A Continuous Secure Scheme in Static Heterogeneous Sensor Networks

Boqing Zhou, Jianxin Wang, *Senior member, IEEE*, Sujun Li, Yun Cheng, and Jie Wu, *Fellow, IEEE*

***Abstract*—In heterogeneous sensor networks (HSNs), which use existing key predistribution schemes, network security will significantly decline with time. In this paper, a continuous secure scheme is proposed based on two-dimensional backward key chains. In the scheme, powerful sensors do not need to be equipped with tamper-resistant hardware. Analysis and simulations indicate that the proposed scheme can significantly improve the performance of existing schemes in resilience against node capture attacks throughout the lifecycle of static HSNs.**

***Index Terms*—heterogeneous sensor network, two-dimensional backward key chain, pairwise key.**

I. Introduction

HSNS, which consist of a small number of powerful H-sensors (e.g., PDAs) and a large number of L-sensors (e.g., the MICA2-DOT [1]), have attracted much attention due to their better performance and scalability compared with homogeneous sensor networks. Security is a critical issue in the deployment of HSNs in hostile environments. However, due to the resource constraints on nodes, achieving secure communi-cations between nodes are non-trivial.

Public-key operations (both software and hardware implementations), albeit computationally feasible, consume energy approximately three orders of magnitude higher than symmetric key encryption [2]. Therefore, in the last few years, different pairwise key distribution schemes using symmetric key algorithms have been developed for HSNs [3]-[6].

However, the problem of continuous secure is still not solved for HSNs. Continuous secure denotes that HSNs have a good performance in the resilient against node capture attacks throughout their lifecycle. In [3] and [4], due to the repeated use of fixed key pool, a fraction of keys known by an attacker increases with the capture of each node. As a result, network security significantly declines with time.

This paper is partially supported by the postdoctoral grant of Central South University with grant No. QT1211, the National Natural Science Foundation of China under grant no. 61232001 and 11071272, the Hunan Provincial Natural Science Foundation of China (12JJ2040), the Research Foundation of Educa-tion Committee of Hunan Province, China(09A046, 10A062 and 13B068).

Boqing Zhou and Jianxin Wang are with the School of Information Science and Engineering, Central South University, Changsha Hunan 410083, China (Email: jxwang@csu.edu.cn).

Boqing Zhou is also with the Department of Information Science and Engineering, Hunan Institute of Humanities, Science and Technology, Loudi Hunan 417000, China.

Sujun Li and Yun Cheng are with the Department of Information Science and Engineering, Hunan Institute of Humanities, Science and Technology, Loudi Hunan 417000, China .

Jie Wu is with the Department of Computer and Information Sciences, Temple University, USA

In this paper, a continuous secure scheme is proposed for static HSNs (CSS-SH). The contribution of this paper is summmarized as follows: (1) We are the first to apply two-dimensional backward key chains technique [11] to HSNs; (2) n disjoint and interrelated key pools are constructed; (3) A new key predistribution scheme is proposed. Analysis and simulations indicate that the proposed scheme can significantly improve the performance of existing schemes in continuous secure of static HSNs.

The paper is organized as follows. Section II reviews the related work. Section III presents our scheme, and Section IV analyzes the scheme. Section V concludes the paper.

II. Related work

For sensor networks (SNs), the basic scheme [8] was proposed by Eschenauer and Gligor, in which each sensor picks some keys randomly from a large key pool before deployment. Two sensors can establish a shared key, if they have at least one common key. To enhance the security of the basic scheme against small-scale attacks, q-composite scheme was proposed [9], in which q common keys are required for two nodes to establish a shared key. To improve the network resilience against node capture throughout the lifecycle of SNs, Zhou et al. proposed a secure scheme using deployment knowledge [10]. Li et al. proposed a continuous secure scheme based on two-dimensional backward key chains [11].

For HSNs, Du et al. proposed *AP-D* scheme based on an asymmetric predistribution key management (*AP*) method. In the *AP* method, an L-sensor and an H-sensor pick t_1 and t_2 ($t_1 \ll t_2$) keys from a large key pool respectively before deployment. Two nodes can establish a shared key through either the basic scheme [8] or the q-composite scheme [9]. Lu et al. proposed a key management scheme (*AP-L*) using AP method [4]. In *AP-L*, the key pool of L-sensors is a subset of the key pool of H-sensors. In the two schemes [3]-[4], all nodes choose their keys from the same key pool. An attacker can easily obtain a large number of keys by capturing a small fraction of nodes, which can make HSN ineffective in continuous security. To improve the performance in continuous security, constructing disjoint and interrelated key pools is a simple and suitable method.

III. CSS-SH SCHEME

Clusters are formed in HSNs. Clustering-base schemes are promising techniques for sensor networks because of their good scalability and support for data aggregation. For HSNs,

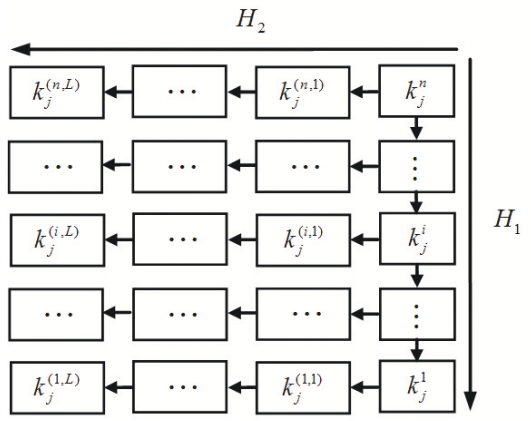

Fig. 1. Two-dimensional key chain

it is natural to let powerful H-sensors serve as cluster heads and form clusters around them [13]. The formation of clusters in HSNs is as follows: Each L-sensor selects an H-sensor whose Hello message has the best signal strength as its cluster head. Simultaneously each L-sensor also records other H-sensors from which it has received Hello messages, and these H-sensors will serve as backup cluster heads in the case that the cluster head fails. In a cluster, the cluster head can communicate with all L-sensors directly, but an L-sensor may need one or more hops to communicate with its cluster head. Cluster heads, which are farther away from the Base Station (BS), can communicate with the BS through hop-by-hop with the help of neighboring cluster heads. In CSS-SH, we make use of the following assumptions:

1. Nodes to be deployed in the target field are not mobile i.e., the heterogeneous sensor network is static.

2. Only a limited number of nodes may be compromised by an attacker during the short time period of the direct key establishment phase [14].

3. BS will not be compromised by an attacker.

CSS-SH has three phases: key predistribution, shared key establishment, and path key establishment.

A. Two-dimensional backward key chain

In [11], a two-dimensional backward key chain is constructed (see figure 1). For a two-dimensional backward key chain C_j, if the key $k_j^{i_1}$ is known, the key $k_j^{i_2}(i_2 \leq i_1)$, the generation key $g_j^{i_2}$ and the first key $k_j^{(i_2,0)}$ of the second dimensional key chain can be calculated as follows, respectively: $k_j^{i_2} = H_1^{i_1-i_2}(k_j^{i_1})$, $g_j^{i_2} = H_2(k_j^{i_2},0)$ and $k_j^{(i_2,0)} = H_2(k_j^{i_2},1)$, where H_1 and H_2 are two independent hash functions. So, the key $k_j^{(i_1,i_2)}$ $(l_2 \geq 1)$ can be computed as follows:

$$k_j^{(i_2,l_2)} = H_2^{l_2}(g_j^{i_2}, k_j^{(i_2,0)}),\ when\ l_2 \geq 1 \tag{1}$$

If the keys $k_j^{i_1}$ and $k_j^{(i_2,l_1)}$ are known, the key $k_j^{(i_2,l_2)}$ $(l_1 < l_2)$can be computed using the following equation:

$$k_j^{(i_2,l_2)} = H_2^{l_2-l_1}(g_j^{i_2}, k_j^{(i_2,l_1)}),\ when\ l_2 > l_1 \tag{2}$$

B. Key pool

The key pool consists of m two-dimensional backward hash key chains, which is divided into n disjoint sub-key pools. A sub-key P^i consists of two parts: a generation key pool P_1^i =$\{k_j^i,\ 1 \leq j \leq m\}$ and an ordinary key pool P_2^i =$\{k_j^{(i,l)},\ 1 \leq j \leq m, 1 \leq l \leq L\}$.

C. Key pre-distribution phase

This stage is performed offline before nodes are deployed. BS is predistributed a master key k_{BS} and all keys of the key pool P_1^n. An H-sensor H^i, which will be deployed in the i^{th} phase, is predistributed the following keys: 1. t_1 and t_2 $(t_1 \ll t_2)$ keys that are selected randomly and uniformly from the key pool P_1^i and P_2^i respectively; 2. A unique key k_{H^i-BS} = $H_2(k_{BS},ID_{H^i})$ that is shared with BS (where ID_{H^i} is the identification of H^i). An L-sensor L^i, which will be deployed in the i^{th} phase, is predistributed t_3 $(t_3 \leq t_1)$ keys selected randomly and uniformly from the sub-key pool P_1^i.

D. Shared key establishment phase

In CSS-SH, after shared key establishment ends, any node should save the hashed keys in its key ring. For example, suppose an H-sensor H^i is predistributed two keys $k_{j_1}^j$ and $k_{j_2}^{(i,l)}$. As soon as the shared keys establishment between H^i and other nodes are finished, H^i saves the two following hashed keys: $H_2(k_{j_1}^i,ID_{H^i})$, and $H_2(k_{j_2}^{(i,l)},ID_{H^i})$.

For any two nodes a^{i_1} and b^{i_2} (Without loss of generality, we assume $i_1 \geq i_2$), the shared key between them consists of two parts: 1. x_1 generation keys which come from the generation key pool $P_1^{i_2}$. For example, suppose keys $k_{j_1}^{i_1},...,k_{j_{x_1}}^{i_1}$ are pre-distributed to node a^{i_1}. If $i_1 = i_2$, b^{i_2} saves the following common keys $k_{j_1}^{i_2},...,k_{j_{x_1}}^{i_2}$ with a^{i_1} ; if $i_1 > i_2$, b^{i_2} saves keys $H_2(k_{j_1}^{i_2}, ID_{b^{i_2}}), ..., H_2(k_{x1}^{i_2}, ID_{b^{i_2}})$, and a^{i_1} can calculate these keys by using the methods in section III-A and section III-D; 2. x_2 ordinary keys which come from the ordinary key pool $P_2^{i_2}$. For example, let keys $k_{j_1'}^{(i_2,l_1)},...,k_{j_{x_2}'}^{(i_1,l_{x_2})}$ are predistributed to node a^{i_1}. b^{i_2} saves keys in line with one of the following key lists: a) $k_{j_1'}^{(i_2,l_1)},...,k_{j'_{x_2}}^{(i_2,l_{x_2})}$; b) $H_2(k_{j_1'}^{(i_2,l_1)},ID_{b^{i_2}}),..., H_2(k_{j'_{x_2}}^{(i_2,l_{x_2})},ID_{b^{i_2}})$, a^{i_1} can calculate these keys using the methods in section III-A and section III-D. As a result, if the number of common keys is more than 0, i.e. $x_1 + x_2 \geq 1$, the shared key between a^{i_1} and b^{i_2} is hashed by all common keys.

E. Path key establishment phase

If direct shared key establishment between two H-sensors fails, the procedure of the path key establishment is the same as the schemes [8]-[9]. For example, if direct key establishment between two H-sensors $H_S^{i_1}$ and $H_D^{i_2}$ fails,$H_S^{i_1}$ needs to find a key path from $H_S^{i_1}$ to $H_D^{i_2}$. In the key path, any two adjacent nodes can establish a direct key. Assume that the key path is $H_S^{i_1}$,$H_1^{i_1'}$,..., $H_v^{i_v'}$,$H_D^{i_2}$. $H_S^{i_1}$ generates a random key k and sends it to $H_1^{i_1'}$ using their secure link; $H_1^{i_1'}$ sends the key to $H_2^{i_2'}$ using the secure link between $H_1^{i_1'}$ and $H_2^{i_2'}$, and so

on until $H_D^{i_2}$ receives the key from $H_v^{i_v'}$. The key k is their common key.

If direct shared key establishment between an H-sensor H^{i_1} and an L-sensor L^{i_2} in its cluster fails, H^{i_1} sends to BS a Request Message, which includes one key identification of L^{i_2}, is encrypted by the key $k_{H^{i_1}-BS}$. BS gets the key identification by decrypting the Request Message with the key $k_{H^{i_1}-BS}$, and sends the corresponding key encrypted by $k_{H^{i_1}-BS}$ to H^{i_1} . To reduce the communication overhead, H^{i_1} can collect the identifications of the keys which need to be obtained from BS, and send only one Request Message to BS. BS sends the corresponding keys encrypted by the key $k_{H^{i_1}-BS}$ to H^{i_1} .

IV. PERFORMANCE AND SECURITY ANALYSIS

In this section, we analyze the performance and the security of our scheme, including the probability that an H-sensor can establish a shared key with an L-sensor in its cluster, and the probability that communications between an H-sensor and L-sensors in its cluster can be compromised by an attacker by the information retrieved from the X compromised nodes. For the sake of the analytical convenience, we suppose that nodes are distributed in the HSN randomly, and an attacker captures nodes from the HSN randomly. N_H^i and N_L^i represent H-sensors and L-sensors deployed in the i^{th} phase, respectively. N_{CH}^k and N_{CL}^k represent H-sensors and L-sensors captured in the k^{th} capture, respectively. After the k^{th} capture, the expectation value of H-sensors, which are deployed in the i^{th} phase and are not captured, is:

$$N_{SH}^{(i,k)} = N_{SH}^{(i,k-1)} - \frac{N_{SH}^{(i,k-1)}}{\sum_{i_1=1}^{k} N_{SH}^{(i_1,k-1)}} N_{CH}^k \tag{3}$$

where $i \geq k$ and $N_{SH}^{(i,i-1)} = N_H^i$.

Similarly, after the k^{th} capture, the expectation value of L-sensors, which are deployed in the i^{th} phase and are not captured, is:

$$N_{SL}^{(i,k)} = N_{SL}^{(i,k-1)} - \frac{N_{SL}^{(i,k-1)}}{\sum_{i_1=1}^{k} N_{SL}^{(i_1,k-1)}} N_{CL}^k \tag{4}$$

where $i \geq k$ and $N_{SL}^{(i,i-1)} = N_L^i$.

A. Performance analysis

In CSS-SH, after the shared key establishment phase, keys from the key pool P_1^i saved in a node are hashed. Therefore, the probability that an L-sensor L^{i_1} can establish a shared key with an H-sensor H^{i_1} is influenced by the time that they are deployed. The probability can be calculated using the following equation:

$$P_{HL} = \begin{cases} P_{HL}^1 = \sum_{x=1}^{t_3} \sum_{x_1+x_2=x} P_{(x_1,x_2)}^1, \ when\ i_1 \leq i_2 \\ P_{HL}^2 = \sum_{x_1=1}^{t_3} P_{(x_1)}^2, \ when\ i_1 > i_2 \end{cases} \tag{5}$$

where

$$P_{(x_1,x_2)}^1 = \frac{\binom{m}{x}\binom{x}{x_1}\binom{m-x}{t_1+t_2+t_3-2x}\binom{t_1+t_2+t_3-2x}{t_3-x}\binom{t_1+t_2-x}{t_1-x_1}}{\binom{m}{t_1+t_2}\binom{t_1+t_2}{t_1}\binom{m}{t_3}}$$

, and

$$P_{(x_1)}^2 = \frac{\binom{m}{x_1}\binom{m-x_1}{t_1+t_3-2x_1}\binom{t_1+t_3-2x_1}{t_3-x_1}\binom{m-t_1}{t_2}}{\binom{m}{t_1+t_2}\binom{t_1+t_2}{t_1}\binom{m}{t_3}}$$

Therefore, in the k^{th} deployment phase, the probability that an H-sensor can establish a pairwise key with an L-sensor is:

$$P_{sk} = \sum_{i_1=1}^{k} P_{SH}^{i_1} \cdot \left(P_{SL}^{i_2 \geq i_1} \cdot P_{HL}^1 + \left(1 - P_{SL}^{i_2 \geq i_1}\right) \cdot P_{HL}^2 \right) \tag{6}$$

where

$$P_{SH}^{i_1} = \frac{N_{SH}^{(i_1,k-1)}}{\sum_{i_3=1}^{k} N_{SH}^{(i_3,k-1)}}, P_{SL}^{i_2 \geq i_1} = \frac{\sum_{i_2=i_1}^{k} N_{SL}^{(i_2,k-1)}}{\sum_{i_3=1}^{k} N_{SL}^{(i_3,k-1)}}.$$

B. Security analysis

The probability that a shared key, which is established using x_1 generation keys from the key pool P_1^i and is compromised, can be calculated as follows:

$$P_{x_1}^i = \left(1 - \left(1 - \frac{t_1}{m}\right)^{\sum_{i_3=i}^{k} N_{CH-B}^{i_3}} \cdot \left(1 - \frac{t_3}{m}\right)^{\sum_{i_3=i}^{k} N_{CL-B}^{i_3}}\right)^{x_1} \tag{7}$$

where N_{CH-B}^i and N_{CL-B}^i are the number of H-sensors and L-sensors which are deployed in the i^{th} phase and are captured, respectively. Similarly, the probability that a pairwise key, which is established using x_2 ordinary keys from the key pool $P_{i,2}$ and is compromised, can be calculated as follows:

$$P_{x_2}^i = \left(1 - \left(1 - \frac{t_1}{m} - \frac{t_2}{m \times L}\right)^{N_{CH-B}^i} \cdot \left(1 - \frac{t_1}{m}\right)^{\sum_{i_3=i+1}^{k} N_{CH-B}^{i_3}} \cdot \left(1 - \frac{t_3}{m}\right)^{\sum_{i_3=i}^{k} N_{CL-B}^{i_3}}\right)^{x_2} \tag{8}$$

So, the probability that a pairwise key between an H-sensor and an L-sensor in its cluster is compromised can be calculated as follows:

$$P_r^k = \sum_{i=1}^{k} P_{SH}^{i_1} \left(P_{SL}^{i_1 \geq i} \cdot \sum_{x=1}^{t3} \sum_{x_1+x_2=x} \frac{P_{(x_1,x_2)}^1}{P_{HL}^1} \cdot P_{x_1}^i \cdot P_{x_2}^i + \sum_{i_1=1}^{i-1} P_{SL}^{i_1} \cdot \sum_{x_1=1}^{t3} \frac{P_{(x_1)}^2}{P_{HL}^2} \cdot P_{x_1}^{i_1} \right) / P_{sk} \tag{9}$$

C. Comparisons

In this section, performance and security between our scheme and AP-L, and AP-D, are compared. For the sake of fairness, in AP-L and AP-D, predistribution keys are hashed as soon as the pairwise key establishment ends. The settings of our experiments can be summarized as follows.

1. Deployment area is 600m×600m.

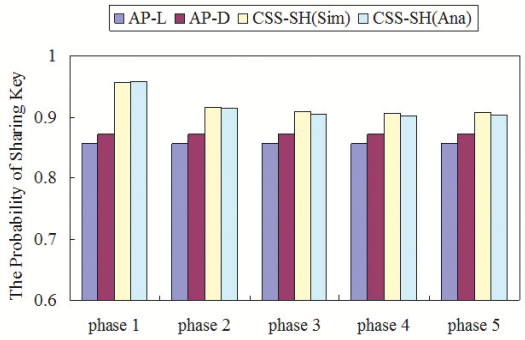

Fig. 2. The probability of sharing key comparisons. In AP-L, the size of the key pool is 8,000. In AP-D, the size of the key pool for L-sensors, namely P_L, is 7,000, and the size of the key pool for H-sensors, namely P_H ($P_H \supset P_L$), is 8,000. In AP-L and AP-D, the number of keys predistributed to an L-sensor and an H-sensor is 30 and 500, respectively.

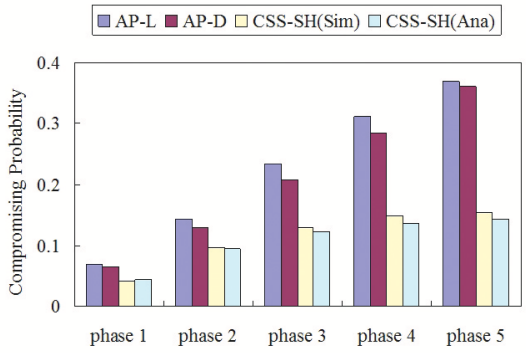

Fig. 3. Resilience comparisons. The parameters are the same as in Figure 2.

2. The number of L-sensors and H-sensors is 5,700 and 300, respectively.

3. Node deployment includes 5 phases. In the first phase, there are 1,900 L-sensors and 100 H-sensors deployed, respectively. In each subsequent phase, it is assumed that there are 950 L-sensors and 50 H-sensors deployed, respectively. There are 950 L-sensors and 50 H-sensors compromised in each phase.

4. The number of key chains is 5,000 (m=5,000), and the length of forward key chains is 100 (L=100).

5. The number of keys predistributed to an L-sensor and an H-sensor are 30 and 500, respectively.

6. During the bootstrapping phase, the number of captured L-sensors and H-sensors is 20 and 2, respectively.

Figure 2 shows that the probability of shared key establishment in the second phase is less than that in the first phase. The larger the deployment phase, the smaller the decline is. For example: from the first phase to the second phase, and from the third phase to the fourth phase, the decline of the probability of shared key establishment is 0.04 and 0.003, respectively. At the same time, compared with scheme AP-L and AP-D, the proba-bility of shared key establishment in CSS-SH is highest.

In AP-D and AP-L, the key pool is fixed. Therefore, increases in the number of captured nodes diminish network resilience. For example, for the scheme AP-D, Figure 3 shows the probability that a shared key is compromised in the first phase and the 5^{th} phase is 0.06 and 0.36, respectively. In our scheme, the sub-key pool of the i^{th} phase and the phase is disjoint, that is, $P_i \cap P_{i'} = \emptyset$ ($i \neq i'$). Therefore, our scheme can improve the performance in continuous secure. As an example, in CSS-SH, the probability that shared keys are compromised in the 5^{th} phase is 0.16.

V. CONCLUSION

In the paper, we proposed a continuous secure scheme for static heterogeneous sensor networks. H-sensors do not need to be equipped with tamper-resistant hardware. Analysis and simulations indicate that the probability that shared keys are compromised drops slightly with time. For example, when the settings of CSS-SH is the same as the section IV-C, in the 5^{th} phase, the probability that shared keys are compromised is only 0.16. Compared with schemes AP-D and AP-L, the continuous secure of CSS-SH scheme is more than double.

REFERENCES

[1] rossbow Technology Inc., www.xbow.com.
[2] K. Piotrowski, P. Langendoerfer, and S. Peter, "How Public Key Cryptography Influences Wireless Sensor Node Lifetime," Proc. 4th ACM SASN 2006.
[3] X. Du, Y. Xiao, M. Guizani, and H.H. Chen, "An effective key management scheme for heterogeneous sensor networks," ad hoc networks, vol. 5, no. 1, pp. 24-34 , 2007.
[4] K. Lu, Y. Qian, M. Guizani, et al., "A Framework for a Distributed Key Management Scheme in Heterogeneous Wireless Sensor Networks," IEEE Trans. On Wireless Communications, vol. 7, no. 2, pp. 639-647, 2008.
[5] Q. Shi, N. Zhang, M. Merabti, et al., "Resource-efficient authentic key establishment in heterogeneous wireless sensor networks," Journal of parallel and distributed computing, vol. 73(2013), pp. 235-249, 2013.
[6] Y. Cheng, and D. P. Agrawal, "An improved key distribution mechanism for large-scale hierarchical wireless sensor networks," Ad Hoc Networks, vol. 5, no. 1, pp. 35-48, 2007.
[7] R. Anderson and M. Kuhn, "Tamper resistance-a cautionary note," Proc. 2nd Usenix Workshop Electronic Commerce, 1996, pp. 1-11.
[8] L. Eschenauer, and V. D. Gligor, "A Key-Management Scheme for Distributed Sensor Networks," in ACM CCS'02 Proceedings, 2002, pp. 243-254.
[9] H. Chan, A. Perring, and D. Song, "Random Key Pre-distribution Schemes for Sensor Networks," in Proc. IEEE Symposium on Security and Privacy, Berkeley, 2003, pp. 197-215.
[10] B. Zhou, S. Li, Q. Li, X. Sun, et al., "An efficient and scalable pairwise key pre-distribution scheme for sensor networks using deployment knowledge," computer communications, vol. 32, no. 1, pp. 124-133, 2009.
[11] S. Li, B. Zhou, J. Dai, et al., "A secure scheme of continuity based on two-dimensional backward hash key chains for sensor networks," IEEE Wireless Communications Letters, vol. 1, no. 5, pp. 416-419, 2012.
[12] C. Blundo, A.D. Santis, A. Herzberg, et al., "Perfectly-secure key distribution for dynamic conferences," Lecture Notes in Computer Science 740 (1993), pp. 471-486, 1993.
[13] X. Du and F. Lin, "Maintaining differentiated coverage in heterogeneous sensor networks," EURASIP Journal on Wireless Communications and Networking, Vol. 4, pp. 565-572, 2005.
[14] S. Zhu, S. Setia, and S. Jajodia, "LEAP+: efficient security mechanisms for large-scale distributed sensor networks," ACM Trans. on Sensor Networks, vol. 2, no. 4, pp. 500-528, 2006.