Regular paper

Journal of the KIECS. pp. 611-618, vol. 11, no. 6, Jun. 30 2016, t. 74, pISSN 1975-8170 | eISSN 2288-2189
http://dx.doi.org/10.13067/JKIECS.2016.11.6.611

하이브리드형 클라우드 시스템에 관한 연구

장재열* · 김도문** · 최철재***

Study on Hybrid Type Cloud System

Jae-Youl Jang* · Do-Moon Kim** · Chul-Jae Choi***

요 약

제안한 논문은 통신 네트워크 및 관련 시스템 기술에 관한 연구로 USB메모리와 클라우드 스토리지 영역을 동시에 동기화하여 네트워크 오류에 따른 클라우드 스토리지 영역 사용부재 또는 USB 메모리를 분실하는 상황이 발생되더라도 데이터를 안전하게 유지관리하기 위한 기술설계이다. 클라우드를 활용하는 사용자들의 안전한 문서관리 정책의 필요성을 기반으로 매체의 분실 및 네트워크의 오류에 따른 대책을 하이브리드형 클라우드 시스템으로 설계구축하고, 사용자의 편리성에 따른 자동 및 수동 동기화 방법을 설계한다. 마지막으로 윈도우즈 환경에 적합한 사용자의 편의보장을 위해 탐색기형 스토리지 UI를 설계함으로써 점차 늘어나는 클라우드 사용자의 안전성과 편리성을 모두 보장해주기 위한 시스템설계이다.

ABSTRACT

The suggested paper studies communications network and system technology, designing data to sync to both USB memories and cloud storages at the same time, which would allow users to safely keep and manage data even in case of network troubles, affecting cloud storages, and/or loss of physical USB memories, resulting in lost data in the physical memory. The need of secure data management policy for cloud storage users form the basis of this study, offering solutions to network failures and loss of physical storage by creating hybrid cloud system. To provide convenience to windows users, the UI design should integrate that of windows explorer to maximize security and convenience.

키워드
Cloud, Synchronization, Security, Hybrid, USB
클라우드, 동기화, 보안, 하이브리드, USB

Ⅰ. 서 론

클라우드 컴퓨팅이라는 용어가 IT 분야에서 가장 중요한 이슈로 부각 된지 오래되었다. 가상화가 시급히 추진해야할 프로젝트로 대두 되었고, 이제 가상화는 클라우드와 동일한 개념처럼 인식될 만큼 IT의 관심이 클라우드로 집중되고 있다.

가상화, 클라우드, 빅데이터 뿐만 아니라 최근의 해킹 사고와 같은 보안사고 방지를 위해 망분리와 클라우드 도입의 필요성이 점증하고 시대적요구로 클라우

* 경동대학교 정보보안학과(ccy@kduniv.ac.kr)
** 경동대학교 정보보안학과(cj-choi@kduniv.ac.kr)
*** 교신저자 : 경동대학교 정보보안학과
· 접 수 일 : 2016. 05. 12
· 수정완료일 : 2016. 06. 13
· 게재확정일 : 2016. 06. 24
· Received : May. 12, 2016, Revised : Jun. 13, 2016, Accepted : Jun. 24, 2016
· Corresponding Author : Do-Moon Kim
Dept. of Cyber Security for Information, Kyungdong University,
Email : dmkim@kduniv.ac.kr

드 컴퓨팅의 발전은 거듭하고 있다[1-5].

이러한 클라우드는 가상 원격 컴퓨팅 자원기반의 IaaS(: Infrastructure as a Service), 개발환경 OS 및 서버기반의 PaaS(: Platform as a Service), 웹 기반 원격접근 App.기반의 SaaS(: Software as a Service)등의 기술을 바탕으로 여러 응용분야의 기술로 거듭나고 있다.

그러나 클라우드 서비스의 가장 큰 단점은 네트워크가 부재중일 때 자료를 공유할 수 없을 뿐 아니라, 사용자 부주의로 USB를 분실하였을 경우 데이터 손실 우려를 여전히 안고 있다.

제안한 논문에서는 두 가지 정책을 통하여 사용자의 안전한 파일관리를 정책화 한다. 즉, 네트워크를 기반으로 하는 클라우드 서비스의 경우 네트워크가 부재중일 경우 USB를 통하여 파일을 관리하며, USB를 분실한 경우는 사용자의 클라우드 환경에서 파일을 동기화하여 USB에 문서를 저장 및 관리한다.

이렇게 되면, 클라우드 또는 USB 중 한 가지가 연결상 문제가 발생되더라도 나머지 하나의 기능을 통해 사용자의 안전한 문서관리가 가능하다. 아울러, USB 또는 클라우드가 동시에 연결된 경우에는 동기화 정책에 따라 파일의 최종 버전을 공유할 수 있으므로 안전성을 확보한 파일관리가 가능하다.

논문구성에서 2장은 관련연구로 클라우드 환경에 대한 이해 및 USB매체를 통한 문서관리 정책에 대하여 정의하며, 3장에서는 클라우드와 USB를 이용한 하이브리드형 클라우드 시스템에 대하여 정의하고, 4장은 향후 확장 가능성을 제시하였다.

II. 관련연구

2.1 클라우드 개요

클라우드 컴퓨팅은 인터넷기술을 활용해 S/W, 스토리지, 서버 등의 IT자원을 웹 기반 서비스로 제공하는 것으로 서비스 범위에 따라 SaaS, PaaS, IaaS로 구분되며, 서비스 제공방법에 따라서 Public Cloud, Private Cloud 등으로 구분된다.

클라우드 컴퓨팅(Cloud Computing)은 개인 또는 기업이 개별적으로 구축해 운영했던 컴퓨팅 자원(S/W, 스토리지, 서버)이나 서비스를 기업이 사용한 만큼 비용을 지급함으로써 IT운영비용을 절감할 수 있는 장점이 있다. 또한, 최근 서비스 확장성과 구축비용 측면의 장점으로 오픈소스 클라우드 컴퓨팅 기술에 대한 관심이 지속적으로 증가하고 있다.

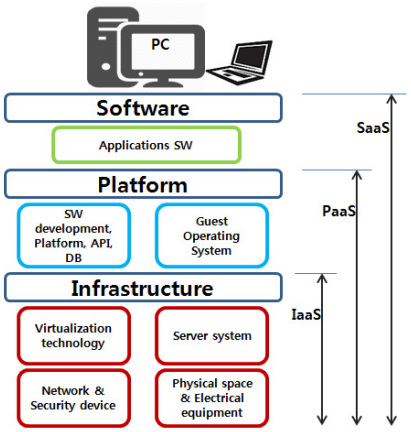

그림 1. 클라우드 컴퓨팅 서비스 구조

Fig. 1 Structure of cloud computing services

2.2 클라우드 기술현황

클라우드 컴퓨팅 서비스의 핵심은 가상화 기술이다. 오픈소스 클라우드 기술은 가상화 기술과 같이 클라우드 컴퓨팅 서비스를 구성하는 필수적인 기능을 개방형 소프트웨어로 제공하는 것을 의미한다. 클라우드 서비스는 컴퓨팅 인프라, 플랫폼, 애플리케이션 등을 제공하고, 사용비용만 지급하는 서비스이다[6-7]. 아울러 IaaS서비스를 예로 들면, 가상의 서버 자원을 제공하고 사용비용을 청구한다. 이러한 IaaS서비스의 핵심은 가상화 기술로 가상머신 모니터 기술, 가상머신 모니터 기능 제어기술, 스토리지 가상화 기술 등 다양한 기술을 필요로 한다.

또한, 오픈소스 클라우드 컴퓨팅은 클라우드 컴퓨팅 서비스를 구성하는 필수적인 기능을 개방형 소프트웨어로 제공하는 것을 의미한다. 가상머신모니터(Hypervisor) 오픈소스 소프트웨어는 Xen, KVM 등이 있고, 스토리지 가상화 및 대용량 스토리지 구축용 오픈소스 소프트웨어는 Hadoop이 있다[8-9]. 오픈소스 클라우드 컴퓨팅의 핵심기술인 가상 자원으로 클라우드를 구축 및 관리하는 제어기술로 OpenStack, CloudStack, OpenNebula, Eucalyptus등이 있다.

오픈소스(Open Source)는 소프트웨어 혹은 하드웨어의 제작자 권리를 지키면서 소스 코드를 누구나 열람할 수 있도록 한 소프트웨어로 저작권자가 오픈 소스 라이선스로 사용자들이 해당 소프트웨어의 연구, 향상, 배포 등을 허용한 소프트웨어를 의미한다.

표 1. 클라우드 컴퓨팅 요소 기술

Table 1. Cloud computing element technology

Classification	Function
The virtual machine monitor (hypervisor) technology	Virtualize server resources
Function control technology of the virtual machine monitor	Interface for using the virtual machine monitor function
Storage virtualization technology	Construction of storage virtualization and Mass Storage
Manage the virtual machine image technology	Registration of the virtual machine, storage management and retrieval
The virtual machine system control technology	To build IaaS cloud to virtual resources & management
Physical system control technology	Control and manage the physical resources
Technology of management automation	Manage a large-scale IaaS cloud system
User authentication technology	Service subscriber management and access control
Service quality management technology	Quality management and troubleshooting of user services
Measurement technology	Accounting according to the usage of resources
Virtual machine security technology	The virtual machine monitor of security

2.3 USB 플래시 드라이브

USB 플래시 드라이브의 외형은 제품마다 차이가 있으나, 일반적인 구성요소는 그림 2와 같다.

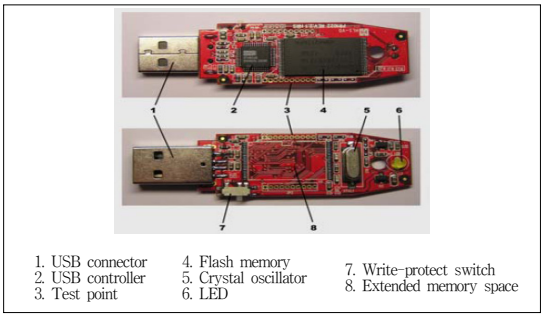

그림 2. USB 드라이브 구조

Fig. 2 Cloud computing element technology

USB 호스트가 되는 PC와 USB 플래시 드라이브의 통신은 그림 2처럼 단계별로 구성되어 있다. 사용자가 실행하는 애플리케이션 단계에서는 USB FlashDrive의 데이터를 읽고 쓰기 위해 운영체제에서 제공하는 API를 이용한다. 윈도우즈에서는 ReadFile, WriteFile, DeviceIOControl의 세 가지 함수를 제공한다.

ReadFile/WriteFile 함수는 USB Flash Drive의 데이터를 읽고 쓰기 위해 임시로 저장하는 버퍼에서 데이터를 읽고, 쓰는 기능을 담당한다[10]. DeviceIO Control 함수는 앞서 소개된 두 함수가 일방향성을 갖는 것과 달리 양방향성을 갖으며 애플리케이션이 IO 명령을 보낼 때 사용한다. Function Driver 단계는 애플리케이션에서 사용자가 내린 명령을 USB Bus-Class Drive에서 인식 가능한 형태로 변환해준다. 드라이버끼리 통신할 때는 IRP(I/O Request Packets) 구조체를 이용하며, USB 통신을 할 때에는 URB(USB RequestBlock) 구조체를 IRP에 담아서 보다 구체적인 프로토콜 규정 등을 한다. USB Hub Driver, USB Bus-ClassDriver와 Host Controller Driver는 운영체제에서 지원해주는 단계로, 개발자나 사용자는 전혀 신경을 쓰지 않아도 되는 부분이다.

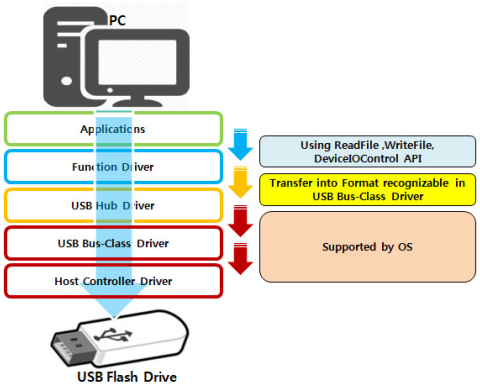

그림 3. PC와 USB 플래시 드라이브 통신

Fig. 3 PC and a USB flash drive communication

USB 플래시 드라이브는 USB 호스트가 될 수 있는 PC 및 모바일 기기와 4가지 방법으로 통신할 수 있다. 제어전송은 USB 호스트와 처음 연결되었을 때 또는 USB 플래시 드라이브에 명령을 내리기 위해 사용된다. 벌크(Bulk) 전송은 많은 양의 데이터를 신뢰

성 있게 전송할 때 사용된다. 인터럽트(Interrupt)전송은 USB 플래시 드라이브에서 호스트로만 전송이 가능하며, 작은 데이터의 빠른 전송에 유리하다.

USB 키보드나 마우스가 인터럽트 전송방법을 사용한다. 등시성(Isochronous) 전송은 데이터가 중간에 손상되더라도 일정한 시간 지속적으로 전송이 필요할 때 사용하는 전송방법으로 UDP와 유사하다.

III. 하이브리드형 클라우드 시스템

본 논문에서는 클라우드 환경과 USB 저장공간 환경을 이용하여 클라이언트의 파일정보를 저장 및 관리를 위해 하이브리드형 클라우드 시스템 설계를 한다. 네트워크가 배제된 환경에서는 PC에 연결된 해당 USB를 통하여 자료를 저장하며, USB를 분실한 경우, 클라우드 영역의 정보를 새로운 USB를 통해 동기화함으로써 안정적인 데이터 관리를 목적으로 한다.

그림 4는 초기에 클라이언트 서비스 시나리오를 정의한 것으로, 초기에 하이브리드형 클라우드 환경을 통해 클라이언트의 수행과정이다.

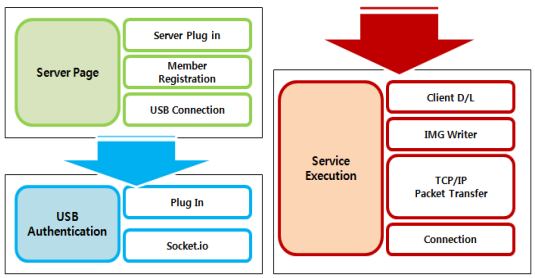

그림 4. 클라이언트 서비스 시나리오

Fig. 4 Client service scenario

① 클라이언트는 본 기술이 적용된 USB 및 도메인을 통해 서버에 접속 및 회원가입을 수행

② 하이브리드형 클라우드 시스템 개발의 기능을 탑재한 USB로 서버에서 Plug In 및 Socket.io을 통해 상호 네트워크 후 클라이언트에 맞는 공유 공간의 환경을 정의

③ 클라이언트는 Client 정보 다운로드 및 Image Writer과정을 통하여 해당 PC에 환경을 구축하며

④ TCP/IP Packet전송을 통하여 해당 PC와 클라우드 서버간의 파일공유를 위한 연결을 수행한다.

⑤ 최종, 클라이언트의 명령에 의해 USB와 클라우드 영역간의 수동 또는 자동 동기화 정책을 수행한다.

동기화 정책 수행에 따른 파일공유는 서버환경 구축과, 해당 파일에 대한 안전한 저장을 위한 보안설계 및 클라이언트 환경에서 동기화 하는 정책에 대하여 다음의 절을 통해 수행한다.

3.1 서버환경

하이브리드형 클라우드 시스템은 클라이언트가 서버에 접속할 때, 그림 5와 같이 접속결과를 보여준다.

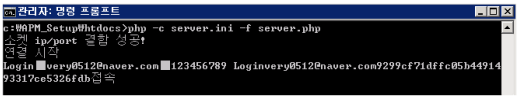

그림 5. 서버 접속 성공

Fig. 5 Success in the server connection

그림 6은 클라이언트가 서버 접속 후 TCP/IP통신과 해당 파일의 정보요청에 따른 Sql쿼리 처리과정을 수행한다. 해당 환경은 아파치 서버를 기반으로 하였으며, 접속과정은 USB제품의 고유정보인 OP코드와 서버에서 클라이언트의 고유정보를 검증하기 위한 클라이언트 ID 및 PW를 테이블로 정의한다.

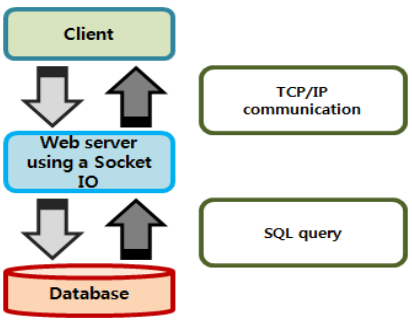

그림 6. 클라이언트의 서버연결

Fig. 6 Client's connection to the server

3.2 보안 모듈

클라이언트는 해당 PC에서 하이브리드형 클라우드 활용을 위한 환경설정 과정을 완료함과 동시에 안전

한 파일관리를 위해 해당 데이터를 인출 및 등록시에 파일의 header에 headerEncrypt를 수행한다.

정보의 중요도에 따라 이 과정은 생략이 가능하며, 암·복호화 대상의 디지털정보는 열람되는 시간을 고려하여 header값에만 Xor연산 암호화를 적용한다.

그림 7은 파일의 크기가 큰 경우 암호화 시간의 비효율적인 방안을 해결하고자 header 부분인 1024 bytes를 암호화 하는 과정이다.

```
#define ENKEY 80
void headerEncrypt(char* fileName)
{
    FILE* fp;
    unsigned int readSize = 0;
    unsigned char buf[1024];
    fp = fopen(fileName,"rb+");
    if(fp == NULL)
    {
        printf("파일이 없습니다.₩n");
    }
    readSize = fread(buf,sizeof(char), sizeof(buf), fp);
    rewind(fp);
    for( int i = 0; i < readSize; i++)
    {
        buf[i] ^= ENKEY;
    }
    fwrite(buf,sizeof(char), readSize, fp);
    fclose(fp);
    printf("readbyte %d ₩n",readSize);
    Sleep(1000);
}
```

그림 7. headerEncrypt

Fig. 7 headerEncrypt

3.3 파일 동기화

디지털정보인 파일 동기화 정책은 자동인 경우, 클라우드 영역 또는 USB의 데이터영역에서 동일 파일 중 가장 최근일을 기준으로 대상 환경에 동일 파일을 복사한다. 즉, FileSystemWqtcher 클래스를 통해 지정된 디렉터리의 변경내용을 조사한다. 이후 NetFramwork를 통해 ReadDirectoryChangesW로 지정된 디렉터리의 변경내용을 버전 관리한다.

그림 8은 감시하고자 하는 디렉터리의 핸들러 생성 과정으로 CreateFile을 통해 디렉터리를 접근한다. FILE_FLAG_BACKUP_SEMANTICS 플래그를 통해 클라이언트는 디렉터리에 접근한다.

```
hDir = CreateFileW(mRootPath, FILE_LIST_DIRECTORY,
    FILE_SHARE_READ | FILE_SHARE_WRITE | FILE_SHARE_DELETE, NULL,
    OPEN_EXISTING, FILE_FLAG_BACKUP_SEMANTICS, NULL);
if (hDir == INVALID_HANDLE_VALUE) {
    wprintf(L"CreateFile failed.");
    return 1;
}
```

그림 8. CreateFileW형 디렉터리 접근

Fig. 8 CreateFileW type directory access

이후, 지정한 디렉터리의 핸들러를 사용하여 해당 디렉터리 내의 파일 시스템을 감시하며, 파일정보의 일자가 변경되어지면 해당 파일에 대한 업데이트 정보를 버퍼에 저장한다.

```
dWNotifyFilter =
  FILE_NOTIFY_CHANGE_FILE_NAME
  FILE_NOTIFY_CHANGE_DIR_NAME
  FILE_NOTIFY_CHANGE_ATTRIBUTES
  //FILE_NOTIFY_CHANGE_SIZE
  //FILE_NOTIFY_CHANGE_SECURITY
  FILE_NOTIFY_CHANGE_LAST_WRITE
  FILE_NOTIFY_CHANGE_LAST_ACCESS
  FILE_NOTIFY_CHANGE_CREATION;

bRet = ReadDirectotyChangesW(hDir, lpBuffer, sizeof(lpBuffer)/sizeof(lpBuffer[0]),
                             TRUE,
                             dWNotifyFilter,
                             &dwBytesReturned, NULL, NULL);
```

그림 9. ReadDirectoryChangeesW형 디렉토리

Fig. 9 ReadDirectoryChangeesW type directory

그림 10은 파일 또는 폴더가 변경된 정보를 확인하기 위해 FILE_NOTIFY_INFROMATION에서 파일 변경 정보를 읽으며, 파일 이름(경로 포함)과 파일 액션 정보로 최근일의 파일 또는 폴더 정보를 검증한다.

```
int i = 0;
while (1) {
  FILE_NOTIFY_INFORMATION *lpInfomation = (FILE_NOTIFY_INFORMATION *)&lpBuffer[i];

  wchar_t filename[1024] = {L'₩0'};
  const size_t length = sizeof(filename)/sizeof(filename[0]);
  lpInfomation->FileName[lpInfomation->FileNameLength/sizeof(wchar_t)] = L'₩0';
  _snwprintf_s(filename, length, length,
      L"%s%s", mRootPath, lpInfomation->FileName);

  switch (lpInfomation->Action) {
    case FILE_ACTION_ADDED:
      wprintf(L"FILE_ACTION_ADDED: %s₩n", filename);
```

그림 10. 파일 변경 정보 읽기

Fig. 10 Read a file of change information

```
CRuntimeClass *pRuntime = RUNTIME_CLASS(CFileWatcher);
pThread = (CFileWatcher *)pRuntime->CreateObject();
pThread->setRootPath(dataRootPath);
pThread->CreateThread(0, 0, NULL);
```

```
FILE_ACTION_ADDED:  E:₩
FILE_ACTION_RENAMED_OLD_NAME:  E:₩
FILE_ACTION_RENAMED_NEW_NAME:  E:₩testDir
FILE_ACTION_ADDED:  E:₩testDir₩
FILE_ACTION_MODIFIED:  E:₩testDir
FILE_ACTION_RENAMED_OLD_NAME:  E:₩testDir₩
FILE_ACTION_RENAMED_NEW_NAME:  E:₩testDir₩a.txt
FILE_ACTION_MODIFIED:  E:₩testDir
FILE_ACTION_MODIFIED:  E:₩testDir₩a.txt
FILE_ACTION_ADDED:  E:₩testDir₩b.txt
FILE_ACTION_MODIFIED:  E:₩testDir
FILE_ACTION_MODIFIED:  E:₩testDir₩b.txt
FILE_ACTION_REMOVED:  E:₩testDir₩a.txt
FILE_ACTION_MODIFIED:  E:₩testDir
FILE_ACTION_REMOVED:  E:₩testDir₩b.txt
FILE_ACTION_REMOVED:  E:₩testDir
_
```

그림 11. 파일 복사

Fig. 11 File copy

그림 11은 클라우드 영역 및 USB의 데이터 영역의 일자 정보 중 가장 최근일로 정의된 해당 파일을 상호 저장환경에 복사해 주는 역할을 수행한다. 위 과정을 통해 Cloud의 영역과 USB의 데이터 영역에서 파일 생성일자를 확인한 후 최종일자의 정보를 동기화 정책에 의해 상호(USB데이터와 Cloud) 복사 및 공유한다.

Ⅳ. 결 론

개인 및 기업은 파일정보를 효율적이고 체계적인 관리를 위해 이메일과 웹 스토리지 및 USB 메모리 등으로 서비스 활용화를 하고 있다. 안전한 정보서비스 정책 중에서도 클라우드 산업은 안정적 성장을 거듭하고 있는 추세이다.

본 논문은 USB메모리와 클라우드 스토리지 영역을 동시에 동기화하여 데이터의 안정적 유지관리 방안을 제시하였다. 네트워크 오류에 따른 클라우드 스토리지 영역 사용부재 또는 USB 메모리의 분실이 발생되더라도 데이터를 안전하게 유지 관리하는 USB메모리 접근에 따른 클라우딩 시스템이다.

USB의 데이터 영역과 Cloud의 영역 중 최근일자의 파일을 비교검증 후 동기화 정책에 따라서 해당 폴더 동기화를 수행과정을 정의하였다. 이로써 USB 및 대용량 외장하드에 솔루션 적용과 테스트베드 구축과정을 통해 안정적 운영이 가능함을 확인하였다.

References

[1] H. Kim, E. Jun, and S. Kim, “Study on security management in cloud computing environment,” *Korean Review of Management Consulting*, vol. 2, no. 1, 2011, pp. 127-144.

[2] C. Kim, “A Study for Transaction Processing Supporting Scalability in the Cloud,” *J. of the Korea Institute of Electronic Communication Sciences*, vol. 7, no. 4, 2012, pp. 873-879.

[3] C. Kim, “An Adaptation of Consistency Criteria for Applications in the Cloud,” *J. of the Korea Institute of Electronic Communication Sciences*, vol. 7, no. 2, 2012, pp. 341-347.

[4] C. Kim, “A Range Query Method using Index in Large-scale Database Systems,” *J. of the Korea Institute of Electronic Communication Sciences*, vol. 7, no. 5, 2012, pp. 1095-1101.

[5] C. Ryu, “Context Inference and Sensor Data Classification of Big Data Stream Environment,” *J. of the Korea Institute of Electronic Communication Sciences*, vol. 9, no. 10, 2014, pp. 1079-1085.

[6] S. Lee and H. Yoon, “The Study on Development of Technology for Electronic Government of S. Korea with Cloud Computing analysed by the Application of Scenario Planning,” *J. of the Korea Institute of Electronic Communication Sciences*, vol. 7, no. 6, 2012, pp. 1245-1258.

[7] K. Park, K. Kim, K. Ban, and W. Kim, “Design and Implementation of Cloud-based Sensor Data Management System,” *J. of the Korea Institute of Electronic Communication Sciences*, vol. 5, no. 6, 2010, pp. 672-677.

[8] B. Cha, D. Kim, J. Kim, N. Kim, and S. Choi, “Design of Searchable Image Encryption System of Streaming Media based on Cloud Computing,” *J. of the Korea Institute of Electronic Communication Sciences*, vol. 7, no. 4, 2012, pp. 811-819.

[9] Y. Kim, S. Him, M. Jo, and W. Kim, “The Bigdata Processing Environment Building for the Learning System,” *J. of the Korea Institute of Electronic Communication Sciences,* vol. 9, no. 6, Aug. 2014, pp. 791-797.

[10] M. Kim, H. Hwang, K. Kim, T. Chang, M. Kim, and B. Noh, “Vulnerability Analysis Method of Software-based Secure USB,” *J. of the Korea Institute of Information Security and Cryptologys*, vol. 22, no. 6, 2012, pp. 1345-1354.

저자소개

장재열(Jae-Yeol Jang)

1984년 동국대학교 전자계산학과 졸업(공학사)
1995년 경희대학교 교육대학원 전자계산교육전공 졸업(석사)
2001년 관동대학교 대학원 전자계산공학과 졸업(공학박사)
1995년~현재 경동대학교 정보보안학과 교수
※ 관심분야: 웹서버보안, 컴퓨터교육

김도문(Do-Moon Kim)

1984년 계명대학교 전자계산학과 졸업(공학사)
1994년 숭실대학교 정보과학대학원 전산공학과 졸업(공학석사)
2004년 숭실대학교 대학원 컴퓨터학과졸업(공학박사)
1997년~현재 경동대학교 정보보안학과교수
※ 관심분야 : 정보보안기술, 컴퓨터교육

최철재(Chul-Jae Choi)

1983년 광운대학교 전자계산학과 졸업(이학사)
1987년 한양대학교 산업대학원 전자계산학전공 졸업(공학석사)
2000년 강원대학교 컴퓨터과학과 졸업(이학박사)
1988년~현재 경동대학교 정보보안학과 교수
2015년~2016 경동대학교 평생교육원장
※ 관심분야 : 데이터처리, 개인정보보호, 웹보안