International Journal of Enhanced Research in Science Technology & Engineering, ISSN: 2319-7463
Vol. 4 Issue 1, January-2015, pp: (25-31), Impact Factor: 1.252, Available online at: www.erpublications.com

An Efficient Pairwise and Group Key Management Scheme For Wireless Sensor Network

Abdalkahik W. Hussain[1], Mahmood K. Ibrahem[2]
[1]Network Engineering Department, College of Information Engineering
[2]Internet Engineering Department, College of Information Engineering
Al-Nahrain University, Baghdad, Iraq

Abstract: Wireless sensor network security is becoming increasingly important especially for sensitive applications; one of the most important mechanisms for ensuring security is key management. This paper presents a pairwise and group key management schemes for wireless sensor network, the aim of the proposed schemes is to reduce the energy consumption for pairwise and group key establishment. The pairwise key establishment is based on a modified Blom's scheme where ID based circular matrix is used instead of vandermonde matrix to generate the public matrix of Blom's scheme to reduce computation and storage overhead. The group key establishmentenables every member of the group to participate in the group key agreement to minimize the overhead on the group head and other nodes, while at the same time enable every member to verify the authenticity of the group key.

Keywords: Wireless sensor network (WSN), Blom's scheme, Circular matrix, Key establishment, Key management, Security.

Introduction

Advances in micromechanical systems, vary large scale integration and wireless communication had led to the development of tiny, low cost network of sensing devices that provides the ability to monitor and take an action to events that take places in the environment. The basic components of a wireless sensor network are: (1) sensing devices; typically have a limited computation, communication and storage capability, (2) the base station for collecting information gathered by sensing devices. Wireless sensor network usually deployed in a hostile environment, security in that case becomes a necessary requirement.
One of the most important areas for ensuring security is key management. The purpose of key management is to enable nodes to find secret keys to secure transmitted information. Most of the key management schemes proposed for WSN are based on symmetric keys because it requires less computation, memory and communication overhead which make it more suited for this type of networks. Eschenauer and Gligor in [1] proposed a random key predistribution scheme, where each node is preloaded with a subset of keys from global key pool. Zhu, Setia and Jajodia proposed LEAP [2] which supports the establishment of four types of keys for each sensor node – key shared with another sensor node, a key shared with the base station and another sensor node, cluster key and global key shared with by all nodes in the network. Chan, Perrig and Song proposed q-composite random key predistribution scheme [3] where nodes to establish a pairwise key, have to shares at least q keys. Du et al. [4] combined random key predistribution scheme with a modified Blom's scheme to make more resilient to node compromise; however it requires a lot of computation, memory and communication overhead due the use vandermonde matrix. Rahman, Sampalli and Hussain in [5] proposed a pairwise and group key management protocol based on Blom's scheme which support new node addition, and key refresh, it also supports on-the-fly dynamic secure group creation. Yu, Chia-Mu, Chun-Shien and Sy-Yen in [6] proposed CARPY+ scheme a non-interactive key establishment scheme based on Blom's scheme with great resilience to a large number of node compromises due to the addition of random noise to break the direct relation between the private and public matrix. Reddy in [7] proposed a key management scheme based on Blom's predistribution scheme, the author presents a solution to reduce computation overhead based on the use of the Hadamard matrix instead of the Vandermonde matrix as the public matrix of Blom's scheme.

In this paper an efficient pairwise key management scheme based on Blom's scheme is proposed, which enable any two nodes to find a shared key without any interaction by constructing Blom's public matrix using ID's based circular matrix, in addition an efficient group key agreement mechanisms is proposed, where each group member participate in the group key agreement, so the load of establishing the group key is distributed among members of the group, at the same times it enables node to verify the source of the group key by employing an authenticated Bloom filter. The rest of the paper is organized as follow the first part provide an overview of the original Blom's scheme. Second part presents the proposed pairwise and group key distribution schemes, third part describe the performance analysis and simulation results, and finally concluding remark is provided.

Notation

Table 1: Summary of Notations

d	Number of neighbors
$MAC_{Kij}(msg)$	Message authentication code of msg using shared key between node i and j
$E_{Kij}(msg)$	Encryption of msg using shared key between node i and j
$GF(q)$	Finite Field Size
N	Number of Nodes in the network
τ	Resilience Factor
BFV	Bloom Filter Vector
K_{ij}	Key Shared between node I and J
G	Public matrix of Blom's scheme
D	Private matrix of Blom's scheme
M	Bloom filter size
N	Number of element inserted into Bloom Filter
K	Number of Hash function
H	Size of MAC function output
ID_i	ID of node *i*

Overview of Blom's Scheme

Blom's scheme is a matrix based key predistribution scheme that enables any two nodes to find a shared secret key as long as no more τ nodes are compromised [8]. Du et al. [4] modified this scheme to make it more suitable for resource constrained sensor networks. Blom's scheme involves two phases:

A. Secret Information Pre-deployment Phase

The Network administrator before deploying the network, generates $(\tau+1) \times N$ public matrix G over $GF(q)$ where N is the number of nodes in the network, τ is the resilience factor and q is very large prime number. Then the Network administrator generates $(\tau+1) \times (\tau+1)$ private matrix D over $GF(q)$ and computes $A = (D.G)^T$. after that the row of matrix A is distributed to each sensor node in the network.

B. Key Agreement Phase

When two nodes want to find a shared secret key between them, they exchange their column of G matrix and multiply the received column with row of matrix A stored at the node to generate the secret key K. Suppose that node i and j wants to find a pairwise key, after exchanging their column, node i computes $K_{ij} = A_{i,-}.\ G_{-,j}$ and node j computes $K_{ji} = A_{j,-}.\ G_{-,i}$ because D is a symmetric matrix then it's easy to see that K is also a symmetric matrix:

$$A.G = (D.G)^T.G = G^T.D.G = (A.G)^T$$

Proposed Scheme

In this section proposed pairwise and group key management schemes are shown. We assume that the network consists of N sensor nodes with the same memory, processing and communication capabilities. The nodes identifications are $\{0, 1, 2, \ldots, N-1\}$ and no deployment knowledge of the network assumed.

A. Modified Blom Scheme

The Implementation of Blom's key predistribution scheme in WSN relays on Vandermonde matrix to generate public matrix. However using this type of matrix, sensor node has to preform 2τ modular multiplications and exponentiation operation which is a costly operation for limited resources sensor node especially for large value of τ. Reddy [7] proposed using non-

binary Hadamard matrix as public matrix, which significantly reduced the computation overhead because only addition and subtraction operation are required to generate a pairwise key, however the overhead is still high, because the whole Hadamard matrix has to be generated at the sensor node. To further reduce the computation overhead we propose using a circular matrix to generate the public matrix of Blom's scheme, the characteristics is matrix is that any node given the ID of destination node can generate that node corresponding column. Also in circular matrix every τ+1column are linearly independent which is a necessary condition for security guarantee. Algorithm.1 shows the steps required to generate the proposed matrix:

Algorithm 1: *Generation of the Proposed G Matrix*
Input: N, τ, q **Output:** G
Loop from $i = 0$ to $N-1$
Loop from $j = 0$ to τ
$G(j, i) = (i - j) \bmod N$
End
End

Fig. 1 shows an example of ID based circular matrix public matrix construction of size $N*N$ for nodes ID $(0, 1, \ldots\ldots, N)$:

$$\begin{bmatrix} 0 & 1 & 2 & 3 & \ldots\ldots & N \\ N & 0 & 1 & 2 & \ldots\ldots & N-1 \\ . & . & . & . & & . \\ . & . & . & . & & . \\ . & . & . & . & & . \\ 1 & 2 & 3 & 4 & \ldots\ldots & 0 \end{bmatrix}$$

Fig. 1: Example of G matrix construction

B. Pairwise Key Management

To enable any two nodes to find a pairwise key between them, the following phases are preformed:

1. Key Predistribution Phase: This phase is similar to that of the Blom's scheme, but instead of using Vandermonde or Hadamard matrix to generate G public matrix, the public matrix is constructed using ID based circular matrix, the central authority generates circular public matrix and $(\tau+1) \times (\tau+1)$ D matrix over $GF(q)$ and then computes and stores row of A matrix in the sensor nodes.
2. Key Agreement Phase: when a pair of nodes (i, j) wants to agree on a pairwise key, our scheme doesn't require any interaction between them because each node itself can generate the corresponding column of the public matrix given only the ID of the destination node using algorithm 2, then each node multiply its row of matrix A by the generated public matrix column of the destination node

Algorithm 2: *Generating i-node Column of G matrix*
Input: N, τ, i, j **Output:** $G_{i,-}$
Loop from j = 0 to τ
$G(j, i) = (i - j) \bmod N$
End

C. Group Key Agreement

The proposed group key agreement to enable a group of sensor nodes to agree on a group key in an efficient manner, which consists of the following steps:

1. The group head u first generates the group key, and computes the d-1MACs over it using pairwise keys it shares with other nodes $v_1, v_2, \ldots, v_{d-1}$ in the group.

2. The generated MACsare compressed in Bloom filter [9] to reduce the space at the cost of usually negligible false positive rate (*fpr*), that is, with certain probability unauthentic key could be falsely considered as a member of the set, and then the generated Bloom filter is broadcast to all nodes in the group.
3. Now, the group head sends the group key to node v_1 encrypted and authenticated with the pairwise key both of them sharing it
4. When destination node receive the group key, first it checks the MACs of the received packet then decrypt the group key, then check the Bloom Filter, if its authenticated it re-encrypt and computes the MAC over the whole packet and send it to the next node v_2 if it's not authenticated it discard the packet and send a report to the group head.

The group key agreement process can be written as follows:

$$u \to *: BFV(MAC_{u,v1}(K_G \parallel C_{u,v1}) \parallel MAC_{u,v2}(K_G \parallel C_{u,v2}) \parallel ..)$$
$$u \to v_1 : E_{Ku,v1}(K_G \parallel C_{u,v1}) \parallel MAC_{Ku,v1}(K_G \parallel C_{u,v1})$$
$$v_1 \to v_2 : E_{Kv1,v2}(K_G \parallel C_{v1,v2}) \parallel MAC_{Kv1,v2}(K_G \parallel C_{v1,v2})$$

.....

Using secure Bloom filter enable group member to authenticate the source of the group key, but introducesfalse positive probability (*fpr*), however the total *fpr* value for the whole group in the proposed scheme is very small and it's shown in Eq. (1) (assuming an optimal number of MACs functions (*B*) is used) this is due to the fact that member nodes are checking the same value, which is the group key and if some node failed to authenticate the Bloom filter, it immediately send a warning message to the group head, which invalidate the current group key and start a new group key agreement. Notices, an increase in *d* or *m* significantly reduce the *fpr* value, at the cost of increase in computation or communication overhead respectively although this overhead is negligible.

$$fpr = \prod_{i=0}^{d-1} (0.618)^{m/n} \quad (1)$$

Due to the use of Bloom Filter, B hash functions need to be implemented, which is not practical, instead only one MAC function is used and the previous MAC output is feedback to the current MAC calculation. However even using this scheme requires high computation overhead, because node has to compute *(d-1)*B*MAC function for each neighbor. Instead of computing *B*MACs for a single neighbor, the sender can break the MAC output into k parts, each part is of size 4 bytes (32 bits) which provides 32 bit randomness and insert each part into the Bloom Filter in this case the total number of computed MAC functions (*M*) is given in Eq. (2):

$$M = (d-1) * \left\lceil \frac{B*4}{H} \right\rceil \quad (2)$$

Where *H* is the size of MAC function output, for example if*d* = 10, *B* =5 and MAC function output size to 256 bit, the required MAC function computation is equal to 9 instead of 45. In addition the size of Bloom filter increases, with the increase in number of group members for the same false positive probability, to reduce the size of Bloom filter only half of the node's MAC is inserted into Bloom filter, at the cost of only half of nodes in the group can authenticate the group key and the total *fpr* value is shown in Eq. (3):

$$fpr = \prod_{i=0}^{\left\lfloor \frac{d-1}{2} \right\rfloor} (0.618)^{m/n} \quad (3)$$

In addition the number of computed MACs (*M*) is reduced to half shown in Eq. (4):

$$M = \left\lceil \frac{(d-1)}{2} \right\rceil * \left\lceil \frac{B*4}{H} \right\rceil \quad (4)$$

International Journal of Enhanced Research in Science Technology & Engineering, ISSN: 2319-7463
Vol. 4 Issue 1, January-2015, pp: (25-31), Impact Factor: 1.252, Available online at: www.erpublications.com

Performance Evaluation

In this section the performance analysis of the proposed pairwise and group key management scheme in terms of, energy overhead and memory usage with schemes with the scheme proposed in [5] is presented.

A. Energy Overhead

The proposed pairwise key establishment is compared with Robust Blom scheme proposed in [5]. The energy consumption is computed according to the number of CPU cycles each schemes requires to compute a single pairwise key were simulated using Cross-Studio for T1 MSP340F1611 microcontroller simulator (http://www.rowley.co.uk/) and multiplying it with the energy consumed per single clock cycle. By varyingfor q equal to 64 bits in length the result is shown in Fig. 3 and then varying the q and τ is fixed at 48 the result is shown Fig. 4. Because both schemes doesn't requires any communication overhead to establish a pairwise key between communicating parties the communication overhead is negligible.

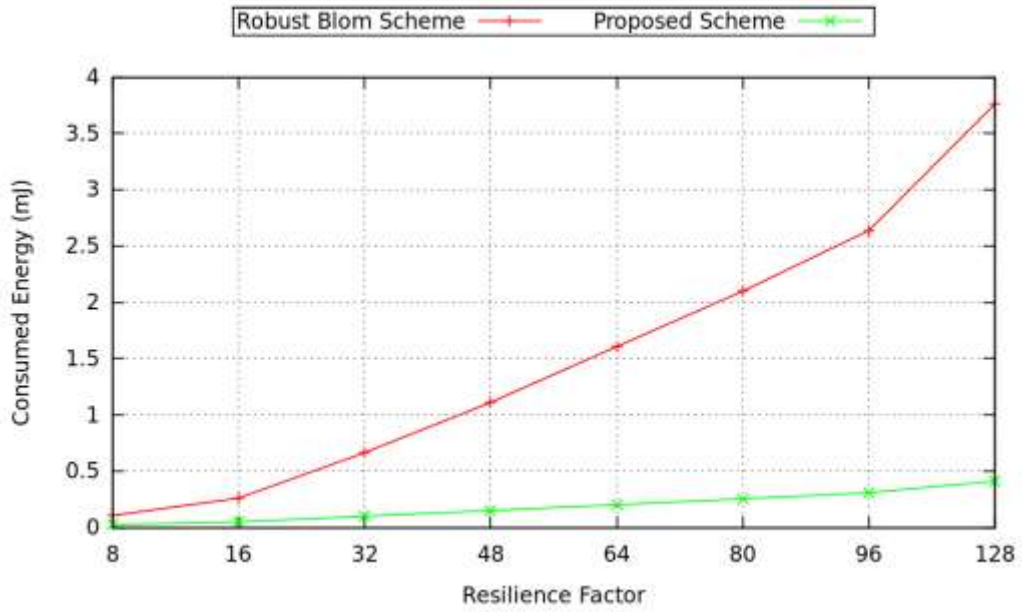

Fig. 2. : Comparison of the amount of consumed energy (mJ) to compute a pairwise key for different Resilience factor

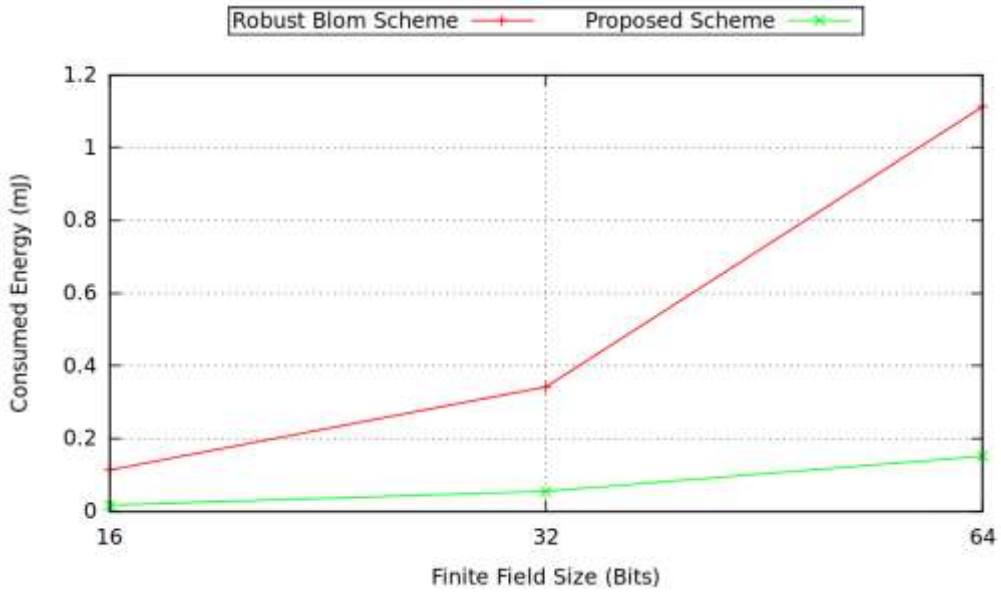

Fig. 3. : Comparison of the amount of consumed energy (mJ) to compute a pairwise key for different key size

The results shows that the proposed scheme more efficient than Robust Blom's scheme [5] in the term of energy saving. Notice that the increase in resilience factor or key size greatly effect scheme in [5] due to costly modular exponentiation operation to generate the G matrix column and modular multiplication to generate the shared key, while its effect is negligible on the proposed scheme.

The Energy consumption of the proposed group key agreement scheme in comparison with Robust Blom's scheme is shown in Fig. 5. The size of group key is 64 bits, RC5 [10] in OFB [11] mode of operation is used to provide encryption and Blake2B [12] is used as a MAC function (although any Encryption and MAC mechanisms can used) and simulated using Castalia [13]. The results shows that the proposed scheme consumes less energy than Robust Blom's scheme this due to the high communication and computation overhead incurred for broadcasting group key agreement information and pairwise key computation, We also note that the higher d is, the higher the consumed energy, while proposed scheme the overhead increases slightly because of hash function calculation at the group head and Bloom filter size, however the impact on the energy consumption is very small, even if number of neighbors increased.

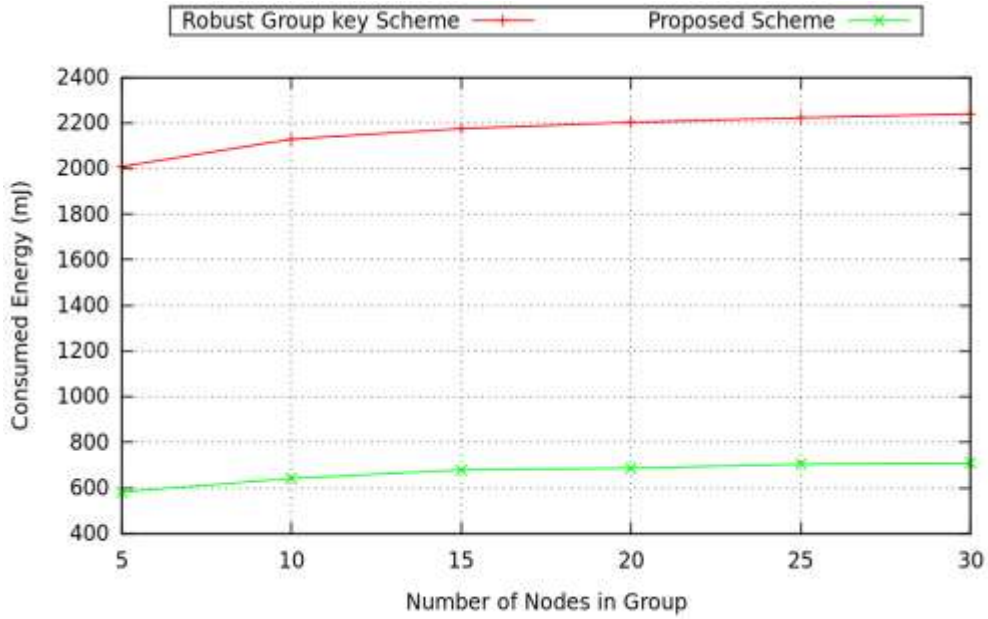

Fig. 4. : Comparison of the amount of consumed energy (mJ) to establish a group key for different number of nodes in a group of d nodes

B. Memory Usage

In the term of memory usage the most memory overhead in our proposed protocol comes from the fact that every node needs to store $(\tau + 1).q$ bits as the pre-loaded secret, for example, if q = 64 bits and τ = 47, then memory overhead equals to 384 bytes, the scheme in [5] has the same storage overhead. In addition our scheme requires the implementation of a MAC function and an encryption algorithm; however most of the proposed security protocols for WSNs for example [9] and [10] provide these functions.

Conclusion

This paper presented an efficient pairwise key pre-distribution wireless sensor networks based on modified Blom's scheme, by using circular matrix and its individual elements are nodes ID, the public matrix column can be easily generated at nodes without the need for storing any information, in addition it doesn't requires a lot of computation to generate a shared secret key. Also we presented a new group key agreement scheme for a group of sensor nodes which balance the load of group key agreement across all nodes in the group. The proposed schemes are compared with the Robust Blom's scheme and Simulation result shows that our schemes surpassit in the term of energy efficiency.

References

[1]. L. Eschenauer, and V. D. Gligor, "A Key-Management Scheme for Distributed Sensor Networks," in 9th ACM conference on Computer and Communications Security, 2002.
[2]. S. Zhu , S. Setia and, S. Jajodia, "Leap: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks.," in 10th ACM Conference on Computer and Communications Security (CCS '03), 2003.
[3]. H. Chan, A. Perrig, and Song, "Random Key Predistribution Schemes for Sensor Networks," in Proceedings of the 2003 IEEE Symposium on Security and Privacy (SP '03), Washington, DC, 2003, p. 197.
[4]. W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz, and A. Khalili, "A pairwise key predistribution scheme for wireless sensor networks," ACM Transactions on Information and System Security (TISSEC), vol. 8, no. 2, pp. 228 - 258, May 2005.
[5]. M. Rahman, S. Sampalli, and S. Hussain, "A robust pair-wise and group key management protocol for wireless sensor network," in GLOBECOM Workshops (GC Wkshps), Miami, FL, 2010, pp. 1528 - 1532.
[6]. C. Yu, C. Lu, and S. Kuo, "Non-interactive Pairwise Key Establishment for Sensor Networks," IEEE Transactions on Information Forensics and Security, vol. 3, no. 5, pp. 556 - 569, September 2010.
[7]. RohithSingi Reddy, "Key Management in Wireless Sensor Networks Using a Modified Blom Scheme," ARXIV, March 2011.
[8]. R. Blom, "An optimal class of symmetric key generation systems," in Advances in Cryptology: Proceedings of EUROCRYPT, Stockholm, 1984, pp. 335 – 338.
[9]. H. B. Bloom, "Space/time trade-offs in hash coding with allowable errors," Communications of the ACM, vol. 13, no. 7, pp. 422-426, July 1970.
[10]. J. Aumasson, S. Neves, Z. O'Hearn, and C. Winnerlein, "BLAKE2: Simpler, Smaller, Fast as MD5," in ACNS , Banff, 2013, pp. 119 - 135.
[11]. W. Stallings, Network Security Essentials: Applications and Standards, 4th ed. United States of America: Prentice Hall, 2011.
[12]. R. L. Rivest, "The RC5 Encryption Algorithm," in 1994 Leuven Workshop on Fast Software Encryption, 1995, pp. 86 – 96.
[13]. A. Boulis, "Castalia: revealing pitfalls in designing distributed algorithms in WSN," in Proceedings of the 5th international conference on Embedded networked sensor systems SenSys '07, New York, 2007, pp. 407-408.
[14]. Burton H. Bloom, "Space/time trade-offs in hash coding with allowable errors," Communications of the ACM, vol. 13, no. 7, pp. 422-426, July 1970.