J Inf Process Syst, Vol.13, No.6, pp.1516~1526, December 2017
https://doi.org/10.3745/JIPS.03.0084

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Patch Integrity Verification Method Using Dual Electronic Signatures

JunHee Kim* and Yoojae Won*

Abstract

Many organizations today use patch management systems to uniformly manage software vulnerabilities. However, the patch management system does not guarantee the integrity of the patch in the process of providing the patch to the client. In this paper, we propose a method to guarantee patch integrity through dual electronic signatures. The dual electronic signatures are performed by the primary distribution server with the first digital signature and the secondary distribution server with the second digital signature. The dual electronic signature ensures ensure that there is no forgery or falsification in the patch transmission process, so that the client can verify that the patch provided is a normal patch. The dual electronic signatures can enhance the security of the patch management system, providing a secure environment for clients.

Keywords

Digital Signature, Electronic Signature, Integrity, Patch Distribution, Patch Management

1. Introduction

The scale of the global software market is expanding. As a result, various software weaknesses are being discovered and damage from these weaknesses is on the rise. Slammer Worm, which was distributed in 2003, infected many systems by using weaknesses in MS-SQL Server 2000 and MSDE 2000, resulting in the loss of 1.2 billion dollars worldwide [1]. In this era where serious damage is caused by software weaknesses, the coming of the Internet of Things (IoT) generation will lead to many more weaknesses in software, which implies that there is a need to effectively address such weaknesses.

There are many limitations when a company tries to independently manage patches in order to supplement their various software weaknesses. If the frequency of patch updates is increased, security risks are decreased, but operational costs will increase in return. On the other hand, reducing the frequency of patch updates will reduce operational costs in exchange for increased security risks. In order to achieve optimal patch management levels, there is a need to modify the patch distribution policies of patch supply companies as well as company patch update policies [2].

Many companies use the patch management systems offered by security agencies in order to resolve these limitations. The patch management system that offers optimal patch management to its clients will offer the following four-stage patch management process [3] (Fig. 1).

Manuscript received April 3, 2017; first revision July 3, 2017; accepted August 27, 2017.

Corresponding Author: Yoojae Won (yjwon@cnu.ac.kr)
* Dept. of Computer Science Engineering, Chungnam National University, Daejeon, Korea ({kjh89, yjwon}@cnu.ac.kr)

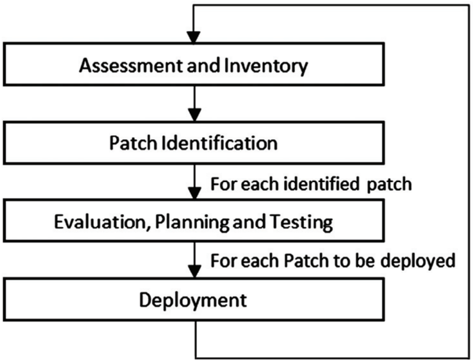
Assessment and Inventory
Patch Identification
For each identified patch
Evaluation, Planning and Testing
For each Patch to be deployed
Deployment

Fig. 1. Four-stage patch management process.

- *Assessment and Inventory*. The purpose of this phase is to accurately record what software components comprise the operational environment, what security threats and vulnerabilities exist, and whether an organization is prepared to respond to new software updates.
- *Patch Identification*. The purpose of this phase is to identify patches and software updates as they are released, determine whether they are relevant to the organization, and determine whether an update represents a normal or emergency change.
- *Evaluation, Planning and Testing*. The purpose of this phase is to decide, for any given patch, whether to deploy that patch into the operational environment, to plan how and when that deployment will take place, and to test the software update in a realistic operational environment to confirm that it does not compromise business critical systems or applications.
- *Deployment*. The purpose of this phase is to successfully roll out the approved software update into the operational environment while minimizing impact on system users.

Of the four stages, it is the last deployment stage that influences the resolution of weaknesses. In the distribution stage, the integrity of the patch delivered to the client must be guaranteed in order for the deployment to be regarded as successful. Thus, there must be a method that can absolutely guarantee the integrity of patches in the patch management system. However, the current patch management systems provided by security agencies are problematic because they are unable to perfectly guarantee integrity in the patches they offer to the client.

This paper will analyze related studies, patch management systems, and security incidents that occurred in patch management systems and propose a method that can guarantee the integrity of patches received by the client.

2. Related Studies

2.1 Design the Normalized Secure Patch Distribution & Management System [4]

This paper proposes seven requirements in a system for effective security patch distribution and management. The proposed requirements include security, safety, dependency, expandability, versatility, management, and convenience. Of these, security refers to the guarantee of patch integrity. In this paper, SSL communication based on authentication certificates between the server and client was used to secure security, and the hash function and CRM check method were used to guarantee the integrity of patches received from the server.

2.2 Design and Implementation of a Secure Software Architecture for Security Patch Distribution [5]

This paper proposes a safe security patch distribution system in order to provide patch distribution, authentication, integrity, and confidentiality when multiple vendors distribute patches. The proposed distribution system includes server-client authentication based on authentication certificates, securing confidentiality through DH key exchanges, and patch integrity guarantees through verifying electronic signatures and message digests.

2.3 Design the Multi-Platform Based Automatic Distribution Method of Security Patches with RMI and SSL [6]

If the security patch distribution system does not work properly, it may pose a significant threat to the security of the network. Thus, this paper has proposed a method that guarantees safety in security patch distribution systems. This paper has also proposed a system that authenticates the communication target by guaranteeing confidentiality and integrity regarding messages by using SSL protocols, and offers non-repudiation functions to guarantee safety.

Until now, research that was conducted for the purpose of guaranteeing patch integrity was able to resolve threats that may occur between the client and server that distributes patches. However, if the patch distribution server was hacked and the patch is already forged, the client will receive a forged patch from the patch distribution server and will be open to being infected by malicious code.

3. Cases of Patch Management System and Security Accidents

3.1 Patch Management System

Fig. 2 shows the patch distribution structure of patch management systems. Before the patch is delivered to the client, there is the process of patch collection, patch testing, and patch distribution [7].

- *Patch Collection.* Managers of security agencies check if new patches have been released through patch sites from vendors or independent patch provision services, then quickly collect patch files and information if a new patch has been released.
- *Patch Testing.* A test is conducted on the collected patch. The test examines the patch file's integrity and compatibility under various environments, then runs the patch in a virtual environment that is similar to the client that will be receiving it. Patches that have completed testing are uploaded to the patch distribution server.
- *Patch Distribution.* This process involves the distribution of patches that have completed testing to the client through the distribution server. The patch distribution server may take a hierarchical structure according to the scale or characteristics of the organization with which the client is affiliated, and the patch is distributed to the client in accordance with distribution

policies. Electronic signatures, encryption, and other techniques are used to safely distribute patches.

This patch distribution process may seem to have no issues. However, there is a critical problem in guaranteeing patch integrity because the client regards the patches received from the patch distribution server as completely reliable. The patch distribution server may use electronic signatures and encryption techniques during the process of delivering patches to the client, but if the patch distribution server is hacked and the patch becomes forged, the client will end up receiving a forged patch without any suspicion.

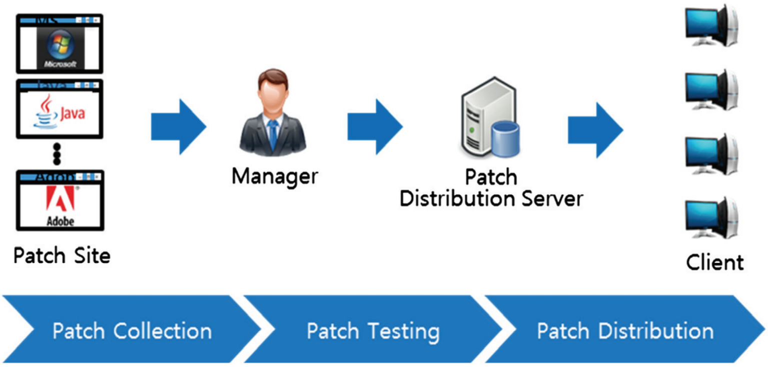

Fig. 2. Patch distribution structure of patch management systems.

3.2 Cases of Security Incidents

On March 20, 2013, in Korea, the financial network was completely paralyzed by malicious code because the integrity of patches provided by the patch management system was not properly verified. The number of PCs that were affected is estimated to be about 26,000. The master boot record (MBR) and the volume boot record (VBR) of these PCs that were infected by the malicious code were destroyed so that they could not be booted, and all existing data was lost [1].

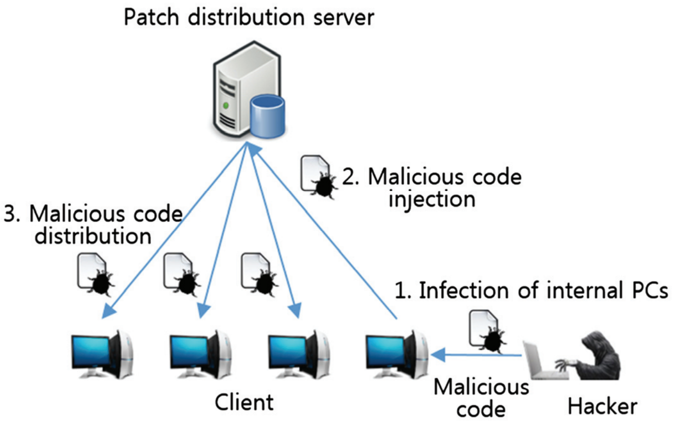

Fig. 3. Process of malicious code circulation.

Fig. 3 shows the process of how a terminal on the financial network got infected by malicious code.

First, the hacker planted malicious code on a PC inside the financial network. The hacker used the PC infected with the malicious code to insert this malicious code into the patch distribution server, then spread the malicious code to the client that received a patch from the patch distribution server. This client that trusted patches provided by the patch distribution server took the patch without any suspicion of malicious code, at which point the hacker paralyzed the financial network.

Although the patch distribution server delivered the patch after going through an electronic signature process, if it electronically signs malicious code instead of a patch, the client will still recognize this as a patch.

This patch distribution method has two issues. First is that the server that distributes the patch is connected to the client. This structure implies that the patch distribution server is open to hacking risks at any time. Second, the patch received by the client from the patch distribution server cannot be fully trusted to be a patch that was collected from the security agency.

In order to resolve these issues, there must be a method that can guarantee the integrity of patches even if the patch distribution server is infected by malicious code.

4. Security Requirements of Patch Distribution

The security requirements, which need to be satisfied in order to send a normal patch to a client during the patch distribution process within a patch management system, are verification and integrity, and their details are as follows.

4.1 Patch Sender Verification

It is an important problem in terms of security to decide whether a right patch distributor sends a patch. A client has to be provided with a patch from a trustworthy patch distributor. It is hard to tell whether a patch transmitted from an untrustworthy patch distributor is normal, so a client should be able to confirm that a patch sender is verified.

4.2 Patch Integrity

Even if a trustworthy patch distributor sends a patch to a client, the client should be able to verify whether the patch is normal. Patch integrity implies that a patch that a distributor intends to send gets transmitted safely to a client. If a patch is forged while the distributor stores or sends it, the client who receives it should be able to decide whether the patch is tampered.

5. Patch Integrity Verification Method Using Dual Electronic Signatures

During patch distribution, the existing patch management systems guaranteed the integrity of patches that are provided for clients by using digital signature, encryption, and packet inspection between a patch distribution server and client. However, if such a patch distribution server is hacked and its patch files are forged, there is no way that its client can confirm the integrity of the patch. The double digital signature suggested in this paper enables clients to check if a patch was forged by verifying the digital signature of a patch collection server even if a patch distribution server gets hacked.

This section explains the composition of the patch management system, the public key distribution method, and the patch integrity verification process for dual electronic signatures.

5.1 Patch Management System Composition

Fig. 4 shows the composition of a patch management system that guarantees patch integrity through dual electronic signatures. In the existing patch management system, there were security issues because the patch distribution system that is connected to the client began the direct distribution of patches to the client. To resolve this issue, the patch collection server, which is the server that begins the direct patch distribution, was placed behind the patch distribution server. The action carried out by each composition factor is as follows.

- *Patch Collection Server*. Manages patches that completed the collection and testing process, and receives the first electronic signature when the patch is delivered to the patch distribution server.
- *Patch Distribution Server*. Manages the patch signed with the first electronic signature from the patch collection server, and distributes the patch to the client in accordance with distribution policies. A second electronic signature is received upon distribution.
- *Client*. Verifies the dual electronic signatures, and checks the integrity of the patch received from the patch distribution system.

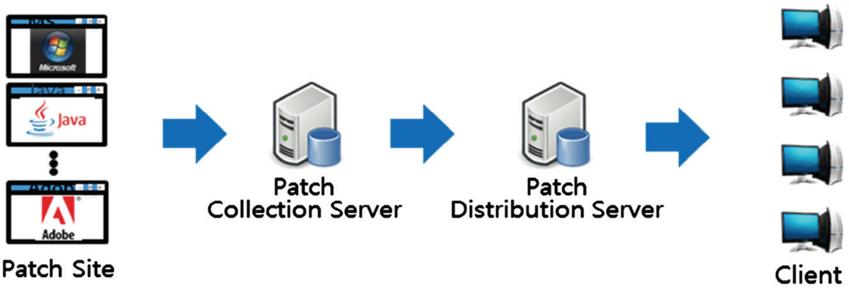

Fig. 4. Patch management system composition.

5.2 Public Key Distribution Method

Because the authentication-based public key distribution method must store authentication certificates and manage discarded certificates, it is difficult to handle the cost of storing, deleting, and reissuing authentication certifications. To overcome this issue, a public key was distributed from the patch collection server and patch distribution server using the method proposed in “Certificateless-Based Public Key Infrastructure using a DNSSE” [8].

To apply the method proposed in “Certificateless-Based Public Key Infrastructure using a DNSSEC” to the patch management system, the DNS server was replaced with the IP management server, and a public key distribution between the client, patch distribution server, and patch collection server was used instead of between the user and web server. The IP management server is a server that manages the IP of servers in the patch management system. Fig. 5 shows the public key distribution process.

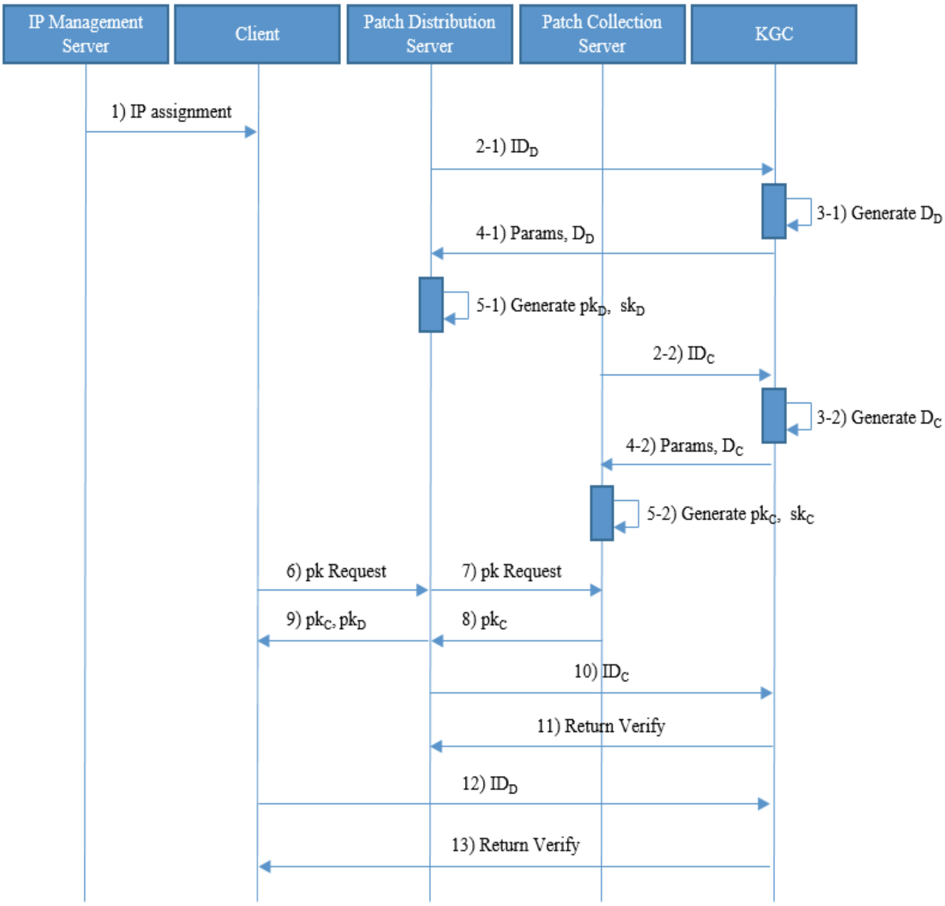

Fig. 5. Public key distribution process.

1. IP management server → Client : Sending an IP address
   When a client tries to obtain public keys, the IP management server sends an IP.

2-1. Patch distribution server → KGC : Sending an identifier
   The Patch distribution server sends its IP to KGC and requests information needed for creating a key.

2-2. Patch Collection Server → KGC : Sending an identifier
   The Patch collection server sends its IP to KGC and requests information needed for creating a key.

3-1. KGC : Generating D_D
   A partial key is generated based on the IP address sent by the patch distribution server.

3-2. KGC : Generating D_C
   A partial key is generated based on the IP address sent by the patch collection server.

4-1. KGC → Patch distribution server : Sending a partial-private key

KGC sends the partial-private key and params calculated from step 3-1 to the patch distribution server.

4-2. KGC → Patch collection server : Sending a partial-private key
KGC sends the partial-private key and params calculated from step 3-2 to the patch collection server.

5-1. Patch distribution server : Generating the public key and private key
Using the partial-private key and params it received, the patch distribution server creates its own public key and private key.

5-2. Patch collection server : Generating the public key and private key
Using the partial-private key and params it received, the patch collection server creates its own public key and private key.

6. Client → Patch distribution server : Requesting public keys
When a client first runs, a client requests public keys to the patch distribution server.

7. Patch distribution server → Patch collection server : Requesting a public key
When the patch distribution server receives the pk request, the patch distribution server requests a public key to patch collection server.

8. Patch collection server → Patch distribution server : Sending a public key
When the patch collection server receives the pk request, the patch collection server sends the created public-key.

9. Patch distribution server → Client : Sending public keys
When the patch distribution server receives the pk_C, the patch distribution server sends the pk_C and the own public key.

10. Patch distribution server → KGC : Sending an IP address and patch collection server's public key
To verify the public key received from the patch collection server, the patch distribution server sends the IP and public key of the patch collection server to KGC.

11. KGC → Patch distribution server : sending verified information
By sending verified information to the patch distribution server, KGC verifies that the patch distribution server has received the public key of patch collection server accordingly.

12. Client → KGC : Sending an IP address and patch distribution server's public key
To verify the public key received from the patch distribution server, the client sends the IP and public key of the patch distribution server to KGC.

13. KGC → Client : sending verified information
By sending verified information to the client, KGC verifies that the client has received the public key of patch distribution server accordingly.

The key point of the public key distribution process is that the client receives a public key from the patch collection server through the patch distribution server. By doing this, the patch collection server is disconnected from the client, keeping it safe from external threats.

5.3 Patch Integrity Verification Process

Fig. 6 shows the process of verifying patch integrity. There are a total of two electronic signatures in order to verify patch integrity, and the client verifies each electronic signature to verify the integrity of the patch. The following is a more detailed explanation.

- *1st Electronic Signature*. When a patch that was collected and tested by a security agency manager is uploaded to the patch collection server, the manager uses his or her personal key (sk_C) to electronically sign the patch. Patches with an electronic signature are sent to the patch distribution server.
- *2nd Electronic Signature*. The patch received from the patch collection server is managed in the patch distribution server while keeping the electronic signature from the patch collection server. The patch distribution server distributes the patch to the client in accordance with distribution policies. When the patch is distributed, a personal key (sk_D) is used to add another electronic signature on the patch.
- *Verification*. When the client receives a patch from the patch distribution server, the patch will be electronically signed through a personal key (sk_C) from the patch collection server and a personal key (sk_D) from the patch distribution server. The client will then use a public key (pk_C) from the patch collection server and a public key (pk_D) from the patch distribution server that was received in advance to verify the dual electronic signatures. If the dual electronic signatures are successfully verified, the patch's integrity can be guaranteed.

By verifying patch integrity using dual electronic signatures, the client can check the patch's integrity even if the patch distribution server is hacked and the patch becomes forged.

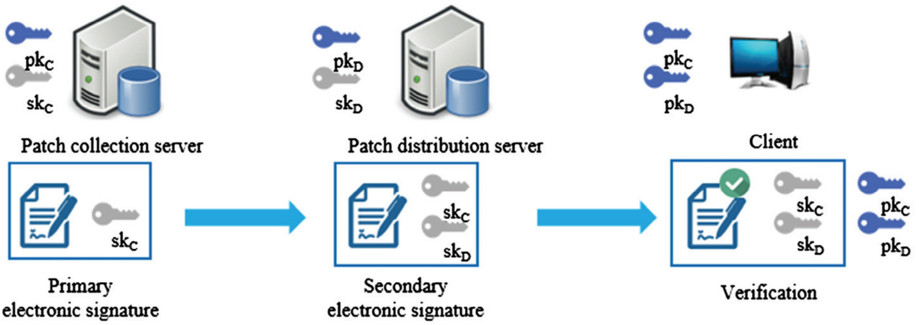
pkC
skC
Patch collection server
skC
Primary electronic signature
pkD
skD
Patch distribution server
skC
skD
Secondary electronic signature
pkC
pkD
Client
skC
skD
pkC
pkD
Verification

Fig. 6. Patch integrity verification process.

6. Evaluation of Security Requirements of Patch Distribution

This study conducts a digital signature for a patch in order to satisfy the security requirements for a patch distribution process.

A digital signature is created using a pair of private and public keys. A private key is its owner's unique key; if one performs a digital signature using their private key, the public key that pairs with the private key is the only means to verify the digital signature. If the public key fails to verify the signature, it is possible to decide that the patch sender cannot be trusted. This study guarantees that a patch sender is trustworthy by performing a primary digital signature in a patch collection server and a secondary one in a patch distribution server.

In addition, digital signatures can be utilized to assure patch integrity. Because a digital signature is applied to the hash value of a patch, the comparison of the decoded value of the digital signature and the hash value extracted from the patch enables a client to inspect the integrity of the patch file received.

7. Conclusions

As the importance of patches increases, patch management systems that can effectively manage these patches have become the center of attention. One of the most important factors in the patch management system is guaranteeing the integrity of patches that are provided to the client. Many studies have been conducted in order to achieve this, but the patches that are provided from patch management systems are still unreliable.

This paper has studied a method of guaranteeing the integrity of patches that are received by the client from the patch management system. Studies on patch distribution, the patch management system, and cases of security incidents were analyzed to derive relevant issues. To resolve the deduced issues, this paper proposed a method where dual electronic signatures are received from the patch collection server and patch distribution server, and the client verifies these electronic signatures. By using this method, patch integrity can be guaranteed and a safer patch management system can be built

Acknowledgement

This work was supported by the Institute for Information & Communications Technology Promotion (IITP) grant, funded by the Korea government (MSIP) (No. B0717-16-0099, IoT Security Vulnerabilities Search, Sharing and Testing Technology Development).

References

[1] J. W. Shin, “Status of infringement accidents through major internet accident experiences in South Korea,” *Internet & Security Focus*, no. 9, pp. 36-53, 2013.

[2] H. Cavusoglu, H. Cavusoglu, and J. Zhang, “Economics of security patch management.” in *Proceedings of 5th Workshop on the Economics of Information Security (WEIS 2006)*, Cambridge, UK, 2006.

[3] Centre for the Protection of National Infrastructure, *Good Practice Guide Patch Management*. London: Centre for the Protection of National Infrastructure, 2006.

[4] S. Lee, Y. J. Kim, T. S. Sohn, J. S. Moon, J. T. Seo, E. Y. Lee, and D. H. Lee, “Design the normalized secure patch distribution & management system,” *Journal of the Korean Institute of Information Scientists and Engineers*, vo. 31, no. 2I, pp. 502-504, 2004.

[5] T. S. Sohn, J. W. Seo, J. S. Moon, J. T. Seo, E. G. Im, and C. W. Lee, “Design and implementation of a secure software architecture for security patch distribution,” *Journal of the Korea Institute of Information Security and Cryptology*, vol. 13, no. 4, pp. 47-62, 2003.

[6] S. Lee, Y. J. Kim, J. S. Moon, J. T. Seo, D. S. Choi, and E. K. Park, “Design the multi-platform based automatic distribution method of security patches with RMI and SSL,” *Journal of the Korean Institute of Information Scientists and Engineers*, vol. 31, no.1A, pp. 283-285, 2004.

[7] T. Bartoletti, L. A. Dobbs, and M. Kelley, “Secure software distribution system,” in *Proceedings of 20th NIST-NCSC National Information Systems Security Conference*, Baltimore, MD, 1997, pp. 191-201.

[8] H. Im, J. Kang, and J. H. Park, “Certificateless based public key infrastructure using a DNSSEC,” *Journal of Convergence*, vol. 6, no. 3, pp. 26-33, 2015.

JunHee Kim

He received his B.S. and M.S. degrees in the Department of Computer Science Engineering from Chungnam National University, Korea, in 2015 and 2017, respectively. He is currently developing a big data collection and analysis system at the Korea Research Institute of Bioscience and Biotechnology.

Yoojae Won

He received his B.S. and M.S. degrees all in the Department of Computational Statistics from Chungnam National University, Korea in 1985 and 1987, respectively. and received his Ph.D. in the Department of Computer Science Engineering from Chungnam National University, Korea, in 1998. From February 1987 to February 2001, he researched wireless Internet information security at Electronics and Telecommunications Research Institute. From March 2001 to August 2004, he researched mobile security at AhnLab. From September 2004 to February 2014, he researched incident handling and was in charge of management planning at Korea Internet & Security Agency. He is currently a professor of the Department of Computer Science Engineering in at Chungnam National University.