Journal of Korea Multimedia Society Vol. 19, No. 2, February 2016(pp. 300-307)
http://dx.doi.org/10.9717/kmms.2016.19.2.300

클라우드 컴퓨팅 환경에서의 안전한 로그 시스템 설계

이병도†, 신상욱††

Design of Secure Log System in Cloud Computing Environment

Byung-Do Lee†, Sang Uk Shin††

ABSTRACT

Cloud computing that provide a elastic computing service is more complex compared to the existing computing systems. Accordingly, it has become increasingly important to maintain the stability and reliability of the computing system. And troubleshooting and real-time monitoring to address these challenges must be performed essentially. For these goals, the handling of the log data is needed, but this task in cloud computing environment may be more difficult compared to the traditional logging system. In addition, there are another challenges in order to have the admissibility of the collected log data in court. In this paper, we design secure logging service that provides the management and reliability of log data in a cloud computing environment and then analyze the proposed system.

Key words: Cloud Computing, Secure Log, Log Data, Logging Service

1. 서 론

클라우드 컴퓨팅은 ICT(Information and Communications Technologies) 분야에서 아직도 성장 가능성이 높은 매력적인 분야로 평가받고 있다. 가트너(Gartner)는 2014년에 이어 2015년에도 주목할 만한 '10대 전략 기술'[1]에 클라우드 컴퓨팅을 포함시켰으며, IDC(International Data Corporation) 또한 'IDC 2015 예측'[2]에서 클라우드 컴퓨팅은 2015년 거의 700억 달러의 고성장을 기록하고 2018년에는 1,260억 달러 규모로 늘어날 것이라고 전망했다. 뿐만 아니라 미국, 영국, 일본, 호주 등 주요 선진국의 정부기관이 클라우드 컴퓨팅을 우선 도입하는 정책을 추진하고 있으며, 우리나라도 다소 늦었지만 클라우드 컴퓨팅의 발전 및 이용을 촉진하고 서비스를 안전하게 이용할 수 있는 환경을 조성하고자 2015년부터 '클라우드컴퓨팅 발전 및 이용자 보호에 관한 법률'을 시행했다. 이처럼 클라우드 컴퓨팅은 여전히 매력적이며 그 가능성은 또한 그만큼 높다.

한편 가트너의 2015 하이프 사이클(Hype Cycle) 보고서에 의하면 클라우드 컴퓨팅이 각성기(Trough of Disillusionment) 단계로 접어들었음을 알 수 있다[3]. 이는 신기술에 대한 기대가 높은 버블기(Peak of Inflated Expectations) 단계를 지나 지나친 관심과 현실적인 제약에 부딪쳐 재조정기에 들었음을 의미한다. 이러한 제약의 중요한 부분으로서 클라우드 컴퓨팅이 본질적으로 가지고 있는 제어권 손실, 법리상의 문제, 기존 컴퓨팅 플랫폼 환경에서 클라우드 컴퓨팅 플랫폼으로의 서드 파티 전환, 여러 보안 문제 등과 같은 한계가 포함되어 있다.

※ Corresponding Author : Sang Uk Shin, Address: (48513) 45, Yongso-ro, Nam-Gu. Busan, Korea, TEL : +82-51-629-6249, FAX : +82-51-629-6230, E-mail : shinsu@pknu. ac.kr
Receipt date : Jan. 22, 2016, Approval date : Feb. 5, 2016

† Dept. of Interdisciplinary Program of Information Security, Graduate School, Pukyong National University (E-mail : trdo413@gmail.com)
†† Dept. of IT Convergence and Application Eng., Pukyong National University
※ This work was supported by a Research Grant of Pukyong National University(2015 year)

클라우드 컴퓨팅 환경은 기존에 컴퓨팅 환경에 비해 더 큰 복잡성과 고유한 특성들로 인하여 다수의 보안 문제와 관리상의 문제를 일으키게 된다. 이러한 문제점들 중 하나가 로그 메시지를 다루는 문제다. 로그 메시지는 데이터 분석에서 중요한 역할을 수행하는 요소이며, 일반적으로 서비스의 상태를 나타내거나 문제에 대응하기 위한 트러블 슈팅, 그리고 실시간 모니터링 등에 이용된다. 또한 로그 데이터는 디지털 포렌식 분야에서도 중요하게 이용된다. 그런데 현재 기존의 로깅 솔루션들은 클라우드 컴퓨팅과 같은 분산된 아웃소싱 플랫폼에서 발생하는 막대한 양의 로그 메시지에 대해서 적절하고 유연한 처리, 선택적 조회, 전송 및 저장 중에 보안 서비스 등이 부족한 실정이다.

따라서 본 논문에서는 클라우드 컴퓨팅 환경의 로깅 서비스의 문제점과 요구 사항을 분석하고, 이러한 요구 사항을 만족할 수 있는 안전한 로깅 시스템을 설계하여 분석을 수행한다. 본 논문의 구성은 다음과 같다. 2장에서는 클라우드 컴퓨팅 환경에서 발생하는 로그 메시지에 대한 분석과 클라우드 포렌식 상에 어려움에 대해 기술하고, 이를 대응하기 위한 요구사항들을 분석한다. 3장에서는 이러한 요구사항들을 만족할 수 있는 안전한 로그 시스템을 설계한다. 제안된 로깅 시스템의 각 구성요소의 기능에 대해 기술하고, 생성된 로그 메시지가 수집되어 보관되기까지의 과정을 기술한 후, 제안된 로깅 시스템을 분석한다. 마지막 4장에서는 결론과 향후 연구 과제를 제시한다.

2. 관련연구

2.1 클라우드 컴퓨팅 환경에서의 로깅의 어려움

증거 능력(Admissibility)은 어떤 증거로의 수용을 판단하는 요구 사항을 규정하는 법적 개념이다. 그래서 증거 능력에 관한 기본 증거주의는 '증거와 관련된 신뢰성 보장'이 있다는 것을 판단하는 것이다[4]. 이러한 신뢰성을 획득하기 위해서 클라우드 환경에서 생성되는 디지털 데이터는 기존의 디지털 포렌식 방법이 아니라 클라우드 포렌식[5]으로 다루어져야 한다. 일반적인 디지털 포렌식은 클라우드 컴퓨팅 시스템의 특성인 분산처리, 멀티 테넌시, 가상화, 동적인 환경으로 인하여 디지털 증거의 식별, 보존, 수집하는 것을 어렵게 하기 때문에 시스템에 직접 적용하는 것은 적절하지 않다. 이러한 문제점을 해결하기 위해 클라우드 환경에 적합한 디지털 포렌식 절차인 클라우드 포렌식이 필요하며, 디지털 포렌식 과정에 있어서 수사에 기본이 되는 로깅 또한 기존 방식과는 다른 방식으로 진행되어야 한다.

지금까지 디지털 포렌식 절차는 다양한 방식[6-8]으로 제안되어있으며, 일반적으로 식별(Identification), 보존(Preservation), 수집(Collection), 조사(Examination), 분석(Analysis), 표현(Presentation) 단계로 구성된 [6]의 DIP(Digital Investigative Process)방식에 따라 진행한다. 이러한 DIP 방식에 따라 [9]은 클라우드 포렌식에 대한 기술적 어려움과 해결책 그리고 비교 분석에 대해 기술하고 있다. 각 단계별로 기술된 어려움들 가운데 로깅에 관련된 어려움으로 식별 단계에서 분산된 데이터(Decentralized data)와 의존성 체인(Dependency chain), 보존 단계에서 연계 보관성(Chain of custody), 수집 단계에서 전문가들이 사용하기 위한 상용도구의 결여(Lack of specialist commercial tools), 조사와 분석단계에서 로그 프레임워크의 결여(Lack of log framework)로 기술하고 있다. 이와 더불어 [10]은 디지털 포렌식을 위해 수행하는 클라우드 환경에서의 로그 분석의 어려움에 대해 기술하고 있다. 분석의 어려움으로는 분산된 로그, 로그 휘발성, 많은 단계와 계층, 보관 및 유지, 로그 접근성, 로그 비존재성, 로그 중요 정보 부재, 비호환 및 임의 로그 형식으로 기술하고 있다.

2.2 클라우드 환경에서 로그 관리 요구사항

클라우드 환경에서 생성되는 로그 메시지에 대한 분석 어려움을 해결하기 위해서는 적절한 로그 관리가 선행되어야 한다. [10]에서는 로그 관리를 위한 요구사항으로 모든 로그의 중앙 집중화, 확장성 있는 로그 저장소, 신속한 데이터 접근과 검색, 일부 로그 포맷 지원, 데이터 분석 작업 수행, 로그 레코드 유지, 로그 장기 보관과 필요시 복원, 접근 제어를 통한 분할된 데이터 접근 제어, 로그 무결성 보존, 로그 접근에 대한 감사 추적에 대해 기술하고 있다.

이러한 로그 관리가 바탕이 되었을 때, 로그 메시지의 신뢰성 위한 [4]의 보안 요구사항이 적용되어야 한다. 보안 요구사항은 전송 단계와 저장 단계 각각에서 충족되어야 한다. 먼저 전송 단계에서는 출처

인증, 메시지 기밀성, 메시지 무결성, 메시지 고유성, 신뢰 전송이 이루어져야 한다. 저장 단계에서는 엔트리(Entry) 책임 추적성, 엔트리 무결성, 엔트리 기밀성이 이루어져야 한다.

클라우드 환경에서 실질적이고 가치 있는 로깅 서비스는 로그 관리 요구사항인 [10]과 신뢰성 요구사항인 [4] 그리고 클라우드 포렌식에 해결책을 준수하는 [5]의 SLA(Service Level Agreement)을 바탕으로 구성되어야 한다. 이러한 로깅 서비스는 클라우드 다양한 사용자에게 유의미한 정보를 제공함은 물론이고, 다양한 보안 서비스와 법적 효력까지 제공해 줄 것이다.

3. 클라우드 컴퓨팅 환경에서의 로깅 시스템 설계

이 장에서는 앞에서 기술된 클라우드 컴퓨팅 환경에서의 로그 시스템의 요구 사항을 만족하는 안전한 로그 시스템을 설계한다. 제안 시스템의 전체적인 구조는 Fig. 1과 같다. 클라우드 컴퓨팅 서비스를 제공하는 인프라에서는 클라우드 구성 요소들이 생성하는 상당히 많은 양과 다양한 종류의 로그 메시지가 산재되어 발생한다. 이에 클라우드 인프라 내에 물리적으로 구분되어 위치한 각 구성요소에 로그 메시지 전송 모듈을 배치한다. 이 전송 모듈은 로그 메시지가 발생할 때마다 실시간으로 로깅 시스템에 전송한다. 이러한 방식으로 다수의 모듈에서 로그 메시지가 로깅 시스템으로 중앙 집중화된 형태로 수집이 되며, 수집된 로그 메시지는 메타 데이터 추가, 정규화, 파싱 등의 과정이 수행되어 저장된다. 이후에 사용자는 통합된 뷰를 통하여 저장된 로그에 대한 로그 정보, 질의, 시각화 등의 서비스를 제공받는다.

3.1 로깅 시스템 구성요소

로깅 시스템(LS: Logging System)은 Message handler, Logging core, Log viewer, Audit trail로 구성되며, 클라우드 인프라의 각 호스트에 있는 Message forwarder는 Message handler에게 안전한 방식으로 로그 메시지를 전달하여 Log viewer가 로그 뷰를 사용자에게 제공한다. Fig. 2는 로깅 시스템의 구성요소를 나타낸 것이며, 각 구성요소의 역할은 다음과 같다.

- Message forwarder(MF): 클라우드 컴퓨팅 인프라를 구성하는 호스트에 배치되며, 로컬 호스트에 있는 로그와 연관된 메타 데이터를 수집하여 Message handler에 전송하는 역할을 담당한다. 이때, 호스트의 낮은 자원 사용율과 안전한 로그 메시지 전송, 낮은 지연시간 그리고 전송 신뢰성을 제공한다.
- Message handler(MH): 수신된 로그 메시지와 연관 메타 데이터에 대하여 정규화를 수행하며 Logging core에게 정규화된 로그 메시지와 레코드를 구성하는데 필요한 레코드 메타 데이터를 전달하는 역할을 담당한다.
- Logging core(LC): Message handler로부터 전달받은 정규화된 로그 메시지와 레코드 메타 데이터를 이용하여 실시간 분석을 수행한다. LC는 다수의 노드로 구성되는 클러스터로 동작하며, 노드는 하나

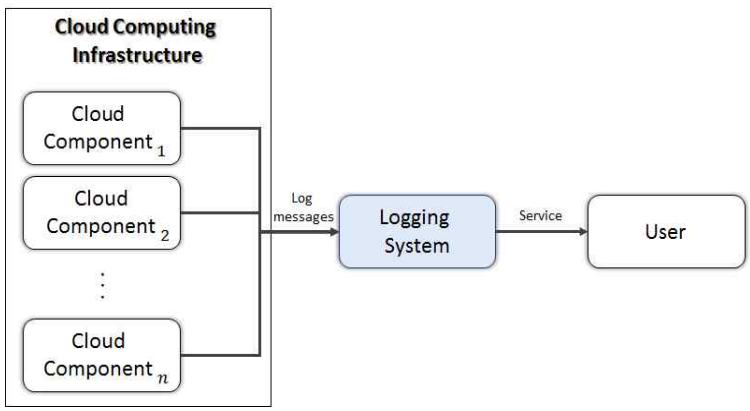

Fig. 1. Overview of the proposed logging system.

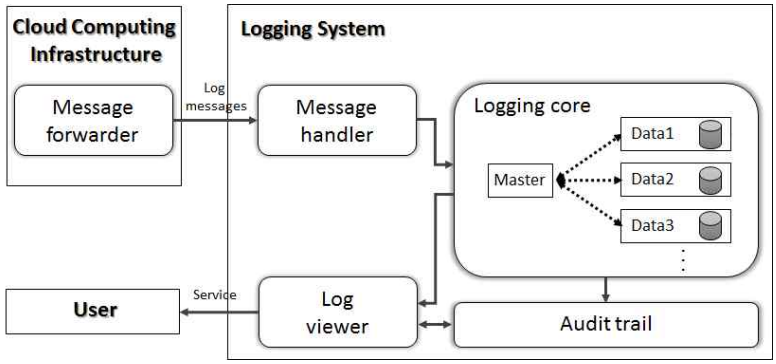

Fig. 2. Components of Logging System.

의 마스터 노드(Master node)와 다수의 데이터 노드(Data node)로 분류된다. 마스터 노드는 클러스터 전체를 관리하고 외부의 명령을 전달 받으며, 데이터 노드는 로그 메시지 인덱싱과 검색을 수행하여 인덱싱된 데이터를 저장한다.

• Log viewer(LV): 이용자에게 로그 데이터를 볼 수 있도록 하는 뷰어로써 질의와 로그 정보 및 시각화를 제공하는 인터페이스 역할을 수행한다.

• Audit trail(AT): 로깅 시스템의 운용과 작업에 관한 로깅을 담당하며, 추가적으로 LC에 저장된 로그 파일에 접근하는 사용자에 대한 로깅을 수행한다.

제안 시스템의 신뢰성을 보장하기 위해서는 클라우드 인프라와 로깅 시스템의 구성 요소들의 동작 모듈이 신뢰성이 있다고 가정해야 한다. 이들 동작 모듈의 보안에 관한 논의는 이 논문의 범위를 벗어나며, TRM(Tamper-Resistant Module)과 같은 신뢰 컴퓨팅 요소에 의해 보장될 수 있다. 클라우드 컴퓨팅 인프라와 로깅 시스템 간의 통신 채널, 좀더 구체적으로 말하면, MF와 MH 간의 통신은 3.2절의 초기화 과정에서 기술될 것처럼 공개키 암호 기법에 의해 안전하게 보호된다. 또한 로깅 시스템 내부 구성 요소들, 즉 MH, LC, LV, AT가 개별 시스템으로 구성된다면, 이들 간의 정보 전달 역시 안전하고 신뢰성 있는 채널로 이루어진다고 가정한다. 이는 시스템 환경 구축 과정에서 구성 요소들 간에 OpenSSL이나 IPsec과 같은 보안 채널 설정을 통해 구성될 수 있다.

3.2 로그 메시지 전송과 로그 데이터 정규화

구성요소들의 정상적인 동작을 위해 4 단계로 이루어진 초기화를 먼저 수행하여야 한다.

(1) 첫 번째 단계는 각 MF의 공개키 PK_{MF}와 개인키 SK_{MF} 그리고 로깅 시스템 LS의 공개키 PK_{LS}와 개인키 SK_{LS}를 생성하며, 이때 공개키 관리와 신뢰성 확보를 위해 공개키 인프라 PKI를 이용할 수 있다.

(2) 두 번째 단계에서 로깅 시스템과 각 MF는 내부시간을 신뢰되는 시간으로 동기화한다.

(3) 세 번째 단계에서 LC는 로그 데이터를 암호화하기 위해 이용되는 초기 IV와 블록암호 알고리즘 비밀키 K를 임의로 선택하여 안전하게 유지한다고 가정한다.

(4) 네 번째 단계에서는 수행한 초기화 작업의 내용과 결과를 AT에 기록한다.

클라우드 컴퓨팅 인프라의 각 구성 요소에 의해 생성된 로그 메시지는 MF에서 수집되어 MH에게 전송된다. 우선 MF와 MH는 로그 메시지의 안전한 전송을 위해 상호 인증을 수행하고 적절한 세션 키를 설정한다. 로그 데이터가 발생하면 MF는 로그 데이터 원본 $OriLog_n$과 연관된 로그 메타 데이터 $LM-Data$를 생성하며, 이때 $LM-Data$는 로그를 전송한 클라우드 인프라의 호스트 식별자 $host$, 로그가 생성된 파일의 절대 경로 $file$, 로그 데이터 타입 $LType$으로 구성된다. 그리고 MF는 설정된 세션 키로 $OriLog_n$와 $LM-Data$에 암호화를 수행하며 이 암호문의 해시 값 $hashVal$을 생성하고 개인키를 이용하여 서명을 수행한다. 이렇게 생성된 암호문과 전자서명을 MH에 전송한다. 이러한 과정은 로그 데이터가 발생되는 즉시 수행된다.

암호문과 전자서명을 전달받은 MH는 먼저 MF의 공개키 PK_{MF}를 이용하여 전자서명 검증을 수행한다. 검증이 성공하면, 암호문을 복호화하여 $OriLog_n$

과 $LM-Data$를 이용하여 정규화를 수행한다. 로그 데이터 원본 $OriLog_n$는 클라우드 인프라의 구성 요소들이 포맷이 표준화되지 않은 다양한 로그 데이터를 생성하므로, 이러한 로그 데이터들을 통합하여 처리하기 위해서 MH는 로그 데이터의 정규화를 수행한다. 이 과정으로 $OriLog_n$와 $LM-Data$를 필드로 포함하는 로그 데이터 $NorLog_n$으로 변환하고, 각 $NorLog_n$에 대한 레코드를 구성하기 위해 필요한 메타 데이터 $RM-Data$를 생성한다. 여기서 $RM-Data$에는 레코드가 속하는 인덱스 $Index$, 레코드 타입 $RType$, 레코드 Id 등이 포함된다. 이렇게 정규화된 로그 데이터 $\{NorLog_n, RM-Data\}$를 LC에게 전달한다.

3.3 안전한 로그 파일 생성

Fig. 3은 일정 시간 구간 t동안 수집된 로그 메시지를 로그 파일로 나타낸 것이다. 로그 파일은 생성된 시간 순으로 로그 레코드가 추가되며, 각 로그 레코드는 $RM-Data$, 토큰 $Token$, $NorLog_n$, 앞의 3개의 필드와 이전 레코드들의 유효성을 검증하는 태그 $VerTag_n$으로 구성된다.

LC의 마스터 노드는 MH로부터 $RM-Data$와 $NorLog_n$를 전달받으며, $NorLog_n$에 대한 레코드를 생성하기 위해 다음 작업을 수행한다. 먼저 첫 번째 필드 $RM-Data$는 MH로부터 전달 받은 $RM-Data$를 그대로 이용한다. 두 번째 필드인 $Token$는 $NorLog_n$를 다수의 데이터 노드에 인덱싱 작업을 할당하여 검색 가능한 키워드를 생성해 구성한다. 마지막으로 $VerTag_n$ 필드는 각 레코드의 유효성을 판단하기 위해 $RM-Data$와 $Token$ 그리고 $NorLog_n$ 필드로 구성된 페이로드에 CCM(Counter with CBC-MAC)[11] 방식의 암호화를 적용하여 생성한다.

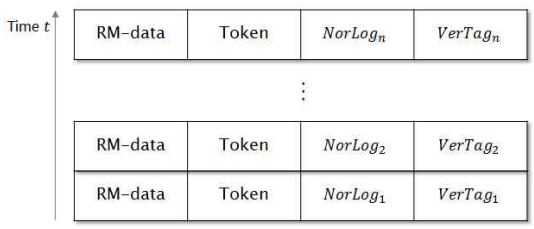

Fig. 3. Log File Structure.

로그 파일은 저장 단계에서 기밀성과 무결성을 유지하기 위해 AES-CCM 방식을 이용하여 각 레코드의 암호화와 검증 태그를 추가하여 저장한다. n 번째 로그 데이터에 대한 AES-CCM 방식을 이용한 안전한 로그 생성 과정은 다음과 같다.

① 먼저 페이로드 $P=RM-Data \parallel Token \parallel NorLog_n$를 생성하고(페이로드 P의 길이는 $Plen$으로 정의한다), 페이로드 P를 포맷 함수 β에 입력하여 블록 B_1, B_2. .., B_r를 생성한다.

② B_0은 이전 검증 태그 $VerTag_{n-1}$를 이용하여 설정한다. 단, 최초 B_0는 랜덤한 IV를 이용하여 생성한다.

③ 생성된 각 블록 B_i에 AES 알고리즘을 적용하여 Y_i를 생성한다.

$$Y_0 = AES_K(B_0)$$

$$Y_i = AES_K(B_i \oplus Y_{i-1}) \quad (1 \le i \le r)$$

④ 마지막에 생성되는 Y_r에서 인증에 필요한 비트 길이 $Tlen$ 만큼 잘라서 인증 태그 T를 생성한다.

$$T = MSB_{Tlen}(Y_r)$$

⑤ 카운터 생성 함수를 이용하여 Ctr_0, ..., Ctr_m를 생성한다($m = \lceil Plen/128 \rceil$). 여기에서 Ctr_0는 이전 검증 태그 $VerTag_{n-1}$를 이용하여 생성한다. 단, 최초 Ctr_0는 랜덤한 IV를 이용하여 생성한다.

⑥ 생성된 카운터 블록들을 AES 암호화 알고리즘에 적용하여, S_0와 $S=S_1 \parallel S_2 \parallel .. \parallel S_m$을 생성한다.

$$S_0 = AES_K(Ctr_0)$$

$$S_j = AES_K(Ctr_j) \quad (1 \le j \le m)$$

⑦ 생성된 S_0와 S를 이용하여 페이로드 P를 암호화하여 암호문 C와 인증 태그 $VerTag_n$를 획득한다.

$$C = (P \oplus MSB_{Plen}(S))$$

$$VerTag_n = T \oplus MSB_{Tlen}(S_0)$$

3.4 로그 파일 복호화와 검증

로그 레코드의 검증은 암호문 C가 복호화되었을 때의 평문이 유효한 것인지를 판단하는 역할을 한다. 로그 파일의 최상위 레코드 검증 태그 $VerTag_n$가 유효하다는 것은 해당 로그 파일의 복호화된 모든 레코드가 유효하다는 것을 의미한다. n 번째 로그 데이터에 대해 AES-CCM 방식을 이용한 로그 복호화 및

검증 과정은 다음과 같다.

① 암호화된 레코드의 복호화를 수행하기 위해 최초에 C의 길이가 $Clen \leq 0$ 인지 확인한다. 암호문의 길이는 0이거나 음의 정수이면 유효하지 않으므로 복호화 및 검증 과정을 중단한다.

② 암호화 과정과 같은 방법으로 카운터 블록들 Ctr_0, ..., Ctr_m를 생성한다. Ctr_0는 이전 검증 태그 $VerTag_{n-1}$를 이용하여 생성한다. 단, 최초 Ctr_0는 랜덤한 IV를 이용하여 생성한다.

③ 생성된 카운터 블록들을 AES 암호화 알고리즘에 적용하여 S_0와 $S=S_1 \parallel S_2 \parallel .. \parallel S_m$을 생성한다.

$S_0 = AES_K(Ctr_0)$

$S_j = AES_K(Ctr_j) \quad (1 \leq j \leq m)$

④ 생성된 S_0와 S를 이용하여 복호화된 페이로드 P'와 검증 태그 T를 획득한다.

$P' = (C \oplus MSB_{Clen}(S))$

$T = VerTag_n \oplus MSB_{Tlen}(S_0)$

⑤ 획득한 P'과 T가 유효한지 판단하기 위한 검증 과정을 수행한다. 먼저, 획득한 P'를 포맷함수 β에 입력하여 블록 B_1, B_2. .., B_r를 생성한다. B_0은 이전 검증 태그 $VerTag_{n-1}$를 이용하여 설정한다. 단, 최초 B_0는 랜덤한 IV를 이용하여 생성한다.

⑥ 생성된 각 블록 B_i에 AES 알고리즘을 적용하여 Y_i를 생성한다.

$Y_0 = AES_K(B_0)$

$Y_i = AES_K(B_i \oplus Y_{i-1}) \quad (1 \leq i \leq r)$

⑦ 마지막에 생성되는 Y_r에서 인증에 필요한 비트 길이 $Tlen$만큼 잘라서 태그 T'를 생성한다.

$T' = MSB_{Tlen}(Y_r)$

⑧ 복원한 검증 태그 T와 T'를 비교하여 일치하지 않으면 복호화된 페이로드 P'이 유효하지 않다고 판단하고, 일치하면 유효하다고 인증한다.

3.5 로그 파일 관리와 레코드 구성

로그 파일의 관리는 LC 내에 하나의 마스터 노드와 다수의 데이터 노드로 구성되는 클러스터를 통해 이루어진다. 이 클러스터는 로그 파일을 위에서 기술한 CCM 방식으로 암호화하여 다수의 데이터 노드에 분산 저장한다. 그리고 신뢰성 보장을 위한 백업을 위해 사본을 생성하여 원본과 마찬가지로 분산 저장한다. 이러한 작업은 이후 사용자가 검색을 수행할 때 분산된 원본과 사본 모두를 이용할 수 있기 때문에 성능 향상 효과를 볼 수 있다.

로그 레코드의 정규화는 다양한 형식으로 표현된 로그들을 일관된 형식으로 유지하도록 하여 동일한 작업을 가능하게 한다. 그래서 정규화를 수행하기 위해서는 로그 레코드의 형식을 먼저 정의해야 한다. 로그 레코드는 위에서 기술한 것처럼 $RM-Data$, $Token$, $NorLog_n$, $VerTag_n$ 필드로 구성된다. 그리고 레코드의 메타 데이터 $RM-Data$는 인덱스 $Index$, 레코드 타입 $RType$, 레코드 식별자 역할을 하는 Id로 구성되며, 관계형 데이터베이스와 비교했을 때에 $Index$는 데이터베이스, $RType$는 테이블, 레코드는 행으로 비교될 수 있다. 그리고 $Token$ 필드에는 검색에 이용되는 키워드가 배열과 같은 형태로 존재한다. 다음으로 $NorLog_n$는 원본 로그 $OriLog_n$와 로그 메시지에 대한 메타 데이터 $LM-Data$로 분류되며, $LM-Data$는 로그를 전송한 클라우드 인프라의 호스트 식별자 $host$, 로그가 생성된 파일의 절대 경로 $file$, 로그 데이터 타입 $LType$으로 구성된다. 마지막으로 $VerTag_n$는 각 레코드의 유효성과 전체 로그 레코드의 인증을 보장하기 위해 제공되며 AES-CCM 방식을 이용하여 인증 태그의 체이닝(chaining) 방식으로 생성된다.

4. 제안 로깅 시스템 분석

이 장에서는 클라우드 환경에서의 로그 메시지에 대한 요구사항을 제안 시스템이 얼마나 충족하는지에 대해 기술한다.

- 모든 로그의 중앙 집중화: 로그 데이터는 MH를 통해 LC에 모두 집중화되어 저장된다.

- 다양한 로그 포맷 지원 : 클라우드 컴퓨팅 인프라의 다양한 구성 요소들에서 생성되어 수집된 다양한 포맷의 로그들에 대해 정규화를 적용하고 파싱과 색인을 수행하여 저장함으로서 다양한 로그 분석을 지원한다.

- 확장성 있는 로그 저장소, 로그 레코드 유지 : LC의 클러스터에 새로운 데이터 노드를 지속적으로 참여시킴으로써 선형적으로 인덱싱 성능을 향상시킴과 동시에 저장소의 확장성을 증가시킨다. 그리고 로그 레코드의 효율적인 유지를 위해 인덱싱된 로그 파일을 ISO 8601 표준에 따라 날짜별로 생성된 인덱스에 분리하여 저장하고 인덱스 내에 클라우드 컴퓨

팅 로그, 웹 서버 로그, 부팅 로그 등과 같이 여러 유형으로 분류하여 저장한다.

- 신속한 데이터 접근과 검색 : 다수의 데이터 노드에 인덱싱되어 분산 저장된 로그 파일과 그 사본을 이용하여 신속한 접근과 검색이 가능하다.

- 전송 단계에서 기밀성, 무결성, 출처 인증 : 소스에서 생성된 로그 메시지를 MF가 MH에 전송할 때에 암호화와 서명을 통해 전송함으로써 전송 중 로그 메시지의 기밀성과 무결성을 유지한다. 그리고 MF가 로그 메시지를 전송할 때 서명 수행과 로그 메시지를 생성한 호스트 이름과 같은 메타 데이터를 포함시키고 이를 전자 서명함으로써 출처 인증을 제공할 수 있다.

- 저장 단계에서 기밀성과 무결성 : 로그 메시지를 저장 할 때에도 AES-CCM 방식을 적용함으로써 로그 데이터의 기밀성을 제공하며, 또한 생성된 검증 태그 $VerTag_n$를 이용하여 무결성을 지원한다.

- 로그 접근에 대한 감사 추적 : 감사 추적에 대한 요구사항은 AT에서 로그 파일에 접근하는 사용자에 대한 로깅을 수행함으로써 만족한다.

제안 로그 시스템이 2장에서 기술한 요구사항 대부분을 충족하였지만, 로그 메시지의 접근제어와 책임 추적성을 만족시키지는 못한다. 이는 저장된 로그 데이터에 대한 접근 제어 메커니즘이 결여되었기 때문에 나타난 결과이다. 이 문제점을 해결하기 위해 클라우드 컴퓨팅 환경에 적절한 접근 제어 메커니즘의 도입이 필요하지만, 이것은 단순하게 해결될 수 있는 문제가 아니다. NIST에서는 클라우드 조사관이 인가되지 않고 데이터에 접근하는 개체를 분명하게 식별하는 것은 사용자의 클라우드 계정에 대한 인가와 접근 제어가 데이터 보호 규칙을 충족하지 않을 수 있기 때문에 어려운 문제라고 기술하고 있다[5]. 이러한 주장은 [12-17]에서도 뒷받침되고 있으며 이 문제를 해결하기 위해 많은 연구가 진행되고 있으며 앞으로도 지속적인 연구가 필요할 것으로 보인다. 또한 [18]과 같이 대용량 로그에 대한 마이닝과 고속 검색에 관한 연구도 필요하다.

5. 결 론

본 논문에서는 클라우드 컴퓨팅 환경에서 제공하는 로깅 서비스가 직면하고 있는 로그 분석과 신뢰성에 대한 어려움과 이를 극복하기 위한 요구사항들을 통하여 로깅 서비스가 지향해야할 방향에 대해 기술하였다. 또한 이러한 요구사항들을 부합하기 위한 로깅 시스템 설계를 수행하였다.

그 결과 설계한 로깅 시스템의 분석에서 로그 관리와 신뢰성 요구사항의 대부분을 충족하였지만 저장 단계에서의 엔트리 책임 추적성과 분리된 데이터 접근 제어를 만족시키기에는 충분하지 못하였으며, 이를 해결하기 위해서 저장 단계에서의 로그 접근 제어 메커니즘이 필요함을 확인할 수 있었다. 그러나 제안 로깅 시스템은 클라우드 컴퓨팅 환경에서의 실질적인 로깅 서비스가 지향해야할 방향을 제시하였을 뿐만 아니라 로깅 서비스를 제공하는 주체인 클라우드 서비스 제공자(CSP)에게 로깅 서비스에 대한 새로운 모델을 제공한다. 뿐만 아니라 로그 데이터를 획득할 수 없었거나 단편적으로 획득할 수 있었을 뿐이었던 기존의 클라우드 컴퓨팅 플랫폼과 비교하여 로그 데이터에 실시간으로 종합적이고 지능적인 분석을 수행하며 로깅 서비스를 제공받는 다양한 사용자들에게 기존 로깅 서비스에서는 제공받지 못한 새로운 사용자 경험을 느끼게 해 줄 수 있을 것이다. 이는 결국에 현재 상황에 대한 넓은 통찰력을 제공할 수 있을 것이라고 사료된다.

향후에는 클라우드 컴퓨팅 환경에 적절한 로그 접근 제어 메커니즘을 구성하여 로깅 시스템에 배치함으로써, 모든 요구사항 조건을 만족하는 로깅 시스템으로 보완할 할 것이며 이러한 제안 로깅 시스템이 얼마나 많은 클라우드 컴퓨팅 플랫폼에서 이식될 수 있는지에 대한 이식성과 시스템 성능을 향상시키기 위한 방안에 대해 연구할 계획이다.

REFERENCE

[1] Gartner Identifies the Top 10 Strategic Technology Trends for 2015, http://www.gartner.com/newsroom/id/2867917 (accessed Feb., 1, 2016).

[2] F. Gens, *IDC Predictions 2015: Accelerating Innovation - and Growth - on the 3rd Platform*, Top 10 Predictions, IDC, 252700, 2014.

[3] B. Burton and M.J. Walker, *Hype Cycle for Emerging Technologies, 2015,* Gartner's

Hype Cycle Special Report for 2015, 3100227, Gartner, 2015.

[4] R. Accorsi, "Log Data as Digital Evidence: What Secure Logging Protocols Have to Offer?," *Proceeding of 33rd Annual IEEE International Computer Software and Applications Conference*, Vol. 2, pp. 398-403, 2009.

[5] P. Mell and T. Grance, *Nist Cloud Computing Forensic Science Challenges*, National Institute of Standards and Technology, U.S. Department of Commerce, 2014.

[6] G. Palmer, *A Road Map for Digital Forensics Research-report from the First Digital Forensics Research Workshop*, Utica, New York, 2001.

[7] R. McKemmish, *What is Forensic Computing?*, Australian Institute of Criminology, Canberra, 1999.

[8] K. Kent, S. Chevalier, T. Grance, and H. Dang, *Guide to Integrating Forensic Techniques into Incident Response*, NIST Special Publication, U.S., 2006.

[9] A. Pichan, M. Lazarescu, and S.T. Soh, "Cloud Forensics: Technical Challenges, Solutions and Comparative Analysis," *Digital Investigation*, Vol. 13, pp. 38-57, 2015.

[10] R. Marty, "Cloud Application Logging for Forensics," *Proceedings of the 2011 ACM Symposium on Applied Computing*, pp. 178-184, 2011.

[11] M. Dworkin, *Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality*, NIST SP. 800-38C, U.S., 2004.

[12] K. Ruan, J. Carthy, T. Kechadi, and I. Baggili, "Cloud Forensics Definitions and Critical Criteria for Cloud Forensic Capability: An Overview of Survey Results," *Digital Investigation*, Vol. 10, No. 1, pp. 34-43, 2013.

[14] K. Ruan and J. Carthy, "Cloud Computing Reference Architecture and Its Forensic Implications: a Preliminary Analysis," *Digital Forensics and Cyber Crime, LNICST*, Vol. 114, pp. 1-21, 2013.

[15] K. Ruan, J. James, J. Carthy, and T. Kechadi, "Key Terms for Service Level Agreements to Support Cloud Forensics," *Advances in Digital Forensics VIII, IFIP AICT*, Vol. 383, pp. 201-212, 2012.

[16] K. Ruan and J. Carthy, "Cloud forensic maturity model," *Digital Forensics and Cyber Crime, LNICST*, Vol. 114, pp. 22-41, 2013.

[17] Y. Chen, V. Paxson, and R.H. Katz, *What's New about Cloud Computing Security*, EECS Dept., University of California, Berkeley Tech., Rep., UCB/EECS- 2010-5, Jan., 2010.

[18] H. Lee, and T. Kim, "High-Speed Search Mechanism based on B-Tree Index Vector for Huge Web Log Mining and Web Attack Detection," *Journal of Korea Multimedia Society*, Vol. 11, No. 11, pp. 1601-1614, 2008.

이 병 도

2013년 8월 동명대학교 정보보호학과(학사)
2014년 3월~현재 부경대학교 대학원 정보보호학협동과정(석사)
관심분야 : 클라우드 컴퓨팅, 네트워크 보안, 디지털 포렌식

신 상 욱

1995년 2월 부경대학교 전자계산학과(학사)
1997년 2월 부경대학교 전자계산학과(석사)
2000년 2월 부경대학교 전자계산학과(박사)
2000년 4월~2003년 8월 한국전자통신연구원 선임연구원
2003년 9월~현재 부경대학교 IT융합응용공학과 교수
관심분야 : 암호 프로토콜, 디지털 포렌식, IT융합보안