*Journal of The Korea Institute of Information Security & Cryptology* ISSN 1598-3986(Print)
VOL.28, NO.2, Apr. 2018 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2018.28.2.429

# BubbleDoc: 클라우드 환경에서의 agent-free 파일시스템 분석을 통한 문서 위/변조 탐지*

전 우 진,[1†] 홍 도 원,[2] 박 기 웅[3‡]
[1]세종대학교 시스템보안연구실, [2]공주대학교 응용수학과, [3]세종대학교 정보보호학과

# BubbleDoc: Document Forgery and Tamper Detection through the Agent-Free File System-Awareness in Cloud Environment*

Woo-Jin Jeon,[1†] Dowon Hong,[2] Ki-Woong Park[3‡]
[1]Sejong Univ. SysCore Lab.,
[2]Department of Applied Mathematics, Kongju National University,
[3]Sejong Univ. Dept. of Computer and Information Security

## 요 약

전자문서는 생성 및 관리가 효율적이나 유통 및 전달 과정에서 사본이 생성되기 때문에 원본성을 상실하기 쉽다. 이러한 이유로 전자문서에 대한 다양한 보안 기술이 적용되었으나, 현재 사용되고 있는 보안 기술은 대부분 파일 접근 권한 제어, 파일 버전 및 이력 관리 등과 같은 문서 관리에 대한 것이므로 기밀문서와 같이 원본성 확보가 절대적으로 요구되어지는 환경에서는 사용이 불가능하다. 따라서 본 논문에서는 클라우드 컴퓨팅 환경에서 인스턴스 운영체제 내부에 별도의 에이전트 설치 없이 파일시스템 분석을 통하여 문서 위/변조를 탐지하는 기법을 제안한다. BubbleDoc은 인스턴스의 가상 볼륨 스토리지의 최소 영역을 모니터링하기 때문에 문서에 대한 위/변조를 효율적으로 탐지할 수 있다. 실험 결과에 따르면 본 논문에서 제안한 기술은 문서 위/변조 탐지를 위한 모니터링 수행에 있어서 1,000ms 주기로 설정하였을 때 0.16%의 디스크 읽기 연산 오버헤드를 보였다.

## ABSTRACT

Electronic documents are efficient to be created and managed, but they are liable to lose their originality because copies are created during distribution and delivery. For this reason, various security technologies for electronic documents have been applied. However, most security technologies currently used are for document management such as file access privilege control, file version and history management, and therefore can not be used in environments where authenticity is absolutely required, such as confidential documents. In this paper, we propose a method to detect document forgery and tampering through analysis of file system without installing an agent inside the instance operating system in cloud computing environment. BubbleDoc monitors the minimum amount of virtual volume storage in an instance, so it can efficiently detect forgery and tampering of documents. Experimental results show that the proposed technique has 0.16% disk read operation overhead when it is set to 1,000ms cycle for monitoring for document falsification and modulation detection.

**Keywords:** Cloud Computing, Disk Monitoring, File System

Received(09. 27. 2017), Modified(03. 20. 2018), Accepted(03. 20. 2018)

* 본 연구는 한국연구재단 연구과제(NRF-2017R1C1B2003957, NRF-2016R1A4A1011761) 지원으로 수행됨

* 본 논문은 2017년도 한국정보보호학회 하계학술대회에서 발표한 우수논문을 개선 및 확장한 것임

† 주저자, woojinjeon929@gmail.com

‡ 교신저자, woongbak@sejong.ac.kr(Corresponding author)

## I. 서 론

IT 기술의 발전으로 전자시스템을 통한 운영이 확대되면서 공공과 금융 기관을 중심으로 전자문서의 사용이 일반화되었다[1, 2]. 전자문서는 기존의 종이문서에 비해 생산 및 보관의 효율성과 문서 유통 및 전달이 수월하게 이루어진다는 장점을 가지고 있다. 하지만 유통 및 전달 과정에서 사본이 생성되거나 위/변조가 용이하여 원본성을 상실하게 된다[3]. 따라서 전자문서의 원본을 증명할 수 있는 보안 기능 강화와 이에 대한 신뢰성이 보장되어야 한다. 특히 기업의 계약서, 국가적 외교문서와 같이 민감한 데이터 혹은 중요한 데이터가 포함되어 있는 기밀문서에 대한 강력한 특수 보안이 필요하다.

현재 사용되는 전자문서 보안 서비스는 크게 두 가지의 문제점을 가지고 있다. 첫째, 제공하는 보안 기능이 파일 접근 권한 제어, 파일 버전 및 이력 관리, 자동 체크인/체크아웃, 개인정보보호 등과 같은 문서 관리에 대하여 주를 이루고 있다[4]. 이는 기밀문서와 같이 위/변조에 대한 원본성 확보가 절대적으로 요구되어지는 환경에서는 사용이 불가능하다. 둘째, 이러한 보안 서비스는 대부분 에이전트 기반으로 작동되기 때문에 사용자는 시스템 측면에서 에이전트 설치에 대한 부담 및 오버헤드로 인한 성능 저하가 발생되며, 공격자는 에이전트 무력화를 발생시켜 공격 포인트로 활용할 수 있다[5, 6].

따라서 본 논문에서는 Fig. 1.과 같은 클라우드 컴퓨팅[7, 8] 환경에서 별도의 에이전트 설치 없이 파일 시스템 분석을 통한 문서 위/변조 탐지 기법을 제안한다. 제안하는 기법은 위에서 제기하였던 기존의 보안 서비스에 대한 문제점을 해결하기 위하여 다음과 같이 설계 및 구현하였다. 파일 시스템 분석을 통해 파일에 변화가 감지되었을 때 메타 데이터 정보가 변경되는 특징을 추출하여 탐지하도록 하였으며, 파일 위/변조 탐지를 위한 모니터링 시 발생되는 시스템 오버헤드를 최소화하였다. 이는 파일 위/변조 탐지를 위한 별도의 에이전트 설치가 불필요하기 때문에 여러 인스턴스를 구동하고 있는 클라우드 서버에 효과적으로 사용이 가능하며, 가상화된 블록 디스크에 대해 모니터링하기 때문에 문서 위/변조 탐지 시 해당 블록 디스크를 마운트 해제함으로써 악성 행위에 대해 즉각적인 대응이 가능하다.

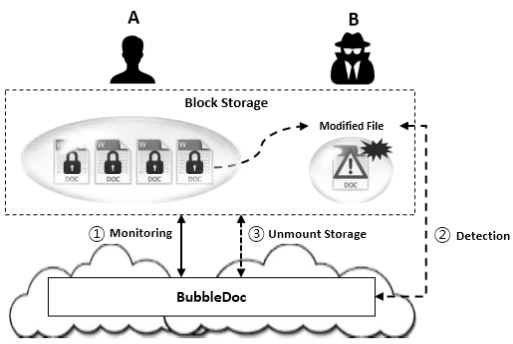


Fig. 1. Overview of BubbleDoc Framework

본 논문의 구성은 다음과 같다. 2장에서는 현재 사용되고 있는 보안 서비스와 파일 변조 보안에 관련된 기존 연구에 대해 서술하고, 3장에서는 클라우드 컴퓨팅 환경에서의 파일 시스템 분석을 통한 문서 위/변조 탐지 기법을 제안한다. 4장에서는 본 논문에서 제안한 기법의 실험 결과 및 성능 평가에 대해 설명하고, 5장에서는 결론을 기술한다.

## II. 관련 연구

본 장에서는 클라우드 컴퓨팅 환경에서 발생할 수 있는 문서 위/변조 탐지를 위한 기술에 대해 설명하고 이에 대한 관련 연구와 한계점을 도출한다.

### 2.1 문서 통합 관리 시스템 보안 기술

기업을 중심으로 전자문서 형태의 정보자산에 대한 효율적인 관리의 필요성이 제기되면서 문서 통합 관리 시스템인 문서중앙화 기술이 등장하였다[9, 10]. 문서중앙화 서비스는 사용자 PC에서 문서에 대한 작업이 가능하지만 작업한 문서의 저장 위치는 중앙화된 서버에 저장되도록 제어한다. 그러나 현재 사용되고 있는 대부분의 문서중앙화 시스템의 보안 기능은 파일에 대한 접근 권한 제어, 파일 버전 및 이력 관리, 자동 체크인/체크아웃 등과 같은 문서 관리 및 유출에 대한 보안 기능을 제공하고 있지만, 중요문서의 파일 위/변조 감지 및 대응을 위한 보안 기능은 제공하지 않고 있어 강력한 보안이 요구되는 환경에서의 사용은 적합하지 않다는 한계점을 가지고 있다.

### 2.2 SVDD 기반 중요문서 변조 유출 탐지 알고리즘

중요문서의 변조 및 유출탐지를 위해 SVDD

(Support Vector Data Description)기반의 탐지 알고리즘을 제안하였다 [11, 12]. 제안하는 알고리즘은 중요문서에 대한 다양한 가상 변조벡터를 생성하여 SVDD기법을 적용하여 학습 후 비교문서의 벡터화를 통해 중요문서에 대한 변조유출을 탐지한다. 하지만 해당 논문에서는 문서 변조율이 증가하는 경우 학습 시간이 증가하는 문제가 발생하며, 탐지 정확도가 감소하는 한계점을 가지고 있다.

이와 같이 기존 기술 및 연구에서는 문서 위/변조 감지 및 대응에 대한 보안기능이 제공되지 않으며, 변조 탐지를 위한 학습 등과 같은 전처리 과정이 요구되어 효율성이 낮고 문서의 변조도에 따라 정확도가 가변적이다. 이에 따라 본 논문에서는 클라우드 컴퓨팅 환경에서 가상 디스크의 파일시스템 분석을 통하여 문서 위/변조에 대한 즉각적인 탐지와 대응을 할 수 있는 기법을 제안한다.

## III. 파일시스템 메커니즘을 활용한 파일 위/변조 탐지 및 차단 기술

본 장에서는 클라우드 컴퓨팅 환경의 호스트 머신 레벨에서 인스턴스의 파일 시스템 분석을 통한 문서 위/변조 탐지 및 차단 기법인 BubbleDoc을 제안한다. 우리가 제안하는 기법은 파일 위/변조 탐지를 위해 수행하여야하는 모니터링 시 발생하는 시스템 오버헤드를 최소화하기 위해 파일이 변경되었을 때 파일시스템의 메타 데이터 변화에 대한 특징 정보가 저장되는 최소 영역만을 모니터링 하도록 설계하였다.

본 논문에서는 여러 파일시스템 중 NTFS(New Technology File System)[13, 14]를 활용한 기법에 대해 연구 제안하였으나, 리눅스 파일시스템인 Ext3/Ext4 등과 같이 저널링(Journaling) 기법이 적용되어있는 다양한 파일시스템에 적용 가능하다.

### 3.1 파일시스템 메커니즘 분석

파일시스템은 사용자의 데이터를 조직화하고 계층 구조로 저장하는 메커니즘이다. 이러한 파일시스템의 특징에 따라 파일 생성, 수정, 삭제, 암호화 등과 같은 작업을 진행할 때 파일시스템의 메타 영역에는 파일 관리를 위한 여러 메타 정보가 기록된다. 우리는 파일시스템 영역에 저장되어 있는 파일들이 내부 데이터가 변경이 될 때마다 메타정보가 규칙적으로 파

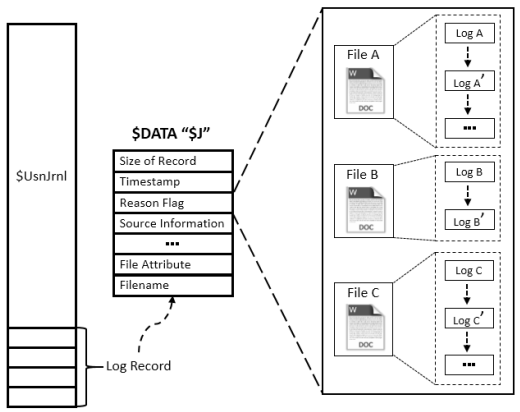


Fig. 2. Illustration of FileSystem Log Monitoring Region

일시스템 특정 영역에 업데이트 된다는 특징을 파악하였다. 이와 같은 파일 시스템의 특징을 활용하여 파일시스템 내부의 특정 메타 데이터 영역을 모니터링 함으로써 파일 변경 여부에 대해 효율적으로 탐지할 수 있다.

본 논문에서는 윈도우즈 계열에서 대중적으로 사용되고 있는 파일시스템인 NTFS를 선정하여 파일시스템 메커니즘 분석을 진행하였다.

NTFS의 내부구조는 크게 VBR(Volume Boot Record), MFT(Master File Table), Data Area로 구성되어 있다. MFT에는 파일시스템 내부에 저장되어 있는 모든 메타 데이터 정보를 테이블 형태로 관리한다. 이에 따라 우리는 Fig. 2.와 같이 우리는 MFT의 여러 메타 데이터 테이블 중 파일시스템 파일 업데이트에 관한 시그니처를 파악할 수 있는 $UsnJrnl 속성[15, 16]의 $J 항목을 모니터링 하는 방법을 고안하였다. $J 속성 내부는 시간 순서 기준으로 각 파일에 대한 이벤트 발생 로그 레코드 정보 모두가 기록된다. 따라서 운영체제에서 파일에 대한 이벤트가 발생하면 해당 이벤트에 대한 로그 정보가 $J 속성 내부에 업데이트 되며 이는 플래그 값으로 표현된다. 즉 $J 속성의 상태 플래그 값을 모니터링 한다면 여러 파일에 대하여 각각 모니터링 해야 하는 번거로움 없이 파일 위/변조에 대해 효율적으로 정확하게 탐지 가능하다.

### 3.2 파일 위/변조 탐지 알고리즘

클라우드 플랫폼에서 인스턴스의 파일시스템을 효

율적으로 모니터링하기 위한 BubbleDoc의 프로세스는 Fig. 3.과 같다. 파일 시스템을 모니터링하기 위한 프로세스는 크게 두 가지의 과정으로 나눌 수 있다.

첫 번째는 스토리지에서 $UsnJrnl 항목의 $J 속성정보 위치를 찾는 과정이고, 두 번째 과정은 인스턴스에 할당된 디스크로부터 $J 속성을 Tran. Mark(Transaction Mark)를 활용하여 모니터하는 것이다.

먼저 파일의 변조를 탐지하기 위해서는 NTFS 전체 영역에서 $UsnJrnl 영역을 찾는다. NTFS에서는 파일에 대한 정보가 MFT(Master File Table) 영역에 일괄적으로 기록되어 있기 때문에 해당 테이블을 통해 $UsnJrnl의 디스크 오프셋 정보를 찾을 수 있다. 이후 $UsnJrnl 파일의 정보를 활용하여 $J 속성의 Reason Flag 정보가 추가되는 위치를 찾는다.

$J 속성에서 Reason Flag가 업데이트되는 4byte를 모니터링하여 Flag 값을 추출한다. 하지만 Reason Flag의 경우 파일의 변경 이벤트가 발생할 때마다 이에 대한 로그가 추가되기 때문에 이전에 모니터링이 완료된 영역에 대해서는 다시 탐색할 필요가 없다. 따라서 Tran. Mark 기법을 적용하여 이전 탐색 이후 추가된 Flag 값만을 확인한다. Flag 값을 모니터링 할 경우 데이터 변조 여부에 대해서 알 수 있을 뿐만 아니라 데이터의 추가, 삭제 등과 같은 이벤트에 대해 상세히 탐지할 수 있다.

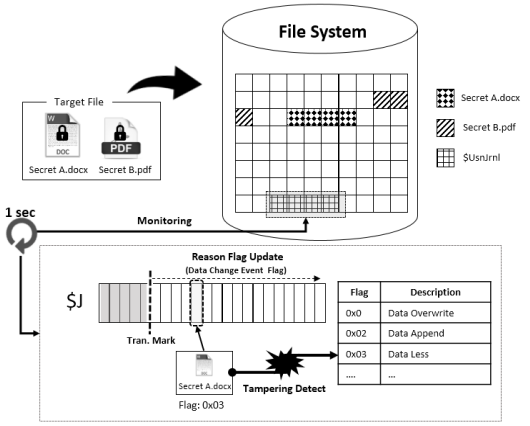


Fig. 3. Illustration of Tampering Detection of BubbleDoc

### 3.3 클라우드 환경에서의 BubbleDoc 적용 프레임워크

본 논문에서 제안한 문서 위/변조 탐지 시스템인 BubbleDoc을 클라우드 플랫폼에 적용하면 Fig. 4.와 같다.

클라우드 플랫폼[17]에서는 크게 두 가지 종류의 노드로 구성되는데, 첫 번째는 CPU와 메모리를 인스턴스에 제공하기 위한 하이퍼바이저가 설치 된 컴퓨트 노드이며, 두 번째는 LVM(Logical Volume Manager)을 활용하여 물리적 스토리지를 논리적으로 가상화하여 인스턴스에 제공해주는 스토리지 노드이다. 본 논문에서 제안한 BubbleDoc은 크게 네 가지의 장점이 있다.

첫째, 모니터링 시 인스턴스의 내부 자원을 사용하지 않는다. 우리가 제안한 BubbleDoc은 인스턴스의 운영체제 내부가 아닌 호스트 머신의 스토리지 노드에서 모니터링을 수행하게 된다. 따라서 모니터링을 수행함에 있어서 하이퍼바이저[18, 19]를 통하여 제공되는 가상화된 자원에는 영향을 미치지 않는다는 장점이 있다.

둘째, 인스턴스 사용자는 모니터링 회피가 불가능하다. BubbleDoc은 기존의 파일 위/변조 시스템과 달리 별도의 에이전트 설치가 필요하지 않는 agent-free 모니터링 시스템이다. 파일시스템의 메타 데이터 정보를 활용하여 탐지하기 때문에 공격 경로(surface)가 줄어들며, 이에 따라 클라우드 환경 내부의 인스턴스 사용자는 모니터링에 대한 회피 및 우회가 불가능하다.

셋째, 파일 위/변조 탐지에 필요한 파일시스템 내부의 최소 영역만을 모니터링하기 때문에 파일에 대하여 각각 모니터링 해야 하는 기존의 방법에 비해

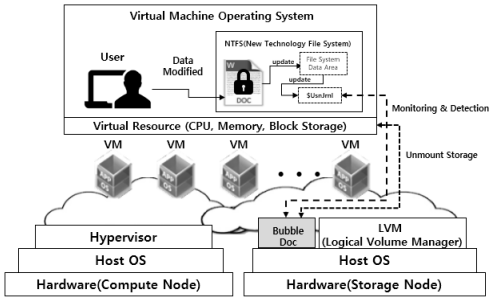


Fig. 4. Application of BubbleDoc in Cloud Environment

효율적이다. BubbleDoc은 인스턴스에 할당된 모든 스토리지 영역을 모니터링하는 것이 아닌 $UsnJrnl 영역에 업데이트 되는 플래그만 모니터링한다. 따라서 문서 위/변조 탐지를 위한 모니터링 시 시간적, 연산적 효율성이 극대화 된다.

넷째, 클라우드 환경에서의 가상화된 블록 디스크에 대해 모니터링하기 때문에 문서 위/변조 탐지 시 해당 블록 디스크 마운트 해제를 통하여 즉각적인 대응에 용이하다.

그러므로, 클라우드 플랫폼에 BubbleDoc 적용 시 클라우드 플랫폼 내의 구동중인 모든 인스턴스 스토리지 내부의 문서에 대하여 효율적으로 위/변조 모니터링 수행이 가능하며, 탐지 및 대응이 가능하다.

## IV. 성능 평가

본 장에서는 클라우드 환경의 디스크 내에서 발생하는 전자문서 위/변조를 탐지하기 위하여 BubbleDoc 프레임워크를 사용하였을 때 모니터링 수행으로 인한 시스템 오버헤드를 측정하였다.

측정 방법은 첫째, 모니터링이 수행되지 않을 때, 둘째, 기존의 파일 위/변조 탐지 시스템에서 일반적으로 사용되고 있는 tree traverse 알고리즘을 사용하여 모니터링을 수행하였을 때, 셋째, 본 논문에서 제안한 BubbleDoc 기법을 사용하였을 때의 CPU 성능과 디스크 읽기 성능을 측정하여 평가하였다.

### 4.1 실험 환경

본 논문에서 제안된 프레임워크의 성능을 평가하기 위해 클라우드 플랫폼 오픈소스 프로젝트인 OpenStack Newton 환경을 구축하여 실험을 진행하였다. 클라우드 플랫폼이 구동되는 호스트 머신의 운영체제는 Ubuntu 16.04-64bit를 사용하였으며, 하이퍼바이저는 KVM(QEMU)을 사용하였다. 문서의 위/변조 모니터링 대상이 되는 가상머신은 50GB의 블록디스크(HDD)를 할당하였고, 운영체제는 Windows7 64bit를 설치하였다.

### 4.2 파일 위/변조를 위한 모니터링 적용에 따른 시스템 오버헤드 성능 평가

클라우드 컴퓨팅 환경에서 디스크 내 발생하는 문서 위/변조 탐지를 위한 모니터링 시 모니터링 주기에 따른 시스템 오버헤드에 관한 성능을 측정하기 위하여 가상머신 내부에 디스크 벤치마크 툴인 HD Tune[20]을 설치하여 CPU 성능과 읽기 속도를 측정하였다.

#### 4.2.1 모니터링 주기 변화에 따른 CPU 오버헤드 측정

본 실험에서는 디스크 영역에 대한 모니터링 주기를 100ms, 200ms, 500ms, 1,000ms로 설정하였을 때, 인스턴스의 CPU 사용률(%)를 측정하여 비교한다.

실험 결과는 Fig. 5.와 같다. 모니터링이 수행되지 않을 때 가상머신의 CPU 사용률은 9.4%를 보였다. 이를 기준으로 모니터링의 주기를 0.1초로 설정하여 BubbleDoc 기법을 활용한 모니터링을 수행하였을 때 CPU 사용률이 9.4%로 모니터링이 수행되지 않았을 때와 동일하였으며, tree traverse 알고리즘을 사용하여 모니터링 하였을 때에는 34.2%로 24.8%의 성능 차이를 보였다. 모니터링 주기를 1초로 늘린 후 BubbleDoc 기법을 사용하여 모니터링 하였을 때에도 마찬가지로 9.4%로 모니터링을 수행하지 않았을 때와 동일하게 나타났으며, tree traverse 알고리즘을 사용한 모니터링의 CPU 사용률은 18.0%가 됨을 확인하였다.

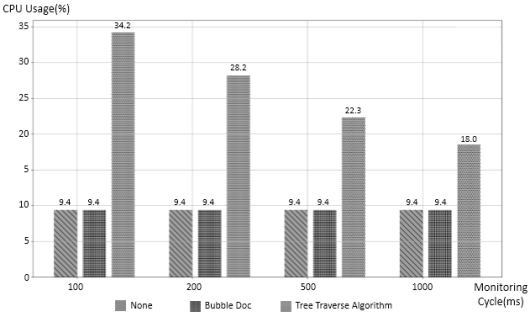


Fig. 5. CPU Utilization according to Disk Monitoring Cycles

#### 4.2.2 모니터링 적용에 따른 디스크 오버헤드 측정

본 실험에서는 디스크 영역의 데이터에 대한 모니터링을 100ms, 200ms, 500ms, 1,000ms 주기로 설정하였을 때 디스크의 읽기 속도(MB/s)의 오버헤드를 측정하여 비교한다.

실험 결과는 Fig. 6.과 같다. 모니터링이 수행되

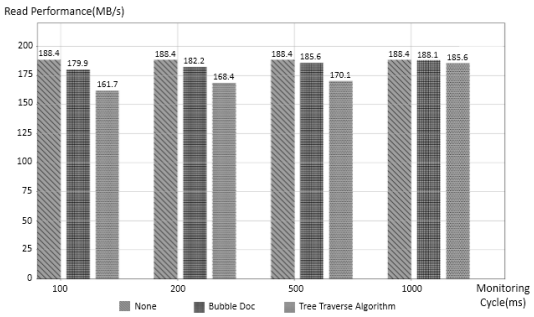


Fig. 6. Disk Overhead Measurement according to Disk Monitoring Cycles

지 않을 때 가상머신의 디스크 성능은 최대 188.4 MB/s 읽기 속도를 보였다. 이를 기준으로 모니터링의 주기를 100ms로 설정하고 본 논문에서 제안하는 기법을 사용하여 모니터링 하였을 때 179.9MB/s 읽기 속도를 나타냈으며, 모니터링 하지 않았을 때의 읽기 성능과 비교하여 오버헤드는 4.72%로 측정되었다. tree traverse 알고리즘을 사용하여 모니터링 하였을 때에는 161.7MB/s 읽기 속도를 나타냈으며, 읽기 성능 오버헤드는 16.51%로 측정되었다. 모니터링 주기를 1,000ms로 설정 후 BubbleDoc 기법을 사용하여 모니터링 하였을 때에는 188.1MB/s 읽기 속도를 보였으며, 오버헤드는 0.16%을 나타내었다. 반면, tree traverse 알고리즘을 사용하여 모니터링 하였을 때 185.6MB/s 읽기 속도를 보였으며, 오버헤드는 1.51%를 나타내었다.

성능 평가에서는 가상머신의 운영체제 내부가 아닌 호스트 머신에서 모니터링을 수행할 때 시스템 오버헤드가 최소화된다는 것을 실험을 통하여 입증하였다. 기존 Agent 방식의 tree traverse 알고리즘을 통한 파일 위/변조 탐지 모니터링에 비하여 시스템 성능 오버헤드를 효과적으로 최소화하였다.

## V. 결 론

본 논문에서는 기밀문서와 같이 특수 보안이 요구되어지는 문서에 대한 위/변조 탐지 기법을 제안하였다. 기존의 파일 위/변조 탐지 기법의 알고리즘으로 발생할 수 있는 공격 경로(Surface)에 대한 위협과 모니터링으로 인한 시스템 성능 오버헤드 문제를 클라우드 환경에서의 호스트 레벨 파일시스템 분석을 활용하여 해결하도록 구현하였다. 이는 모니터링 주기에 따라 본 논문에서 구현한 기법과 모니터링이 수행되지 않았을 때, 기존 시스템 기법을 비교 대상으로 설정하여 실험한 결과에서 CPU 사용률은 모니터링을 수행하지 않았을 때와 동일하게 나타났고, 디스크 읽기 속도는 0.16%의 오버헤드가 발생되는 것을 확인하였다. 본 논문에서 제안한 클라우드 환경에서의 파일시스템 분석을 통한 문서 위/변조 탐지 기법은 실험 결과를 통해 시스템 성능 오버헤드를 최소화하였음을 입증하였으며, 이를 통해 향후 다수의 인스턴스가 구동되어지는 클라우드 서버에서 파일 위/변조 탐지 및 대응에 효과적으로 활용이 가능할 것으로 기대된다.

## References


[1] Chung Wan-yong, “Study on the definition and legal effect requirements of the electronic document,” Lawyers Association journal, Vol.65 , No.8, pp. 10-75, 2016.

[2] Bong-geun Sung, “The law political review on the change from paper document to electronic document,” Korean Association Of Law And Politics, Vol.16 , No.2, pp. 33-71, 2016.

[3] Kim Yeon, “An evidence investigation process into electronic document,” IT & Law Research Institute, Vol.3, pp. 251-278, 2009

[4] Yeyoung Jung, Gab-Young Shim and Yong Kim, “A study on design and application of central archives management system based on cloud computing,” Journal of the Korean Biblia Society for Library and Information Science, Vol.25 , No.4, pp. 209-223, 2014.

[5] Kim, Jong-Tae, Dong-Keun Kim, and Seung-Ryol Maeng. “Document management monitoring system in P2P environment,” Journal of the Korea Academia-Industrial cooperation Society Vol.14 , No.3, pp. 1402-1408, 2013.

[6] Young Seung Kim, In Suk Park and Soon J. Hyun, “Design and im-

plementation of link server for efficient management of virtual documents," Korean Institute of Information Scientists and Engineers ,Vol.29 , No.1B, pp. 175-177, 2002.

〔7〕 Mell, Peter, and Tim Grance. "The NIST definition of cloud computing," National Institute of Standards and Technology's, 2011.

〔8〕 Armbrust, Michael, et al. "A view of cloud computing," Communications of the ACM Vol.53 .No4, pp. 50-58, 2010.

〔9〕 Yong Yoon, Young sung Ki, "An operation of cloud system for the centralization services of business documents," Journal of the Korea society of IT services, Vol.13 , No.4, pp. 309-324, 2014.

〔10〕 Keedong Yoo, "Cloud storage-based intelligent archiving system applying automatic document summarization," Journal of the Korea industrial information systems society, Vol.17 , No.3, pp. 59-68, 2012.

〔11〕 Tax, David MJ, and Robert PW Duin. "Support vector data description," Machine learning 54.1, pp. 45-66, 2004.

〔12〕 Chang, Wei-Cheng, Ching-Pei Lee, and Chih-Jen Lin. "A revisit to support vector data description," Dept. Comput. Sci., Nat. Taiwan Univ., Taipei, Taiwan, Tech. Rep, 2013.

〔13〕 Yue, Liang Jinqian Zhang. "The main data structure of ntfs file system," Computer Engineering and Applications, Vol.8, pp.0-38, 2003.

〔14〕 Chow, Kam-Pui, et al. "The rules of time on NTFS file system," Systematic Approaches to Digital Forensic Engineering, pp. 71-85, 2007.

〔15〕 Uijtewaal, Frank, and Jeroen van Prooijen. "UsnJrnl parsing for file system history project report," 2016.

〔16〕 Zhang, Zilu, Jinqiao Shi, and Lanlan Hu. "Towards enumeration of NTFS using USN journals under UEFI," International Conference on Trustworthy Computing and Services, pp. 166-174, 2014.

〔17〕 Sefraoui, Omar, Mohammed Aissaoui, and Mohsine Eleuldj, "OpenStack: toward an open-source solution for cloud computing," International Journal of Computer Applications, Vol.55 , No.3, pp. 38-42, 2012.

〔18〕 Habib, Irfan. "Virtualization with kvm," Linux Journal Vol. 166, No. 8, 2008.

〔19〕 Kivity, Avi, et al. "kvm: the linux virtual machine monitor," Proceedings of the Linux symposium, Vol.1, pp. 255-230, 2007.

〔20〕 HD Tune: www.hdtune.com/

## 〈저자소개〉

전 우 진 (Woo-Jin Jeon) 학생회원
2015년 2월: 대전대학교 정보보안학과 학사
2017년 3월~현재: 세종대학교 정보보호학과 석사과정
〈관심분야〉 클라우드 시스템 보안, 스토리지 시스템 보안, 시스템 모니터링

홍 도 원 (Dowon Hong) 종신회원
1994년 2월: 고려대학교 수학과 학사
2000년 2월: 고려대학교 수학과 박사
2000년 4월~2012년 2월: 한국전자통신연구원 팀장, 책임연구원
2012년 3월~현재: 공주대학교 응용수학과 교수
〈관심분야〉 암호기술, 프라이버시 보호기술

박 기 웅 (Ki-Woong Park) 종신회원
연세대학교 Computer Science 학사
KAIST Electrical Engineering 석사 (시스템보안 전공)
KAIST Electrical Engineering 박사 (시스템보안 전공)
~2009년 10월: Microsoft Research, Graduate Research Fellow
~2012년 8월: 국가보안기술연구소
~2016년 8월: 대전대학교 정보보안학과 조교수
2016년 9월~현재: 세종대학교 정보보호학과 조교수
〈관심분야〉 클라우드 시스템 보안, 초고속 보안 시스템, 시스템 심층관제