and a value N which refers to the chain length (chain length can be the number of sensor nodes like in [7], or fixed as will be seen herein). The OHC protects the j^{th} transaction by computing an authentication token $V_j=H^{N-j+1}(S_0)$, where the notation $H^m(x)$ implies applying the hash function m times, for example, $H^2(x)= H(H(x))$. For instance, if N=100, then the authentication tokens for the 1st, 2nd, and 3rd transactions are $V_1=H^{100}(S_0)$, $V_2=H^{99}(S_0)$, $V_3=H^{98}(S_0)$, respectively. The main drawback of the OHC approach is its high computational overhead especially with high numbers of N.

If used in this fashion for WSNs, the one-way hash scheme would have two major shortcomings. First, the above description of OHC incurs very high computation overhead especially as the number of N increases which sensor nodes cannot afford due to memory and energy limitations. Second, there is a high risk that the shared secret gets compromised. If this secret is sniffed or accessed, the whole network might be jeopardized. Our solution addresses these two shortcomings.

To solve the problem of computational overhead, we essentially shorten the length of the key chain to a degree that can easily be handled by the sensor nodes. As for the risk of secret compromise, we basically use the secret once in the first communication iteration. Subsequently and instead of exposing the secret throughout the connection, we only use it once. The secret is used in the first round of communication between each sensor node and the coordinating entity in a hashed function. In the subsequent rounds of communication, we use the previous authentication token in a hash function, concatenated with other parameters, to create the next authentication token for each sensor. In other words, the sensor node is only required to save the previous authentication token and use it in the next iteration. Thus, authentication in our solution is ongoing as long as the communication is active. At any given point, the base station and the sensor nodes will compute the authentication tokens and they have to match in both sides.

III. NETWORK MODEL

For our protocol, we envision a star topology network where communication between the user and the WSN is mediated by a central entity, called *coordinator.* Essentially, the *coordinator* is a sensor node with special capabilities. The role of the coordinator is to manage communication in the WSN by receiving queries from the users, communicating with sensor nodes to get the sensed information, and fetching

the results of users' queries back to the user. With this in mind, we assume the coordinator to be responsible for authentication in both directions. In the user's direction, the coordinator is expected to only allow identified users holding the right credentials to access the network. In the sensor nodes direction, on the other hand, the coordinator(s) delegates only authenticated sensors that hold the correct authentication tokens at the time of the communication.

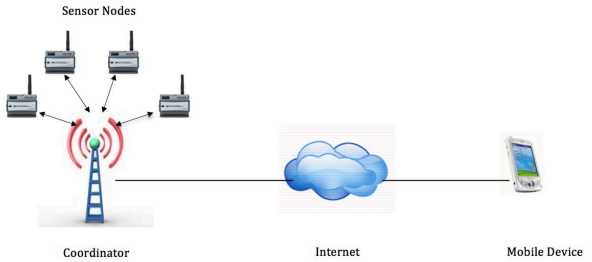

Figure 1. Network model external structure

Figure 1 captures the main design aspects of the network external structure. Such network is ideal in a medical or industry setting where the proximity of the sensor nodes makes it almost impossible for physical capture of sensor nodes to occur. Moreover, given coordinators are administered by network administrators who assign users such as physicians, nurses, quality control workers, etc. access to the WSN, we can make sure that only authorized personnel can access the network.

IV. THE PROPOSED *MOHC* AUTHENTICATION SCHEME

In this section, we introduce the formal description of our mini OHC (MOHC) authentication scheme. There are three components of the proposed solution: the user's mobile device, the coordinator CO, and the sensor nodes SN. The user's mobile device can be configured by the network administrator and is equipped with network access capabilities in the form of a username and a password. The user is anyone who is allowed to have access to the data (e.g. in the case of WSN used in the medical field, it could be the physician, the patient or the nurse etc.).

The network communication scenario consists of the following steps:

1) The network administrator provides WSN users with temporary login credentials that are changed by the respective users after initial login. This step ensures that only authorized personnel are allowed to have access to the WSN. By changing the password, we make sure that administrators' role is just to establish service.
2) Once logged in, the user communicates with the coordinator by sending a query asking for data from the sensor nodes.
3) The coordinator sends requests to the respective sensor nodes.
4) Once gathered, the sensor node(s) delivers the data to the coordinator who checks that each replying sensor has the correct credentials, accepting data if this condition holds, otherwise, data will be denied.
5) The coordinator replies to the user with the requested data.
6) Steps 2-5 are repeated for each query generated by the user in the current user session.

With this communication scenario in mind, we designed the protocol to achieve user and broadcast authentication throughout the WSN session. Therefore, the protocol works in two directions: the user direction and the SNs direction. In the user direction, once successfully logged in to the network, login credentials will be substituted by a secret value (*Secret*), which will be generated by the coordinator. This *Secret* automatically expires once the user logs out. Each time the user logs in, there will be a new generated *Secret*. This secret will be used to generate *authentication tokens* used to secure subsequent communication. The first authentication token V_1 will be the result of applying a hash function *h(.)* to the *Secret*||*C,* where *C* is the current number of communications between CO and the user; initially C=0. Thus, the first authentication token will have the value $V_1 := H^K(Secret||\ C)$, where *K,* the length of the hash chain, is decremented after each iteration. Before each communication between the coordinator and the user, the C index is incremented. In the subsequent transactions, and to avoid exposing the *Secret* and potentially compromising the WSN, we replace the *Secret* in the hash function with the preceding authentication token used in the previous communication. Thus, the second authentication token *V* has the value $V_2 := H^{K-1}(V_1||\ C+1)$, then V_1 is disposed, and V_2 is used instead to generate the third authentication token which is $V_3 := H^{K-2}(V_2||\ C+2)$ and so on. The length of the chain *K* used in our implementation is 10. In [14], we proposed a scheme where one-way hash chains are divided into smaller hash chains. The best length of a chain striking a good balance in terms of hash count falls between 10 and 25. Each time the chain length is exhausted, the index *K* is reset.

In the sensor nodes direction, the protocol works a little differently. It actually works in phases: the first phase occurs during deployment when sensor nodes identify themselves to the coordinator. The coordinator subsequently distributes $Seed_0$ among the sensor nodes which will be used in a hash function at the sensor node's side. The value of $Seed_0$ is never exposed, but used in a way similar to what we did with the *Secret* for the user authentication; i.e., as input for a hash function *h(.)* to generate subsequent *authentication tokens*. The second phase is when the actual communication begins when the coordinator relays a user's query to SNs. In the first round of communication, we use $Seed_0$ to generate the first *authentication token* at the sensor node's side $V_1 := H^K(Seed_0||\ C_i)$, where *K* is the hash chain length, decremented after each round, and C_i is the number of communication rounds between sensor_node [*i*] and the coordinator; initially $C_i = 0$. The value of C_i is incremented before each round of communication. When C_i reaches a certain value, *C_limit,* we reset its value to 0 to avoid large numbers, and also update and broadcast $Seed_0$ to sensor_node [*i*]. In the subsequent rounds, and to avoid exposing of $Seed_0$ and potentially

compromising the WSN, we replace $Seed_0$ in the hash function with the preceding authentication token. Thus, the second authentication token will be $V_2 := H^{K-1}(V_1 || C_i + 1)$, then, V_1 is discarded. The third authentication token will be $V_3 := H^{K-2}(V_2 || C_i + 2)$, V_2 is discarded, and so on. Once the hash chain length is exhausted, the index K is reset. Figure 2 presents a high level pseudo code of the authentication token scheme described in this section.

Authentication token

Phase 1: Deployment Phase

Input: number of sensor nodes

Coordinator creates $Seed_0$

Attach $Seed_0$ to each Sensor node

Set number of communications C_i between Sensor node [i] and coordinator to 0

Phase 2: Operational Phase

Input: sensor_node [i] or coordinator

Increment C_i

If Authentication token in sensor_node [i] or coordinator is the first authentication

$K := 10$

Create 1st token $V := H^K(Seed_0 \| C_i)$

Else

$Prev\_V = V$

$V := H^K(Prev\_V \| C_i)$, End if

Decrement K

If K is equal to 0

$K := 10$; End_if

If C_i is equal to C_limit

Set C_i to 0

If input is coordinator

Create random $Seed_0$

Broadcast $Seed_0$ to sensor_node [i]

End If

End If

Return (V)

Figure 2. Authentication protocol pseudo code

The above description is for the protocol when utilized in lenient environments. We also developed the code with the potential of being deployed in harsh environments where more stringent authentication is desired. We achieve this functionality by determining and presetting an expiration to the *Seed*. The expiration could be as simple as a period of time, or after a certain number of rounds of communication between the coordinator and sensor node. Once the *Seed* is updated, the communication count has to be reset to 0.

V. SIMULATION AND EVALUATION

In order to evaluate the performance of our proposed scheme, we designed a testbed where the network model described in section III is depicted. The whole network structure has been translated in a Java code and run without the authentication protocol. The performance of the simulated WSN is measured and reported. To test how our scheme impacts the simulated WSN network, we also wrote a Java code implementing the authentication protocol described in section IV. As a benchmark, we also simulated the authentication scheme if a straightforward OHC was used. A comparison between these scenarios is made and will be presented in the next section.

In our simulation, the design consisted of two configurations: serial and parallel communication. One possible example where the serial mode can be found is in industry/manufacturing quality control where a product stays on the assembly line for a period of time and moves from one point to another for different phases of processing. In each phase, there is a sensor to measure the product temperature and report it to the coordinator. For that purpose, a set of serial transactions is useful. In this configuration, the coordinator sends a request message to the first sensor node, and receives a response message from it. Then, the coordinator sends a request message to the second sensor node, and receives a response message from it, and so on until all sensors reply with their data. The coordinator, then, sends the response message to the mobile device or the quality control manager. In the parallel mode, the coordinator sends a request message to all sensor nodes at the same time by creating multiple threads. After all response messages are received, it sends a response message to the mobile device. An example where parallel configuration is ideal is a medical setting where a physician, for instance, needs to monitor a patient's electrocardiogram (EKG) by placing the sensor nodes around the body. Thus, multiple threads are initiated between the coordinator and sensor nodes.

To measure the effectiveness of our proposed WSN authentication scheme, we used the following performance metrics in our simulations:

- Time delay: the time was measured based on rounds of communication between the coordinator and N sensor nodes. Figure 3 shows one round of communication for N sensors in the serial mode. For example a round of communication between the coordinator and sensor node 1 is the difference between the time that coordinator sends a

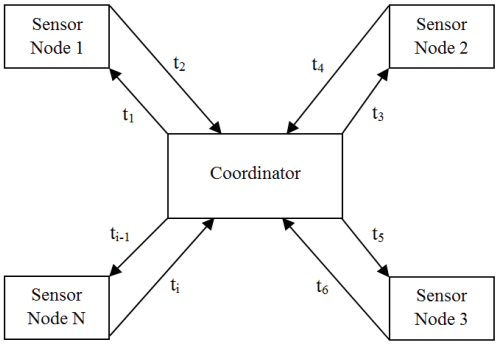

Figure 3. Example of time calculation for serial mode.

request message to sensor node (t_1) and receives a response from the sensor node t_2. We measured the WSN performance, in the serial and parallel modes, once without the proposed authentication scheme and another time when the authentication scheme is plugged either in the OHC or our MOHC format.

- Average number of hashes: the average number of hash operations performed during a round of communication. Given the fact that each component of the WSN (coordinator and sensor nodes) is going to calculate the same number of hashes in each round of communication, we used an average number of hashes for each of these components and presented comparisons based on them.
- Energy consumption per sensor node: energy consumed is measured using the cost of one SHA-1 operation described in [15] where different energy consumption characteristics of various cryptographic approaches are investigated. According to [15], one SHA-1 operation consumes 0.76 microjoule/byte of energy.

VI. PERFORMANCE AND RESUTLS

In this section, we present the performance evaluation results of our protocol. Let us first look at how the MOHC protocol performed when compared with the case of no authentication at all, and with the straightforward OHC. Table 1 illustrates the difference in milliseconds between these three cases in the serial mode. Here, we compared the time in the case of no authentication, straight forward OHC and our protocol based on the number of rounds of communication between the coordinator and the sensor nodes. In this case, the coordinator sends a query to the first sensor node and waits for its response before sending the next request to the second sensor node. The results indicate that adding authentication increases the time. This increase tends to correlate linearly with the duration of the WSN session which is not unexpected. However, our focus was to compare our idea of mini hash chains on the performance of WSN with the straightforward implementation of OHC. When the number of rounds of communication is small, as in the case of 50 rounds, it can be seen that the difference between OHC and MOHC protocol is approximately 2 milliseconds. However, this difference becomes more salient with higher rounds of communication numbers. In real WSN traffic, communication is expected to be higher, and, therefore, rounds of communication number increases, which makes our solutions an obvious better choice as it outperforms its counterpart by a margin.

Table 1. Time comparisons showing equal execution overheads for MHOC and OHC (Serial Mode)

Rounds of communication	No-Authentication*	OHC*	MOHC*
50	1192	1201	1199
150	2513	2531	2526
300	5011	5047	5029
600	9767	9874	9793

* In milliseconds

Similarly, Table 2 summarizes the performance comparison in the parallel mode. As described above, the coordinator initiates multiple threads to send a request message to all sensor nodes. The response messages are received in no particular order from the sensor nodes. Here, we can see the time required to run the simulation without authentication is much less. However, the time required for authentication, either in the serial or parallel mode, is approximately the same. What is important for our comparison is the time difference between OHC and our MOHC. The difference in time shows our protocol still outperforms the OHC as the table shows.

Table 2. Time comparisons showing equal execution overheads for MHOC and OHC (Parallel Mode)

Rounds of communication	No-Authentication*	OHC*	MOHC*
50	297	306	304
150	939	957	951
300	1545	1581	1563
600	3104	3211	3131

* In milliseconds

In order to further demonstrate the improvement we achieved in our protocol over OHC, we measured the time difference between the performance of the two schemes and present it in Figure 4. This figure represents the performance in the serial mode.

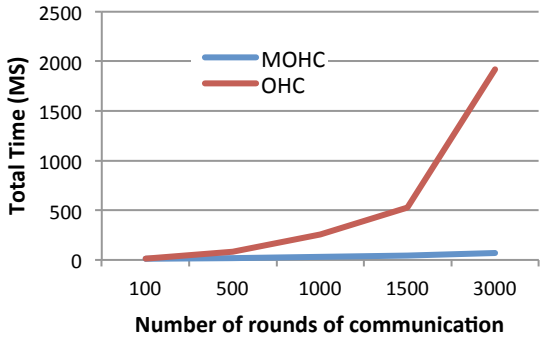

Figure 4. Time comparisons between OHC and MOHC protocol with higher rounds of communication (Serial Mode).

As can be seen Figure 4, the higher the number of rounds of communication, the wider the difference margin is. This is another indication that using our MOHC protocol definitely improves performance and makes deploying a one-way hash chain based protocol more attractive in WSN.

When the average number of hashes required for a round of communication is compared as in Figure 5, we still can see how our MOHC scheme outperforms the traditional OHC. Similar to the time delay compared above, as the number of rounds of communication increases, so does the average hash count. This is indicative of the high computation cost of the one-way hash chain schemes if configured as is. With minor modifications, like the ones we have proposed in this paper, we were able to achieve noticeable improvements. It should

be noted though that regardless of configuration options, the number of hashes is going to be the same. Each component of the WSN will calculate the same number of hashes.

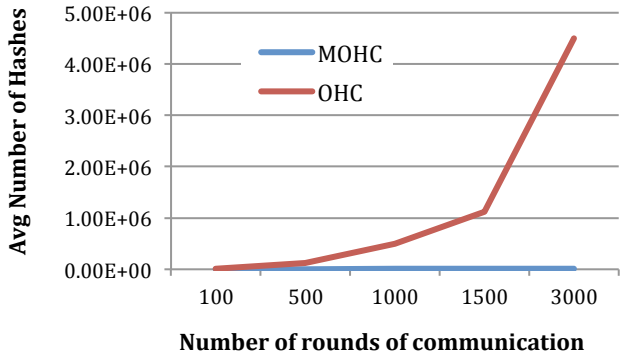

Figure 5. Average number of hashes comparison between OHC and MOHC

This leads us to the final finding in Figure 6. In this figure, we illustrate the energy consumption differences per sensor node between the OHC and our MOHC protocol. Our MOHC consumes much less energy in the long run. It is known that sensor nodes are typically not equipped with changeable batteries. In this case, our protocol drains energy at a much lower rate compared to OHC.

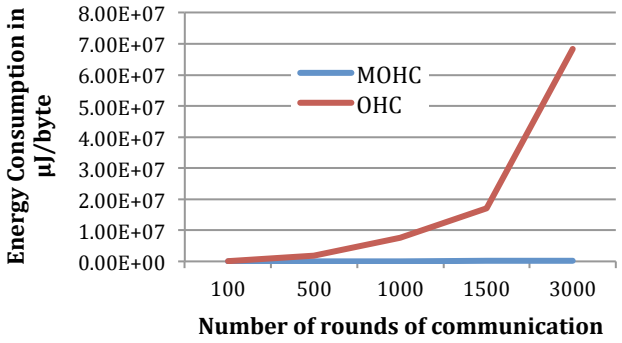

Figure 6. Energy consumption comparison between MOHC and OHC

VII. CONCLUSION

In this paper, we have presented a lightweight mini one-way hash chain based solution for WSN user and broadcast authentication. Our objective of the proposed solution was to mitigate the high computational cost of the straightforwardly configured one-way hash chain and avoid exposing the initial secret. We have shown through simulation and evaluative experiments that with economic modifications to one-way hash chains, we were able to achieve lower computational overhead measured by computation time and number of hash operations required to provide protection. Additionally, by substituting the initial secret with easily computed authentication tokens, we were able to maintain the initial secret less exposed. We have also shown that our protocol is attractive form an energy consumption perspective as it lowers consumption of sensor nodes' battery lives. In the future, we plan to improve our solution and apply it in alternative WSN topologies.

References

[1] Z. Benenson, N. Gedicke, and O. Raivio, "Realizing robust user authentication in sensor networks," *Real-World Wireless Sensor Networks (REALWSN),* vol. 14, 2005.

[2] I. Butun and R. Sankar, "Advanced two tier user authentication scheme for heterogeneous wireless sensor networks," in *Consumer Communications and Networking Conference (CCNC), 2011 IEEE*, 2011, pp. 169-171.

[3] R. Wang, W. Du, X. Liu, and P. Ning, "ShortPK: A short-term public key scheme for broadcast authentication in sensor networks," *ACM Transactions on Sensor Networks (TOSN),* vol. 6, p. 9, 2009.

[4] K. Arikumar and K. Thirumoorthy, "Improved user authentication in wireless sensor networks," in *Emerging Trends in Electrical and Computer Technology (ICETECT), 2011 International Conference on*, 2011, pp. 1010-1015.

[5] O. Cheikhrouhou, A. Koubaa, M. Boujelben, and M. Abid, "A lightweight user authentication scheme for Wireless Sensor Networks," in *Computer Systems and Applications (AICCSA), 2010 IEEE/ACS International Conference on*, 2010, pp. 1-7.

[6] Y. Liu, J. Li, and M. Guizani, "PKC Based Broadcast Authentication using Signature Amortization for WSNs," *Wireless Communications, IEEE Transactions on,* vol. 11, pp. 2106-2115, 2012.

[7] A. Perrig, R. Szewczyk, J. Tygar, V. Wen, and D. E. Culler, "SPINS: Security protocols for sensor networks," *Wireless networks,* vol. 8, pp. 521-534, 2002.

[8] D. Liu and P. Ning, "Efficient Distribution of Key Chain Commitments for Broadcast Authentication in Distributed Sensor Networks," in *NDSS*, 2003.

[9] M. L. Das, "Two-factor user authentication in wireless sensor networks," *Wireless Communications, IEEE Transactions on,* vol. 8, pp. 1086-1090, 2009.

[10] M. K. Khan and K. Alghathbar, "Cryptanalysis and security improvements of 'two-factor user authentication in wireless sensor networks'," *Sensors,* vol. 10, pp. 2450-2459, 2010.

[11] B. Vaidya, D. Makrakis, and H. T. Mouftah, "Improved two-factor user authentication in wireless sensor networks," in *Wireless and Mobile Computing, Networking and Communications (WiMob), 2010 IEEE 6th International Conference on*, 2010, pp. 600-606.

[12] K. H. Wong, Y. Zheng, J. Cao, and S. Wang, "A dynamic user authentication scheme for wireless sensor networks," in *Sensor Networks, Ubiquitous, and Trustworthy Computing, 2006. IEEE International Conference on*, 2006, p. 8 pp.

[13] I. Dacosta, S. Chakradeo, M. Ahamad, and P. Traynor, "One-time cookies: Preventing session hijacking attacks with disposable credentials," *ACM Transactions on Internet Technology (TOIT),* vol. 12, 2012.

[14] A. Alabrah and M. Bassiouni, "A hierarchical two-tier one-way hash chain protocol for secure internet transactions," in *Global Communications Conference (GLOBECOM), 2012 IEEE*, 2012, pp. 868-873.

[15] N. R. Potlapally, S. Ravi, A. Raghunathan, and N. K. Jha, "Analyzing the energy consumption of security protocols," in *Proceedings of the 2003 international symposium on Low power electronics and design*, 2003, pp. 30-35.

[16] M. Li, S. Yu, J. D. Guttman, W. Lou, and K. Ren, "Secure ad hoc trust initialization and key management in wireless body area networks," *ACM Transactions on Sensor Networks (TOSN),* vol. 9, p. 18, 2013.

[17] A. Alabrah, J. Cashion and M. Bassiouni "Enhancing security of cookie-based sessions in mobile networks using sparse caching" International Journal of Information Security- Springer Publishing, online version published December 2013, DOI: 10.1007/s10207-013-0223-8 (12 pages).