TPS: TPM 및 파일 가상화를 통한 개인정보보호 자동화 시스템 디자인 및 구현

Design and Implementation of an Automated Privacy Protection System over TPM and File Virtualization

정혜림*, 안성규*, 김문성*, 박기웅** [1)]
Hye-Lim Jeong, Sung-Kyu Ahn, Mun Sung Kim, Ki-Woong Park

(34520) 대전광역시 동구 대학로 62 대전대학교 전산정보보호학과, 융합컨설팅학과*
(05006) 서울특별시 광진구 능동로 209 세종대학교 정보보호학과**
{hyello13, yiimfn}@gmail.com, k4022@daum.net, woongbak@sejong.ac.kr

요 약

본 논문에서는 TPM(Trusted Platform Module) 기반의 개인정보보호 자동화 시스템, TPS(TPM-enhanced Privacy Protection System)를 제안한다. TPS는 클라이언트 내 저장된 문서 중 개인정보를 포함하는 문서를 주기적으로 탐지하고 개인정보가 탐지된 문서를 암호화하여 서버에서 이를 관리하도록 하는데, TPM 기반의 키 관리 기법 및 클라이언트 시스템의 무결성 검증을 통해 비정상 상태의 클라이언트에 대한 개인정보 포함 문서의 열람을 제한하여 보안성을 높였다. 또한 개인정보가 포함된 문서가 암호화 되어 원격 서버에 저장이 되나, 사용자에게는 일반 문서 접근과 동일한 사용자 인터페이스를 제공하기 위한 VTF(Virtual Trusted File) 인터페이스를 제안하고 이를 구현하였다. 이를 통해 TPS는 개인정보 탐지, 암호화, 원격 서버로의 저장까지의 일련의 과정을 자동 수행하도록 하여 사용자 관점에서의 개입 없이 개인정보보호법을 준수를 자동화 하는 시스템을 구현하였다.

Abstract

In this paper, we propose the TPS (TPM-enhanced Privacy Protection System) which is an automated privacy protection system enhanced with a TPM (Trusted Platform Module). The TPS detects documents including personal information by periodic scanning the disk of clients at regular intervals and encrypts them. Hence, system manages the encrypted documents in the server. In particular, the security of TPS was greatly enhanced by limiting the access of documents including the personal information with regard to the client in an abnormal state through the TPM-based platform verification mechanism of the client system. In addition, we proposed and implemented a VTF (Virtual Trusted File) interface to provide users with the almost identical user interface as

※ This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government (MSIP) (NRF-2017R1C1B2003957, NRF-2016R1A4A1011761)

1) corresponding author(교신저자), * : co-first author

논문투고일자 : 2017. 03. 24 심사일자 : 2017. 03. 30 게재확정일자 : 2017. 04. 21

general document access even though documents containing personal information are encrypted and stored on the remote server. Consequently, the TPS automates the compliance of the personal information protection acts without additional users' interventions.

키워드: 개인정보보호, TPM, 접근 제어
Keyword: Privacy Protection, TPM, Access Control

1. Introduction

Personal information protection act was amended in 2015 as a reaction for the increased number of personal information leaks in Korea [1] [2] [3]. The personal information protection act of Korea consists of three main parts. The first act is that any document containing personal information must be encrypted [4]. The second act is that a document containing personal information must be stored on an isolated storage area or unit [5]. At last, when deleting documents containing personal information, the deletion must be performed in such a manner that cannot be recovered and deleted permanently [6]. These legislative acts cause inconveniences in managing documents containing of personal information to administrators in charge of such task. As our previous work, we have conducted a research on a system, termed TEAM [7] to resolve these inconveniences from the users' perspective. Our previous work, TEAM, allows users to comply with the personal information protection act in regard to documents containing personal information stored in clients' PCs. However, the system requires high level of security to the key management used in the encryption algorithm when it comes to the encryption process of documents containing personal information. In result, it could be a cause of further leaks when encryption key is managed with a lower security standards, which leads to leaks of stored documents within storage by decrypting the encrypted storage system through using the leaked key. It can lead to the leakage of stored documents in storage through decryption of the encrypted storage system, and leakage of documents containing personal information can bring a legal issue. Therefore, high security for managing keys for documents containing personal information and security storage must be qualified.

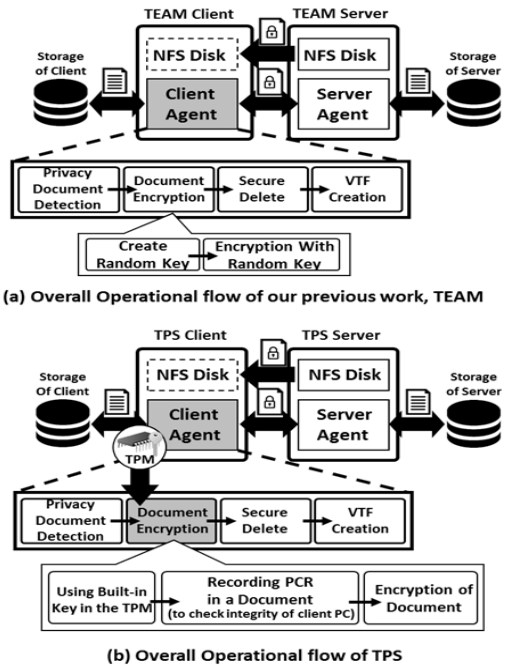


Fig 1. Comparison of TEAM system and TPS

In this paper, we propose an automated privacy protection system enhanced with a TPM, termed TPS (TPM-enhanced Privacy System) as a remedy to solve the key management issue of TEAM. As shown in Fig 1., our previous work, TEAM, proposed software-based key management. TPS that is a system proposed in this paper makes use of TPM to perform hardware-based key management and platform integrity verification for enhancing the security of the privacy protection system. Specifically, TPS performs hardware-based encryption and decryption through *PCR* (Platform Configuration Register) and *Seal/Unseal* function of TPM. The *PCR* is used as a value for integrity verification of client system. The *Seal* has encrypted documents containing personal information through *PCR* value. The processes are performed by hardware-based method and security is enhanced in management of key since the key uses the status value of the client system. In addition, if the hard disk stored with encrypted documents through *Seal* process moves to another computer system, decryption of documents cannot be performed due to the differences shown in *PCR* values. If the TPS encrypts the documents containing personal information through *Seal* function of TPM, the TPS sends and stores them to an isolated storage server. After such process has taken place, the TPS deletes the original files from the client, and creates VTF (Virtual Trusted File) in the client PC. The VTF is able to accesses documents stored in a storage server of TPS and receives them from a storage server. Moreover, the VTF performs the *Unseal* function which is a decryption function of the TPM from the client. It is applied to allow user to open documents containing personal information. During the *Unseal* process, if the *PCR* which is the status value of client PC is checked and matched to a current value of *PCR*, it will conduct decryption function. As such, we propose a personal information protection system that is more secure than the TEAM system proposed in the previous work.

This paper consists of five sections. In section 2, we describe the TEAM system that was used in the previous study using TPM and TPS which differs from the proposed system using TPS in this paper. Section 3 describes the implementation, structure, and data structure of TPS. In section 4, we measured the delay time taken to view documents using TPS as a hardware and software-based encryption in the previous study while measuring the time delay when viewing documents by file size. Section 5 is the conclusions of this paper.

2. Related Work and Concept of TPM

As described above, in this paper, we propose a TPS system that conducts hardware-based encryption to solve the key management problem of TEAM, which is a system for providing convenience of complying with the personal information protection act proposed in the previous research. However, a lot of related works on the system for such key management and personal information have been conducted[8] [9]. In this section, we describe TEAM system and discuss related works. After that, we describe the TPM, which is a hardware-based encryption used by the system.

2.1 Related Work

Mowbray, et al [10] proposed a client-based privacy management mechanism. They implemented

a client-based privacy manager to reduce the risk of leakage of both important data and personal information. Before the user sends crucial data and personal information to the cloud server, the privacy manager obfuscates and encrypts them. After that, the privacy manager sends them to the cloud server. The user accesses documents through the web to view them uploaded to the cloud server.

However, in case of the TPS proposed in this paper, the user does not need to check whether the personal information is included in the document. The agent of TPS automatically detects it so that user does not directly involve in upload to the storage server. The TPS has transparency because user can conduct the same operation as using the existing client environment to view the documents stored on the storage server by accessing with the VTF of the virtualization concept of the original document without accessing through the web.

Ji, Yi-mu, et al [11] proposed a personal information protection solution. It is definitely designed to differentiate security levels in accordance with the security requirements of users' personal information data stored in the cloud system. Each security level is classified into three levels such as high security level, intermediate security level, and low security level. This solution provides the ability to store user's personal information more securely. However, when compared to the method presented in this paper, the following differences occur when it comes to security and efficiency.

When the security of the personal information stored in the client is processed by the server, overhead such as key management occurs in the server. The TPS proposed in this paper has solved this problem by applying the encryption process for personal information on the client side. In addition, there is a problem that high security has to be applied from the server side for security of the data stored in the cloud server. If data is leaked due to security problems on the server side, there is a risk that user's personal information may be exposed outside. However, in the TPS system proposed in this paper, as each data was encrypted and uploaded by using TPM as a hardware-based encryption of the client, this problem was solved through designing the original data to be recovered by the client even though the data on the server was leaked.

Rahumed, Arthur, et al [12] proposes a cloud backup system for configuring the safe backup environment of data backup system using cloud storage activated in personal and enterprise computing environment. It performs the encryption and decryption process during data backup and ensures complete deletion of data. If a user deletes a particular backup version or data stored in the cloud storage, everyone cannot even access that data permanently. In case of this related work, there is an advantage that a safe and efficient environment can be configured when it comes to personal information protection using the data backup environment of cloud storage.

However, in case of this related work's environment, there is a problem that vulnerability may happen in client environment before data backup. In case of the TPS proposed in this paper, we have configured more secure personal information protection environment by configuring the data to be decrypted only in the PC of the same environment using *Seal* and *UnSeal* operation of the TPM.

2.2 Concept of TPM and how to use TPM in TPS

TPM is a hardware-based security chip that carries out encryption and key management functions which was created by an international industry standard organization within computer operation and cyber security called TCG. It is considered to be a standard function for a security chip to solve a vulnerability of software-based encryption key management. The TPM functions are carried out for ensuring platform integrity, disk encryption, and encryption. The TPS, a proposed system in this paper, takes advantage of *PCR* and *Seal* technology of TPM [13].

The *PCR* stands for Platform Configuration Registers, which records the state of all PCs from BIOS to Boot as the value for integrity check for client system. *Seal* and *Unseal* function conducts data encryption with TPM hardware built-in key and assigned specific *PCR* value. In case of *Unseal*, when the *PCR* is encrypted with a specific *PCR* value through the *Seal* of the document, decryption is carried out by matching *PCR* value and *AIK(Attestation Identity Key)* as a TPM built-in Private key. If the value of *PCR* match the value at the time of performing the *Seal*, the decryption is performed as well.

It conducts the encryption through the *PCR* value from TPM and the *Seal* function when the TPS detects documents containing the personal information through the agent. Furthermore, *Unseal* function must be performed to decrypt documents containing personal information which the *Seal* function is conducted. As described above, decryption is carried out if *Unseal* matches the *PCR* value from the time when *Seal* was performed. It can enhance security for key management and system integrity when both are properly used.

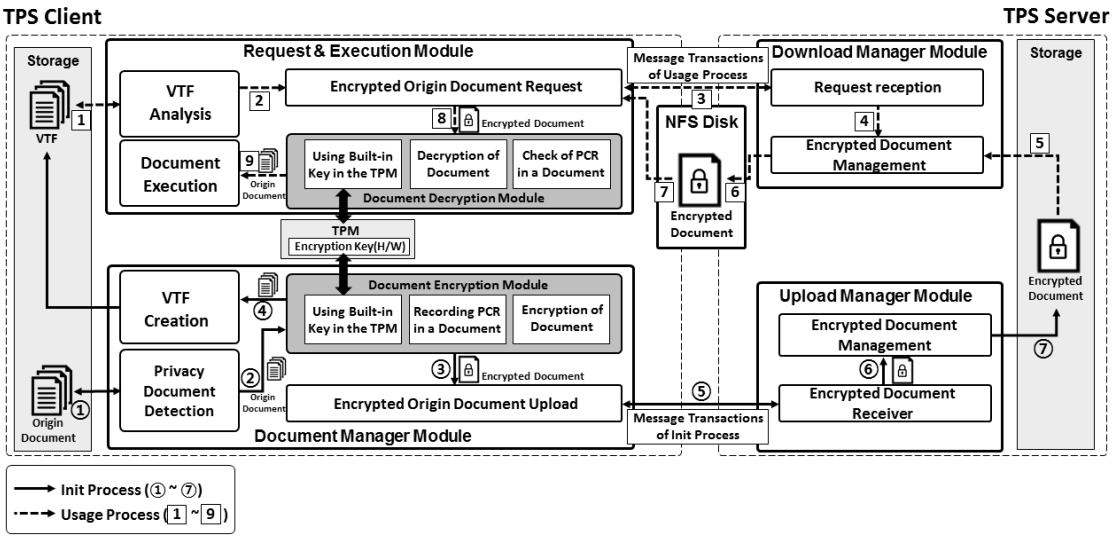


Fig 2. Overall Architecture and Operational Flow of TPS

3. Design and Implementation of TPS

3.1 Overall Architecture of TPS

The TPS is composed of agent in client, TPM, and VTF storage server. The implementation process of the system is divided into an *Init* process and a *Usage* process. In the *Init* process, the client agent retrieves the character string of personal information stored with the client and detects the character string containing personal information. Furthermore, the client agent conducts a process to ensure that the detected documents are encrypted and complies with personal information protection act. After such process has taken place, it is uploaded to the storage server to manage documents containing personal information. The VTF acts as a catalyst to access documents containing personal information uploaded to the storage server and producing virtual document from the original document. In the *Usage* process of the TPS, a document is provided to end user by using decryption function of TPM after a user executes VTF function to access the personal information containing document stored in the storage server.

3.2 Message Transaction of *Init* Process

Init process shown in Fig 3. presents the *Init* process of the TPS and it shows how the documents are uploaded from the client to the storage server. The client agent detects the documents containing personal information in the storage location of the client PC regularly during the *Init* process. The process for detected documents are then carried out by the agent to comply with the encryption and personal information protection act through the *Seal* function within the TPM. The process to comply with personal information protection act includes 'encryption', 'permanent deletion', and 'storing in separate storage'. The encryption part of it was carried out through hardware-based encryption in the TPM from a client. Documents containing personal information are encrypted through the *Seal* function of TPM. At this precise moment, encryption is performed with *PCR* value. In addition, the hash value and name of the document are recorded using the Mapping Table. The encrypted documents are uploaded to the storage server and the original documents in the client have to be deleted by overwrite deletion [14] in forensic style to limit any potential for recovery. The client creates VTF files in the client PC to access the documents stored on the storage server. This allows the client to request access to the documents through the VTF following virtualization concept of the original document.

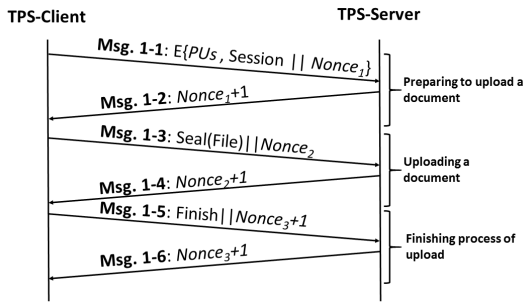


Fig 3. Message Transaction of *Init* Process

Fig 3. shows the Message transaction of the *Init* process to detect documents containing personal information in the documents stored from the client storage at the beginning of the *Init* process while complying with the personal information protection act. As shown in Fig 3., the client sends their session to the server in

preparation for document transmission (*Msg. 1-1*). The storage server not only receives such request but also verifies the session while sending a confirmation message to the client (*Msg. 1-2*). The encrypted personal information documents are uploaded, and the client can send a sealed document to the storage server afterwards (*Msg. 1-3* and *Msg. 1-4*). Following the above process, the client conducts safe deletion and VTF creation of the original document. If all documents containing personal information are sent to the storage server, upload completion signal is sent to the storage server with the session (*Msg. 1-5*). The storage server completes the *Init* process by confirming such and sending a termination confirmation message (*Msg.1-6*) to the client.

3.3 Message Transaction of *Usage* Process

Usage process in Fig 2. shows the *Usage* process of TPS. If the user requests the documents by running VTF, it will perform document delivery process from the storage server to the client. Agent requests the viewing of document to the storage server through the document information of the VTF from the agent when the VTF from the document requested is executed. The storage server sends the document in accordance with the request made from the client and NFS. The client receives requested information and allows the agent to perform the *Unseal* process to provide what was requested. The *Unseal* process functions to check whether information is matched with the *PCR* from current value of the client PC when the *Seal* was implemented, and provides user with information by decrypting stored information.

When documents are completed by users, the agent verifies whether they are modified by comparing the hash value of the documents from the mapping table. It, then will perform *Seal* process again as how it was done during the *Init* process to update the mapping table if it is confirmed that the document has been modified.

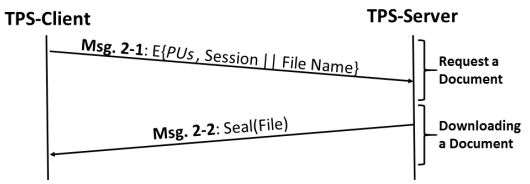


Fig 4. Message transaction of the *Usage* Process

Fig 4. shows the Message transaction of the *Usage* process. The *Usage* process sends the access request message to the storage server (*Msg. 2-1*) to view the original documents connected to executed VTF while the client user runs VTF during the entire process is taking place. The document request function order is sent to the storage server with the client's session value and the requested document name. The server receives above information and sends the corresponding Sealed document to the client (*Msg. 2-2*). Following the request process, the client agent conducts *Unseal* function which is a decryption function to allow user to access the document. Once document modification by user is completed, the agent extracts the hash value of the document and verifies whether any modifi-cation has been implemented by comparing to contained records in the mapping table of client. If it is confirmed that the modification has been made, the agent conducts *Seal* function in the same manner as the *Init* process. Furthermore, the agent uploads it to the server and carries out the process of deleting the original document.

This concludes the access process to documents containing personal information stored in the storage server from the client.

3.4 Implementation of TPS

The client implementation environment of TPS proposed in this paper was composed of Intel i5-4690 and 8GB DDR3 RAM, and was developed in 64bit window. The TPM in the client is TPM 2.0 version. The server implementation environment of TPS consists of Intel i3-4370 and 12GB DDR3 RAM and the operating system is Windows 64-bits version.

Encryption in the client has conducted the *PCR*, *Seal* and *Unseal* of TPM as a hardware-based encryption. Deletion of original document was performed by overwrite deletion with Guttmann algorithm[15]. SHA-1[16] technology was implemented to extract hash values for documents including personal information. NFS[17] technology was applied to send and receive documents between client and storage server.

4. Evaluation

4.1 Comparison to delay time of encryption operations

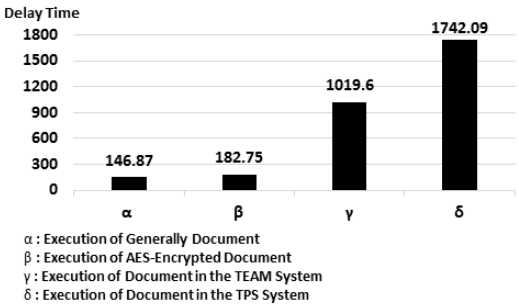


Fig 5. Comparison of file execution delay time for each system

We have compared to the TPS proposed in this paper and system in the previous research in terms of delay time. As for case 'α', the delay time was generally measured when the document was executed in Fig 5., and in case of 'β', delay time about process of the document execution was measured by performing decryption of AES[17] encrypted document. In case of 'γ', TEAM system in previous study has accessed the document stored in the server. As to 'δ', after performing of both AES decryption and decompression, the delay time was measured when the document was provided to the user. In all cases, it was measured by 1 KB document.

The measurement results are as follows. The case 'α' was measured at 146.879 ms and the case 'β' was measured at 182.758 ms. The case 'γ' was measured at 1019.6 ms and the case of 'δ' was measured at 1242.93 ms. In case of 'γ', the delay time of the TEAM system is approximately five times higher than decryption and delay time of document execution of case 'β'. Unlike the case 'β' and case 'γ' was performed by additional functions such as automatic input of decryption key and document compression. In case of 'β', the user key input time was not measured. The previous research system has advantage of providing users with convenience by automatically complying with personal information protection act.

In case of 'γ', the delay time of decryption and document execution in TPS proposed in this paper was measured to be 200ms higher than the delay time of c since it performs the process of Unsealing using *PCR* which is a hardware-based TPM. However, TPS has higher security than TEAM system as the previous research when it comes to key management.

4.2 Measuring delay time of viewing file by file size

When the VTF of TPS proposed in this paper is executed, the delay time until it is provided to the user by file size was measured and illustrated (Fig 6.). In case of 1KB, the delay time of 1242.93 ms was measured and in case of 100 KB, the delay time of 1470.84 ms was measured. In case of 1000 KB (1MB), the delay time of 1966.89 ms was measured. In case of 5000 KB (5MB), the delay time of 2641.73 ms was measured. Furthermore, in case of 10000 KB (10MB), the delay time of 3849.47 ms was measured as well.

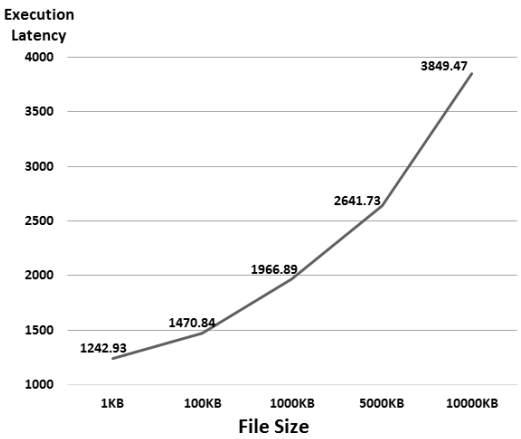


Fig 6. Comparing file execution Latency for each file sizes

The delay time of 1 KB and 10000 KB (10 MB) was about three times higher. Depending on the file size, the delay time in performing the TPM's *Unseal* function has increased. It is generally equal to increasing the decryption algorithm execution time in accordance with the file size. Thus, as the file size increases, it is a common result that the execution time of *Unseal* function increases in the TPS.

5. Conclusion

In this paper, we propose the TPS (TPM-enhanced Privacy Protection System) which is an automated privacy protection system enhanced with a TPM (Trusted Platform Module). TPS has performed encryption of documents containing personal information and enhanced security regarding key and document leaks through *PCR* as well as *Seal/Unseal* function of TPM. The original data can be decrypted only by the client system even though the data is leaked through the TPM. A comparison test was performed for measuring performance in terms of delay time of the TPS and conventional system, TEAM. As a result, it was measured at 200 ms higher than previous work, TEAM, due to the operational overhead of *Seal/Unsealing* using the *PCR* of TPM. However, TPS brings better security than TEAM. Our further work will accelerate the detection of stored personal information in the client through the GPGPU and will provide an efficient mecha-nism for storage and user data processing through server-side cloud environment configuration.

Acknowledgement

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIP) (NRF-2017R1C1B2003957, NRF-2016R1A4A1011761)

참고문헌

〔1〕 Sophia Yan, K.J. Kwon, "Massive data theft hits 40% of South Koreans", CNN Money, January 21, 2014

〔2〕 Min-Jeong Kim, Namgil Heo, Jinho Yoo, "A

Study on the Stock Price Fluctuation of Information Security Companies in Personal Information Leakage", Journal of the Korea Institute of Information Security and Cryptology 26(1), 2016.2, 275-283(9 pages)

〔3〕 Y.C-W, "Risk of Information Leak Growing in Korea", KOREA IT TIMES, July 10th, (2015)

〔4〕 Korea Personal Information Protection Act. Article 24, paragraph 3

〔5〕 Korea Personal Information Protection Act. Article 21, paragraph 2

〔6〕 Korea Personal Information Protection Act. Article 21, paragraph 3

〔7〕 Hye-Lim Jeong, Ki-Woong Park, "TEAM : Virtual Synchronized File-based Transparent and Privacy-Enhanced Storage System", IJSIA, 2016.09 , pp.285-294

〔8〕 박기웅, "TPM 기반 위변조 방지형 디지털 운행기록 장치 설계 및 구현", 한국차세대컴퓨팅학회 논문지, Vol.9, No.4, pp. 6-13, 2013.08

〔9〕 서병문, "랜덤 넌스와 타임스탬프 기반의 개선된 사용자 인증 스킴, 한국차세대컴퓨팅 학회 논문지, KCI, Vol.8, No.6, pp.69-75, 2012.02

〔10〕 Mowbray, Miranda, and Siani Pearson. "A client-based privacy manager for cloud computing." Proceedings of the fourth international ICST conference on COMmunication system softWAre and middlewaRE. ACM, 2009, pp.5.

〔11〕 Ji, Yi-mu, et al. "A privacy protection method based on CP-ABE and KP-ABE for cloud computing." Journal of Software 9.6, 2014, pp.1367-1375.

〔12〕 Rahumed, Arthur, et al. "A secure cloud backup system with assured dletion and version control." 2011 40th International Conference on Parallel Processing Workshops. IEEE, 2011, pp.160-167.

〔13〕 Park, Ki-Woong, et al. "THEMIS: A Mutually verifiable billing system for the cloud computing environment." IEEE Transactions on Services Computing 6.3, 2013, pp. 300-313.

〔14〕 Joukov, Nikolai, and Erez Zadok. "Adding secure deletion to your favorite file system." Security in Storage Workshop, 2005. SISW'05. Third IEEE International. IEEE, 2005.

〔15〕 GUTMANN, Peter. Secure deletion of data from magnetic and solid-state memory. In: Proceedings of the Sixth USENIX Security Symposium, San Jose, CA. 1996. p. 77-89.

〔16〕 Eastlake 3rd, D., and Paul Jones. "US secure hash algorithm 1 (SHA1)", No. RFC 3174, 2001.

〔17〕 MUTHITACHAROEN, Athicha; CHEN, Benjie; MAZIERES, David. A low-bandwidth network file system. In: ACM SIGOPS Operating Systems Review. ACM, 2001. p. 174-187.

〔18〕 Daemen, Joan, and Vincent Rijmen. "AES proposal: Rijndael", 1999

〔19〕 최동훈, 조희승, 박기웅 " 이질적 고성능 클라우드 컴퓨팅을 위한 확장형 Openstack의 개발 및 평가", 한국차세대컴퓨팅학회,KCI, 2016.09, pp44-49,Vol.12, No3

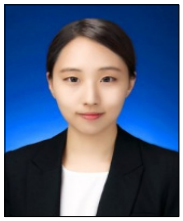
〔20〕 Cong Thuan Do, Dong Oh Son, Jong Myon Kim, Cheol Hong Kim, "A New Thread Block Scheduling Technique for Improving the Performance of GPGPU Architecture", 한국차세대컴퓨팅학회, 2015.08, pp7-14.

〔21〕 박달용, 최상훈, 최동훈, 박기웅 "워크플로우 모델링을 통한 IaaS 클라우드 플랫폼 구축 최적화", 한국차세대컴퓨팅학회, 2014.12, pp.36-45, Vol.10, No.6

〔22〕 Jintaek Kim, Susie Jo, Dongha Oh, JunGil Noh, "An Analysis on The Technology Stack of Cloud Computing", 한국차세대컴퓨팅학회, KCI, 2015.45, pp 79-89

Author

◆ 정혜림



- 2017년 대전대학교 전산정보보호학과 석사
- 관심분야: 임베디드시스템 보안

◆ 안성규

- 2017년 대전대학교 전산정보보호학과 석사
- 2017년~현재 세종대학교 정보보호학과 박사과정
- 관심분야: 임베디드 시스템 보안

◆ 김문성

- 1996년 한밭대학교 전산학과 학사
- 2013년 한밭대학교 창업경영대학원 창업학 석사
- 2013년~현재 엠에스텍 대표
- 2016년~현재 대전대학교 일반대학원 융합컨설팅학과 박사과정

◆ 박 기 웅

- 2005년 연세대학교 Computer Science 학사
- 2007년 KAIST Electrical Engineering 석사
- 2012년 KAIST Electrical Engineering 박사
- 2008년 Microsoft Research Asia, Wireless and Networking Grourp, Research Intern
- 2009년 Microsoft Research, Network Research Group, Graduate Research Fellow
- 2012년 국가보안기술연구소 연구원
- 2012년~2016년 대전대학교 정보보안학과 교수
- 2016년~현재 세종대학교 정보보호학과 교수
- 관심분야: 시스템 보안, 모바일-클라우드 컴퓨팅, 보안 프로토콜, 디지털 포렌식 등