Efficient Mobile Sensor Authentication In Smart Home and WPAN

Kyusuk Han, Taeshik Shon, *Member*, IEEE, and Kwangjo Kim, *Member*, IEEE

Abstract — *Currently, it is rapidly increasing convergence services based on various mobile devices with sensors like Smart Home. Specifically the mobility of the sensors in Smart Home merged with wireless sensor networks (WSN) brings security issues such as re-authentication and tracing the node movement. We extend our novel and efficient node authentication and key exchange protocol that support Irregular distribution. Compared with previous protocols, our protocol has only a third of communication and computational overhead. We expect our protocol to be the efficient solution that increases the lifetime of sensor network*[1].

Index Terms — Wireless Sensor Networks, Authentication, Mobile node, Untraceability, Key Distribution.

I. INTRODUCTION

Wireless Sensor Network (WSN) is the network that consists of lightweight devices with short-ranged wireless communication and battery-powered. The devices have the sensor that gathers the environmental information and etc. After sensing this information, the devices send the information to the networks. The recent advance made the WSN technologies be applied in various areas such as Smart Digital Home Network [13]-[14], Wireless Personal Area Network (WPAN) and Wireless Sensor and Actor Network (WSAN) [7]-[8]. Recently, RF4CE also deploy Zigbee (IEEE 802.15.4) [11] as underlying communication technologies, which is designed to substitute the current IR communication. In such environments, handling a large overhead from frequent node re-authentication requests due to the continuous node movements and the threats of tracing the node movement are important security issues.

While most security researches on the WSN remain on how to efficiently utilize the limited resources in static network environments [1,5,9,10,12], a few researches begin to consider the security in the dynamic environments. Reference [6] argued the possible presence of mobile node, and proposed the authentication protocol supporting node mobility that does not require any sink or base station for authentication and key distribution. Their model requires the large communication and computation cost when the node is continuously moved though. In order to minimize such overhead, we proposed efficient node authentication and key exchange model that reduces communication and computational costs for node re-authentication and also provides untraceablity to mobile nodes [2]. In the model, once a mobile node is firstly authenticated by a static sink, the node can be efficiently authenticated by the neighbor sinks of the firstly connected sink.

However, the previous model has the limit that the protocol may not properly work in the environment that the sensors are irregularly distributed. In case of the smart home, the electric devices that attach sensors may be distributed irregularly as in Fig. 1. In such environment, the remote controller may fail to be re-authenticated depending on the node movements.

Therefore, our motivation is to provide the improved node authentication and key exchange model suitable for such irregularly distribution. Applying our improvement, the mobile node can be authenticated by the sink that is not the neighbor of the formerly connected sink.

The paper organized as follows: Section II describes the mobility of the sensor network and the previous authentication and key exchange protocol. We argue the problem in the irregularly distributed environments and show the improved protocol in Section III. Section IV shows the analysis of the protocol, and Section V concludes this paper.

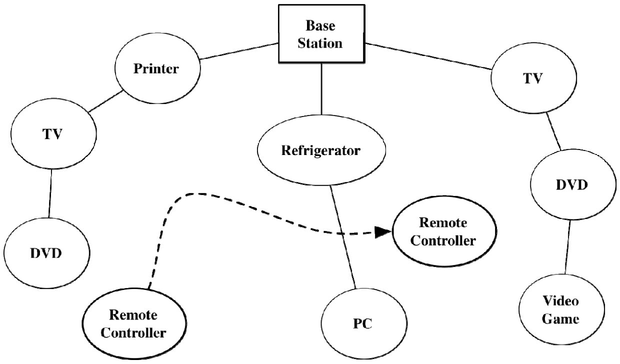


Fig. 1 RF4CE deploys Zigbee based sensor network technologies as underlying communication technology. In such environment, the sensor in the remote controller has the mobility.

II. PROBLEMS IN THE SMART HOME

In this section, we describe the brief procedure of original protocol, and claim problems applying to the smart home.

A. Overall Process of Previous Protocol

The overall protocol is divided in to five phases: Periodical Neighbor Discovery (Phase 0), Neighbor Sink Setup (Phase 1), Neighbor Group Key Distribution (Phase 2), Node Initial

K. Han is with the department of Computer Science, KAIST, Daejeon, Korea (e-mail :hankyusuk@kaist.ac.kr).
Corresponding Author : T. Shon is with Convergence Solution Team, Digital Media & Communication R&D Center, Samsung Electronics, Suwon, Korea (e-mail : ts.shon@samsung.com).
Corresponding Author : K. Kim is with the department of Computer Science, KAIST, Daejeon, Korea (e-mail :kkj@kaist.ac.kr).

Contributed Paper
Manuscript received April 13, 2010
Current version published 06 29 2010;
Electronic version published 07 06 2010.

Authentication (Phase 3), and Node Re-authentication (Phase 4).

Assume that there are a base station *BS*, a sink S_1, a neighbor sink S_2, and a mobile node *N* in the network. We define the neighbor sink as the sink that is in the 1 hop communication range.

During phase 0, every sink such as S_1 and S_2 periodically broadcasts HELLO. If no attempt happens, phase 0 is just discarded.

When S_2 receives HELLO from S_1, S_2 initiates the neighbor relationship if S_1 is a newly discovered sink. After the pairwise key between S_1 and S_2 has been exchanged in phase 1, S_1 and S_2 exchange the authentication key that is used to verify the authenticated user in phase 2. Phase 1 and phase 2 are only required during establishing the static sensor network. We let the establishing the static sensor network follows the any previous protocol such as [4].

When *N* firstly joins the network, *N* may be connected to S_1 in the network as in Fig. 2. After receiving HELLO of S_1, *N* initiates the initial authentication with S_1 in Phase 3. Once *N* is authenticated S_1, *N* only needs the re-authentication in Phase 4 when *N* continuously moves and request the authentication again. The authentication process in Phase 3 is only necessary when the re-authentication fails due to the certain case that the neighbor sink is not available.

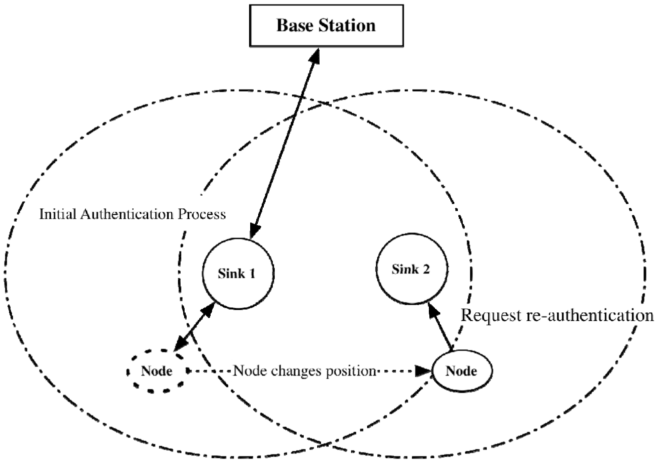


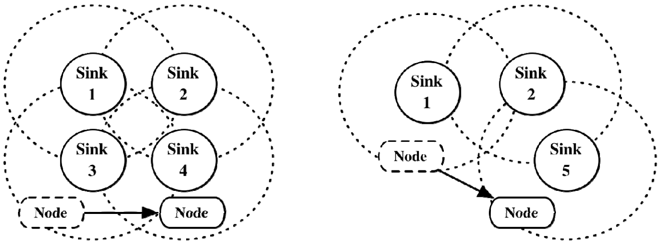
Fig. 2 The base station is only involved when the sink 1 firstly authenticates the node (Phase 3). Next time, the node is directly authenticated by sink 2 without the base station (Phase 4).

B. Authentication Ticket

The previous protocol introduced the Authentication Ticket that is the proof of the node is authenticated. A node receives the authentication ticket from the sinks during the authentication process. The ticket is used for the next authentication by the neighbor of the sink. The neighbor sinks verify the ticket and sends the updated one to the sink. The verification of the ticket is done using the authentication key inherited 'cluster key' in [9]-[10]. The main difference is that the key is used for broadcast communication in the cluster, while the key in our protocol is used for verifying the authentication ticket.

C. Problems in The Smart Home

Since the resident of the smart home does not consider the regularity arranging the devices such as TVs, DVDs, and microwaves, the regular distribution is not expected in real environments.



(a) Ideal Environments **(b) Real Environments**

Fig. 3 Sinks are regularly distributed in the ideal environments as in (a). However, in the real environments such as the smart home, the sinks may be distributed irregularly as in (b).

The previous protocol works well in the ideal environments as in Fig. 3 (a). However, the node may fail to be re-authenticated in case sinks are irregularly distributed. The node authenticated by S_1 may move and reconnected to S_5. However, S_5 is not the neighbor of S_1, the node cannot be re-authenticated as in Fig. 3 (b).

III. Improvement for Smart Home

In this section, we show the improvement of the previous protocol for the smart home. We introduce the concept of 'Neighbor Sink List (NSL)' in order to make our protocol be applicable in the real environments, and show the improvement with NSL.

A. Neighbor Sink List

When a sink finds the neighbor sinks, the sink stores the list of the neighbors. The neighbor sink list (NSL) of a sink S_1 is denoted as NSL_{S_1}, where $\mathrm{NSL}_{S_1} = S_i \| h(S_j) \| \ldots \| h(S_{j+k}) \| M$, and $M = MAC_{AK_{S_i}}(S_i \| h(S_j) \| \ldots \| h(S_{j+k}))$.

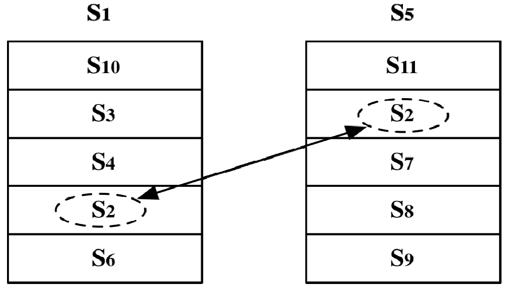


Fig. 4 S_5 finds that S_1 and S_5 have the common neighbor S_2 by checking NSL of S_1.

NSL is sent to the node during authentication process. When a node authenticated by S_1 is reconnected to S_5, the node sends NSL_{S_1} to S_5. Although S_1 is not the neighbor of S_1, S_5 finds out that S_2 is the common neighbor of both S_1 and S_5 as in Fig. 4.

B. Improved Protocol

1) Periodical Neighbor Discovery Procedure

S_1 periodically generates a random nonce R_0. S_1 also generates u_0 and v_0, where $u_0 = E_{K_{S_1}}(R_0 \| TS_0)$ and $v_0 = MAC_{IK_{S_1}}(S_1 \| \mathrm{HELLO} \| u_0)$. TS_0 is timestamp. Then S_1 broadcasts u_0 and v_0 with HELLO. We have no change from the previous protocol.

2) Establishing Neighbor Sink List

Assume another sink S_2 receives HELLO message. S_2 checks the sender of HELLO whether S_1 is known or not. If S_2 already knows S_1, S_2 discards the message. Otherwise, S_2 requests the setting up the neighbor relationship as follows:

S_2 randomly selects R_1 and generates u_1 and v_1, where $u_1 = E_{K_{S_2}}\{R_1 \| u_0\}$ and $v_1 = MAC_{IK_{S_2}}(S_2 \| \mathrm{BS} \| S_1 \| u_1 \| v_0)$. After verifying v_1, BS decrypts u_1 and retrieves R_1 and u_0. Then, BS verifies v_0 and decrypts u_0. Finally, BS retrieves R_0 and TS_0 , and then generates u_3, u_4, v_4, and v_3, where $u_3 = E_{K_{S_1}}\{R_1 \| h(TS_0)\}$, $v_3 = MAC_{IK_{S1}}(\mathrm{BS} \| S_1 \| u_3)$, $u_4 = E_{K_2}\{R_1 \| u_3\}$ and $v_4 = MAC_{IK_2}(\mathrm{BS} \| S_2 \| R_1 \| u_4 \| v_3)$. And then BS sends u_4, v_4, and v_3 to S_2. Then S_2 verifies v_4 and decrypts u_4, and retrieves R_1 and u_3. S_2 generates the encryption key $K_{S_1S_2}$ and the integrity key $IK_{S_1S_2}$ shared between S_1 and S_2, where $K_{S_1S_2} = KDF(0 \| R_0 \| R_1)$ and $IK_{S_1S_2} = KDF(1 \| R_0 \| R_1)$. Then S_2 generates $v_{5,}$ where $v_5 = MAC_{IK_{S_1S_2}}(S_2 \| S_1 \| R_0 \| R_1)$, and sends u_3, v_3, and v_5 to S_1. After verifying v_3, S_1 decrypts u_3 and retrieves R_1. S_1 also generates $K_{S_1S_2}$ and $IK_{S_1S_2}$. Then S_1 verifies v_5. S_1 generates $v_6 = MAC_{IK_{S_1S_2}}(S_1 \| S_2 \| ACK \| R_0 \| R_1)$ and sends v_6 with ACK to S_2. S_2 verifies v_6 and shares pairwise keys $K_{S_1S_2}$ and $IK_{S_1S_2}$. As a result, S_1 and S_2 update their NSL.

***3)** Distribution of Authentication Key*

After neighbor sinks are found, the sink S_1 may distribute the authentication key (AK). S_1 randomly selects two nonce $ASEED_{S_1}$ and R_1. Then S1 generates u_1 and v_1, where $u_1 = E_{K_{S_1S_2}}\{ASEED_{S_1} \| R_1\}$ and $v_1 = MAC_{IK_{S_1S_2}}(S_1 \| S_2 \| u_1)$. After verifying v_1, S_2 decrypts u_1, and retrieves $ASEED_{S_1}$ and R_1. Then S_2 generates $AK_{S_1} = KDF(0 \| ASEED_{S_1})$ and $AIK_{S_1} = KDF(1 \| ASEED_{S_1})$. S_2 also generates v_2 using AIK_{S_1} , where $v_2 = MAC_{AIK_{S_1}}(S_2 \| S_1 \| \mathrm{ACK} \| AR_1)$. Then S_1 verifies v_2.

4) Initial Node Authentication

Assume a node N is firstly joining the sensor network. When N receives HELLO of S_1, N randomly selects R_1 and generates u_1 and v_1 and sends them to S_1, where $u_1 = E_{K_N}\{R_1 \| u_0 \| v_0\}$ and $v_1 = MAC_{IK_N}(N_1 \| S_1 \| u_1)$. Then, S_1 generates v_2, where $v_2 = MAC_{IK_{S_1}}(S_1 \| \mathrm{BS} \| N \| u_1 \| v_1)$, and sends it to BS. After verifying v_2 and v_1, BS decrypts u_1, and retrieves R_0, u_0 and v_0. After verifying v_0, BS decrypts u_0, and retrieves R_0 and TS. BS checks the validity of TS and generates u_3, v_3, u_4, and v_4, where $u_3 = E_{K_N}\{R_0\}$, $v_3 = MAC_{IK_N}(BS \| N \| S_1 \| u_3)$, $u_4 = E_{K_{S_1}}\{R_1 \| u_3 \| v_3\}$ and $v_4 = MAC_{IK_{S_1}}(BS \| S_1 \| N \| R_0 \| u_4)$.

After verifying v_4, S_1 decrypts u_4, and retrieves R_1, u_3 and v_3. Then S_1 generates $NK_N = KDF(R_0 \| R_1)$. S_1 generates authentication ticket T = (t, w), where $t = E_{AK_{S_1}}\{TS \| R_1 \| NK_N\}$ and $w = MAC_{AIK_{S_1}}(N \| t)$. S_1 also generates u_5 and v_5, where $u_5 = E_{NK_N}\{TS \| T \| NSL_{S_1}\}$ and $v_5 = MAC_{NIK_N}(S_1 \| N \| R_0 \| u_5)$. S_1 sends v_3, u_5, and v_5 to N.

After verifying v_3, N decrypts u_3 and retrieves R_0. Then N also generates NK_N and verifies v_5. N decrypts u_5 and retrieves TS, T and NSL_{S_1} . N generates v_6, where $v_6 = MAC_{NK_N}(N \| S_1 \| ACK \| R_0 \| R_1)$. S_1 verifies v_6.

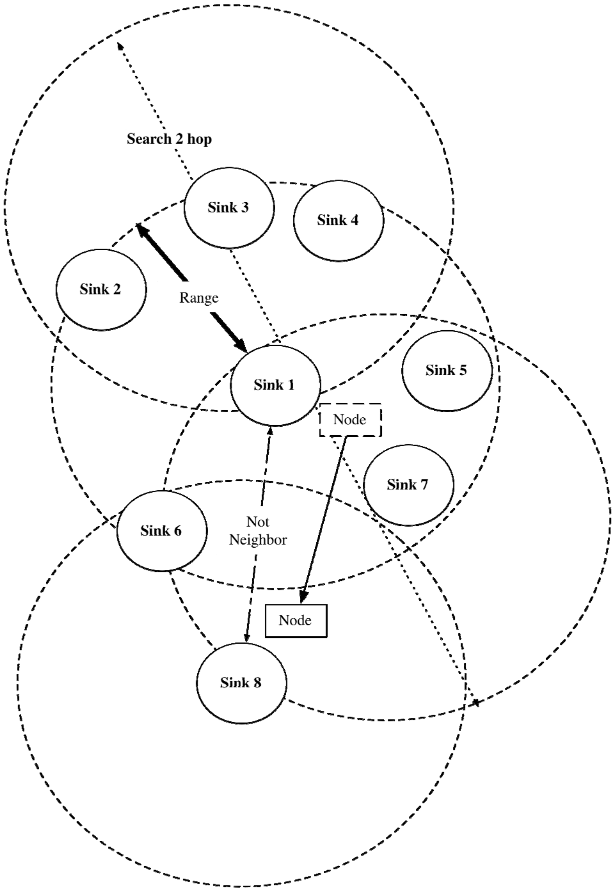


Fig. 5. When the node authenticated by S_1 is reconnected S_8, S_8 authenticates the node by finding the common neighbor S_6 comparing NSL of S_1 and S_8, and requesting the authentication of the node to S_6.

5) Node Re-authentication in Ideal Environments

Once the node N is authenticated, N can have the reduced overhead for the following authentication. Assume N moves and receives HELLO from S_2. N generates v_1, and sends T, v_1 and NSL_{S_1} to S_2, where $v_1 = MAC_{NIK_N}(N \| S_2 \| T \| NSL_{S_1} \| v_0)$. Then S_2 check NSL_{S_1} if S_1 is the neighbor of S_2.

When S_1 is the neighbor of S_2, S_2 verifies T and decrypts t using the authentication key AK_{S_1} . S_2 retrieves R_1, NK_N and TS. Using NK_N, S_2 verifies v_1. Then S_2 generates new shared key $NK'_N = KDF(R_1 \| R_0)$, also generates the new authentication ticket *T' = (t',w')*, where $t' = E_{AK_{S_2}}\{R_1 \| NK'_N\}$ and $w' = MAC_{AIK_{S_2}}(N \| t')$. S_2 generates $v_2 = h(NK'_N \| R_0)$ and $u_3 = E_{NK_N}\{R_0 \| v_2 \| T' \| NSL_{S_2}\}$, $v_3 = MAC_{NIK_N}(S_2 \| N \| u_3)$.

After verifying v_3, *N* decrypts u_3 and retrieves R_0, v_2, *T'* and NSL_{S_2} . Then *N* generates NK'_N and verifies v_2. *N* generates v_4, where $v_4 = MAC_{NIK'_N}(N \| S_2 \| \mathrm{ACK} \| R_0 \| R_1)$, and sends v_4 with ACK to S_2. After verifying v_4, S_2 authenticates *N*.

6) Node Re-authentication in Real Environments

In case the node *N* that was authenticated by the sink S_1 is reconnected to other sink S_8 as in Fig, the node may fail to be authenticated in the previous scheme, since S_8 is not the neighbor sink of S_1. However, our improvement enables the efficient re-authentication of *N*.

When S_8 receives NSL_{S_1} , S_8 identify that S_1 is not the neighbor. Instead, S_8 finds that the neighbor sink S_6 is also the neighbor of S_1. (Refer Fig. 4.) Thus, S_8 sends the authentication ticket *T* to S_6 and request verification, then S_6 verifies *T* using AK_{S_1} of S_1 and returns the results to S_8. With the results from S_6, S_8 generates NK'_N and *T'*. The remaining follows the process in ideal environments.

IV. Analysis

In this Section, we analyze our improved protocol with comparing the previous protocol. For the performance analysis, we compare the number of communication passes, the required message sizes, and the number of computation of the protocol. We do not count the overhead in the neighbor discovery procedure, since the node just ignores this procedure when the node receives HELLO from the sink that already authenticated the node.

A. Communication Pass

We compared the required number of communication passes with Fantacci *et al.*'s model [6], Ibriq and Mahgoub's model [4], and original model [2]. TABLE I shows the comparison of communication passes for node re-authentication, where *n* denotes the number of nodes and *t* denotes the number of sinks. Since nodes act as the authentication server (the base station) and the authenticator (the sink), all the communications in [6] are operated among nodes.

Comparison of required number of communication pass in initial authentication is as same as the previous models. In re-authentication of the nodes, Improved model requires 2 more communication for re-authentication in real environments than the original model, while it is still much efficient than [3]-[4].

TABLE I
COMPARISON OF COMMUNICATION PASS FOR RE-AUTHENTICATION

	Fantacci , *et al.*'s [6]	Ibriq *et al.'s* [4]	Previous Model [2]	Proposed model
Node	2	2n	2	2
Sink	2t+1	2t	1	1 (3)
Base station	-	2	-	

B. Message Size

We compared Abraham and Ramanatha's model [3], [2] and [4] for the required message size for authentication. Based on the results in [3], we approximately compared the message sizes based on the message size with MAC size as 4 bytes, the time stamp as 8 bytes, nonce as 8 bytes, and key size as 16 bytes. We also set the source and target IDs as 1 byte, respectively.

TABLE II
COMPARISON OF MESSAGE SIZE FOR INITIAL AUTHENTICATION (BYTES)

	Abraham 's model [3]	Ibriq and Mahgoub's model [4]	Previous Model [2]	Proposed Model
Node to Sink	46	68	56	56
Sink to Sink	70	76	62	62
Sink to Base station	70	76	62	62
Base station to Node	92	188	192	204
Total message size	278	408	314	326

TABLE II and III show the message sizes in initial authentication and the message sizes in re-authentication with 2 hops between sink and base station, respectively. TABLE II shows that the performance for the initial authentication is similar to other protocols. In initial authentication (Phase 3), Abraham and Ramanatha's model [3] showed the best result that 30 bytes less message sizes than our protocol. However, as the TABLE III shows, our protocol achieves about a third overall message sizes than other protocols. Even we increase the size of each parameter, our protocol is still much efficient than any other protocols in node re-authentication.

TABLE III
COMPARISON OF MESSAGE SIZE FOR RE-AUTHENTICATION (BYTES)

	Abraham's model [3]	Ibriq and Mahgoub's model [4]	Previous model [2]	Proposed model
Node to Sink	46	68	44	56
Sink to Sink	70	76		-
Sink to Base station	70	76		-
Base station to Node	92	188	64	76
Total message size	278	408	108	132

Fig. 6 shows the comparison of our improved model with the previous models. While the message cost is increasing with the longer hop distance in the static models [3]-[4], the original model [2] and the improved model have the constant cost.

Fig. 7 shows the comparison of the proposed protocols in several environments. The result of initial authentication shows the increasing cost depends on the hop distance. The re-authentication cases show the constant result although overall cost increases depending on the rate that the sink is not the neighbor of the former sink.

C. Security Analysis

Since the proposed protocol improves the previous protocol [2], most security features such as confidentiality, key freshness, and node/sink resiliency are inherited. Thus, we only concentrated on the analysis of the changes.

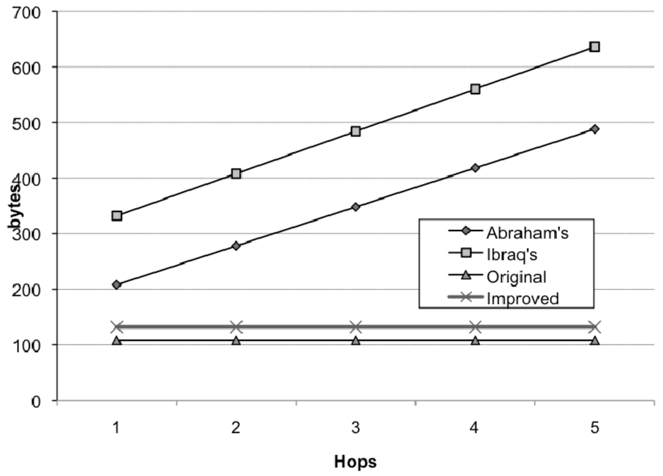


Fig. 6 Comparison of message sizes with static models [3]-[4], previous model [2] and improved model per hop distance between a sink and a base station.

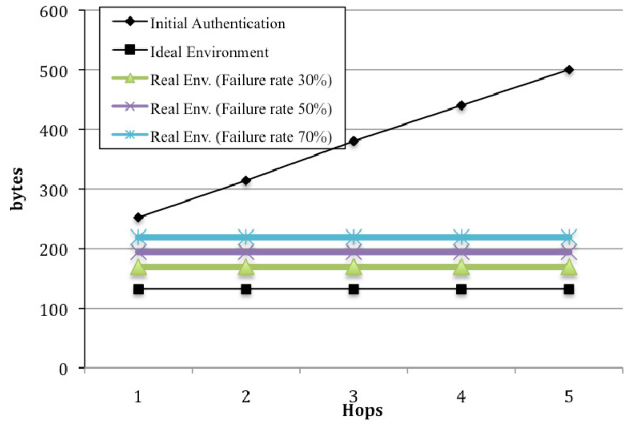


Fig. 7 Communication cost for proposed protocol. Depending on the failure rate, the cost increases, but still the cost shows the constant when the hop distance increases.

1) Re-authentication using Neighbor Sink List

After a node *N* is initially authenticated by a sink S_1 in phase 3, the node receives the authentication ticket *T* and v_1. When *N* moves and requests re-authentication to the neighbor sink S_8, S_8 may fail to verify *T* since S_1 is not the neighbor. However, S_8 and S_1 have the common neighbor *S2*, and the authentication key of S_1, AK_{S_1} is shared to S_2. Thus, with help of S_2, S_8 can authenticate *N* and exchange the key. In the re-authentication phase, the base station is not involved.

2) Untraceability using Neighbor Sink List

When S_8 authenticates *N*, S_2 involves in the protocol. However, the role of S_2 is just verifying and decrypting *T*. Therefore S_2 cannot predict *N*'s next movement.

3) Security against known attacks

The sinkhole attack against our protocol fails without knowing the keys. An adversary *A* may capture the authentication ticket *T* that *N* initially sent to S_2, and *A* send *T* to S_2 or other sink S_5 that is also a neighbor sink of S_1. However, *A* fails in such attack without knowing AK_{S_1} . Wormhole attack on our protocol fails since the adversary cannot send the confirmation message. Spoofed, altered or replayed routing information attack also fails with our knowing encrypted nonce in our protocol. To succeed in the replay attack, the adversary has to be able to re-use the intercepted packet. We don't consider relaying through the attackers as successful attack. Sybil attacks also fails from verification of identity of nodes through sinks and the base station. And for HELLO flood attacks, we can apply the global key shared to all entities in the network that many researches such as [4], [9], [10] used for the efficient message broadcast and DoS attack protection.

V. Conclusion

Recently, Smart Home is emerging and extending rapidly as new converged paradigms including fusion & convergence, smart grid, machine-to-machine, and peer-to-peer pervasive computing to provide fully always-connected services with mobility. Thus, it is very important to support dynamic topology among various CE and IT devices. Specifically, the failure of the node re-authentication can be occurred frequently because the previous works only considered the environment that the sensors are regularly distributed ideally.

In this paper, our proposed improvement enables the efficient node re-authentication and key exchange even when the sensors are irregularly distributed to the smart home and WPAN for supporting various convergence services. In order to verify the proposed approach, we perform three kinds of validation according to communication pass, message size, and security analysis. From the analysis, we can say that our improvement guarantees the longer lifetime of Smart Home Devices and WPAN while providing security solutions.

In future work we will deploy the proposed approach to real Smart home environments and confirm the authentication operations for supporting NSL.

VI. References

[1] H. Chan, A. Perrig, and D. Song, "Random Key Predistribution Schemes for Sensor Networks", *in IEEE Symposium on Security and Privacy*, Berkeley, California, pp. 197–213, 2003.

[2] K. Han, K, Kim, and T. Shon, "Untraceable Mobile Node Authentication in WSN," accepted to Sensors 2010 (ISSN 1424-8220; CODEN: SENSC9), Molecular Diversity Preservation International (MDPI), 2010

[3] J. Abraham, and K.S. Ramanatha, "An Efficient Protocol for Authentication and Initial Shared Key Establishment in Clustered Wireless Sensor Networks," *Proceeding of Third IFIP/IEEE International Conference on Wireless and Optical Communications Networks*, 2006.

[4] J. Ibriq, and I. Mahgoub, "A Hierarchical Key Establishment Scheme for Wireless Sensor Networks," *Proceedings of 21st International Conference on Advanced Networking and applications (AINA'07)*, 2007, pp. 210–219.

[5] L. Eschenauer, and V. Gligor, "A key management scheme for distributed sensor networks," *in Proceedings of the 9th ACM conference on Computer and Communications Security (CCS)*, Washington. DC. USA 2002, pp. 41–47.

[6] R. Fantacci, F. Chiti, and L. Maccari, “Fast distributed bi-directional authentication for wireless sensor networks”, *Security and Communication Networks*, John Wiley & Sons, pp. 17–24, 2008.

[7] S. Das, H. Liu, A. Kamath, A. Nayak, and I. Stojmenovic, "Localized Movement Control For Fault Tolerance of Mobile Robot Networks," *in IFIP International Federation for Information Processing, Wireless Sensor and Actor Networks*, eds. L. Orozco-Barbosa, Olivares, T., Casado, R., Bermudez, A., (Boston:Springer) 2007, 248.

[8] S. S. Krishnakumar, and R. T. Abler, "Intelligent Actor Mobility in Wireless Sensor and Actor Networks," *in IFIP International Federation for Information Processing*, Wireless Sensor and Actor Networks, eds. L. Orozco-Barbosa, Olivares, T., Casado, R., Bermudez, A., (Boston:Springer) 2007, pp. 13– 22.

[9] S. Zhu, S. Setia, and S. Jajodia, "LEAP: efficient security mechanisms for large-scale distributed sensor networks," *In CCS '03: Proceedings of the 10th ACM conference on Computer and communications security*. ACM: New York, NY, USA, 2003, pp. 62–72.

[10] S. Zhu, S. Setia, and S. Jajodia, "LEAP+: Efficient security mechanisms for large-scale distributed sensor networks," *ACM Trans. Sen. Netw*. 2006, 2, 500–528.

[11] W. C. Craig, "Zigbee:Wireless Control That Simply Works," Zigbee Alliance 2005.

[12] W. Du, J. Deng, Y. S. Han, and P. K. Varshney, "A Pairwise Key Pre-distribution Scheme for Wireless Sensor Networks," *in Proceedings of the 10th ACM conference on Computer and Communications Security (CCS)*, Washington. DC. USA 2003, pp. 42–51.

[13] E. Callaway, P. Gorday, and L. Hester, “Home Networking with IEEE 802.15.4: A Developing Standard for Low-Rate Wireless Personal Area Networks", *IEEE Communications Magazine*, vol. 40, no. 8, pp. 70-77, 2002.

[14] G. K., S. Yang, F. Yao, and X. Lu, "A zigbee-based home automation system", *IEEE Transactions on Consumer Electronics*, vol. 55, no. 2, pp 422 - 430, 2009.

BIOGRAPHIES

Kyusuk Han received the B.S. degree in Mechanical Engineering from Hongik University, Korea and the M.S. degree in Computer Science from Information and Communications University, Korea, respectively in 2001 and 2004. He is presently Doctorate course student in School of Engineering, KAIST, Korea. His interests are in cryptography and information security.

Taeshik Shon (M’10) is a senior engineer in the Convergence Solution Team, DMC R&D Center of Samsung Electronics Co., Ltd. He received his Ph.D. degree in Information Security from Korea University, Seoul, Korea, 2005 and his M.S. and B.S. degree in computer engineering from Ajou University, Suwon, Korea, 2000 and 2002, respectively. While he was working toward his Ph.D. degree, he was awarded a KOSEF scholarship to be a research scholar in the Digital Technology Center, University of Minnesota, Minneapolis, USA, from February 2004 to February 2005. He was awarded the Gold Prize for the Sixth Information Security Best Paper Award from the Korea Information Security Agency in 2003, the Honorable Prize for the 24th Student Best Paper Award from Microsoft-KISS, 2005, the Bronze Prize for the Samsung Best Paper Award, 2006, and the Second Level of TRIZ Specialist certification in compliance with the International TRIZ Association requirements, 2008. He is also serving as an editorial staff and review committee of the Journal of The Korea Institute of Information Security and Cryptology, IAENG International Journal of Computer Science, and other journals. His research interests include Mobile/Wireless Network Security, WPAN/WSN Network Security, network intrusion detection systems, and machine learning.

Kwangjo Kim received the B.S and M.S. degree of Electronic Engineering in Yonsei University, Korea, and Ph.D of Div. of Electrical and Computer Engineering in Yokohama National University, Japan. Currently he is Professor at School of Computer Science in KAIST, Korea. He is also the president of Korean institute on Information Security and Cryptography.