A Robust Key Management Scheme with Strong Connectivity for Wireless Sensor Network

Avinash Chaurasia, Utkarsh Dubey, R. K. Ghosh
Department of Computer Science & Engineering
Indian Institute of Technology, Kanpur 208016, India

***Abstract*—In a wireless sensor network (WSN), creating pair-wise secure links between resource constrained sensor nodes is a major challenge. In this paper we point out three serious weaknesses, namely, lack of scalability, high memory requirements and vulnerability to node capture attacks in an earlier scheme for designing pair-wise key agreement between any pair of sensor nodes in a WSN proposed by Chien et al. [1]. We develop a new scheme based on Blom's key agreement protocol [2] which is free from the weaknesses mentioned above. Compared to [1], our scheme not only achieves better resilience against the node capture attack, but also requires less memory and at the same time provides more secure accesses to the network. Furthermore, the computational efficiency of the new scheme is comparable to that of [1].**

***Keywords*—WSN; node capture attack; key agreement; Moore-Penrose pseudo-inverse; pairwise key; one-way hash function.**

I. INTRODUCTION

Wireless sensor networks (WSNs) are generally deployed in remote, sensitive and sometimes in unattended hostile environments where the risk of physical attacks is very high. The topology, the size of the network and the density of sensor nodes are unknown before the deployment. Furthermore, radio based communications use spatial multiplexing, where a sensor node shares the medium with many other nodes. Therefore, providing security in a WSN is very challenging. The distributed nature of information flow in a WSN can quickly spread any attack and compromise the entire network in a short time. Therefore, apart from securing wireless links between sensor nodes, limiting the extent of a node capture attack is an extremely important issue in a WSN.

Establishing a secure link between communication end-points, and maintaining such a link over time are the two fundamental steps of any security protocol. Secure links can be established by using encryption keys. Key agreement is a challenging task in itself. A lot of research has been done on the key agreement protocols for wired networks [3], [4] as well as for WSNs [5]–[7]. Although cryptographic techniques, such as RSA [3], based on asymmetric keys offer reasonably high level of security against any kind of attack, these techniques can not easily be extended to WSN due to resource and computational constraints of sensor nodes.

This research was supported by Research I Foundation, IIT-Kanpur

A simple way to establish encryption keys is to pre-store a globally shared master key in all the nodes in a WSN before deployment. After deployment, each pair of nodes establishes a pair of keys by using the master key. The scheme is energy efficient and provides greater connectivity. However, the network becomes highly vulnerable. An adversary can access all the communications in the network just by capturing a single node and extracting the master key.

Another simple procedure for establishing encryption keys is to have each node pre-store a distinct key for every other node in the network. In a network having N nodes, it requires every node to pre-store $N-1$ keys. A node can communicate with another node using the key corresponding to it. In this case, if an attacker captures any node, then only the nodes which communicate with the captured node are compromised, while the rest of the network remains unaffected. Though this scheme is feasible for small networks, it cannot be used in networks having large number of sensor nodes due to a small storage capacity of a sensor node. On the other hand, in a small network, a substantial portion the network can be compromised by capturing even a single node.

A key management scheme based on random key pre-distribution was proposed by Eschenauer and Gligor [6]. The scheme consists of three phases: (i) key pre-distribution phase, (ii) shared key discovery phase and (iii) path key establishment phase. In the key pre-distribution phase, before deployment, a random subset of keys selected from a large key pool is stored in each sensor node. In the shared key discovery phase, each node broadcasts the indices of its keys to the neighbors. If two nodes share a common key then they use that as the secret key between them. Otherwise they use the path key establishment phase to establish a key between them. In this phase, if a secure path exists between the two nodes that do not have any key common between them, then one of two nodes sends a key along that path to the other.

Varshney et al. [5] developed an improved scheme based on Eschenauer and Gligor's key management scheme. They exploited the knowledge of spatial closeness of deployment assuming that a group of sensors dropped from air are likely to be spatially close. Du et al. [7] also developed their key

978-1-4673-1382-7/12/$31.00 ©2012 IEEE

agreement protocol following Eschenauer and Gligor's idea of random key pre-distribution. They argued that in certain cases deployment knowledge may be available. So, using deployment knowledge, keys can be established between neighbouring sensor nodes. This scheme provides perfect security if no more than λ ($1 \leq \lambda \leq n$) nodes are compromised.

Chien et al. [1] showed that although the EPKEM scheme proposed by Cheng and Agarwal [8] can support large networks and provide full network connectivity, it is vulnerable to node capture attack. They tried to improve the network's resilience against node capture attack. Unfortunately, the scheme was still unable to provide immunity from node capture attack. The second problem which made this scheme unusable is the lack of scalability. The third problem with the scheme is that it requires a substantial storage space at each sensor node; and hence, unsuitable for deployment on WSNs having large number of nodes.

In this paper, we have provided a cryptanalysis of Chien et al.'s scheme [1] to substantiate the point why it is susceptible to node capture attack. We then propose a new key management protocol which fixes both the problems. Our scheme is based on Blom's key agreement protocol [2]. We have also provided a theoretical analysis to show that the newly proposed scheme is more robust than Chien et al.'s scheme. Our scheme not only achieves better resilience against node capture attack, but requires less memory per node yet provides more secure access to the network. Furthermore, the computational efficiencies of the two schemes are comparable.

II. BACKGROUND

A. Blom's Key Agreement Protocol

Blom's scheme [2] was designed for a large network. Blom suggested that instead of storing $N-1$ keys at each node in a network of size N, a small set of secret keys can be shared among the nodes. These shared keys are used for key generation. As the keys are generated from a small set of secret keys, there will be dependencies between one another. Blom's scheme is based on MDS (Maximum Distance Separable) code. It uses a generator matrix $\mathbf{G}$ [9] of an (N, k) linear code over $GF(q)$, where N denotes the length of the codewords, and k denotes the dimension of code. That is, G is a $(k \times N)$ matrix having elements from $GF(q)$, where q is a power of prime. MDS code [10] is defined by a distance $d = N-k+1$, which is equivalent to saying that any set of k columns in $\mathbf{G}$ are linearly independent. Therefore, the code can only be determined by knowing a set of k elements of a row, any number less than that can not reveal any information about the code.

Blom's scheme allows any pair of nodes to calculate pair-wise secret keys. The matrix $\mathbf{G}$ (as explained above) gives a set of public keys and the rows of another matrix $\mathbf{A}$ act as private keys of nodes. The matrix $\mathbf{A}$ is calculated as:

$$\mathbf{A} = (\mathbf{D} \times \mathbf{G})^T,$$

where $\mathbf{D}$ is a random private symmetric matrix of size $(k \times k)$. After this phase is over, a node N_i can calculate the key between itself and N_j as follows:

$$K_{i,j} = \mathbf{A}[i] \times \mathbf{G}[j]^T,$$

where $\mathbf{A}[i]$ denotes the ith row of $\mathbf{A}$ and $\mathbf{G}[j]^T$ denotes the jth column of $\mathbf{G}$. Since, $\mathbf{D}$ is symmetric,

$$\begin{aligned} \mathbf{A} \times \mathbf{G} &= (\mathbf{D} \times \mathbf{G})^T \times \mathbf{G} \\ &= \mathbf{G}^T \times \mathbf{D} \times \mathbf{G} \\ &= \mathbf{G}^T \times \mathbf{A}^T \\ &= (\mathbf{A} \times \mathbf{G})^T. \end{aligned} \quad (1)$$

Implying that $\mathbf{K} = \mathbf{A} \times \mathbf{G}$ is symmetric. Therefore, $K_{i,j}$ is same as $K_{j,i}$.

B. Pseudo-inverse of Matrix

Pseudo-inverse of a matrix is the generalized inverse of the matrix. It exists even for a non-square matrix. Many algorithms have been proposed for calculating pseudo-inverse. Two of the popular ones are proposed by Moore [11] and Penrose [12]. Generally pseudo-inverse is referred to as Moore-Penrose pseudo-inverse. The pseudo-inverse is unique for each matrix, and it can be calculated by SVD (singular value decomposition). We use this property later in our scheme.

C. Chien et al.'s Scheme

Let us now examine how Chien et al.'s scheme suffers from the serious problems mentioned in Section I. Before analyzing the problems, let us go through a quick review of the scheme.

Chien et al.'s [1] key agreement scheme ensures full connectivity of WSN by establishing a pair-wise session key between any two nodes. It has the following three phases: (i) parameter pre-assignment, (ii) pair-wise key establishment and (iii) obsolete parameter erasure.

i. Parameter pre-assignment:
In this phase an $(N \times m)$ public matrix $\mathbf{G}$ and an $(N \times N)$ private symmetric matrix $\mathbf{D}$ are constructed by the base station (BS), where N is the size of network and $m = \lceil \sqrt{N} \rceil$. Next an $(m \times N)$ matrix $\mathbf{A} = (\mathbf{D} \times \mathbf{G})^T$ is computed. For each node N_i, the BS stores the following into N_i's memory:

1) ith row of $\mathbf{A}$,
2) a random number r_s (which is common to all nodes),
3) the shared key K_{BS} between BS and N_i, and
4) the generator s.

ii. Pair-wise key establishment:
Suppose the two nodes N_a and N_b (which are close to each other) want to establish a session key between them. Both the nodes generate random numbers r_a and r_b respectively. Then N_a computes $r_t = r_s \oplus r_a$, and broadcasts (ID_a,r_t) to N_b. Similarly, N_b broadcasts (ID_b,r_u), where $r_u = r_s \oplus r_b$. After receiving the broadcast message from N_b, N_a computes $K_{a,b} = \mathbf{A}[\text{ID}_a] \times \mathbf{G}[\text{ID}_b]^T$ and retrieves r_b using $r_u \oplus r_s = r_b \oplus r_s \oplus r_s = r_b$, where $\mathbf{A}[\text{ID}_a]$ is the ID_ath row of matrix $\mathbf{A}$ and $\mathbf{G}[\text{ID}_b]^T$ denotes the ID_bth column of matrix $\mathbf{G}$. Similarly, N_b computes $\text{K}_{b,a}$ = $\mathbf{A}[\text{ID}_b]\times\mathbf{G}[\text{ID}_a]^T$ and will be able to retrieve r_a using $r_t \oplus r_s$ = $r_a \oplus r_s \oplus r_s = r_a$. As $\mathbf{K} = \mathbf{A} \times \mathbf{G}$ is symmetric, $K_{b,a}$ will be same as $K_{a,b}$. Hence both nodes can calculate the pair-wise key as $\text{EK}_{a,b}$ = H($K_{a,b}$·r_a·r_b) = H($\text{K}_{b,a}$·r_b·r_a) = $\text{EK}_{b,a}$

iii. Obsolete parameter erasure:
After the pair-wise key establishment phase is over, each node erases all the pre-assigned parameters from its memory and retains the pair-wise keys with its neighbours plus the secret key with the BS.

III. Problems with Chien et al.'s Scheme

Chien et al.'s scheme suffers from the following three weaknesses: (i) it does not consider memory constraints on sensors nodes, (ii) it can not scale up to large size network, and (iii) it fails to ensure secure communication if any of the nodes gets captured.

A. Memory Constraint

The scheme demands a storage space of size $(N+3)\times|key|$ at each node, where $|key|$ is size of each number stored in the matrix cell. Furthermore, though in practice the matrix G can be created from the generator p of $GF(q)$, for the kth column of matrix $\mathbf{G}$ the seed p^k should also be stored at the node. Since sensor nodes typically have low memory capacities, the above scheme cannot be used for large sensor networks (where N is large).

B. Scalability Problem

Since the network size is N, and the number of rows in matrix $\mathbf{A}$ is only m, Chien et al.'s scheme can not assign a unique row to each sensor node. On an average each row of $\mathbf{A}$ will be replicated in m nodes. This will give rise to high vulnerability, as capturing a single node can provide the keys of m nodes. Under the scenario, if m different keys can be captured it will expose the whole network. In contrast, Blom's [2] original scheme has a better scalability as it is designed to reach all nodes in the network.

C. Cryptanalysis

Now we examine why the scheme is not secure against the node capture attack. If an adversary captures a node before *obsolete parameter erasure phase*, then not only the communications of captured node is compromised, but the adversary can also use the extracted information from the captured node to compromise the secret communications between the non-captured nodes of the network by calculating pseudo-inverse [11], [12] of the public matrix $\mathbf{G}$.

The following scenario explains the weakness of the scheme. Suppose the two nodes N_a and N_b want to establish a session key between them. If node N_a is captured by an adversary during the key establishment phase then the adversary will get $\mathbf{A}[\text{ID}_a]$, r_a and r_s, as the node is captured before completion of the third phase. Now, as (ID_b,$r_u = r_s \oplus r_b$) is globally broadcast from N_b to N_a, the adversary can overhear ID_b, and extract r_b using $r_u \oplus r_s$.

So after capturing node N_a, the attacker obtains $\mathbf{A}[\text{ID}_a]$, r_a, r_s, and r_u. Then he/she uses these values to retrieve r_b as explained earlier. After gathering the required information the attacker proceeds to calculate $K_{b,a}$, as explained below.

Since matrix $\mathbf{G}$ is public, the attacker can calculate $K_{a,b} = \mathbf{A}[\text{ID}_a] \times\mathbf{G}[\text{ID}_b]^T$ which is equal to $K_{b,a}$ = $\mathbf{A}[\text{ID}_b] \times \mathbf{G}[\text{ID}_a]^T$. Once the attacker has $K_{b,a}$, he/she can calculate $\mathbf{A}[\text{ID}_b]$ by multiplying $K_{b,a}$ with pseudo-inverse of $\mathbf{G}[\text{ID}_a]^T$.

$$\begin{aligned} K_{b,a} \times (\mathbf{G}[\text{ID}_a]^T)^+ &= \mathbf{A}[\text{ID}_b] \times \mathbf{G}[\text{ID}_a]^T \times (\mathbf{G}[\text{ID}_a]^T)^+ \\ &= \mathbf{A}[\text{ID}_b] \times \mathbf{I} \\ &= \mathbf{A}[\text{ID}_b], \end{aligned}$$

where $(\mathbf{G}[\text{ID}_a]^T)^+$ is pseudo-inverse of $\mathbf{G}[\text{ID}_a]^T$ and $\mathbf{I}$ is identity matrix. The pseudo-inverse can be determined by using techniques discussed in [11]–[13]. Now, the attacker knows $\mathbf{A}[\text{ID}_b]$, r_s and r_b, he/she can apply the same process on all the remaining nodes which established a pair-wise key with either N_a or N_b. The attacker can keep on repeating the process until whole network is compromised.

IV. Our Scheme

Now we propose a new scheme that overcomes the deficiencies of the two schemes [1], [2] and improves the network's resilience against the node capture attack. The proposed scheme provides full network connectivity, i.e., pair-wise key establishment for every pair of sensors within the communication range of each other. It also provides greater support for scalability.

Our scheme is based on following assumptions:

- All sensor nodes are homogeneous.

- The location of a sensor node cannot be predicted before deployment.
- If a node is captured then all the information inside it can be extracted by the adversary.
- The BS (base station) does not suffer from resource constraints including communication and computation capabilities. It can communicate directly with any node in the network.

The proposed scheme consists of two phases:

i. Parameter pre-assignment phase:
The BS computes an $(m \times N)$ private matrix $\mathbf{G}$ and an $(m \times m)$ private symmetric matrix $\mathbf{D}$, where N is the size of the network and $m = \lceil \sqrt{N} \rceil$. Then it calculates an $(N \times m)$ matrix $\mathbf{A} = (\mathbf{D} \times \mathbf{G})^T$. Before deployment, the BS pre-stores the following values in each node N_i:

(i) ith row of $\mathbf{A}$,
(ii) a random number r_i and
(iii) public identity ID_i (associated to a row number of $\mathbf{A}$ stored in the node).

The BS also keeps a copy of r_i with itself.

ii. Pair-wise key establishment phase:
Corresponding to each node N_i, the BS periodically transmits $M_i^{BS} = \mathbf{G}[\text{ID}_i]^T \times hash(r_i^{t_j})|t_j$, where t_j is some random number (that keeps changing on every transmission), and "|" denotes concatenation operation. The overall matrix to be transmitted over network is:

$$\begin{bmatrix} G_{11} \times hash(r_1^{t_j}) & \dots & G_{1N} \times hash(r_1^{t_j}) \\ G_{21} \times hash(r_2^{t_j}) & \dots & G_{2N} \times hash(r_2^{t_j}) \\ \vdots & \vdots & \vdots \\ G_{m1} \times hash(r_m^{t_j}) & \dots & G_{mN} \times hash(r_m^{t_j}) \end{bmatrix},$$

where G_{pq} is value of the p^{th} row and q^{th} column of generator matrix $\mathbf{G}$. Before broadcasting the matrix, BS concatenates it with t_j. This whole message is used by every other node that wishes to establish pair-wise key between itself and N_i. Suppose node N_a wants to establish a key with node N_b, then N_a captures message $M_b^{BS} = \mathbf{G}[\text{ID}_b]^T \times hash(r_b^{t_k})|t_k$ from BS, and sends the message m_a = $(\text{ID}_a|\text{t}_k)$ to N_b. On receiving the message from N_a, N_b knows that N_a used the message of BS broadcast at time t_k. Likewise, node N_b captures $M_a^{BS} = \mathbf{G}[\text{ID}_a]^T \times hash(r_a^{t_l})|t_l$ from broadcast by BS, and sends the message m_b=$(\text{ID}_b|\text{t}_l)$ to N_a. Since $\mathbf{A} \times \mathbf{G}$ is symmetric (ref. equation 1), after receiving each other's message, nodes N_a and N_b compute the pair-wise keys as follows:

N_a computes:

$$\begin{aligned} K_{a,b} &= \mathbf{A}[ID_a] \times M_b^{BS} \times hash(r_a^{t_l}) \\ &= \mathbf{A}[ID_a] \times \mathbf{G}[ID_b]^T \times hash(r_b^{t_k}) \times hash(r_a^{t_l}) \end{aligned}$$

N_b computes:

$$\begin{aligned} K_{b,a} &= \mathbf{A}[ID_b] \times M_a^{BS} \times hash(r_b^{t_k}) \\ &= \mathbf{A}[ID_b] \times \mathbf{G}[ID_a]^T \times hash(r_a^{t_l}) \times hash(r_b^{t_k}) \end{aligned}$$

Since $\mathbf{A} \times \mathbf{G}$ is symmetric, from Blom's Scheme [2]

$$\mathbf{A}[ID_a] \times \mathbf{G}[ID_b]^T = \mathbf{A}[ID_b] \times \mathbf{G}[ID_a]^T.$$

Therefore,

$$\begin{aligned} &hash(r_a^{t_l}) \times hash(r_b^{t_k}) \times \mathbf{A}[ID_a] \times \mathbf{G}[ID_b]^T \\ &\quad = hash(r_a^{t_l}) \times hash(r_b^{t_k}) \times \mathbf{A}[ID_b] \times \mathbf{G}[ID_a]^T, \end{aligned}$$

and $K_{a,b} = K_{b,a}$. So, the above key can be used as secret key between node N_a and node N_b.

V. KEY REVOCATION AND REPLACEMENT PHASE

On detection of a node capture, the session key related to the captured node will be dropped. The process for detection of node capture attack can be found in [14].

Sometimes we may want to replace an energy starved node or a captured node or may want to add more node for enhancing the information gathering. But in any of the cases, the number of nodes should not exceed N (being limited by the number of rows N in matrix $\mathbf{A}$).

When a new node N_a is added, it will be mapped to one of the unassigned rows (say i) of $\mathbf{A}$, random number r_i and node public identity ID_i (row i of $\mathbf{A}$). Any node which wants to communicate with this node can use phase (ii) of the proposed scheme for establishing the pair-wise secret key.

VI. ANALYSIS OF THE PROPOSED SCHEME

Suppose a node N_a is captured then all the communication going through this node will no more be secure. This will not lead to insecure communication among other nodes. It can be illustrated as follows. Let $K_{a,b}$ be the secret key between node N_a and node N_b and $K_{b,c}$ be the secret key between N_b and N_c. Now suppose the attacker captures N_a and retrieves the information stored inside it, i.e., r_a, ID_a, $\mathbf{A}[\text{ID}_a]$ and $K_{a,b}$. After getting this information the attacker will try to extract other communication keys. He/she will need the secret keys which are stored on other nodes. But he/she can retrieve only limited information, namely, $K_{a,b}$, $\mathbf{G}[ID_b]^T \times hash(r_b^{t_k})$. Since $\mathbf{G}$ is private, the attacker will not get $\mathbf{A}[ID_b]$. If the attacker somehow gets hold of $\mathbf{G}$, he/she can calculate $\mathbf{A}[ID_b]$ by using following procedure. Since, $\mathbf{G}$ is known, $hash(r_b^{t_k})$ can be determined by simply dividing the message M_a^{BS} by $\mathbf{G}[ID_b]^T$. Now that the attacker knows $K_{b,a}$, $hash(r_b^{t_k})$ and $\mathbf{G}[ID_a]^T \times hash(r_a^{t_l})$; he/she can

calculate $\mathbf{A}[ID_b]$ by dividing $K_{b,a}$ with product of $hash(r_b^{t_k})$ and $\mathbf{G}[ID_a]^T \times hash(r_a^{t_l})$.

$$\mathbf{A}[ID_b] = \left(\frac{K_{b,a}}{hash(r_b^{t_k}) \times \mathbf{G}[ID_a]^T \times hash(r_a^{t_l})} \right) \quad (2)$$

Therefore, the security of the proposed scheme is enhanced due to security of both $\mathbf{A}$ and $\mathbf{G}$.

VII. COMPARISON OF TWO SCHEMES

A comparison of both the schemes is provided in Table I. We made the proposed scheme more robust and resistant to node capture attack at the cost of a little increase in computation, as exponentiation is more expensive than simple XOR operation. However, the size of the network supported by our scheme is to N. Compared to Chien et al.'s scheme the memory requirement is reduced considerably by storing reduced number of keys ($\sqrt{N}$) at each node. So the new scheme provides increased support for scalability.

TABLE I
PERFORMANCE COMPARISON

Parameter	Chien et al.'s scheme	Our scheme
Resistant to node capture attack even when node is captured during deployment	No	Yes
Computation cost for pair wise session key by one node	$NT_{mul}+T_h+(N+1)T_{add}$	$(N+1)T_{mul}+T_h+T_{pow}+(N-1)T_{add}$
Storage space	$(N+3)\|key\|$	$(\lceil\sqrt{N}\rceil+2)\|key\|$
Maximum supported network size	$\sqrt{(N)}$	N

In the table,

T_{mul}: the cost of single multiplication,
T_h: the cost of computing one way hash,
T_{add}: the cost of single addition,
T_{pow}: the cost of raising a number to some power, and
$|key|$: the size of number stored.

VIII. CONCLUSION

In this paper we pointed out the three serious weaknesses of the scheme proposed by Chien et al. [1]. We proposed a new scheme based on Blom's [2] scheme in order to remove the weaknesses of the former scheme. The proposed scheme provides better resilience against node capture attacks compared to Chien et al.'s scheme. The computational cost is slightly higher, but it has much lower memory requirements for storing keys, and it can support more number of nodes in the network compared to Chien et al.'s scheme. Therefore, the proposed key agreement protocol provides relatively better support for scalability than Chien et al.'s scheme.

Our scheme incurs a slightly higher computational overhead than Chien et al.'s scheme. Therefore, one argument that can go against the new scheme may be that the extra overhead would affect the limited battery capacities of the sensor nodes. However, the overhead being limited to the use of exponentiation operation in the place of XOR, is not really substantial. Yet reducing computational overhead could be an important starting point for the future work. Similarly, though the new scheme provides better support for scalability, perhaps this issue can still be worked on.

REFERENCES

[1] H. Y. Chien, R.-C. Chen, and A. Shen, "Efficient key pre-distribution for sensor nodes with strong connectivity and low storage space," in *Proceedings of the 22nd International Conference on Advanced Information Networking and Applications*. Washington DC, USA: IEEE Computer Society, 2008, pp. 327–333. [Online]. Available: http://portal.acm.org/citation.cfm?id=1395079.1395185

[2] R. Blom, "An optimal class of symmetric key generation systems," in *Proc. of the EUROCRYPT 84 workshop on Advances in cryptology: theory and application of cryptographic techniques*, 1985, pp. 335–338.

[3] R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, pp. 120–126, 1978.

[4] W. Diffie and M. Hellman, "New directions in cryptography," *Information Theory, IEEE Transactions on*, vol. 22, no. 6, pp. 644 – 654, Nov 1976.

[5] W. Du, J. Deng, Y. Han, S. Chen, and P. Varshney, "A key management scheme for wireless sensor networks using deployment knowledge," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 1, march 2004, pp. 4 vol. (xxxv+2866).

[6] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proceedings of the 9th ACM conference on Computer and communications security*, ser. CCS '02. New York, NY, USA: ACM, 2002, pp. 41–47. [Online]. Available: http://doi.acm.org/10.1145/586110.586117

[7] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz, and A. Khalili, "A pairwise key pre-distribution scheme for wireless sensor networks," in *Proceedings of the 10th ACM conference on Computer and communications security*, 2003, pp. 42–51.

[8] Y. Cheng and D. Agrawal, "Efficient pairwise key establishment and management in static wireless sensor networks," *IEEE International Conference on Mobile Adhoc and Sensor Systems Conference*, p. 550, 2005.

[9] J. T. D. Joyner, R. Kreminski, "Applied abstract algebra," http://wdjoyner.com/teach/book/node125.html, January 2003.

[10] F. MacWilliams and N. Sloane, *The Theory of Error-Correcting Codes*. New York: North-Holland, 1977.

[11] E. H. Moore, "On the reciprocal of the general algebraic matrix," *Bulletin of the American Mathematical Society*, vol. 26, pp. 394–395, 1920.

[12] R. Penrose, "A generalized inverse for matrices," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 51, no. 03, pp. 406–413, 1955. [Online]. Available: http://dx.doi.org/10.1017/S0305004100030401

[13] K. K. Schmidt, "The Moore-Penrose inverse of a vector: coping with a sometimes tricky case differentiation," http://faculty.pepperdine.edu/dstrong/LinearAlgebra/2009/JMM2009Schmidt.pdf, 2009.

[14] W.-T. Su, K.-M. Chang, and Y.-H. Kuo, "ehip: An energy-efficient hybrid intrusion prohibition system for cluster-based wireless sensor networks," *Comput. Netw.*, vol. 51, pp. 1151–1168, March 2007. [Online]. Available: http://portal.acm.org/citation.cfm?id=1223683.1223800