Journal of Information Security, **2012, 3, 272-280**
http://dx.doi.org/10.4236/jis.2012.34034 Published Online October 2012 (http://www.SciRP.org/journal/jis)

C3SM: Information Assurance Based on Cryptographic Checksum with Clustering Security Management Protocol

Moad Mowafi[1], Lo'ai Tawalbeh[2], Walid Aljoby[1], Mohammad Al-Rousan[1]
[1]Department of Network Engineering and Security, Jordan University of Science and Technology, Irbid, Jordan
[2]Department of Computer Engineering, Jordan University of Science and Technology, Irbid, Jordan
Email: mowafi@just.edu.jo, tawalbeh@just.edu.jo, walid_aljoby@yahoo.com, alrousan@just.edu.jo

Received March 25, 2012; revised April 26, 2012; accepted May 20, 2012

ABSTRACT

Wireless Sensor Networks (WSNs) are resource-constrained networks in which sensor nodes operate in an aggressive and uncontrolled environment and interact with sensitive data. Traffic aggregated by sensor nodes is susceptible to attacks and, due to the nature of WSNs, security mechanisms used in wired networks and other types of wireless networks are not suitable for WSNs. In this paper, we propose a mechanism to assure information security against security attacks and particularly node capturing attacks. We propose a cluster security management protocol, called Cryptographic Checksum Clustering Security Management (C3SM), to provide an efficient decentralized security management for hierarchal networks. In C3SM, every cluster selects dynamically and alternately a node as a Cluster Security Manager (CSM) which distributes a periodic shared secrete key for all nodes in the cluster. The cluster head, then, authenticates identity of the nodes and derive a unique pairwise key for each node in the cluster. C3SM provides sufficient security regardless how many nodes are compromised, and achieves high connectivity with low memory cost and low energy consumption. Compared to existing protocols, our protocol provides stronger resilience against node capture with lower key storage overhead.

Keywords: Wireless Sensor Networks; Security; Message Authentication Code; Cryptographic; Node Capture Attack

1. Introduction

Wireless Sensor Networks (WSNs) are highly distributed and self-organized system that is based on collaborative effort of a large number of nodes, where each node has the ability of sensing, computation, and communication. WSNs suffer from various malicious attacks because the environment is open to the public. Thus, an enemy can easily listen to the wireless communication and intercept the traffic. To prevent such malicious attacks, secret keys should be used to encrypt wireless communication and establish data confidentiality, integrity and authentication among sensor nodes. An enemy can also capture a sensor node and access its data and communication keys. This type of attacks is called node capture attack, and forms a main challenge to develop a security mechanism for WSNs.

In wired and wireless networks, information assurance is attained by data encryption and authentication. Many complex security algorithms are developed such as public-key cryptography (e.g., RSA [1] and Diffie-Hellman [2]), digital signature and trusted third-party authentication schemes [3]. In WSNs, the sensor node does not have sufficient memory to store a lot of keys or support a complex public key algorithm. Moreover, the computation overhead and energy consumption make traditional security mechanisms not suitable for WSNs. Therefore, it is necessary to develop an appropriate security mechanism for WSNs to distribute secret keys among the nodes, encrypt communication and form authentication. However, the challenge does not lie in the development of a secure mechanism merely, but on how to efficiently create, distribute and manage the secret keys among the nodes.

In this paper, we introduce a new security protocol called Cryptographic Checksum Clustering Security Management (C3SM) that operates under clustered hierarchical network architecture. The proposed scheme provides sufficient security regardless of how many nodes are compromised and achieves efficient energy consumption with low key storage overhead. In C3SM, every cluster selects dynamically and alternately a node called Cluster Security Manager (CSM) which distributes a periodic shared secrete key for all nodes in the cluster. Then, the cluster head (*CH*) authenticates the identity of the cluster nodes, and establishes a unique pairwise key for each node in the cluster. The authentication is achieved by cryptographic checksum or Message Authentication Code (MAC). To enhance confidentiality between the cluster nodes and the *CH*, we design a local, random,

dynamic, periodic and unique pairwise key for each path between the *CH* and sensor node. These key properties make the security in WSNs stronger, and attain high connectivity with low memory cost and low energy consumption. To enhance integrity and authenticity among nodes, we use cryptographic checksum (variable tiny segment of code) appended to control messages. Moreover, the proposed scheme has strong resilience against node capture because it has an alternating CSM that distributes keys at regular period of times in normal (safety) mode, monitors the cluster nodes for attack, and changes keys directly in case of an attack occurs (threat mode).

The rest of this paper is organized as follows. In Section 2, we provide a review of related work in key management for WSNs. In Section 3, we describe the proposed system architecture. In Section 4, we present and analyze the system model. In Section 5, we evaluate the system performance and present simulation results. Finally, we conclude the paper in Section 6.

2. Related Work

The existing approaches for solving the key distribution problem in WSNs can be classified into four categories [4]: Network-wide keys schemes, full pairwise key schemes, matrix-based schemes, polynomial-based schemes, and probabilistic schemes.

In the network-wide keys scheme, a single master key is loaded into all sensor nodes. This scheme provides perfect connectivity since all deployed nodes share the same key, and also new added nodes can be loaded with the same master key and connect simply. Several schemes have adopted this approach [5-7]. The shortcoming of the network-wide scheme is that a capturing of a single node will comprise all the nodes and their communication. Moreover, malicious nodes can be easily injected into the network.

In the full pairwise key scheme, each node from n nodes stores n-1 pairwise keys in order to communicate with every other node. This scheme provides a high level of security but its main drawback is its demand for very large memory storage.

Matrix-based schemes are originally created for establishing a pairwise key by Blom [8]. In Blom's scheme, each node i has the ith row and the ith column of secret and public matrices, respectively. By exchanging their columns, any two nodes i and j can create their pairwise key $K_{ij} = K_{ji}$. In this scheme, if no more than t nodes are compromised, no more keys are compromised. Increasing t can improve the scheme resilience; however more secret information needs to be stored. Extensions of Blom's scheme have been proposed in [9,10].

The polynomial-based key management schemes are originally initiated by Blundo [11] as a special case of Blom's scheme. In Blundo's scheme, each node i has a polynomial $f(x,y)$ over a finite filed. By evaluating their polynomials, any two nodes i and j can create their pairwise key $f(i,j) = f(j,i)$. Several schemes have adopted Blundo's scheme [12-14]. The main drawbacks of the polynomial-based approach are its demand for large memory to store the polynomials, and the computational power of the multiplication and exponentiation operations [4].

In the probabilistic approaches, the security services are divided into phases in order to offer high security as the pairwise key approach and lower storage as the network-wide key approach, and to find suitable tradeoff between security and overhead. In general, the probabilistic approaches pass through three phases [4]: Key predistribution, shared-key discovery, and path-key establishment. Our work belongs to such approaches and presents a security mechanism for each phase, aiming at attaining effective security, efficiency, and flexibility.

Several schemes, related to our work, have been proposed [15-20]. In [15], a random key pre-distribution scheme is proposed. It prepares a very large size key pool, chooses randomly a subset of keys, and then stores them in the node's memory before deployment. After the discovery process performed between the nodes that intend to communicate, the nodes can establish a connection if they share one or more of the common keys stored in their memories. The common key then becomes a shared key for the link between the nodes. Nodes that cannot find a shared key with each other can generate a path key through what so-called a connected secure graph. This scheme requires a large key storage in large scale networks. Moreover it does not support node authentication, and its resilience to node capture attacks is weak since any captured node can compromise other nodes keys.

In [16], a scheme called efficient pairwise key establishment and management (EPKEM) is proposed. In this scheme, each node stores randomly a row and column from a key matrix, and any two nodes create a distinct pairwise key by combining their common keys and node identities. If a node is compromised, the communication between non-compromised nodes remains secure. However, this scheme has high communication overhead in large scale networks, needs large key storage, and consumes energy when adding nodes.

A scheme for large-scale hierarchical WSNs is presented in [17]. It uses polynomial key calculations to create a distinct pairwise key between any two nodes. This approach assumes three phases for key management: key pre-distribution, inter-cluster pairwise key establishment, and intra-cluster pairwise key establishment. The scheme shows good security mechanism against node capture attacks, but the establishment of one pairwise key for each node needs the cluster head to communicate

with other cluster heads to authenticate node connectivity.

In [18], a rekey-boosted security protocol in hierarchical WSNs is proposed. In this approach, clusters are formed based on LEACH, and random key pre-distribution is used to establish node-to-node security and authentication. A cluster key is used to secure the cluster head-to-node communication, and a key created by the cluster head is used to protect the cluster head-to-base station communication. In this scheme, the cluster head carries much overhead because it is used for both routing and security.

In [19], the proposed scheme stores a master key and random vector in each node, and any two nodes create a pairwise key by combining their random vector with the stored master key. In addition, each node stores a cluster key to communicate with the cluster head. The cluster key is derived from the preloaded master key and identification number of the cluster head. Hence, an enemy that knows the master key and the identification number of the cluster head can extract the cluster key and easily attack the cluster.

A protocol for securing the paths among the nodes in WSNs is proposed in [20]. In this protocol, the network area is partitioned into a virtual grid with identical cells. It is assumed that each node stores four keys: individual key to communicate with the base station, cell key to communicate with nodes inside the cell, eight pairwise keys to communicate with other cells, and broadcast keys. By capturing any of the cell nodes, the adversary can extract the key and then easily attack all the nodes inside the cell.

3. System Architecture

The system is organized in a multi-tier architecture according to the recourses and functionality. The resources variability divides the system into two-tier architecture. One tier represents the base station (BS) and another tier represents the deployed sensor nodes. The BS is assumed to have no computational, storage and communication limitations and is located far from the sensor field. The sensor nodes are assumed to be resource-constrained in energy, processing, and storage. The sensor nodes are homogenous, have the same resources, and start with the same level of energy. The sensors nodes are capable of control their power to vary their functionality.

From the functionality viewpoint, the system is divided into four-tier architecture as shown in **Figure 1**. The first tier represents the BS that will be considered as a powerful data processing unit that performs heavily operations, a storage center that collects data, and a key distribution center before deployment. The BS is assumed to be trusted and temper resistant. The second tier represents the *CH* that will be considered as a collector for sensed data from other members in the cluster, an aggregator for the collected data, and a sender for the fused data to the BS in a single-hop path, as depicted in **Figure 2**. The third tier represents the sensor nodes that sense data and report the target field states to the *CH* as depicted in **Figure 2**. The fourth tier, the lower layer of our proposed stack model, represents the contribution of what we target in this research that is the clustering security management layer. This layer appears dynamically in each cluster by targeting one of the sensor nodes other than the *CH*s that is the CSM as depicted in **Figure 2**. The CSM will manage the cluster security because it works as a key distribution center and as a guard for cluster sensor nodes against an adversary. The CSM periodically constructs a key and distributes it to its cluster members and to the *CH*.

4. System Model and Analysis

The C3SM scheme consists of two parts: The first part deals with key distribution and managing methodologies while the second one deals with network monitoring and resilience against the node capture attack and its implications.

4.1. Key Distribution and Managing Model

Before deploying the nodes in the target field, each sensor node will be assigned two types of key. One key to

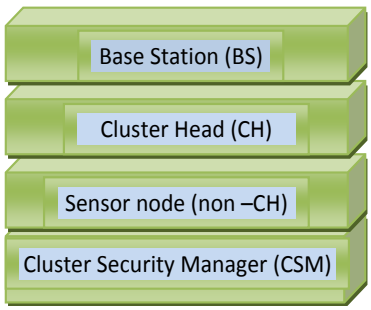

Figure 1. Four-tier clustering security model for WSNs.

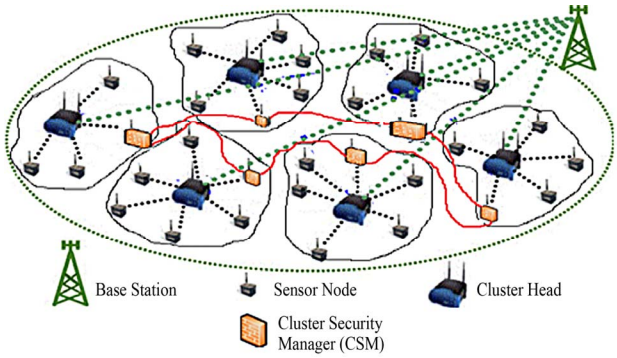

Figure 2. A clustering security architecture for WSNs.

encrypt and authenticate aggregated sensing data from a *CH* to the BS. And the other one used for a period of time after the deployment to encrypt and authenticate exchanged data between the *CH* and its sensor node members. After the deployment of sensor nodes and formation of clusters, each cluster will candidate one node to become a CSM that will protect the cluster against an adversary attacks, and distribute (re-keying) keys for the nodes in the cluster including the *CH*s. This procedure will constitute a distributed cluster among all clusters in the network with cluster members called CSMs. The key management scheme consists of two phases: key setup phase and path key establishment phase.

4.1.1. Key Setup Phase

When the nodes are deployed, each node is preloaded with an initial shared key (K_p) which assumed to be a large number symmetric key assigned for all nodes in the network. The preloaded key can be used by any sensor node to generate its master key as a function of the node *ID*. An authenticator MAC function (C-function) is used to generate the keys. For example, node *i* uses K_p and its *ID* to generate its master key (K_i) as follows:

$$K_i = C\left(K_p, ID_i\right) \tag{1}$$

The symmetric key K_p is just used after the sensor nodes deployment for a short period of time between the *CH* and cluster members to create a master key. This key is changed periodically by the CSM, thus, any attack to any node in the cluster does not affect its security.

After formation of clusters, the *CH* will broadcast its *ID* (ID_{CH}) and an advertisement message (Adv) encrypted by the preloaded key K_p to the cluster members. It is assumed that every node can know the Adv message.

$$CH \rightarrow N: \quad ID_{CH} \| C\left(K_p, \text{Adv} \| ID_{CH}\right) \tag{2}$$

After the above formatted message is broadcasted, each authorized sensor node receives the message and performs the same function, $C\left(K_p, \text{Adv} \| ID_{CH}\right)$, as the *CH* and compares with the received MAC, as depicted in **Figure 3**. Then, each member of the cluster will respond with its *ID* (ID_N), and a message contains a join signal (Ack), the *CH*'s Adv and the node *ID* (ID_N), encrypted by its preloaded key K_p, as shown in **Figure 4**.

$$N \rightarrow CH: \quad ID_N \| C\left(K_p, \text{Adv} \| ID_N \| \text{Ack}\right) \tag{3}$$

When the *CH* receives a reply from its cluster members, all of them can generate a secure master key as shown by the following equations:

$$K_{CH} = C\left(K_p, ID_{CH}\right) \tag{4}$$

$$K_N = C\left(K_p, ID_N\right) \tag{5}$$

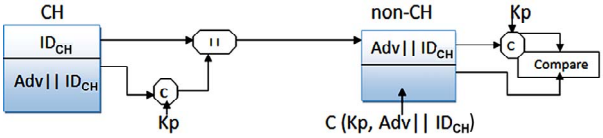

Figure 3. *CH*-to-node message authentication.

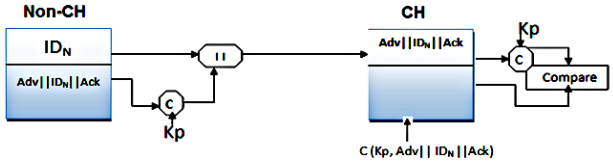

Figure 4. Node-to-*CH* message authentication.

Equation (4) represents the master key of the *CH*, and Equation (5) represents the master key of any member node *N*. After each node in the cluster generates its master key, the cluster will translate into next phase, which is called the path key establishment.

4.1.2. Path Key Establishment Phase

In this phase the pairwise key K_{CH-N} will be established between the *CH* and each cluster member node *N*. The pairwise key maintains a unique key for a path between the *CH* and each node in the cluster. Hence, it provides a sufficient security against node capture attacks since any compromised node will not affect the secure communication among non-compromised nodes. Moreover, this approach does not require a large storage for each node to store the whole pairwise keys in the network because the *CH* node just stores the pairwise keys of its cluster members. The pairwise key is derived as follows:

$$K_{CH-N} = C\left(K_N, ID_{CH}\right) \tag{6}$$

This technique can alleviate the tradeoff between the pairwise key supported security and the key storage overhead.

4.1.3. C-Function

The C-function is a cryptographic checksum function that is usually called message authentication code or MAC. However, the domain of C-function consists of a message of some arbitrary length, whereas the range consists of all possible MACs and all possible keys [21]. There are three types of MACs in our proposed system, as shown in Equations (4)-(6). The left hand side of each equation represents the generated fixed length authenticator that can be exploited to perform the following functionalities: authentication to assure that received messages are from alleged nodes, confidentiality to protect the traffic as long as the generated authenticator used as a unique pairwise key between the *CH* and its cluster members, and resiliency against node capturing because the key for each path is unique and is managed periodically by the CSM. On the other hand, the right hand

side of the equation represents the variable length message which is the ID of the node and either the secret shared key between all nodes in the cluster such as K_p or the unique key between the *CH* and its cluster members such as K_N.

4.2. Clustering Security Management

After completing the formation phase for each cluster in the network, the role of security is triggered in order to protect the network against malicious attacks. The security of the network is managed by distributed nodes throughout the network, forming a security cluster.

The security cluster is a distributed cluster through all the data clusters in the network, as shown in **Figure 2**. In the safe mode, the construction of the cluster is assumed to take place at the beginning of the second half of the current data cluster cycle and remain to the ending of the first half of the next data cluster cycle.

The *CH* in each data cluster can candidate one of its cluster members to be a CSM which carries out three tasks. First, the first CSM creates a schedule in which order the cluster member nodes are elected as CSM and repeats the same process after the member nodes in the cluster are already pass the turn. The CSM checks its energy and if it is less than a threshold, the CSM will broadcast a release message to its cluster nodes. The node in schedule will take the turn and become a CSM. By this property the CSM guarantees the fairness in energy consumption among the cluster nodes. Second, the CSM can work as a key distribution center to construct and distribute periodically a shared master key for each node inside the cluster and also to the *CH* in order to re-keying them. By the master key, all the nodes in the cluster can use this key as a secrete key for establishing a new pairwise key (re-keying) between the *CH* and each sensor node in the cluster. Third, the CSM carries out monitoring and controlling the cluster member nodes against any attack. The CSM will exchange periodic messages with the cluster member nodes, and if one of the nodes does not reply, the CSM assumes an adversary captures the node, and then it will change all the keys in the cluster. In case that any CSM are captured, the nodes can tell through the disappearance of the control message sent by the captured CSM, and consequently the turn for the next appointed CSM arises to work for a period of time equals to the time of the data cluster. **Figure 5** summarizes the C3SM algorithm.

4.3. Malicious Attack and Threat Model

The malicious attacks can be divided into passive and active modes. In the passive mode, the enemy listens to the communication among the nodes to seize private data, while in the active mode it captures nodes in the network. When a node is compromised, the stored secret keys or information are revealed and, hence, false messages can be injected or the transmitted messages can be modified or dropped.

Next, we analyze the system security under node capture attacks, considering three types of sensor nodes: a cluster member, the *CH*, and the CSM.

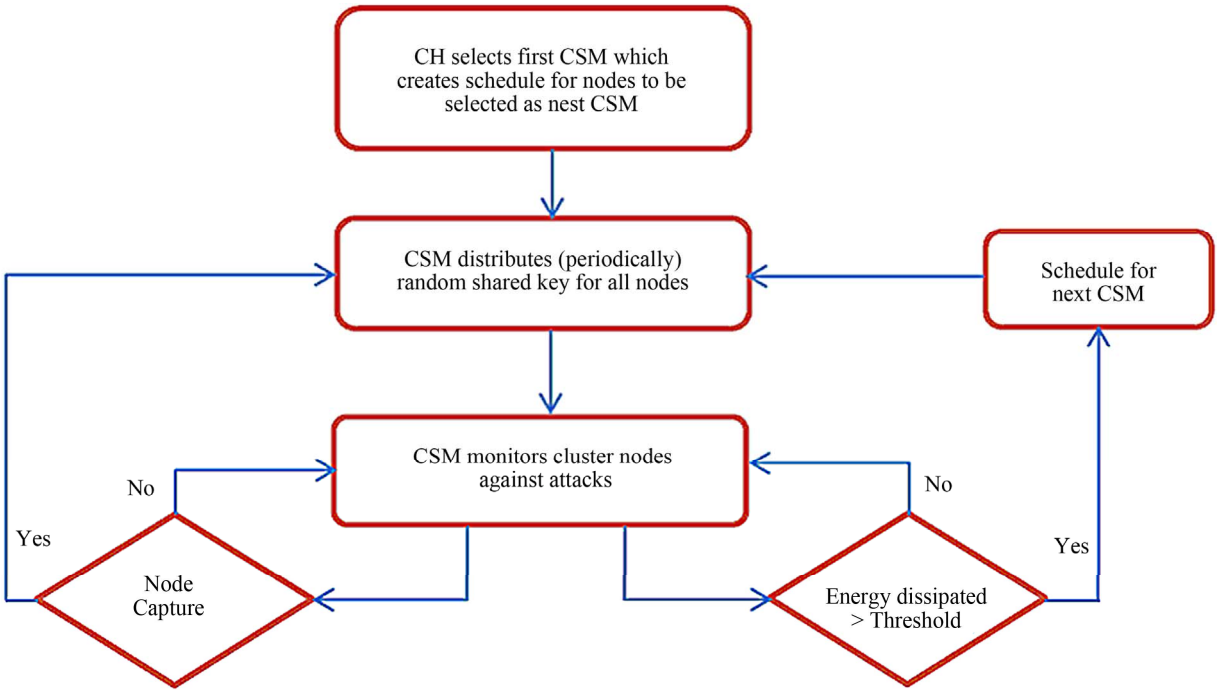

Figure 5. C3SM algorithm.

4.3.1. Sensor Node Capture Attack

In our scheme, after nodes deployment and cluster formations, in a short period of time, each *CH* will establish a unique pairwise key (K_{CH-N}) for each link with a cluster member node. In addition, after the CSM distributes a shared secrete key for the nodes in each cluster, the *CH* can also establish a unique pairwise key (K_{CH-N}) for each link with a cluster member node, and so forth. Thus, if adversaries deploy their own malicious nodes, these malicious nodes cannot be connected to the cluster because the communication with the *CH* requests from the node to know its master key (K_N) which is a cryptographic checksum or MAC from the node ID and the shared key K_p as shown in Equation (5). In case that a sensor node is physically captured, the adversary can read the contents of the node memory and discover its pairwise key with the *CH*, however it cannot compromise other non-captured nodes because the pairwise key is unique for each pair of two communicating parties. On the other hand, the CSM will lose the communication with the captured node, and distributes a new master key for the attacked cluster.

4.3.2. Cluster Head Capture Attack

In our scheme, the *CH* stores all the pairwise keys of the member nodes in the cluster. The CSM monitors the nodes in the cluster and exchanges periodic control messages with them. So, if the *CH* is captured, the CSM will detect the capture and broadcast messages to all the nodes in the cluster to set up a new round, and candidate a new *CH*. The CSM also distributes a new shared key for all the nodes in the cluster. Then, the new *CH* will use the shared key to establish a unique pairwise key with each node in the cluster. So, the adversary cannot compromise any node in the cluster because all the cluster node keys are changed and the communication is also changed to a new *CH*.

4.3.3. CSM Capture Attack

In our scheme, each elected CSM in a cluster can create a schedule to determine when each node in the cluster is elected as a CSM. In case that the CSM is captured, all nodes in the cluster lose communication with the CSM. After a short period of time the node in responsible in the schedule will take the turn to become a CSM.

5. Performance Evaluation

Security algorithms for WSNs include a tradeoff between the security level and resources consumption. In this section, we evaluate by simulation the performance of our proposed scheme, and compare it with current schemes in [15,16]. The simulation was performed by a selfdeveloped simulator using the simulation settings as follows. A deployment region of 100 × 100 m^2 is considered. The frequency of key refreshment is 5 time units, and the frequency of control messages is 1 time unit. The control message size is 50 bytes.

5.1. Communication and Computation Overhead

The communication per bit in WSNs is more costly than computation [22]. In C3SM, communication is performed inside a cluster, which means there is no longer transmission. In addition, the computational operations of key management are simple and performed locally (inside the cluster). A network of 10 clusters with 12 nodes per cluster was used. The Friis free-space model was used to estimate the communication energy consumption [23]. **Figures 6** and **7** show the energy consumed by the CSM, *CH*, and data sensor node (non-*CH*) of a randomly chosen cluster, for performing key management operations: key setup, re-keying and nodes monitoring. **Figure 6** shows the accumulated dissipated energy after completing 100 rounds, while **Figure 7** shows the dissipated energy during one round. As shown in both figures, the CSM consumes more energy compared to *CH* and non-*CH* nodes because it performs monitoring and re-keying tasks frequently while the *CH* only performs key setup and authentication with its cluster nodes.

Figures 8 and **9** show the consumed energy for each phase of key management: Setup, re-keying, and monitoring which are performed at randomly chosen cluster. **Figure 8** shows the accumulated dissipated energy during each phase at a random cluster after completing 100 rounds, while **Figure 9** shows the energy dissipated for each phase during one round. As shown in both figures, the monitoring task consumes approximately four times energy compared to re-keying task because in the monitoring task the CSM exchanges periodic messages with the cluster nodes in order to protect the network against node capture attack. On the other hand, the re-keying task consumes energy only when the CSM discovers an attack or its dissipated energy reaches a threshold. The energy consumed by the setup phase is mostly done when the *CH* performs authentication with its cluster nodes.

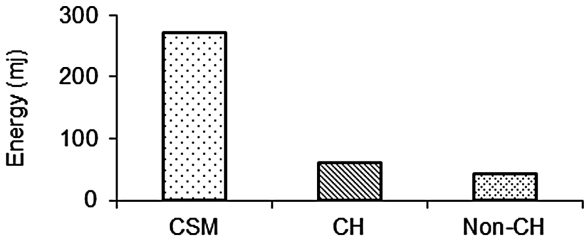

Figure 6. Consumed energy by cluster nodes of a randomly chosen cluster.

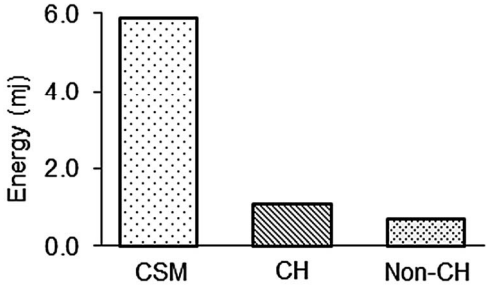

Figure 7. Consumed energy by cluster nodes of a random cluster during one round.

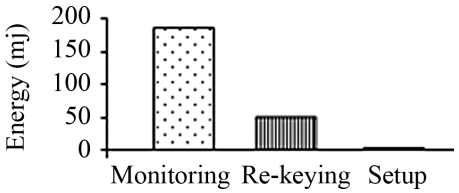

Figure 8. Consumed energy for key management phases at a randomly chosen cluster.

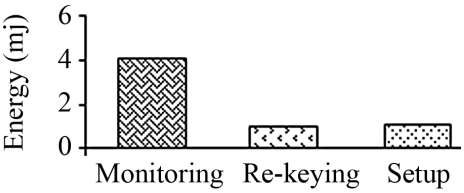

Figure 9. Consumed energy for key management phases at a random cluster during one round.

5.2. Resiliency to Node Capture

Because the resources of a sensor node are very limited, complexity of computations or long term transmission affects the lifetime of the network. Many approaches try to solve the problem of node capture attack. However, most of them still suffer from overhead or compromising nodes attack. In the random key pre-distribution scheme in [15], the same keys are used by different nodes and, hence, if a node is captured, the secure communication among other nodes is compromised. In EPKEM [16] and our proposed C3SM, pairwise keys are stored in every node and, hence, the resilience against node capture is improved. C3SM prevents key compromise for non-compromised nodes, even if many of the sensor nodes are captured. **Figure 10** shows the network resilience against node capture attacks for our C3SM scheme in addition to the random key pre-distribution scheme in [15] and the EPKEM scheme in [16]. It is shown that in the random key pre-distribution scheme, the fraction of compromised keys in non-captured nodes increases as the number of captured nodes increases, while in EPKEM and C3SM it remains at low fraction regardless how many nodes are captured. In C3SM, each sensor node receives periodically a shared key from the CSM, and then the *CH* uses this key to establish a pairwise key with each node in the cluster. Pairwise keys are different for each path and cannot easily be derived because the MAC used is many-to-one function. Namely, there are many keys to produce the correct MAC; consequently the opponent has no way to know the correct key. Furthermore, keys are refreshed periodically.

5.3. Key Storage Overhead

In random key distribution, to achieve the required network connectivity, each sensor node is required to store a certain number of keys in its memory. **Figure 11** shows the number of keys stored in each node versus the network size for the three schemes: C3SM, EPKEM, and the random key pre-distribution scheme. It is shown that the number of keys per node increases linearly in the random key pre-distribution scheme, and increases sub-linearly in

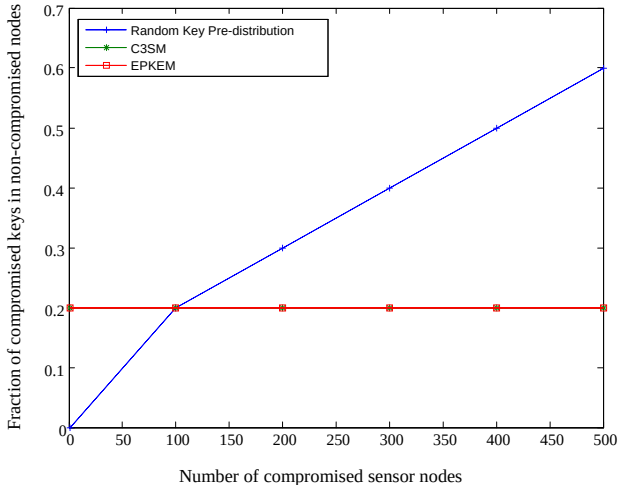

Figure 10. Network resilience against node capture attacks.

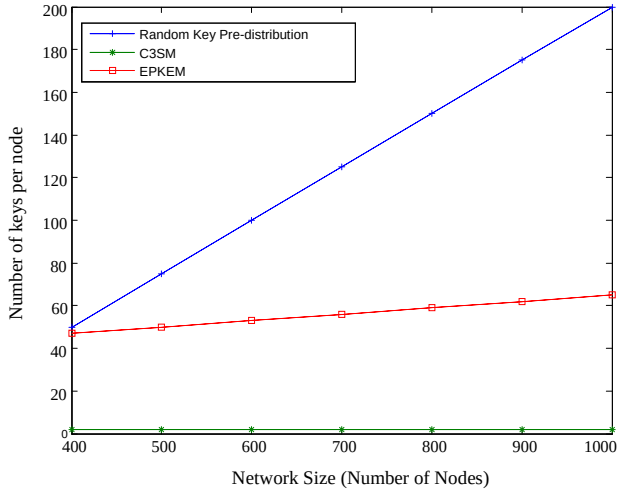

Figure 11. Key storage overhead versus network size.

EPKEM. On the other hand, C3SM has the lowest key storage overhead. In C3SM, each node only needs to store a pairwise key with the *CH* and a key with the BS in its memory no matter how many nodes are in the network.

6. Conclusions

In this work, we propose a cluster-based security protocol for WSNs, called Cryptographic Checksum Clustering Security Management (C3SM). Our protocol uses cryptographic checksum to authenticate communication among nodes. In C3SM, each node only stores two keys despite the network size, which reduces the key storage overhead especially in large scale networks.

To enhance confidentiality between the cluster nodes and the cluster head, we use a local, random, periodic and unique pairwise key for each path between the cluster head and the sensor node. These key properties make the network security stronger while achieving high connectivity with low memory cost and low energy consumption. Compared to existing schemes, C3SM achieves better network resilience against node capture attacks with lower key storage overhead.

7. Acknowledgements

The authors would like to thank Jordan University of Science and Technology, and the Scientific Research Support Fund at the Ministry of High Education in Jordan for supporting this research.

REFERENCES

[1] R. L. Rivest, A. Shamir and L. M. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Communications of the ACM*, Vol. 21, No. 2, 1978, pp. 120-126. doi:10.1145/359340.359342

[2] W. Diffie and M. E. Hellman, "New Directions in Cryptography," *IEEE Transactions on Information Theory*, Vol. 22, No. 6, 1976, pp. 644-654. doi:10.1109/TIT.1976.1055638

[3] Y. Jararweh, L. Tawalbeh, H. Tawalbeh and A. Moh'd, "Hardware Performance Evaluation of SHA-3 Candidate Algorithms," *Journal of Information Security*, Vol. 3, No. 2, 2012, pp. 69-76. doi:10.4236/jis.2012.32008

[4] M. A. Simplício Jr., P. S. Barreto, C. B. Margi and T. C. Carvalho, "A Survey on Key Management Mechanisms for Distributed Wireless Sensor Networks," *Computer Networks*, Vol. 54, No. 15, 2010, pp. 2591-2612. doi:10.1016/j.comnet.2010.04.010

[5] B. Lai, S. Kim and I. Verbauwhede, "Scalable Session Key Construction Protocol for Wireless Sensor Networks," *IEEE Workshop on Large Scale Real-Time and Embedded Systems* (*LARTES*), Austin, December 2002, p. 6.

[6] Y. Zeng, B. Zhao, J. Su, X. Yan and Z. Shao, "A Loop-Based Key Management Scheme for Wireless Sensor Networks," *Proceedings of the* 2007 *Conference on Emerging Direction in Embedded and Ubiquitous Computing* (*EUC*'07), Taipei, 17-20 December 2007, pp. 103-114. doi:10.1007/978-3-540-77090-9_10

[7] B. Dutertre, S. Cheung and J. Levy, "Lightweight Key Management in Wireless Sensor Networks by Leveraging Initial Trust," Technical Report, System Design Laboratory, Menlo Park, 2004.

[8] R. Blom, "An Optimal Class of Symmetric Key Generation Systems," *Proceedings of the Eurocrypt* 84 *Workshop on Advances in Cryptology*, Paris, 9-11 April 1984, pp. 335-338.

[9] H. Chien, R.-C. Chen and A. Shen, "Efficient Key Pre-Distribution for Sensor Nodes with Strong Connectivity and Low Storage Space," *Proceedings of the* 22*nd International Conference on Advanced Information Networking and Applications* (*AINA*'08), Okinawa, 25-28 March 2008, pp. 327-333.

[10] W. Du, J. Deng, Y. Han, P. Varshney, J. Katz and A. Khalili, "A Pairwise Key Pre-Distribution Scheme for Wireless Sensor Networks," *ACM Transactions on Information and System Security*, Vol. 8, No. 2, 2005, pp. 228-258. doi:10.1145/1065545.1065548

[11] C. Blundo, A. Santis, A. Herzberg, S. Kutten, U. Vaccaro and M. Yung, "Perfectly-Secure Key Distribution for Dynamic Conferences," *Proceedings of the* 12*th Annual International Cryptology Conference on Advances in Cryptology* (*CRYPTO*'92), Santa Barbara, 16-20 August 1992, pp. 471-486.

[12] H. Chan and A. Perrig, "PIKE: Peer Intermediaries for Key Establishment in Sensor Networks," *Proceedings of the* 24*th Annual Joint Conference of the IEEE Computer and Communications Societies* (*INFOCOM*'05), Miami, 13-17 March 2005, pp. 524-535.

[13] D. Liu and P. Ning, "Establishing Pairwise Keys in Distributed Sensor Networks," *Proceedings of the* 10*th ACM Conference on Computer and Communications Security* (*CCS*'03), Washington DC, 27-31 October 2003, pp. 52-61.

[14] D. Liu, P. Ning and R. Li, "Establishing Pairwise Keys in Distributed Sensor Networks," *ACM Transactions on Information and System Security*, Vol. 8, No. 1, 2005, pp. 41-77. doi:10.1145/1053283.1053287

[15] L. Eschenauer and V. D. Gligor, "A Key-Management Scheme for Distributed Sensor Networks," *Proceedings of the* 9*th ACM Conference on Computer and Communications Security*, Washington DC, 17-21 November 2002, pp. 41-47. doi:10.1145/586110.586117

[16] Y. Cheng and D. P. Agrawal, "Efficient Pairwise Key Establishment and Management in Static Wireless Sensor Networks," *Proceedings of the* 2*nd IEEE International Conference on Mobile Ad Hoc and Sensor Systems*, Washington DC, 7-10 November 2005, p. 7. doi:10.1109/MAHSS.2005.1542842

[17] Y. Cheng and D. Agrawal, "An Improved Key Distribution Mechanism for Large-Scale Hierarchical Wireless Sensor Networks," *Ad Hoc Networks*, Vol. 5, No. 1, 2007, pp. 35-48. doi:10.1016/j.adhoc.2006.05.011

[18] Y.-Y. Zhang, W.-C. Yang, K.-B. Kim, M.-Y. Cui and M.-S. Park, "A Rekey-Boosted Security Protocol in Hierarchical Wireless Sensor Network," *Proceedings of the* 2*nd International Conference on Multimedia and Ubiquitous Engineering*, Seoul, 24-26 April 2008, pp. 57-61.

[19] D. P. S. E. Christina and R. J. Chitra, "Energy Efficient Secure Routing in Wireless Sensor Networks," *Proceedings of* 2011 *International Conference on Emerging Trends in Electrical and Computer Technology* (*ICETECT*), Tamil Nadu, 23-24 March 2011, pp. 982-986.

[20] S. G. Yoo, S. Kang and J. Kim, "SERA: A Secure Energy and Reliability Aware Data Gathering for Sensor Networks," *Proceedings of* 2010 *International Conference on Information Science and Applications* (*ICISA*), Seoul, 21-23 April 2010, pp. 1-11. doi:10.1109/ICISA.2010.5480347

[21] W. Stallings, "Cryptography and Network Security: Prin ciples and Practice," 5th Edition, Pearson-Prentice Hall, Upper Saddle River, 2011.

[22] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. E. Culler and K. S. J. Pister, "System Architecture Directions for Networked Sensors," *Proceedings of the* 9*th International Conference on Architectural Support for Programming Languages and Operating Systems*, Cambridge, 12-15 November 2000, pp. 93-104.

[23] W. R. Heinzelman, A. Chandrakasan and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," *IEEE Transactions on Wireless Communications*, Vol. 1, No. 4, 2002, pp. 660-670. doi:10.1109/TWC.2002.804190

