

Practical Access Control Protocols (PACPs) for Wireless Sensor Networks: Analysis of Security Pitfalls

Mangal Sain, Hyun Ho Kim, Hoon Jae Lee

Department of Information Engineering, Dongseo University, Busan, South Korea

mangalsain1@gmail.com, feei_@naver.com, hjlee@dongseo.ac.kr

Abstract— Access control protocol is always highly required in wireless sensor networks (WSNs), where low cost sensors are deployed for obtaining the environments sensitive data from the real applications. In this regards, very recently, Lee et al proposed practical access control protocols (also called as PACPs) for WSNs (published in IEEE TCE) and claimed that their PACPs are practical and secure for the applications. However, a single security loophole can become a big menace to the application if possible security threats are not considered properly while designing the access control protocol. Therefore, this paper identifies that PACPs have security pitfalls and new node addition may not be highly scalable for practical (real-world) applications. Hence, PACPs have inherent security flaws, and may not be practical for the WSNs.

Keywords— Access control protocol, authentication, key establishment, wireless sensor networks

I. INTRODUCTION

Wireless sensor networks (WSNs) are known as novel and intelligent systems, and are continuously deploying in wide range of real-world applications (military, healthcare, smart building, security systems, etc) [1]. WSNs consist of number of inexpensive sensor's devices, which contain limited resources (such as, low processing unit, low bandwidth, limited power (i.e., AA batteries), and less memory). In addition, sensors are small in size, and are integrated with a sensing unit and wireless communication capabilities (i.e., a wireless transmitter and receiver). These sensor nodes are deployed in a wide area for performing their intended task efficiently. Due to the novelties of WSNs such as, large scale deployment, resource scarcity and wireless communication nature makes them vulnerable to various attacks. It is possible that an adversary can introduce the malicious nodes into the network and may disturb the network functionality. However, to protect WSNs from adversaries and maintain the network working continuously (life-time), security mechanisms (e.g., access control [2] [3]) are highly desirable for the applications.

In 2009, Kim and Lee proposed an enhanced novel access control protocol (ENACP) which exploits the hash-chain approach and performs the node authentication and key establishment [4]. Unfortunately, Zeng et al [5] and Shen et al [6] demonstrated that ENACP has natural design flaws and vulnerable to many attacks. Very recently (in 2012), Lee et al

pointed out that ENACP is susceptible to message forgery and new node masquerade attacks, and proposed practical access control protocols (also known as PACPs) for WSNs [7]. PACPs consist of two sub-schemes, namely, secure PACP (secPACP) and memory-efficient PACP (ePACP). Moreover, authors claimed that PACPs are secure against many attacks and very practical for the real WSNs.

However, in this paper we demonstrate that PACPs are not secure against message replay attack, Sybil attack and impersonation attack. More importantly, we will show that the new node addition is very limited (i.e., only for certain nodes) and hence, PACPs are not highly scalable. Next section will briefly review the PACPs [7].

II. REVIEW OF PACPS[7]

PACPs have two variant, namely, *secPACP* and *ePACP*.

A. *secPACP (secure PACP)*: It is composed of three phases: initialization, authentication and key establishment, and new node addition.

1) *Initialization phase*: This phase is performed off-line by the base station (BS); it generates a large key space (*LKS*), key identifiers, and identities (IDs) for all sensor nodes (i.e., *N* sensor nodes). BS randomly chooses *Q* nodes for the initial deployment (or network). Thereafter, BS randomly picks one secret key and *m* keys from *LKS* for each node and computes an authentication set (*AS*) (i.e., set of hash values, and their identifiers). Finally, BS installs a secret key and *AS* into the nodes, which are selected of the network deployment. *More general example*, BS randomly chooses K_X and $\{K_{Ri}\}_{i \in \{1,2,\dots,x\}}$ from *LKS* for the node *X*. Then, BS computes $AS_X = \{(HID_i, h(ID_X || K_{Ri}))\}_{i \in \{1,2,\dots,x\}}$ for node *X*. Here, HID_i means the owner of secret key K_{Ri} . Thereafter, BS installs K_X and AS_X into the node *X*. Now sensors are ready for the deployment.

2) *Authentication and key establishment phase*: Assume that two nodes (e.g., node *A* and node *B*) are neighbors and each node recognizes the identities of its neighboring nodes using some beaconing technique which includes the node identity in the beacons. If node *A* shares $h(ID_A || K_B)$ with node *B*, then two nodes (*A* and *B*) start key establishment as follows.

i. Node *A* generates a random integer t_A , and computes the

- point $N_A = t_A P = (N_{x_A}, N_{y_A})$ over the elliptic curve E and $S_A = h(ID_A || N_{x_A} || h(ID_A || K_B))$. Now, it (Node A) broadcasts ID_A , N_A , and S_A .
- ii. After receiving the broadcasted message from the node A, node B checks whether $h(ID_B || N_{x_A} || h(ID_B || K_A))$ is in $AS_B = \{(HID_i, h(ID_B || K_{Ri}))\}_{i \in \{1, 2, \dots, z\}}$ or not. If it is not true then aborts the system. Otherwise, node B verifies $h(ID_A || N_{x_A} || h(ID_A || K_B)) = S_A$ with its own key K_B . If S_A is verified then node B assured that N_A is generated by a legal node who knows the $h(ID_A || K_B)$. After that, node B generates a random integer t_B and computes $N_B = t_B P = (N_{x_B}, N_{y_B})$ and $S_B = h(ID_B || N_{x_B} || h(ID_B || K_A))$. And it broadcasts ID_B , N_B , and S_B .
 - iii. Upon receiving the broadcasted message from the node B, node A checks $h(ID_B || N_{x_B} || h(ID_B || K_A)) = S_B$ with its own key K_A . If S_B is verified then node A assured that N_B is generated by a legal node who knows the $h(ID_B || K_A)$. Thereafter, node A computes $SK_{AB} = t_A N_B = (SK_{x_{AB}}, SK_{y_{AB}})$ and $Z_A = h(ID_A || SK_{x_{AB}} || h(ID_A || K_B))$, and broadcasts Z_A .
 - iv. Node B computes $SK_{AB} = t_B N_A = (SK_{x_{AB}}, SK_{y_{AB}})$ and checks $h(ID_A || SK_{x_{AB}} || h(ID_B || K_A)) = Z_A$. If it is true, then node B approves SK_{AB} . Now node B computes $Z_B = h(ID_B || SK_{x_{AB}} || h(ID_A || K_B))$ and broadcasts it to the node A.
 - v. Finally, node A checks $h(ID_B || SK_{x_{AB}} || h(ID_A || K_B)) = Z_B$. If it holds, then node A also approves SK_{AB} .
- The authentication and key establishment phase of *secPACP* is shown in Fig. 1.

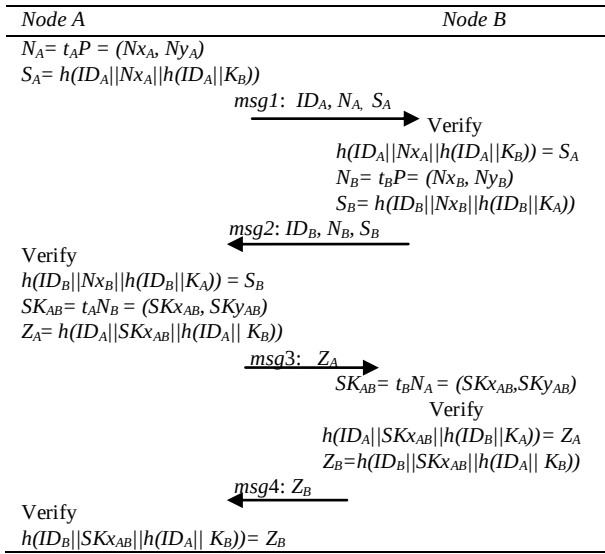


Figure 1. secPACP: Authentication and key establishment phase

3) *Node addition phase*: This phase is invoked when a new node is entering into the existing network. First, BS assigned an identity to the new node (ID_{Q+1}) and also preloads secret key K_{Q+1} and $AS_{Q+1} = \{(HID_i, h(ID_{Q+1} || K_{Ri}))\}_{i \in \{1, 2, \dots, z\}}$. Thereafter, new node will perform the authentication and key establishment phase as shown in Fig. 1, and becomes the legal member of the network.

B. ePACP (memory-efficient PACP): It is composed of two phases, namely, initialization, and authentication and key establishment. This subsection reviews *ePACP*, which is a variant of *secPACP* except the initialization phase.

1) *Initialization phase*: This phase performed offline by the base station (BS); it generates a large key space (LKS), key identifiers, and identities for all N sensor nodes. BS randomly chooses Q nodes for the initial network deployment. Now it is assumed that the identities of all nodes are in a circular order (i.e., the last identity is equal to the first identity). Therefore, each sensor node has its inner nodes and outer nodes in circular order. The number of all candidate node is Q' ($Q \leq Q' \leq LKS$), we describe the *inner* nodes of node X as $\{ID_{Yi}\}_{(X < Yi \leq X + \lceil Q'/2 \rceil)}$ and the other nodes are represented as the *outer* nodes of node X .

Thereafter, BS randomly chooses one secret key from the large key space (LKS) and installs it into the each node. Then, it (BS) chooses m keys from LKS for each sensor's inner nodes; derives an authentication set (AS); and finally, installs AS into its corresponding sensor node. For example, BS randomly chooses K_X and $\{K_{Ri}\}_{i \in \{1, 2, \dots, x\}}$ from LKS for node X . Here X is a node, and K_{Ri} are randomly selected secret keys for node X 's inner nodes. For node X , BS computes $AS_X = \{(HID_i, h(ID_X || K_{Ri}))\}_{i \in \{1, 2, \dots, z\}}$, here HID_i means the owner of secret key K_{Ri} . Thereafter, BS installs K_X and AS_X into node X .

2) *Authentication and key establishment phase*: Assume that two nodes (e.g., node A and B) are neighbors and each node recognizes the identities of its neighboring nodes using some beaconing technique which includes the node identity in beacons. If sensor node B is an inner node of node A, then A starts the pairwise key establishment with node B, otherwise, node B starts. The authentication and key establishment phase is same as in *secPACP* (refer to the *secPACP* authentication and key establishment phase). However, the flow of *ePACP* is depicted in Fig. 2.

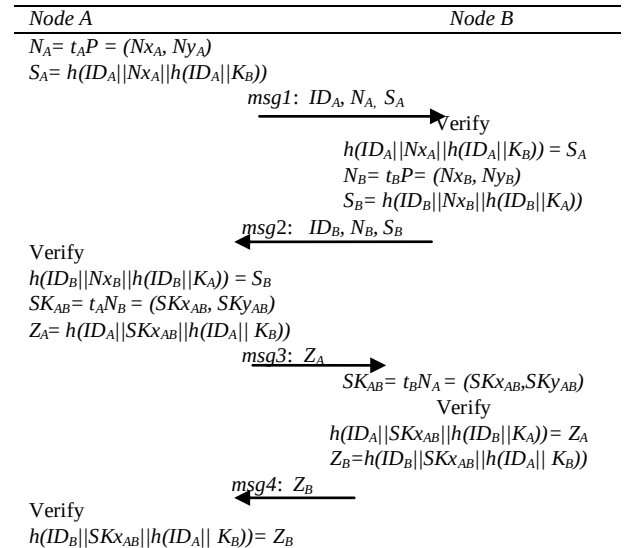


Figure 2. ePACP: Authentication and key establishment phase

Next section will demonstrate the security pitfalls in PACPs.

III. ANALYSIS OF SECURITY PITFALLS IN PACPs

Indeed, PACPs are strong against eavesdropping, message forgery attack, and new node masquerade attack. However, a single loophole can become a big danger to the network, if all possible security threats are not considered (with their destructive impact) while designing the protocol. In this section we present the inherent PACPs security pitfalls, such as, message replay attack, Sybil attack and impersonation attack, and other practical issues. For the comprehensive analysis of PACPs, we have assumed that an attacker has full control over wireless channels (e.g., it can insert, drop, modify or replay the wireless messages). Based on above assumptions, we generalize the message replay attack in PACPs, as follows.

1) Message replay attack: In this attack, an adversary actively captures on-air wireless messages between two communicating entities (e.g., node A and node B) and replays the captured messages, later, as it is. Although, it is a very common attack on wireless communication protocols but it (replay attack) could cause of one of the network destructive denial-of-services attack if it would not be protected efficiently and resultant, node's (AA) battery power depletion. **Attack description:** In PACPs, it is worth noting that, as shown in Fig.1 (*secPACP*) and Fig.2 (*ePACP*), an active adversary easily captures the wireless messages (*msg1*) between the node A and the node B (refer-Section II, authentication and key establishment phase). In *secPACP*, assumed that after some later time adversary transmits, *msg1* (ID_A, N_A, S_A) to the node B.

Upon receiving *msg1* from adversary, node B starts computations as follows: verifies $h(ID_A || N_{X_A} || h(ID_A || K_B)) = S_A$. It will be verified easily because every time node B considers *msg1* as a fresh message (because random number/nonce is not properly verified) and node B computes: $N_B = t_B P = (N_{X_B}, N_{Y_B})$ and $S_B = h(ID_B || N_{X_B} || h(ID_B || K_A))$ and sends *msg2* (ID_B, N_B, S_B) to attacker. *Note that*, here the node B is not aware about that it has sent *msg2* to an attacker or to a legal node. Now upon receiving the *msg2* from the node B, an attacker generates a fake *msg3* ($Z_A' = h(ID_A || SK_{X_{AB}}' || h(ID_A || K_B))$) and sends it to the node B. Here, $SK_{X_{AB}}'$ is attacker's fake key. Now, the node B computes the key ($SK_{AB} = t_B N_A = (SK_{X_{AB}}, SK_{Y_{AB}})$) and verifies the message (Z_A'). Obviously, attacker's fabricated fake message (i.e., Z_A') will not be verified by the node B because $SK_{X_{AB}} \neq SK_{X_{AB}}'$ and hence Z_A' will not be verified. Thus, due to the very late detection of an attacker, *secPACP* is vulnerable to the message replay attack. By imposing the message replay attack again and again, an attacker can make sensor node battery depletion which is not acceptable in the mission-critical WSN applications. Likewise, *ePACP* is also vulnerable to the *replay attack*.

Authors of [8] argued that preloading the number of keys (i.e., either pairwise or not) onto exposed devices (i.e., not

tamper-proofed) strengthens the incentive for attackers to compromise a node. In PACPs, authors exploit the pairwise key pre-distribution scheme and suggested that each PACPs node contains number of keys (e.g., 5,740 keys in *secPACP* and 1650 keys in *ePACP*) (refer Table I [7]). Though, *Kim et al* claimed that *secPACP* and *ePACP* are resilience against node capture attack and node fabrication attacks means if a node is captured then the pairwise keys of non-captured nodes are node revealed. However, the high number of keys in a node motivates to the attackers for corrupting more nodes. Moreover, *Tyler Moore* demonstrated that a small colluding node (less than 5% of the entire network) can control half's of its neighbors' communication channels [8]. Thus in PACPs, an adversary can collect the energy-exhausted sensor nodes from the terrain and can dig outs the all secrets from a node. Based on above assumptions, we generalize the Sybil attack and impersonation attack on *secPACP* and *ePACP*.

2) Sybil attack: In this attack, a malicious sensor node can present itself with multiple fake identities (IDs) and impersonates other legitimate nodes as a legal node [8] [9]. Moreover, it can manifest in a severe form leading to the failure of basic protocols functioning, such as network routing, network resource allocation and network functioning.

Attack description: In mission-critical applications (e.g., military, homeland security, etc) where sensor networks are often deployed in hostile environments. Consider *secPACP* case, where 5,750 keys suggested for an exposed sensor node. Assumed that a motivated adversary collects some energy-exhausted sensor nodes and reprogram them or make replication of the nodes (known as clone). Thereafter adversary deploys these malicious/clone nodes into the terrain, authenticates itself with non-compromised nodes and may control the network, accordingly. Now onwards, we call a malicious node as a *Sybil node*. It is assumed that a *Sybil node* can recognize the identities of its neighboring nodes using some beaconing technique which includes the node identity in beacons. A *Sybil node* illegitimately takes on multiple identities [2]. Moreover these identities may belongs to its *authentication set* (i.e., $AS_X = \{(HID_i, h(ID_X || K_{Ri}))_{i \in \{1,2,...,z\}}\}$ or belong to the existing nodes identities, here, HID_i means the owner of secret key K_{Ri} . Fig.3 depicts the Sybil attack running example.

For the simple generalization of the Sybil attack, we assume the size of a large key space (*LKS*) is 50. As shown in Fig. 3, a *Sybil node* presents its multiple identities to its neighboring nodes and tries to authenticate and establish a pairwise key, as a legal node. For instance, it (*Sybil node*) shows own multiple identities as follows: ID_{42} to the node 25, ID_8 to the node 39, ID_{16} to the node 45 and ID_{46} to the node 20. The solid (red) line represents that the node 25 has HID_{42} , the node 39 has HID_8 , and the node 45 has HID_{16} are corresponding to the *Sybil node*. Hence, the node 25, node 39 and node 45 authenticate to the *Sybil node* as a legitimate node and establish pairwise keys with the *Sybil node*.

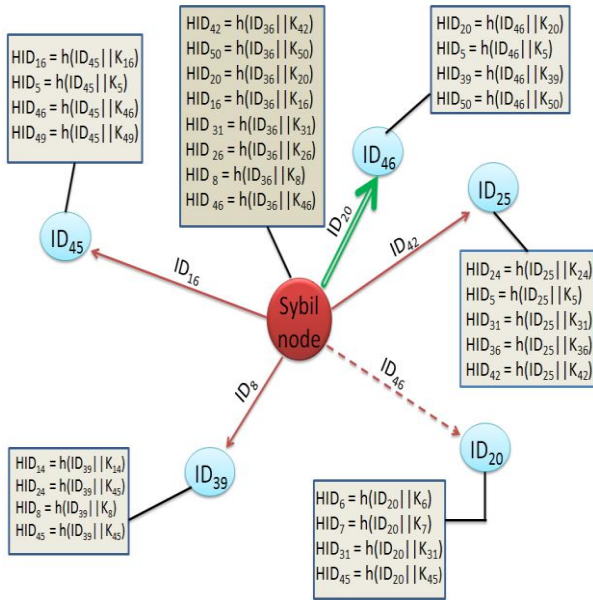


Figure 3. Sybil attack in secPACP scheme

The flow of Sybil attack between the Sybil node (i.e., ID_{42}) and the node 25 (says node B) is as follows.

- Sybil node generates a random integer St_A and computes the point $SN_A = St_A P = (SN_{x_A}, SN_{y_A})$ over the elliptic curve E , and computes $SS_A = h(SID_A || SN_{x_A} || h(SID_A || K_B))$. Now Sybil node sends SID_A , SN_A , and SS_A to the node 25 (i.e., B).
- After receiving the message from the Sybil node, node B checks whether $h(SID_B || K_A)$ is in $AS_B = \{(HID_i, h(SID_B || K_{Ri}))\}_{i \in \{1,2,\dots,z\}}$. Since, the node B holds, and it verifies $h(SID_A || SN_{x_A} || h(SID_A || K_B)) = SS_A$ with its own key K_B . Here, SS_A will be verified and node B assured that SN_A is generated by a legal node. Note that, here the node B does not know whether this message (SID_A , SN_A , and SS_A) is received from legitimate node or an attacker (Sybil node). After that, node B generates a random integer t_B and computes $N_B = t_B P = (N_{x_B}, N_{y_B})$ and $S_B = h(ID_B || N_{x_B} || h(ID_B || K_A))$. And it sends ID_B , N_B , and S_B to the Sybil node.
- Upon receiving the messages from the node B, Sybil node easily checks $h(ID_B || N_{x_B} || h(ID_B || K_A)) = S_B$ with its own key K_A . Thereafter, Sybil node computes $SSK_{AB} = St_A N_B = (SSK_{x_{AB}}, SSK_{y_{AB}})$ and $SZ_A = h(SID_A || SSK_{x_{AB}} || h(SID_A || K_B))$, and sends SZ_A to the node B.
- Node B computes $SK_{AB} = t_B SN_A = (SKK_{x_{AB}}, SKK_{y_{AB}})$ and checks $h(SID_A || SSK_{x_{AB}} || h(ID_B || K_A)) = SZ_A$. Since it will be verified and node B computes $Z_B = h(ID_B || SSK_{x_{AB}} || h(SID_A || K_B))$ and sends it to the Sybil node.
- Now Sybil node computes $h(ID_B || SSK_{x_{AB}} || h(SID_A || K_B))$ and establishes a pairwise key with the legitimate node (i.e., node B).

Similarly, Sybil node can establish a pairwise key with the node 39, 45, and many more. The Sybil node authentication

and key establishment phase is shown in Fig. 4.

Moreover, in Fig. 3, the (red) dotted line represents that the node 20 do not contain any HID_{46} , and hence, cannot authenticate to the Sybil node. The double (green) solid line represents that a Sybil node can impersonates its neighboring nodes. For example, it (Sybil node) sends own neighbor's identity (i.e., ID_{20}) to the node 46 and impersonates as a legal node. Since, the node 46 has HID_{20} ; it authenticates and establishes a pairwise key (as shown in Fig. 4) with the node 46.

Likewise, ePACP is also susceptible to the Sybil attack and impersonation attack, where 1,650 keys are recommended for an exposed sensor node.

Resultant, PACPs are not secure against the Sybil attack and impersonation attack where a sole Sybil node can control PACPs's neighboring nodes communication channels without misbehaviour detections [9].

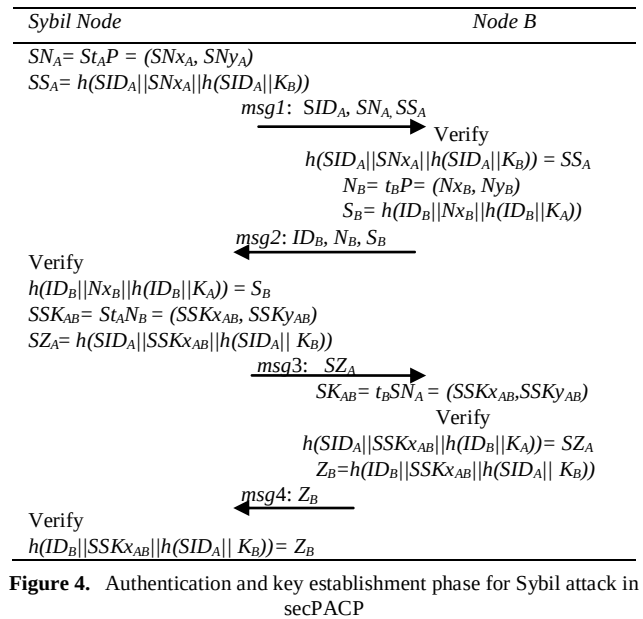


Figure 4. Authentication and key establishment phase for Sybil attack in secPACP

3) Limited scalability in secPACP (new node addition):

Recall a new node addition phase in secPACP (refer section-II), where a new node is entering into the existing networks. The base station (BS) assigned a new identity to the new sensor node (ID_{Q+1}) and also preloads secret key K_{Q+1} and $AS_{Q+1} = \{(HID_i, h(ID_{Q+1} || K_{Ri}))\}_{i \in \{1,2,\dots,z\}}$. However, secPACP allows only limited scalability (i.e., new node addition) to the network. In secPACP network, where N numbers of identities were generated offline for the N nodes and Q nodes were selected for the initial network deployment (recall initialization phase in secPACP, Section-II). Now, only $Q+1$ (i.e., new node) can easily enter into the existing network because it may have shared secrets (i.e., K_{Q+1} and $AS_{Q+1} = \{(HID_i, h(ID_{Q+1} || K_{Ri}))\}_{i \in \{1,2,\dots,z\}}$) with the existing nodes. Note that, here an $N+1$ node can never be entered into the network since it does not contain any secret shared (i.e., K_{N+1} and $AS_{N+1} = \{(HID_{N+1}, h(ID_{N+1} || K_{Ri}))\}_{i \in \{1,2,\dots,z\}}$) with the existing N

nodes. For more simple generalization consider a *simple running example*. Assumed that a BS generates offline 50 nodes (N) identities and the size of key space is 50. Then BS randomly chooses 45 nodes (Q) for the initial deployment (or network). Then only, 5 nodes ($N-Q$) can be easily added into the network, because these ($N-Q$) nodes may have secret shared with the existing (Q) nodes. Therefore, $N+1$ (e.g., node 51) node cannot join the network. Consequently, *secPACP* has limited scalability, which is not practical for the MAMMOTH size distributed WSNs, where scalability is highly required.

Other practical issues: PACPs also have other practical issues, which are highly desirable for the real WSNs, as follows.

- In PACPs, if node A shares $h(ID_A || K_B)$ with the node B , only then both the nodes (A and B) can start key establishment. Otherwise, it is possible that a big part of network may isolates from the entire network, if shared secrets are not found. Hence, in PACPs shared secret is not guaranteed (i.e., 100%).

More importantly, in PACPs, the computation time (or computation cost) is very high (as depicted in Fig.5 [7]), where the average time for establishing a pairwise key is about 9 seconds, which is expensive for the real WSNs.

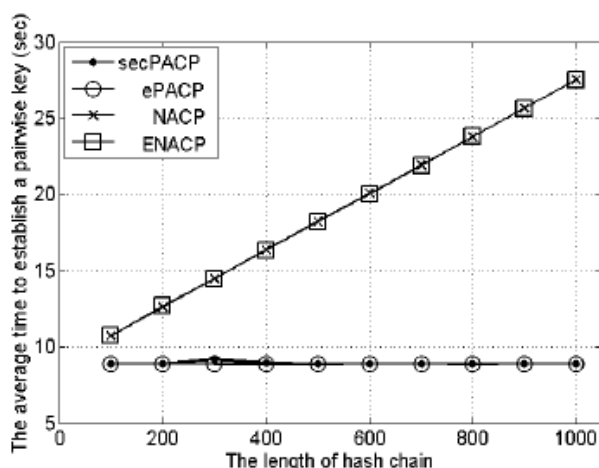


Figure 5. The average time to establish a pairwise key [7]

IV. CONCLUSIONS

In this paper, we have pointed out that PACPs are neither secure nor practical for the real mission-critical WSN applications. PACPs have still inherent security pitfalls; and can give enough incentives to the attackers. We have shown that how a sole energy-exhausted node (i.e., a Sybil node) can easily control the big part of a mission-critical application.

ACKNOWLEDGMENT

This research was financially supported by the Ministry of Trade, Industry and Energy(MOTIE) of the Republic of Korea and Korea Institute for Advancement of Technology(KIAT) through the Regional Specialized Industry Development Program(Grant Number : R0005598).

This work is supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology. (Grant number: NRF-2011-0023076).

REFERENCES

- [1] I.F. Akyildiz, W. Su, and Y. Sankarasubramaniam, "A Survey on sensor Network," *IEEE Comm. Mag.* 2002, 40, pp. 102-114.
- [2] Y. Zhou, Y. Zhang, and Y. Fang, "Access Control in Wireless Sensor Networks," *Ad Hoc Networks* 5 (2007), pp. 3-13.
- [3] H. Huang, "Novel Access Control Protocol for Secure Sensor Networks," *Computer Standard & Interfaces*, 2009, vol. 31, pp. 272-276.
- [4] H-S Kim and S-W Lee, "Enhanced Novel Access Control Protocol over Wireless Sensor Networks," *IEEE Trans. on Consumer Electronics*, vol. 55, No.2, May 2009, pp. 492-498.
- [5] P. Zeng, K-K. R. Choo, and D-Z. Sun, "On the Security of an Enhanced Novel Access Control Protocol for Wireless Sensor Networks," *IEEE Trans. on Consumer Electronics*, vol. 56, No. 2, May 2010, pp. 566-569.
- [6] J. Shen, S. Moh, L. Chung, "Comment: "Enhanced Novel Access Control Protocol over Wireless Sensor Networks", " *IEEE Trans. on Consumer Electronics*, vol. 56, No. 3, August 2010, pp.2019-2021.
- [7] H. Lee, K. Shin, and D-H Lee, "PACPs: Practical Access Control Protocols for Wireless Sensor Networks," *IEEE Trans. on Consumer Electronics*, vol.58, No. 2, May 2012, pp.491-499.
- [8] M. Tyler, "A Collusion Attack on Pairwise Key Predistribution Schemes for Distributed Sensor Networks," *In the proceeding 4th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06)*, 2006, 13-17 March 2006.
- [9] D. Mukhopadhyay, I. Saha, "Location Verification based defense against Sybil attack in Sensor Networks," *In the proceedings of the 8th International Conference on Distributed Computing and Networking*, 2006, pp 509-521.



Mangal Sain received the M.Sc. degree in computer application from India in 2003 and the Ph.D. degree in computer science in 2011. Since 2012, he has been an Assistant Professor with the Department of Computer Engineering, Dongseo University, South Korea. His research interest includes wireless sensor network, cloud computing, Internet of Things, embedded systems, and middleware. He has authored over 30 international publications including journals and international conferences. He is a member of TIIS

and a TPC member in several international conferences

Hyun Ho Kim



2013 : BS at Dongseo University, Republic of Korea
2015 : MS at Dongseo University, Republic of Korea
2015 ~ current: doctor's course Dongseo University, Republic of Korea
Research Interests: Digital Forensic, Information Security, Network Security



Hoon-Jae Lee received his BS, MS, and Ph.D. degrees in Electrical Engineering from Kyungpook National University, Daegu, South Korea, in 1985, 1987, and 1998, respectively. He is currently a professor in the Department of Information and Communication Engineering at Dongseo University. From 1987 to 1998, he was a research associate at the Agency for Defense Development (ADD). His current research interests include developing secure communication system, side-channel attack, and ubiquitous sensor network/radio frequency identification security.