This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2009 proceedings.

On Attack-Resilient Wireless Sensor Networks with Novel Recovery Strategies

Ka-Shun Hung†, Chun-Fai Law†, King-Shan Lui†, and Yu-Kwong Kwok†‡
†Electrical and Electronic Engineering, The University of Hong Kong, Pokfulam Road, Hong Kong
‡Electrical and Computer Engineering, Colorado State University, Fort Collins, CO 80523-1373, USA

***Abstract*— In a wireless sensor network (WSN), when an adversary physically captures one or more sensor nodes, all the information stored on these nodes may be exposed completely. Consequently, the adversary can use the information to attack the remaining part of the network. In this paper, we investigate the effects of different node capture attack patterns on state-of-the-art key management schemes. We find that a compromised WSN can be made resilient to such attacks by introducing new resources, such as new nodes and new keys. Based on this observation, we propose two recovering strategies, namely, *link replacement strategy* and *node replenishment strategy*, to replace the compromised links and the functions of the compromised region, respectively. Simulation results indicate that our proposed strategies can improve the network resilience of a compromised WSN significantly with a small amount of additional resources.**
***Keywords:* wireless sensor networks, node capturing attacks, leaked keys, broken communication links, recovery, node replacement, link replacement.**

I. Introduction

A typical wireless sensor network (WSN) is usually composed of a large number of small battery-powered devices operated in an autonomous manner [1]. Such devices (also called nodes hereafter) are equipped with relatively limited computational power and wireless networking hardware with a limited transmission range. Capable of working collaboratively, devices in a WSN can achieve many useful functions in a hostile environment, where putting human personnel would lead to grave danger [1]. For instance, in a modern combat situation, a WSN would be useful for various military functions such as remote surveillance, intelligence gathering in a hostile terrain, team assault operations, etc. Indeed, the US Army has an intense recent interests in deploying armored unmanned ground vehicles (UGVs) such as the SWORDS robots [7] for carrying out anti-insurgent operations.

Since a WSN is expected to handle important information in a hostile situation, security is of a prime concern. Specifically, every packet transmitted has to be encrypted and every sensor node has to be authenticated. To support encryption and authentication, an efficient and effective key management scheme is a necessity. Currently, there are a number of key management scheme proposed in WSNs [13].

With the support of an efficient and effective key management scheme, the network itself and the information transmitted can be protected from any external attacker. Unfortunately, recent research has shown that it is highly probable for a sensor node to be captured physically [12]. By physically capturing a sensor node, under a worst case assumption, an attacker can presumably get all the keys and the encryption information contained in the node.

In this paper, we evaluate two commonly adopted key management schemes under two attack patterns. Our observation of the performance of these two key management schemes under two different attack models motivates us to design a novel link replacement strategy which performs key update on the links no matter they are compromised or not. In particular, in closest neighbor attack model which is a more realistic attack model intuitively, we observe that captured nodes are generally clustered. By bounding the regions of a node capture attack, we propose to carry out node replenishment in the severely attacked region to restore the secure connectivity of the regions to other parts of the network. By doing so, the network lifetime can be extended even with the presence of compromised nodes.

The organization of this paper is as follow. In Section II, we briefly describe the related work. In Section III, we describe two node capture attack patterns and evaluate their effects to two common adopted key management schemes in terms of network resilience. Afterwards, we propose two recovery strategies—*Link Replacement Strategy* and *Node Replenishment Strategy* in Section IV. Then, we evaluate and demonstrate the feasibility of these two strategies through extensive simulations in Section V. Finally, we conclude our work with some possible future research directions in Section VI.

II. Related Work

Recently, De *et al.* [8] proposed a model of virus and worm spreading in WSNs. Their works were motivated by the recent spread of a kind of virus known as *cabir* [5] through wireless medium (e.g., Bluetooth) in mobile network. They extended this idea to the node compromising attack in WSNs. They investigated this attack with the use of epidemic theory and tried to develop a defensive strategy specifically targeted for the spreading of virus or worms. Unlike their work, we consider the node capture attack due to physically capturing a node by an attacker. This attack is shown to be highly feasible [12].

To prevent physical node capture attack, the most effective technique is to armor each sensor node with tamper-resistance hardware [3], [18]. However, this technique is generally considered as impracticable due to the high cost and large size of the tamper-resistance hardware. In view of this, Alarifi and Du [2] proposed to use code diversifying technique.

978-1-4244-2948-6/09/$25.00 ©2009 IEEE

They proposed to use different memory locations to store sensitive information for different nodes. By doing so, even if an attacker cracks a sensor node by using the node capture attack suggested in [12] and figures out the location of the sensitive information in the memory of that particular node, such memory location information cannot be used to obtain the sensitive information of other nodes in the network. This technique can greatly lengthen the time an attacker needed to compromise a node. Unfortunately, an attacker can still compromise the whole network if he/she is given enough time.

On the other hand, Song *et al.* [17] admitted the seriousness of the node capture attack to WSNs. They pointed out the importance of locating the compromise node in the network. They suggested that most of the current techniques only takes a reactive approach, i.e., to detect the misbehavior caused by the compromised nodes. The rationale is that we do not know which nodes are actually compromised unless the compromised nodes misbehave. By contrast, Song *et al.* proposed a proactive approach to detect node capture. According to their approach, each node tries to measure the differences between the location information of its neighbor node (and also the neighbor list information) before and after they disappear and then reappear. By doing so, a node can actively estimate whether its neighbor node has been compromised.

Currently, most of the proposed work suggested that a compromised node has to be isolated from other network components after they are detected. However, Strasser and Vogt [19] suggested that the isolation approach leads to a wastage of resources because a compromised node can still function normally after it is recovered. They discussed some guidelines for node recovery and proposed a method to reprogram the compromised nodes. As a result, the recovered nodes can be restarted in the "clean" state. However, in [19], they also admitted that this recovery mechanism is quite complicated.

Yang and Cardei [20] and Chatzigiannakis *et al.* [6] proposed the concept of node redeployment. The main thrust of their approaches is to redeploy the nodes in the network to extend the lifetime, enhance delivery ratio, etc. This idea is similar to our node replenishment idea proposed in this paper. However, our work focuses on the feasibility and the effectiveness of node replenishment to replace the functions of the compromised nodes and repair some of the compromised links by link replacement strategy whenever possible.

III. Compromising a WSN

Currently, most existing key management schemes [4], [9], [11], [13], [14], [16] assume a random attack pattern in the sensor network security analysis. However, this may not be realistic in practice. Specifically, sensor nodes are supposed to be largely distributed over a large geographical area. In this section, two commonly adopted key management schemes are first evaluated. Then, we investigate the differences between the *random attack* and the *closest neighbor attack* patterns in terms of network resilience.

A. Key Pre-distribution Schemes

Key pre-distribution schemes refer to key management schemes in which keys are randomly generated and distributed to the sensor nodes before they are deployed. Two commonly adopted key management schemes are the schemes proposed by Eschenauer *et al.* [11] (we refer to this scheme as *basic scheme* in this paper) and Du *et al.* [9] (we refer to this scheme as *deployment knowledge scheme* in this paper), respectively.

1) Basic Scheme: Based on random graph theory [10], Eschenauer *et al.* assumed that a random graph is formed in the sensor network. They proposed to install a random set of keys drawn from a very large key pool into each sensor node. A set of keys is also known as a key ring. Any two neighbors in the network are assumed to be securely connected if they have a key in common in both key rings. Eschenauer *et al.* showed mathematically that a certain connectivity and an acceptable level of network resilience can be obtained by maintaining a certain ratio of key ring size and key pool size.

2) Deployment Knowledge Scheme: Intuitively, the larger the ratio of key ring size and key pool size, the higher will be the connectivity. However, this will also reduce the level of network resilience. In view of this, Du *et al.* [9] exploited the deployment knowledge of the sensor nodes to reduce the ratio needed to maintain the desired connectivity. By knowing the approximate deployment location of each sensor, the whole key pool can be divided into several smaller sub key pools. Each sub key pool is then assigned to a region. Nodes in nearby region can then generate a smaller key ring by drawing the keys randomly from similar sub key pools.

B. Attack Patterns

Only two attack patterns are considered in this paper due to their large discrepancy in performances which motivates our work in this paper. The reader is referred to [15] for more details. The two attack patterns considered are:

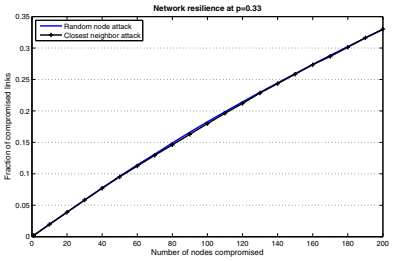
Random attack pattern: This is the most widely used attack pattern in WSN research. Nodes are randomly picked by an attacker to compromise. This pattern can illustrate the network resilience when the keys are randomly pre-distributed in the nodes prior to deployment. However, this attack pattern may not be realistic as it may not be possible for an attacker to randomly pick sensors to compromise.

Closest neighbor attack pattern: When an attacker finishes compromising a sensor node, he/she may try to find another sensor node which is the closest to the current compromised node. This attack pattern is realistic and occurs frequently in a practical scenario.

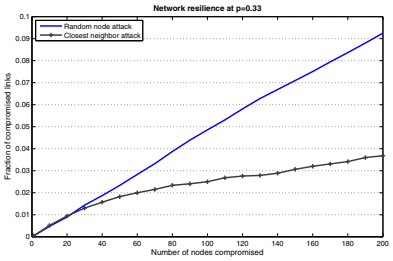
To evaluate the resilience of our framework against node capture attack, we measure the fraction of communication links compromised when x nodes are captured. This method of measuring **network resilience** is generally adopted [11], [9], [14]. Figure 1(a) shows the network resilience of the *basic scheme* under the above attack patterns, while Figure 1(b) shows that of the *deployment knowledge scheme*. The results of these two figures are obtained using the simulation settings discussed in Section 4.4 of [15] (similar settings are also used

in the simulations discussed in Section V in this paper). In Figure 1(a), we can see that the *basic scheme* achieves similar performance under two different attack models. This is mainly due to the fact that keys are evenly distributed among all the nodes in the network.

In Figure 1(b), the deployment knowledge scheme exhibits a very good network resilience against the closest neighbor attack by dividing the key pool in sub-key pools so that different regions have different sub-key pools. This limits the effect of attack within the attack region.



(a) The basic scheme.



(b) The deployment knowledge scheme.

Fig. 1. Illustrations of Network resilience of various scheme under different attack patterns at $p = 0.33$.

IV. Recovery Strategies

In this section, we first briefly discuss the motivations and assumptions of our recovery strategies. Then, we describe these two proposed strategies – *Link Replacement Strategy* and *Node Replenishment Strategy* in detail.

A. Motivation

The results shown in Section III are very interesting in three different aspects. Firstly, the use of different attack patterns does have different effects to the network resilience based on the key management schemes adopted. Secondly, the use of the *basic scheme* generally shows poor performance compared to all other key management schemes in terms of network resilience as it has a larger key ring size per node. Thus, a compromised node will leak out the key information which can be used to attack other parts of networks even if they are geographically far apart. This motivates us to think of a technique to update the key of the whole network so that the remaining part of the network can still be protected even if some of the keys leak out. Thirdly, the use of *deployment knowledge scheme* shows a firewall effect which limits the attack to the attack region, especially, under the closest neighbor attack pattern. However, we argue that the influence of this firewall effect, in fact, is a double edge sword. On one hand, it prevents the effect of node capture from propagating to the whole network. On the other hand, the nodes in the attack region are totally sacrificed. This motivates us to design a technique to replace the functions of the compromised nodes in the attack region so as to restore the normal operation of the network.

B. Assumptions

1. Sensor nodes are randomly distributed in a large area in which secure links are formed between those neighbors with a common key.

2. After a node is captured, the sensitive information inside is leaked out and can be used to attack other parts of the network even if other nodes are geographically far apart.

3. In this paper, since we focus on the effect of applying the recovery strategies to the whole network and the attack region, we assume that the attack model and attack region can be figured out by using some other means [17].

C. Link Replacement Strategy

The main idea of link replacement strategy is to do key update on three neighbor nodes in which secure links are mutually formed between them, so that some of the compromised links can be recovered. In the following subsection, we describe the link replacement procedures in detail.

1) Link Replacement Procedures: Suppose there are three nodes A, B, and C as shown in Figure 2. The links AB, BC, and AC are secured by keys K_c, K_a, and K_b, respectively. To perform link replacement, each link is chosen in turn. Suppose the link BC is chosen in the figure and we need to replace key K_a with another key $K_{a''}$.

Afterwards, the sensor node at the other end of the chosen link is responsible to generate a secret and send the generated secret to the adjacent nodes of the chosen link securely through the links. As can be seen from step 2 of Figure 2, node A is responsible to generate a secret $K_{a'}$ and send $K_{a'}$ to node B and node C through the links AB and AC which are secured by keys K_c and K_b, respectively.

Then, the two nodes adjacent to the chosen link generate the new key of the chosen link independently. The new key can be generated by hashing the original key with that of the new secret received. Now, node B, and C generate the same new key $K_{a''} = hash(K_a, K_{a'})$, respectively. Thereafter, $K_{a''}$ is used to secure the link BC. Finally, the remaining two keys K_b and K_c used to secure the links AC and AB, respectively, can be replaced by following the same procedures described above.

2) Security Analysis: In this section, we present our security analysis on the replacement procedure by considering three cases.

Case 1: None of the links is compromised. Because all three links are still secured, there is no change in the number of the compromised links in the network.

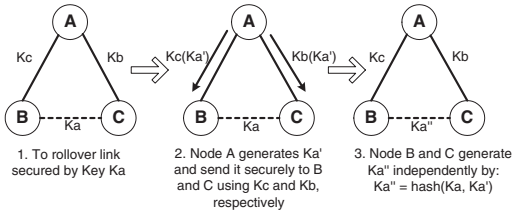


Fig. 2. Link replacement procedure.

Case 2: Either one of the three links is compromised. In this case, the only compromised link is recovered to a normal link. This situation involves two parts. In the first part, the chosen link is a normal link and the newly generated secret must pass through the compromised link. Since the transmission of a secret through a compromised link is supposed to be leaked out, the attacker knows the newly generated secret used to do key update. However, since the original key is still secured, hashing the original key with a leaked secret will still lead to a secure new key. In the second part, the chosen link is a compromised link and the secret passes through two normal links. Because the secret passes through two normal links, the attacker does not know the newly generated secret. As a result, hashing a secure secret and a compromised key will result in a secure new key. By doing so, the compromised link is replaced by a normal link.

Case 3: Two or more links are compromised. Since two or more links are compromised, the secret generated must pass through the compromised link. As a result, the newly generated key will remain in the same status as that of the original key.

3) Mathematical Analysis: Cases 1, 2, and 3 consist of one, three, and four combinations, respectively. From the security analysis in Section IV-C.2, cases 1 and 3 have no effect on the total number of the compromised links in the network, while the case 2 can actually reduce the total number of compromised links. As a result, we are particularly interested in the probability of the existence of the case 2 if there exist three neighbors forming mutually secure links. We denote the case in which three neighbors forming mutually secure links as a *triangle structure* in our paper. In a dense sensor network with an appropriate secure connectivity, there exists a large number of *triangle structures*.

In the attack region, suppose the total number of the compromised links is C, the total number of links is T. With these two parameters, we can assume that a link has a probability of $\frac{C}{T}$ to be compromised in that region. With the existence of a triangle structure, the probability of the existence of case 2 discussed in Section IV-C.2 can be approximated by Equation (1).

$$P(Case_2) = 3 \times (\frac{C}{T}) \times (1 - \frac{C}{T})^2 \quad (1)$$

D. Node Replenishment Strategy

The main idea of the node replenishment strategy is to replace the functions of the compromised nodes as well as the dead nodes in a WSN by deploying some new nodes into the network. In implementing the node replenishment strategy, there are several issues to be considered.

Where should the new nodes be deployed?
With the assumption that the attack pattern and the attack region can be detected, it is possible to limit the target region for node replenishment to a rectangle as shown in Figure 3. The rectangular area can be easily derived with the knowledge of estimated compromised nodes' locations. The new nodes can then be randomly distributed over the region.

How can the newly deployed nodes interact with the existing nodes in the network?
Basically, the key pool of the newly deployed nodes contains both new generated keys and the old keys. To connect the existing nodes in the network, only the old keys can be used. It is possible for a newly deployed node to connect to the normal existing nodes as shown in Figure 3 or the compromised nodes which are not shown in the figure. It is worthwhile to state that if the newly deployed node connects to another newly deployed node or a normal existing node, a normal link is formed. However, if it connects to a compromised node, a compromise link is formed.

In the first case, if a newly deployed node connects to a normal node, it is possible for the newly established link to be secure or compromised. The probability of the existence of a secure (or a compromised link) depends on the ratio of the compromised links in the region. As a result, it will not affect the ratio of compromised links in the region. However, as discussed in Section IV-C, link replacement strategy will be performed together with the link addition process. Consequently, the addition of the new link can help reduce the ratio of compromised links in the region. In the second case, the newly established links are supposed to be compromised if a new node connects to a compromised node. This will increase the ratio of the compromised links in the region. With the assumption that the amount of compromised nodes are small, the increase in the number of compromised links due to this reason is expected to be acceptable.

How can the newly deployed nodes interact among themselves?
A newly deployed node can only use newly generated keys in the key pool to connect other newly deployed nodes in the region as the newly generated keys do not present in the existing nodes, so it is impossible to compromise these new keys before new nodes are deployed. In other words, all the new links formed are guaranteed to be secured.

What should be the appropriate ratio between the new keys and the old keys?
The ratio between new keys and old keys determines the number of different types of new links. The higher this ratio, the larger the number of new links formed between new nodes. These links are resilient to the previously captured nodes. However, the higher this ratio, the smaller amount of links formed between the new nodes and old nodes. This violates our objective to extend the lifetime of the existing network. We believe that finding an optimal ratio is an important research

problem, which is outside the scope of our present paper. In the following, we assume that this ratio is $1:1$.

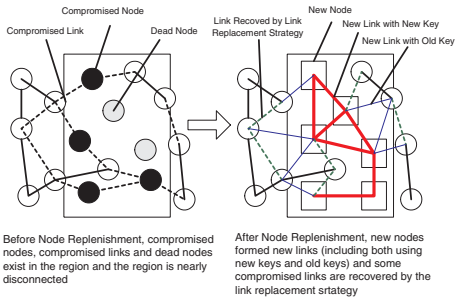


Fig. 3. Node replenishment strategy.

1) Mathematical Analysis: Table I lists the notation used in this section.

TABLE I
NOTATION.

Symbol	Description
N_r	Number of nodes in the region
N_c	Number of compromised nodes in the region
N_n	Number of new nodes in the region
C_r	Number of compromised links in the region
C_n	Number of compromised links after link addition
$C_{n'}$	Number of compromised links after link addition and link replacement
T_r	Number of links in the region
T_n	Number of links after link addition
$T_{n'}$	Number of links after link addition and link replacement
p_o	Secure connectivity ratio using old keys
p_n	Secure connectivity ratio using new keys
d	Expected number of neighbors (Degree) after the node replenishment

Equation (2) illustrates the compromised links ratio in the region after the link addition process is completed. The main idea is to recalculate the total number of compromised links and total number of links after node replenishment, respectively. These two kinds of links can be estimated by summing the old existing links and the new links formed. In recalculating the total number of links (i.e., the denominator of Equation (2)). The term $p_o \times d \times N_n \times \frac{N_r}{N_r+N_n}$ represents the newly established links to the old nodes which is affected by the density of old nodes, the number of new nodes deployed and also the secure connectivity that can be brought by using the old keys. The term $p_o \times d \times N_n \times \frac{N_n}{N_r+N_n}$ represents the newly established links within the new nodes themselves.

On the other hand, in recalculating the total number of compromised links (i.e., the nominator of Equation (2)) $(p_o \times d \times N_n)(\frac{N_c}{N_r+N_n})$ represents the situation in which a new node connects to a compromised node. If a new node connects to a normal node, there is a probability $\frac{C_r}{T_r}$ that the new link formed has already been compromised. The number of compromised links formed due to this case is represented by $(p_o \times d \times N_n)(\frac{N_r-N_c}{N_r+N_n} \times \frac{C_r}{T_r})$.

$$\frac{C_n}{T_n} = \frac{C_r + (p_o \times d \times N_n)(\frac{N_c}{N_r+N_n} + \frac{N_r-N_c}{N_r+N_n} \times \frac{C_r}{T_r})}{T_r + ((p_o \times \frac{N_r}{N_r+N_n}) + (p_n \times \frac{N_n}{N_r+N_n})) \times d \times N_n} \quad (2)$$

After new links are formed with both new nodes and existing nodes (no matter compromised or not), new compromised links ratio $\frac{C_n}{T_n}$ is formed based on Equation (2). At this stage, the link replacement strategy discussed in Section IV-D will then be performed. By applying Equation (1) and Equation (2), the compromised links ratio in the region after the whole node replenishment process can be estimated by Equation (3).

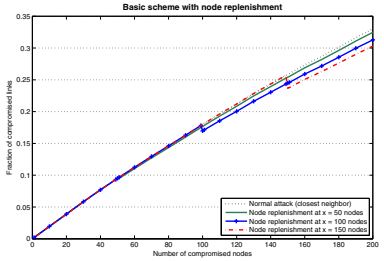
$$\frac{C_{n'}}{T_{n'}} = (1 - P(Case_2)) \times \frac{C_n}{T_n} \quad (3)$$

V. Simulation Results

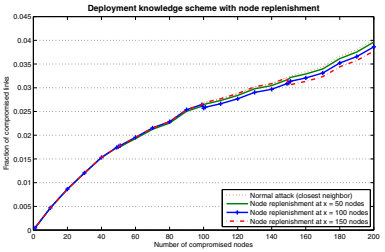
In our simulation, 10000 nodes are generated in a region of 1000 units $\times 1000$ units. The transmission range is 40 units. The key pool size is 100000. The key ring sizes are 200 for *basic scheme* and 46 for *deployment knowledge scheme*, respectively. These two key ring size settings correspond to the secure connectivity of 0.33 in [14], [15]. For these newly deployed nodes, 50% of the keys are new keys.

A. Network-Wide Performance

To study the effect of node replenishment, we measure the network resilience of a key management scheme before and after adding new nodes. In our simulations, $77, 195, 296$ nodes are newly added in the compromised region upon $50, 100, 150$ nodes are compromised, respectively. We can see that when more nodes are compromised, we need to deploy more new nodes as the compromised region is larger in both schemes. Figure 4(a) illustrates the change of network resilience at different attack intervals for the *basic scheme*, while Figure 4(b) shows the change of network resilience as a measure of node replenishment for the *deployment knowledge scheme*. There is a drop in the fraction of compromised links when new nodes are replenished in the compromised region in both figures.



(a) The basic scheme.



(b) The deployment knowledge scheme.

Fig. 4. Performance of node replenishment applied to both schemes.

TABLE II

PERFORMANCE OF THE NODE REPLENISHMENT SCHEME IN THE WHOLE NETWORK.

Basic Scheme		
Replenishment interval	No. of nodes added	Improvement
50	77	0%
100	195	4.72%
150	296	7.93%
Deployment Knowledge Scheme		
Replenishment interval	No. of nodes added	Improvement
50	64	0%
100	168	2.87%
150	243	5.45%

TABLE III

PERFORMANCE OF THE NODE REPLENISHMENT SCHEME IN THE TARGET REGION.

Basic Scheme			
Description	Original	Simulation	Analytical
Compromised Links	154	144	203
Total Links	911	9393	9338
Compromised Ratio	0.169	0.0153	0.0217
Improvement	N/A	10.0	7.79
Deployment Knowledge Scheme			
Description	Original	Simulation	Analytical
Compromised Links	462	565	607
Total Links	931	3265	4087
Compromised Ratio	0.496	0.1730	0.1485
Improvement	N/A	1.87	2.34

On the other hand, Table II illustrates the improvements achieved by our proposed node replenishment strategy for both schemes. Both schemes show a certain percentage of improvement with the use of our strategy. However, the improvement of *deployment knowledge scheme* is less significant than that of the *basic scheme*. This is because the nodes in the vicinity of the compromised area may contain keys from the similar portion in the key pool. Consequently, the link replacement continues to use those compromised keys.

B. Target Region Performance

In this section, we investigate the network resilience in the region where new nodes are deployed. In our simulations, we specifically consider the time durations before and after new nodes are deployed when 100 nodes are compromised and neglect the links connected to the compromised nodes. For the *basic scheme*, we use the following settings obtained from simulation after 100 nodes are compromised for analytical calculations: p_o is 0.0952, p_n is 0.995, d is 83, N_r and N_n are 195. For the *deployment knowledge scheme*, the following settings are used: p_o is 0.418, p_n is 0.273, d is 85, N_r and N_n are 168.

Table III illustrates the simulation and mathematical analysis results on both *basic scheme* and *deployment knowledge scheme*. Both sets of results indicate that the node replenishment schemes can achieve significant improvement in the region in which the improvement in *basic scheme* is larger than that of *deployment knowledge scheme*.

VI. CONCLUSIONS AND FUTURE WORK

We considered two node capture attack patterns and summarized their corresponding characteristics. We then proposed two novel recovery strategies—Link Replacement Strategy and Node Replenishment Strategy. We demonstrated the effectiveness of our proposed strategies through both mathematically analysis and simulation. Our proposed strategies achieve significant improvement in terms of network resilience. We assumed that it is possible to detect the compromised region based on the characteristics of the node capture attack pattern. However, how this can be done in practice remains as one of the most challenging future work. Furthermore, the problem of finding an optimal old key to new key ratio based on different application scenarios is another interesting research problem.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol 11, Aug 2002, pp. 6 – 28.

[2] A. Alarifi and W. Du, "Diversify sensor nodes to improve resilience against node compromise," *Proc. ACM SASN 2006*.

[3] R. Anderson and M. Kuhn, "Tamper resistance: a cautionary note," *Proc. 2nd USENIX Workshop on Electronic Commerce*, vol. 2, pp. 1-11, Nov. 1996.

[4] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," *Proc. IEEE Symposium on Security and Privacy*, pp. 197–213, May 2003.

[5] Cabir, *http://www.f-secure.com/v-descs/cabir.shtml*, 2008.

[6] I. Chatzigiannakis, A. Kinalis, and S. Nikoletseasm, "An adaptive power conservation scheme for heterogeneous wireless sensor networks with node redeployment," *Proc. SPAA 2007*.

[7] D. Crane, "Armed/Weaponized Infantry Robots for Urban Warfare and Counterinsurgency Ops," *DefenseReview.Com*, Dec. 2006, http://www.defensereview.com/article657.html.

[8] P. De, Y. Liu, and S. K. Das, "Modeling Node Compromise Spread in Wireless Sensor Networks Using Epidemic Theory," *Proc. IEEE WoWMoM 2006*.

[9] W. Du, J. Deng, Y. S. Han, S. Chen, and P. K. Varshney, "A key management scheme for wireless sensor networks using deployment knowledge," *Proc. INFOCOM 2004*, vol. 1, Mar. 2004.

[10] P. Erdos and A. Renyi, "On the evolution of random graph," *Publ. Math. Inst. Hung. Acad. Sci.*, 5, pp. 17-61, 1960.

[11] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," *Proc. 9th ACM Conference on Computer and Communications Security*, pp. 41–47, Nov. 2002.

[12] C. Hartung, J. Balasalle, R. Han, "Node compromise in sensor networks: the need for secure systems," *Technical Report CU-CS-990-05*, Department of Computer Science, University of Colorado at Boulder, Jan. 2005.

[13] Y.-K. Kwok, "Key management in wireless sensor networks," in *Security in Distributed and Networking Systems*, Yang Xiao and Yi Pan (eds.), World Scientific Publishing Co., 2007.

[14] C.-F. Law, K.-S. Hung, and Y.-K. Kwok, "A Novel Key Redistribution Scheme in Wireless Sensor Networks," *Proc. IEEE ICC 2007*.

[15] C.-F. Law, *Design and Evaluation of Key Redistribution Mechanisms in Wireless Sensor Networks,* M.Phil Thesis, The University of Hong Kong, Sept. 2007.

[16] 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, pp. 41–77, Feb. 2005.

[17] H. Song, L. Xie, S. Zhu, and G. Cao, "Sensor node compromise detection: the location perspective," *Proc. IWCMC 2007*.

[18] F. Stajano and R. Anderson, "The resurrecting duckling: security issues for ad-hoc wireless networks," *Proc. 7th Int't Workshop Security Protocols*, pp. 172–194, Apr. 1999.

[19] M. Strasser and H. Vogt, "Autonomous and distributed node recovery in wireless sensor networks," *Proc. ACM SASN 2006*.

[20] Y. Yang and M. Cardei, "Movement-assisted sensor redeployment scheme for network lifetime increase," *Proc. MSWiM 2007*.