A Method for Hiding Link Layer Addresses Using Bloom Filter in Wireless Sensor Networks*

Sangho Park[1], Jihyun Bang[2], Mirim Ahn[3], Woomin Lee[3], and Taekyoung Kwon[2†]
[1]Sejong University, Seoul, Korea
superh1@gmail.com
[2]Yonsei University, Seoul, Korea
{jh.bang, taekyoung}@yonsei.ac.kr
[3]Agency for Defense Development, Seoul, Korea
mirimahn@hanmail.net, ewoomin@add.re.kr

Abstract

A unique identifier is necessary for identifying an object, such as a physical equipment used in networks, a user, and a server, in information systems. Such an identifier can be used not only for identification but also for authentication and authorization purposes after binding additional information. However, regarding the wide use of those unique identifiers across multiple systems, a great concern about privacy is seriously growing because of strong possibilities in collecting personal behaviors and transactions histories. Various researches are in progress to solve this problem. However, most of them are related to an identity which is used in software applications. Since it does not guarantee the anonymity of network addresses which are actually the identity in the lower level of communications, the network traffic toward specific devices can be easily analyzed. This problem can be more critical in the networks which should minimize the leakage of traffic information, such as in military wireless sensor networks. In this paper, we propose a new method that hides a receiver's address in the data link layer. Moreover, to prevent attackers from identifying specific traffics, we make it hard to determine receivers of the traffics. The proposed method is suitable to wireless sensor networks due to the use of the bloom filter based on an efficient hash function primitive rather than more complex cryptographic primitives involving heavier computational overhead.

Keywords: Link Layer Anonymity, Bloom Filter, Wireless Sensor Networks

1 Introduction

We use many identities in various societies. When you get a job, you get an employee identification number. You can distinguish a book which has a same title by ISSN number. You can also communicate with others by mobile phone number or email address. In certain information system, an object of a group can be identified with a unique identifier. Specifically, an object which is connected to network should have a unique identity in certain network. An object includes tangibles and intangibles such as physical equipment, human, service and so on. Communication can be made between the two using their own identities.

The identity is necessary to identify an object, but in an attacker's perspective, it can be used to show who the sender is. To prevent the leakage, the anonymity of identity is an important issue and

Journal of Internet Services and Information Security (JISIS), volume: 4, number: 4 (November 2014), pp. 71-81

*This research was supported by the project named "A Research on Dynamic Networking and Security for Wireless Sensor Networks and Mobile Networks" supervised by the ADD(Agency for Defense Development), Korea, and the MSIP(Ministry of Science, ICT&Future Planning), Korea, under the ITRC(Information Technology Research Center) support program (NIPA-2014-H0301-14-1010) supervised by the NIPA(National IT Industry Promotion Agency)

†Corresponding Author: 207, New Millenium Hall, 50 Yonsei-ro, Seodaemun-gu, Seoul, Korea, Tel: +82-(0)2-2123-4523, Web: `http://islab.yonsei.ac.kr`

anonymity and privacy have been studied in various researches. However, most of works are about identity which is used only in application. Link layer is the lowest layer that uses an identity between device communications. Traffic flow can be detected if the anonymity of network address used in link layer is not provided. The important devices can be detected by analyzing the packets if many packets foreword to the certain device even though the message body is encrypted and can be protected from others. Moreover in transport or application layer, data is transferred to an identity of final receiver node, but in link layer it is transferred to an intermediary device. Network structure can be determined by analyzing the receiver's identity of packet in link layer, such as network address. This can be critical in military and surveillance wireless sensor network that should hide the information of network structure.

Wireless sensor network is composed of a few number of base stations and a number of sensor nodes. Sensor node is usually low cost and low specification because it is used a lot. In addition, since it uses battery, it is hard to apply security primitives which use high computation power. Previous researches on link layer security for wireless sensor network is about lightweight cryptography and authentication function. The research on lightweight method that provides anonymity of link layer identity is insufficient.

This paper suggests the method which hides link layer address in wireless network that the range of node is predetermined like wireless sensor network so that an attacker cannot distinguish receiver's devices where the messages come from. The rest of the paper is structured as follows: we discuss related work in chapter 2 about link layer security and identity management, and explains our suggesting scheme in chapter 3. We analyze our proposal in chapter 4 and conclude in chapter 5.

2 Related Works

Many researches about link layer security for wireless sensor network are published. TinySec is designed for TinyOS which an one of the most well-known operating system for sensor network[5][21]. It uses Skipjack encryption algorithm with 80-bit key length and 64-bit block size[3], CBC (Cipher Block Chaining) mode of operation[14], and CBC-MAC (CBC Message Authentication Code). It includes TinySec-AE packet type which provides encryption and authentication and TinySec-Auth packet type which only provides authentication. The network application can choose between TinySec-AE and TinySec-Auth according to its purpose. The TinySec-AE packet consumes about 10% energy more than un-encrypted TinyOS normal packet. TinySec-Auth packet requires 3% energy more than un-encrypted TinyOS normal packet.

SenSec is a link layer security architecture for TinyOS and it is lighter than TinySec-AE[22]. It reduces 1 byte of packet size by modifying a header and it uses XCBC-MAC[7] instead of CBC-MAC. Considering broadcast and unicast, MiniSec contains more information by adding field in header[23]. It uses OCB (Offset Codebook) mode[27] and detects packet replay attack by using counter value. Besides, other protocols such as LLSL[26],L^3Sec[2], FlexiSec[19], and so on were made for link layer security of TinyOS. ContikiSec is a link layer security protocol for Contiki-OS which is an operation system for wireless sensor networks or Internet of Things[12][10]. It uses CBC-CS (Ciphertext Stealing)[13], OCB mode of operation, and AES (Advanced Encryption Standard) with 128-bit key length and 128-bit block size.

Various researches to integrate identities are in progress that have been managed by services or systems. The integrated management is an effective way to prevent defragment of information and increases convenience of users. Especially, many researches are in progress to prevent privacy exposure from identity leakage. PRIME (Privacy and Identity Management for Europe) is one of those research projects funded by the European Commission's 7th Framework Program[25][11]. PRIME basically provides anonymity or pseudonymous interaction. Anonymous or pseudonymous interactions are the default

within PRIME. PRIME is extended to PrimeLife and it finished in October 2011[24]. Liberty Alliance project is an identity management project which focuses on identity of web-based services[4]. It provides a level of pseudonymity by using pseudonyms instead of real identifiers in the communications between identity provider and service provider. Kantara is the next project of Liberty Alliance Project[18].

Ben Greenstein et al. presented an identifier-free link layer protocol to improve wireless security[16]. It uses temporary unlikable address which encrypts address by using shared secret key between device A and service B. This value is calculated differently according to time interval. The address is protected since an attacker not knowing the corresponding key cannot compute the address. Eun-Kyung Ryu et al. proposed an anonymous authentication method with link layer privacy based on bilinear pairings and the Bilinear Diffie-Hellman assumption[30]. It is resilient key compromise impersonation attack. Ronggong Song et al. designed a link layer anonymous access protocol in order to provide secrecy and anonymity[31]. Nodes in the LAA network generate dynamic pseudonyms by itself and use them for anonymous medium access authentication. Host identifier can be protected by using Cryptographically Generated Address in IPv6 (Internet Protocol Version 6) of network layer[6]. Some parts of address is computed cryptographic hash with host public key and message is signed with host private key. Thus, the the receiver can verify the host identity through packets.

Sometimes a bloom filter is used in network for multicast, routing IP traceback and so on[8][29][9]. LIPSIN assigns a link ID generated by inserting delivery tree or path to the bloom filter in network interface and routes packets to the link ID[20]. The researches about self-routing method with in-packet bloom filter which is resilient to Denial of Servie[28] and data-centric routing method with cumulative bloom filter and aggregated bloom filter[17] are in progress as well. We describe more about a bloom filter in the next chapter.

3 Proposed Scheme

When we encrypt a data part of link layer packet by using the existing link layer security protocol, all the packets in upper layer are encrypted. Since addresses which are identities of upper packets are also encrypted, we can hide the receiver from attacker. In wireless environment, message spreads in the air and device receives all the packets. The device decides whether to handle or not by looking at the address of link layer. Since the address is leaked, an attacker can exploit the information of network structure like wireless network routing path by analyzing traffic. Moreover, if there are a lot of packets toward the certain address, the attacker can predict the device with the address is an important device that handles with lots of data.

In a critical environment like military and surveillance wireless sensor network, we need to hide the information of network. When encrypting the address to hide receiver's data, the device need to decrypt first to figure out the packets in its charge and that makes delay in network. When considering a low spec environment like wireless sensor networks, we try not to use complicated mathematics operations. In this paper, we propose the method that can check packets promptly by hiding the address of link layer based on a bloom filter.

3.1 Bloom Filter

In 1970, Burton H. Bloom proposed a bloom filter which can check fast whether data set has a certain element or not by using hash code[8]. When a system performs membership query which checks whether a set A has an element x or not, it should check all the elements in set A. As the number of element which is as same as the size of a set A increases, too much memory is used to store all the elements. It the system uses database or file system, too much time is used compared to using memory.

A bloom filter returns the result in the form of true or false to show whether or not the element x is included. When false, there is no error, but when true, errors can exist. This means that even if the result is true, there exists false positive, which means certain element is not always included. When the result is false, the element is definitely not included. We can control the size of filter, the number hash functions and so on in order to be in false positive acceptable range. Thus, we can trade off the available time and space of a search system and make the false positive probability lower enough to be acceptable. When false positive is not acceptable, non-existing element is checked by a bloom filter and only existing element can be used to follow the method. Since a bloom filter can search faster than exhaustive search, it is more effective to search only for the element that turns to be true rather than search all the elements. Because of this feature, a bloom filter is sometimes used as an assistance tool of data search systems. When the search result is true, we can recheck the existence through database search and when it is false, we can show the element does not exist without any other process.

A bloom filter uses a bit arrangement v which consists of m elements and k independent hash functions which is denoted as h. Each hash function returns one of the values in $0, \cdots, m-1$ with discrete uniform distribution based on the input value. When you store a value in a bloom filter, the value should be put in k hash functions and put 1 at each index which is the output of hash function. When searching whether it has an element or not, we can check the output of all the k hash functions. If corresponding index is 1, then the element exists. If not, it does not exist. The figure 1 is the example of a bloom filter with $m = 18$, $k = 3$. Let's say we have the following result when we put x, y, z as an input of 3 hash functions respectively.

- $h(x) = \{1, 5, 13\}$
- $h(y) = \{4, 11, 16\}$
- $h(z) = \{3, 5, 11\}$

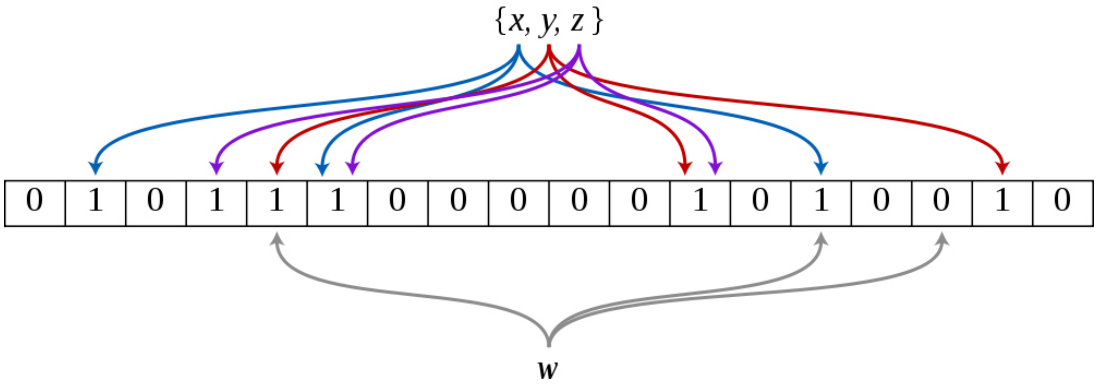

Figure 1: An example of Bloom Filter[1]

To figure out whether or not the element w is in the set, we apply k hash functions and get $h(w) = \{4, 13, 15\}$. Since the bit of index 4 and 13 of array is 1, and the bit of index 15 is 0, we can see that w is not included in the array.

As we see in figure 1, a bloom filter can misjudge the not-stored value as the stored value due to the stored value of the other elements. We decide expected probability of false positive, the size of filter m, and the number of hash functions k. The probability of false positive can be obtained as follows: For arbitrary x, the probability of any index of array A after implementing a hash function is $1/m$. On the

other hand, the probability of not obtaining 1 is $1-(1/m)$. When implementing k hash functions and putting n elements, we can obtain the probability that a certain bit of A is 1 by subtracting the probability of not being 1 from total probability. Since implementing a hash function and putting an element are independent, the formula is shown as follows:

$$1-\left(1-\frac{1}{m}\right)^{kn}$$

False positive occurs when all the index of the result of putting x into A is 1. Since executing a hash function is an independent event, the probability of false positive is k times the probability of a certain bit is 1.

$$\left(1-\left(1-\frac{1}{m}\right)^{kn}\right)^{k}$$

The above formula with the following condition can be estimated as follows:

$$\left(1-e^{\frac{-kn}{m}}\right)^{k} \quad \because \lim_{m\to\infty}\left(1-\frac{1}{m}\right)^{m}=\frac{1}{e}$$

Thus, the probability of occurring false positive is $\left(1-e^{\frac{-kn}{m}}\right)^{k}$, this has the minimum value when $k=\frac{m}{n}\ln 2$. As k increases, the probability of false positive decreases, but it increases again when k exceeds a certain number. Since k stands for the number of hash functions, operation costs should also be considered.

When the range of n, the number of nodes, is roughly determined like wireless sensor networks, we can decide m in consideration of the memory of node and k in consideration of operation performance. To get a proper value of m and k, we need to apply all the possible values of m and k in a valid range and find the trade-off point of memory and electronic consumption in false positive nodes.

It is easy to put a data in a bloom filter, but hard to delete. Since the other element can set 1 to the same index, it cannot be erased when the element setting 1 to a certain index is not unique. To solve this problem, Counting Bloom Filter is proposed[15]. The element can be decreased by putting a bit as an integer and increasing by 1, not by setting a bit as 1. However the weakness of this scheme is that storage space needs to be increased, since the array A has to be composed of several bits.

3.2 Bloom Filtered Address for Link Layer

Wireless sensor networks can decide the range of the number of devices for network according to application. By using this feature, we can apply a bloom filter to link layer address of device. In this paper, we separate preliminary phase and network phase in order to hide the address of link layer.

3.2.1 Preliminary Phase

In this stage, we define necessary factors of wireless sensor networks through the following steps:

- Define function BF() that generates the a bloom filter. BF is composed of k hash functions.
- Define the minimum number of data q which is inserted in a bloom filter. (eg. $q=5$)
- Define an inserted data I. I should be a value that node knows. (eg. MAC address, serial number etc.)

- Base station collects the data of devices in network. They are denoted as $I_1, I_2, \ldots, I_n$.
- Insert collected Is in a bloom filter. If false positive occurs, do revision by changing m, re-defining I and so on. Make sure false positive is not occurred, It can be done by defining MAC address twice.
- Base station generates random data $R_1, R_2, \ldots\ R_o$ which are not overlapped with device data.
- Put collected Is and generated Rs in a bloom filter and see whether or not false positive occurs. The false positive of R can be ignored and false positive of I is only proved. If there exists R that occurs false positive, then delete or change it so that false positive is not occurred.
- Deploy devices at an application environment. Each device generates $bfid$=BF(I) by using own I and stores it.

3.2.2 Network Phase

In network stage, a message is delivered through the following steps.

- To send message, base station put receiver's data in BF(). (eg. BF(I_2))
- For the randomness of BF(), base station put $q-1$ random data in BF(). (eg. BF(I_2, R_1, R_5, R_{10}, R_{11})

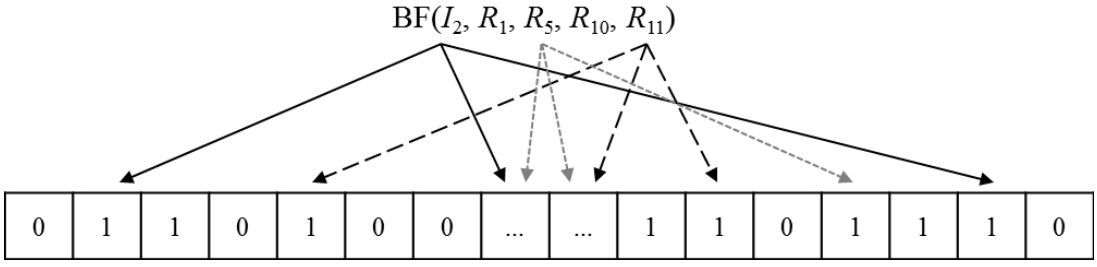

Figure 2: An example of BF() function

- Base station inserts the result of *recvID*=BF() in destination address field in a link layer packet. Since the result of BF() is not changed by input, when the storage space in base station is enough, we can precalculate the results of BF() of each element and save them. In this case, we can obtain *recvID* by implementing bit-or operation to precalculated BF() not by implementing BF() all the time. Figure 3 is an example of inserting *recvID* to ZigBee link layer message.
- Base station sends message.
- It checks whether or not there exists its $bfid$ in *recvID* of message receiver.
- Node deals with certain message when itself is included as a receiver. Otherwise it deletes the message.

4 Analysis

In this chapter, we analyze the proposed method and state how it hides link layer address from attackers.

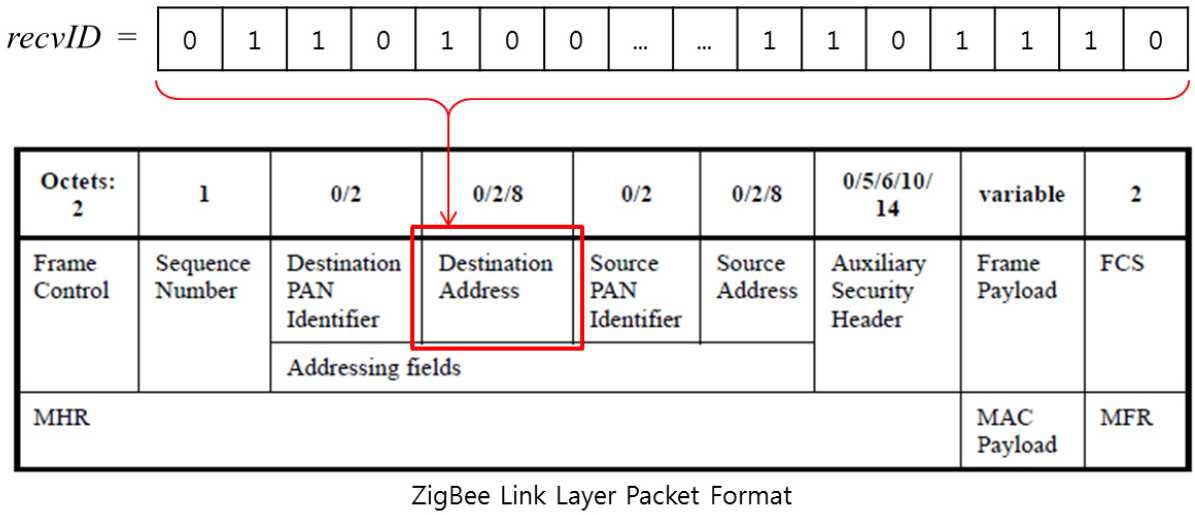

Figure 3: Inserting BF() address into ZigBee packet

4.1 Prevention of Recognizing Receiver Devices

Since we get the same result when we put the same I in BF(), attackers know the messages are pointing to the same receiver. We make the result of BF() different by putting a random value in set R so that attackers cannot distinguish the messages even though they are pointing to the same receiver. Figure 4 is the example of BF() with I and R. Suppose that the packet is sent to the attacker with the address (a). (b). (c), and (d) respectively as in figure 4. The attacker knows packet (a) and (c) are for the same receiver device. However when combining I and R like (e) and (f), even though the attacker eavesdrops the message, he or she cannot distinguish between the two with same I.

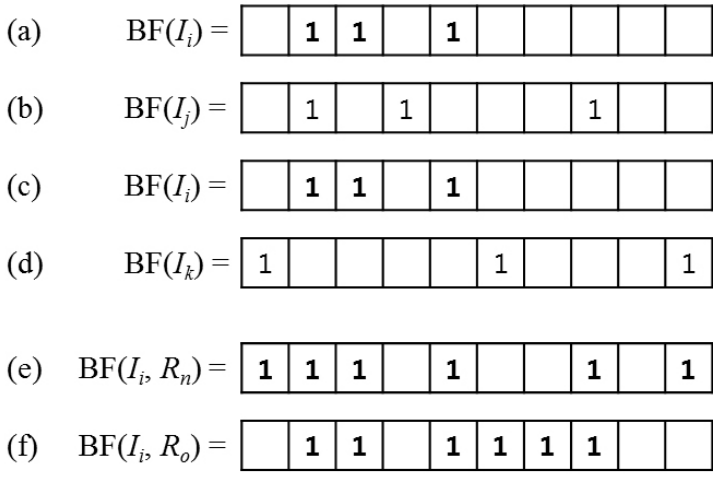

Figure 4: An example of randomization using R

4.2 Comparison to the previous schemes

The previous methods uses receiver's unique information to identify the receiver in link layer. When an attacker eavesdrops a message, he or she can identify the receiver of the message. However, our

method transmits receiver's unique identifier inserted in a bloom filter. The receiver put his information in a bloom filter and check whether his information is there. As long as the attacker do not know the receiver's unique identifier, he cannot identify the message receiver.

In the previous methods, a message header that is transmitted to the receiver is the same when the receiver is the same. When the attacker keeps eavesdropping, he can collect the message that goes to the same receiver. Even though the message is encrypted, it is possible for the attacker to recognize the message flow. Our scheme confirms that the receiver information in the message header changes everytime even though the receiver is the same. The receiver can check whether or not the message header has his information by using his unique information. However, since the attacker do not know the unique information of the receiver, he cannot know the message flow.

The previous methods implement mathematical or cryptographic operation to provide anonymity and perform special protocol. Our method includes fast operation based on hash function, and it takes little effort since each value is pre-computable. Moreover, we do not need to perform or build additional protocol since we only change the address of normal protocols. There is no similar research to our propose but we compare our scheme to the previous researches. Table 1 compares our scheme and previous schemes.

	Our scheme	Ben Greenstein et al.'s	Eun-Kyung Ryu et al.'s	Ronggong Song et al.'s
Goals	Non-distinguishability of receivers	Obfuscating whole packet	Anonymous mutual authentication	Secrecy and anonymity
Target system	Wireless sensor network	802.11 wireless local area network		
Methods	Bloom filter (hash functions)	Encryption and CBC-MAC	Bilinear pairing	Discrete logarithm
Keys per link	None	6	2	3
Compatibility	Any link layer protocol	No		

Table 1: Comparison of schemes for link layer privacy

5 Conclusion

In this paper, we propose the effective way of hiding address of link layer by using a bloom filter. It can be useful in an environment where arranged devices communicate like in wireless sensor networks. By pre-checking false positive availability, we can prevent errors by false positive in network operation. Since hash functions are faster than the other cryptographic operations, it can be used in sensor nodes which have constrained resource. If the storage space is enough, we can pre-calculate the result of hash functions and store before use. Our scheme can be adopted almost link layer protocol because it do not require specific protocol format. Anonymity, confidentiality, and integrity can be guaranteed when combining our proposed scheme with previous link layer security protocol that provide encryption and authentication.

In further research, we will measure the performance with actual nodes or simulation. We will also find the proper size of a bloom filter for the number of nodes, the proper number of hash functions and so on. Moreover, we will expand our research by hiding not only receiver's address but also sender's address.

References

[1] Bloom filter.svg. http://commons.wikimedia.org/wiki/File:Bloom_filter.svg.

[2] Wsn link-layer security frameworks. In J. López and J. Zhou, editors, *Wireless Sensor Network Security*, Cryptology and information security series, pages 142–163. IOS Press, 2008.

[3] N. S. Agency. Skipjack and key algorithm specifications. Technical report, National Security Agency, May 1998.

[4] L. Alliance. Liberty alliance project. http://www.projectliberty.org/.

[5] T. Alliance. Tinyos. http://www.tinyos.net/.

[6] T. Aura. Cryptographically Generated Addresses (CGA). IETF RFC 3972, March 2005. http://www.ietf.org/rfc/rfc3972.txt.

[7] J. Black and P. Rogaway. Cbc macs for arbitrary-length messages: The three-key constructions. In *Proc. of the 20th Annual International Cryptology Conference (CRYPTO'00), Santa Barbara, California, USA, LNCS*, volume 1880, pages 197–215. Springer-Verlag, August 2000.

[8] B. H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, July 1970.

[9] A. Broder, M. Mitzenmacher, and A. B. I. M. Mitzenmacher. Network applications of bloom filters: A survey. In *Internet Mathematics*, pages 636–646, 2002.

[10] L. Casado and P. Tsigas. Contikisec: A secure network layer for wireless sensor networks under the contiki operating system. In *Proc. of the 14th Nordic Conference on Secure IT Systems: Identity and Privacy in the Internet Age (NordSec'09), Oslo, Norway, LNCS*, volume 5838, pages 133–147. Springer-Verlag, October 2009.

[11] E. Commission. Seventh framework programme for research (fp7). http://ec.europa.eu/research/fp7/index_en.cfm.

[12] A. Dunkels. Contiki os. http://www.contiki-os.org/.

[13] M. Dworkin. Recommendation for block cipher modes of operation: Three variants of ciphertext stealing for cbc mode, October 2010.

[14] W. Ehrsam, C. Meyer, J. Smith, and W. Tuchman. Message verification and transmission error detection by block chaining, Feb. 14 1978. US Patent 4,074,066.

[15] L. Fan, P. Cao, J. Almeida, and A. Z. Broder. Summary cache: A scalable wide-area web cache sharing protocol. *IEEE/ACM Transactions on Networking*, 8(3):281–293, January 2000.

[16] B. Greenstein, D. McCoy, J. Pang, T. Kohno, S. Seshan, and D. Wetherall. Improving wireless privacy with an identifier-free link layer protocol. In *Proc. of the 6th International Conference on Mobile Systems, Applications, and Services (MobiSys'08), Breckenridge, CO, USA*, pages 40–53. ACM, June 2008.

[17] D. Guo, Y. He, and P. Yang. Receiver-oriented design of bloom filters for data-centric routing. *Computer Networks*, 54(1):165–174, 2010.

[18] K. initiative. Kantara. http://kantarainitiative.org/.

[19] D. Jinwala, D. Patel, and K. S. Dasgupta. Flexisec: A configurable link layer security architecture for wireless sensor networks. *Journal of Information Assurance and Security*, 4:582–603, 2009.

[20] P. Jokela, A. Zahemszky, C. Esteve Rothenberg, S. Arianfar, and P. Nikander. Lipsin: Line speed publish/-subscribe inter-networking. *SIGCOMM Computer Communication Review*, 39(4):195–206, August 2009.

[21] C. Karlof, N. Sastry, and D. Wagner. Tinysec: A link layer security architecture for wireless sensor networks. In *Proc. of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys'04), Baltimore, MD, USA*, pages 162–175. ACM, November 2004.

[22] T. Li, H. Wu, X. Wang, and F. Bao. Sensec: Sensor security framework for tinyos. In *Proc. of the 2nd International Workshop on Networked Sensing Systems (INSS'05), San Diego, CA, USA*, pages 145–150, June 2005.

[23] M. Luk, G. Mezzour, A. Perrig, and V. Gligor. Minisec: A secure sensor network communication architecture. In *Proc. of the 6th International Conference on Information Processing in Sensor Networks (IPSN'07), Baltimore, MD, USA*, pages 479–488. ACM, April 2007.

[24] PrimeLife. Primelife project. `http://primelife.ercim.eu/`.

[25] P. Project. Privacy and identity management for europe. `https://www.prime-project.eu/`.

[26] J. Ren, T. Li, and D. Aslam. A power efficient link-layer security protocol (llsp) for wireless sensor networks. In *Proc. of the IEEE Military Communications Conference (MILCOM'05), Atlantic City, NJ, USA*, pages 1002–1007. IEEE, October 2005.

[27] P. Rogaway, M. Bellare, and J. Black. Ocb: A block-cipher mode of operation for efficient authenticated encryption. *ACM Transactions on Information and System Security*, 6(3):365–403, August 2003.

[28] C. E. Rothenberg, P. Jokela, P. Nikander, M. Sarela, and J. Ylitalo. Self-routing denial-of-service resistant capabilities using in-packet bloom filters. In *Proc. of the 2009 European Conference on Computer Network Defense (EC2ND'09), Milan, Italy*, pages 46–51. IEEE, November 2009.

[29] C. E. Rothenberg, C. A. B. Macapuna, and A. Wiesmaier. In-packet bloom filters: Design and networking applications. *Computer Networks*, 55(6):1364–1378, 2011.

[30] E.-K. Ryu, E.-J. Yoon, and K.-Y. Yoo. More robust anonymous authentication with link-layer privacy. In *Proc. of the IEEE Asia-Pacific Services Computing Conference (APSCC'10), Hangzhou, China*, pages 441–446. IEEE, December 2010.

[31] R. Song and H. Tang. Laa: Link-layer anonymous access for tactical manets. In *Proc. of the IEEE Military Communications Conference (MILCOM'12), Orlando, NJ, USA*, pages 1–7. IEEE, October 2012.

Author Biography

Sang-ho Park received his B.S. and M.S. degrees in computer science and engineering from Sejong University, Seoul, Korea, in 2004 and 2006, respectively. He is currently pursuing a Ph.D. degree at the department of computer science and engineering, Sejong University, Seoul, Korea. His current research interests include computer network security and mobile security.

Jihyun Bang received her B.S. degree in mathematics and professional English from Ewha Womans University, Seoul, Korea, in 2014. She is currently pursuing a M.S. degree at the Graduate School of Information, Yonsei University, Seoul, Korea. Her current research interests include cryptographic protocol, computer network security, and smart card security.

Mirim Ahn received her B.S., M.S. degrees in computer science from George Mason University, Fairfax, VA, USA, Korea University, Seoul, Korea, in 1985, 1994, respectively. She is currently a principal researcher at Agency for Defense Development, Seoul, Korea. She has involved in projects for developing military conscription and mobilization system, military satellite system, surveillance and reconnaissance sensor network, etc. as a participating researcher and a project manager since 1986. Her current research interests include information security and privacy, network protocol, computer network, and cyber warfare simulation.

Woomin Lee received his B.S., M.S., and Ph.D. degrees in mechanical engineering from Chung-Ang University, Seoul, Korea, in 1990, 1992, and 2002, respectively. He is currently a principal researcher at Agency for Defense Development, Seoul, Korea. He has involved in project for developing surveillance and reconnaissance sensor network. as a participating researcher. His current research interests include network protocol, unmanned ground vehicle, and cyber warfare simulation.

Taekyoung Kwon received his B.S., M.S., and Ph.D. degrees in computer science from Yonsei University, Seoul, Korea, in 1992, 1995, and 1999, respectively. He is currently an Associate Professor of information at Yonsei University, Seoul, Korea. From 1999 to 2000, he was a Post-Doctoral Research Fellow at the University of California, Berkeley, CA, USA, and developed a cryptographic protocol, which was later standardized by IEEE P1363.2 and ISO/IEC JTC1 SC27 11770-4, respectively. From 2001 to 2013, he was a professor of computer engineering at Sejong University, Seoul, Korea. From 2007 to 2008, he was on sabbatical at the University of Maryland, College Park. In 2013, he returned to Yonsei University, Seoul, Korea. His current research interests include information security and privacy, applied cryptography, cryptographic protocol, network protocol, usable security, and human-computer interactions.