전방 안전성이 보장되는 로그 시스템 보안기법 비교분석★

강석규* · 박창섭**

요 약

보안로그의 활용범위가 다양해짐에 따라 저장된 로그 데이터에 대한 무결성의 중요성이 높아지고 있다. 특히, 저장된 로그 데이터는 시스템에 침입한 공격자들이 자신의 흔적을 없애기 위해 우선적으로 조작되는 대상이다. 키 정보가 노출이 된 이후의 로그 데이터의 안전성은 보장하지 못하지만, 그 이전에 축적된 로그 데이터 무결성의 전방 안전성을 보장하는 다양한 이론적 기법들이 소개되었다. 본 논문에서는 기존기법들의 특성을 분석하며, 계산 효율적인 측면에서의 비교분석을 통해 적용될 운영환경에 적합한 기법들의 유효성을 확인한다.

Comparative Analysis of Security Schemes for Log System Providing Forward Security

Seok-Gyu Kang* · Chang-Seop Park**

ABSTRACT

In IT system, logs are an indicator of the previous key events. Therefore, when a security problem occurs in the system, logs are used to find evidence and solution to the problem. So, it is important to ensure the integrity of the stored logs. Existing schemes have been proposed to detect tampering of the stored logs after the key has been exposed. Existing schemes are designed separately in terms of log transmission and storage. We propose a new log system for integrating log transmission with storage. In addition, we prove the security requirements of the proposed scheme and computational efficiency with existing schemes.

Key words : Audit Log, Forward Secrecy, Log Transmission, Log Storage.

접수일(2015년 12월 1일), 게재확정일(2015년 12월 30일)

★ 본 연구는 미래창조과학부 및 한국인터넷진흥원의 "2015년 정보보호 석사과정 지원사업"의 연구결과로 수행되었음.

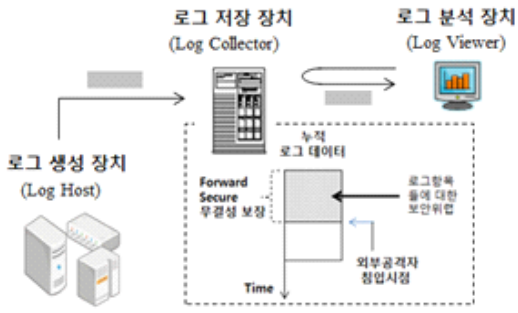
★ 본 연구는 2015년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임.(NRF-2014R1A1A2055074)

* 단국대학교/컴퓨터학과

** 단국대학교/컴퓨터학과(교신저자)

1. 서 론

시스템 및 네트워크 장비 그리고 다양한 보안 및 응용 소프트웨어로부터 발생되는 보안로그의 활용범위는 다양하다. 기본적인 보안사고와 비정상행위 탐지, 내부감사 및 포렌식 분석 그리고 시스템 이용분석을 통한 자원의 효율적 배분과 장기플랜을 확보하는 데에도 이용된다. 뿐만 아니라, 국내외적으로 제정된 다양한 IT 컴플라이언스 (예: 미국 HIPAA, 한국의 개인정보보호법 등)를 준수하기 위해서도 로그 시스템을 통한 전자적 기록물의 유지관리가 의무화되고 있다 [1]. 정부 및 기업뿐 만 아니라 개인적인 차원에서도 로그 시스템의 활용성은 높다고 할 수 있다. 예를 들어, 의료정보분야에 있어서는 진단 방사선 또는 이식형 심장제세동기 등의 IT 의료기기를 통한 환자진단 및 치료에 대한 기록유지는 의료사고 예방 및 사후 분쟁조정을 위한 중요한 토대가 되고 있다 [2]. 수집된 로그의 활용성을 높이기 위해서는 다음 3가지의 로그관리(Log Management)가 선행되어야 한다. 첫째, 제한된 로그관리 자원과 지속적으로 발생되는 대용량 로그 데이터간의 불균형을 조정하는 것이다. 10기가 방화벽의 경우 초당 2만 건의 이벤트 로그가 발생하는데 이러한 빅데이터를 원본 그대로 압축저장할지 아니면 필터링해서 저장 할지는 로그분석의 목적에 따라서 결정되어야 한다. 둘째, 효율적인 그리고 유효한 로그분석을 위해서는 로그분석 정책 및 인프라의 구축이 요구된다. 수집된 로그 데이터를 분석하여 유의미한 정보를 도출하기 위한 다양한 방법론이 존재한다. 셋째, 로그 데이터의 수집 및 저장에 있어서 로그의 기밀성, 무결성 및 가용성 등이 보장되어야 한다. 수집된 로그 데이터에는 민감한 개인정보가 포함될 수도 있고 기업의 영업기밀에 대한 정보 역시 유추가 가능하기에 활용유형에 따라서 로그 데이터의 기밀성이 요구된다. 하지만, 로그 데이터의 무결성은 활용유형에 관계없이 로그 데이터 전송 및 저장관리에 있어서 필수적인 요구사항이 된다. 본 논문에서는 로그관리 중에서 로그 무결성에 초점을 맞춘다.



(그림 1) 로그 시스템 구성

(그림 1)에서와 같이 로그생성장치(Log Host)를 통해 생성된 로그 데이터는 로그저장장치(Log Collector)에 전송/저장되고, 정기적으로 로그분석장치(Log Viewer)를 통해 로그 데이터를 분석하게 되는데, 로그 저장 과정 그리고 로그분석 과정에서의 보안 이슈가 발생한다. 전통적인 대칭키 및 공개키 암호 그리고 MAC 등이 활용되지만, 로그 시스템에 대한 보안위협모델은 로그저장장치 자체가 “외부공격자”에 의해 침투되어 로그 데이터를 보호하는 키 정보가 노출 될 수 있다는 가정을 하고 있다. 결국, 로그저장장치에 보관된 키 정보가 노출이 된 이후의 로그 데이터의 안전성은 보장하지 못하지만, 그 이전에 축적된 로그 데이터 무결성의 전방 안전성(Forward Security)의 확보가 기본적인 보안요구사항이 된다.

로그 시스템 보안에 대한 기존연구는 로그저장장치에 저장된 로그 데이터의 “Forward Secure 무결성” 보장을 지향하고 있다. 이를 위한 가장 단순한 방안은 (그림 1)에서와 같이 로그저장장치에 저장된 로그 데이터를 별도의 WORM(Write-Once Read-Many) 디바이스에 저장하거나 또는 원격로그서버에 실시간으로 연속 전송하여 백업하는 것이다. 하지만, 최근에 발표된 WORM 드라이브 보안상의 취약점 [3] 그리고 대용량의 로그 데이터에 대한 실시간 전송이 여의치 않은 환경 등을 고려하여 궁극적으로 암호기법을 통한 로그저장장치 내의 로그 데이터 보호 메커니즘들이 제안 되고 있다. 로그 시스템이 공격자에 의해 제어되는 상황에서는 키 정보가 노출 될 수 있기 때문에 그 시점부터 생성되는 로그 데이터에 대한 무결성은 보장될 수 없다. 하지만, 그 이전에 축적된 로그 데이터에 대한 무결성은 보장되어야 하는데, 이를 “F

orward Secure 무결성"이라고 한다. 2장에서는 로그 시스템의 위협모델과 보안요구사항을 소개하고, 3장에서는 로그시스템이 적용되어야 할 응용환경과 그 환경에서의 요구사항을 알아본다. 4장과 5장에서는 이미 제안된 대칭키 기반의 Forward Secure 무결성 기법과 공개키 기반의 Forward Secure 무결성 기법을 소개하고 6장에서는 소개한 기법들의 보안요구사항을 비교한다. 7장에서는 소개한 기법들의 계산적인 측면과 메모리적인 측면에서 비교분석과 각 환경에 맞는 기법들을 확인하고, 마지막으로 8장에서 결론을 맺는다.

2. 로그 시스템 위협모델 및 보안요구사항

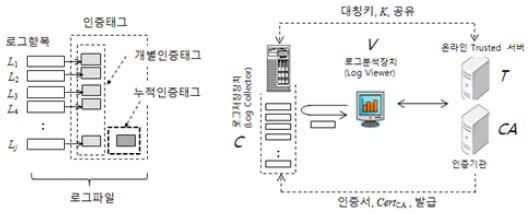
2.1 로그 시스템 보안위협 모델

로그파일은 다수의 로그항목들로 구성된다. 가장 기본적인 보안위협은 첫째, 특정 로그항목들에 대한 변조, 위조된 로그항목들의 삽입 그리고 기존 저장된 일부 로그항목들에 대한 삭제공격을 들 수 있다. 이것들은 넓은 의미에서의 변조공격(Modification Attack)으로 정의된다. 둘째로 저장된 로그항목들의 순서를 변경함으로써 로그파일에 기록된 원래의 의미를 왜곡시키는 순서변경공격 (Re-ordering Attack), 셋째로 로그파일의 끝부분에 존재하는 일정 개수의 로그항목들을 임의로 삭제시키는 절단공격(Truncation Attack)으로 구분할 수 있다. 변조공격에서의 삭제와 절단공격은 물리적인 측면에서의 효과는 동일하지만, 이들에 대응하기 위한 보안기법 설계는 상이하기에 차별성을 두어야 한다. 즉, 전자를 위한 보안기법이 그대로 후자를 위한 보안기법에 적용될 수는 없다.

2.2 로그 시스템 보안요구사항

2.1절에서 언급한 기본적인 보안위협에 대처하기 위해서는 결국 로그파일에 대한 무결성(Integrity)이 보장되어야 한다. 특히, 로그 시스템에 있어서는 외부 공격자가 시스템에 침투하여 무결성을 보장하기 위해 사용되는 키 정보가 노출 될 수 있다는 가정을 하고 있다. 결국, 로그저장장치에 보관된 키 정보가 노출이 된 이후의 로그항목들에 대한 무결성은 보장하지 못하지만, 그 이전에 축적된 로그항목들의 무결성에 대한 전방 안전성(Forward Security)의 확보가 기본적인 보안요구사항이 된다. 따라서 로그파일에 대한 무결성을 "전방 안전성 보장" 무결성(Forward Secure 무결성)이라고 한다. 일반적인 무결성 보장을 위한 보안기법들은 대칭키 방식, $MAC_K(L_i)$ 그리고 공개키 방식, $SIG_{SK}(L_i)$으로 구분된다. 이때, $MAC_K(.)$는 대칭키 K를 이용한 MAC 함수이고 $SIG_{SK}(.)$는 서명용 개인키 SK를 이용한 서명함수이다. (그림 2)는 무결성이 적용된 로그파일의 일반적인 구성을 보여주고 있다. 인증태그(Authentication Tag)는 대칭키 방식 또는 공개키 방식을 통해서 로그항목 L에 대한 무결성을 보장하는 역할을 한다. 인증태그는 개별 로그항목을 보호하는 개별인증태그(Individual Authentication Tag) 그리고 전체 로그항목들을 보호하는 누적인증태그(Aggregated Authentication Tag)로 구분된다. 로그저장장치 C에 저장된 로그항목들에 대한 무결성 검증은 로그분석장치 V가 온라인 Trusted 서버 T (대칭키 방식) 또는 인증기관 CA (공개키 방식)의 도움을 받아 진행된다. 로그항목들에 대한 무결성 검증은 로그 시스템의 운영 및 응용환경에 따라서 다양하게 구분된다. 첫째, 로그항목에 대한 검증이 개별 로그항목이냐 아니면 전체 로그항목들에 대한 것이냐에 따라서 개별검증(Individual Verification)과 누적검증(Aggregated Verification)으로 구분된다. 따라서, 개별검증은 개별인증태그 그리고 누적검증은 누적인증태그를 기반으로 행해진다. 둘째, 검증자체가 시스템 내부적으로 또는 공개적으로 이루어지느냐에 따라 내부검증(Private Verification)과 공개검증(Public Verification)으로 나뉜다. 따라서 무결성 보장기법이 대칭키 방식이면 내부검증이 되어야 하며, 공개키 방식이면 공개검증이 허용된다. 그 이유는 대칭키 방식의 경우는 검증용 키와 인증태그 생성키가 동일하여 조작의 가능성이 있기에 공개검증이 허용되어서는 안 된다. 셋째는 무결성 검증에 있어서 온라인 Trusted 서버의 도움을 받을 경우에는 온라인검증(Online Verification), 독자적으로 이루어질 경우에는 오프라인검증(Offline Verification)이라고 한다. 공개키 방식의 장

점은 오프라인검증이 가능하다는 데에 있다.



(그림 2) 로그파일의 구성 및 로그항목 검증

Forward Secure 무결성이 보장되기 위해서는 대칭키 방식에서는 대칭키, 공개키 방식에서는 서명용 개인키가 로그항목에 대한 인증태그를 생성할 때마다 갱신되어야 한다. 이를 Key Evolution이라고 하는데, 일반적으로 One-way hash function이 대칭키 또는 서명용 개인키에 적용되어 Key Evolution이 작동된다.

3. 적용 응용환경 및 요구사항

방화벽은 내부 네트워크와 외부 네트워크의 경계선에서 전달되는 패킷의 흐름을 조절하므로 보안의 핵심이 된다. 경계선에서의 방화벽은 소통되는 모든 네트워크 트래픽에 대한 정보를 감시하고, 정책에 의해 통제하는 역할을 수행하기 때문에 방화벽에서 생성되는 로그는 네트워크 운영과 보안에 중요한 판단자료가 된다. 또한, 방화벽에서 생성되는 로그는 네트워크 보안사고 발생을 원인들에 대하여 추적할 증거자료를 제공해주거나, 사고 발생 전 이상증후 혹은 외부의 해킹 시도를 탐지할 수 있어 사이버 범죄수사에 있어서 중요한 증거자료로 사용된다. 따라서, 방화벽에서 생성되고 저장된 로그는 공격자의 공격대상이 되고, 이에 대응하기 위해 로그 시스템 보안기법이 적용된다.

방화벽에서는 수없이 많은 로그들이 생성되고, 생성된 로그들을 분석하는 과정에 많은 자원을 사용한다. 따라서 방화벽에서는 로그의 Forward Secure 무결성을 위한 인증 태그 생성에 충분한 자원을 가지고 있지만 대부분의 자원을 로그 분석에 사용되기 때문에 많은 자원을 할당하기는 어려운 측면이 많다. 하지만 생성된 인증 태그 검증에는 많은 자원을 할당할 수 있고, 항상 네트워크에 연결되어 Online 검증과 TTP 서버에 접속이 가능하여 다양한 방식의 검증이 가능하다.

체내 이식형 의료기기는 몸 속에서 신체 상황을 파악하고 치료하는 장비이다. 이식형 의료기기는 고 대역폭의 무선 통신을 통해 원격 모니터링을 위한 데이터 전송을 지원하고, 다른 여러 이식형 의료기기 간의 통신을 통해 다양한 조치를 가능하게 한다. 이러한 체내 이식형 의료기기는 환자의 생명과 직결되기 때문에 의도적인 의료 사고, 보안 및 개인정보 보호 문제, 그리고 의도하지 않은 사고를 방지하기 위한 방안이 필요하다. 환자의 이식형 의료기기 데이터는 기기 내에서 뿐만 아니라 서버에 저장된 이후에도 계속 보호되어야 한다. 또한, 이식형 의료기기에 저장된 보안로그 및 원격 측정된 개인정보는 공격자에게 공개되지 않아야 하고, 공격자로부터 이전에 저장된 로그파일과 개인정보를 삭제하거나 새로운 데이터를 삽입 혹은 변조를 방지해야 한다[4].

공개키 암호화와 같은 강한 보안 메커니즘들은 많은 계산 시간 및 에너지 소비 측면에서 많은 비용을 요구하게 된다. 일반적인 센서 네트워크와 같이, 암호화의 사용은 보안과 이식형 의료기기의 수명과 성능 사이에서 중요한 요인이 된다. 또한, 안전한 통신을 위해 더 많은 자원을 사용할 경우 인증을 반복적으로 시도하는 방법이 같은 악의적인 DoS 공격이 가능하게 된다. 따라서, 안전한 이식형 의료기기의 운용을 위해서는 모든 외부 장치와의 통신이 기록되어야 하고, 이러한 기록에는 공격자의 변조 공격을 방지할 수 있도록 보안기법이 적용되어야 한다[5].

체내 이식형 의료기기에서는 많은 로그가 발생하지 않는다. 많은 양의 로그를 처리할 자원을 필요로 하지 않는다. 하지만 체내 이식형 의료기기에서는 공개키 암호화와 같이 많은 계산 능력을 필요로 하는 메커니즘을 사용할 수 없기 때문에 계산량에 제한적이다. 또한, 체내 이식형 의료기기의 특성상 항상 네트워크에 연결되기에는 어려움이 많다. 따라서 체내 이식형 의료기기에서는 적은 계산량과 적은 메모리를 요구하는 로그 저장 기법을 적용하는 것이 적합할 것이다.

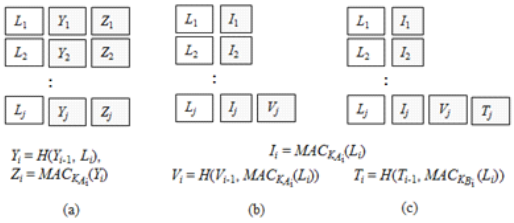
스마트기기의 보급과 네트워크 기술의 발전으로 인

터넷을 통해 금융서비스를 제공하는 전자금융거래의 사용량이 급증하고 있다. 그러나, 전자적인 장치를 이용한 전자금융거래의 대중화의 부작용으로 피싱 및 파밍과 같은 신종 사이버 사기나 사용자 PC에 악성코드를 설치하여 금융정보를 유출하여 비정상 자금이체를 하는 금융사고 건수도 함께 증가하고 있다. 이러한 전자금융거래에서의 로그는 피싱, 파밍 등 전자금융 신종 사기 수법에 의한 사고 예방뿐만 아니라 사고 발생 후 원인 추적 및 범죄수사에 있어서 중요한 증거자료로 사용된다. 또한, 개인 사용자 및 금융회사의 거래사실에 대한 부인을 방지하는 중요한 자료가 된다. 따라서, 전자금융거래 시 생성되는 로그는 보안기법이 적용되어 안전하게 저장되어야 한다.

전자금융거래는 현금자동입출금기(ATM) 또는 현금자동지급기(CD), 컴퓨터, 지급용 단말기, 스마트폰, 디지털 TV 등 다양한 전자기기에서 진행된다. 이러한 전자기기들은 로그의 Forward Secure 무결성 인증태그 생성을 위한 충분한 자원을 가지고 있지 않은 경우도 있지만, 전자금융거래 시 많은 로그가 발생하지 않고 기기들의 특성상 전원의 연결이 충분하기 때문에 계산량 및 에너지 소비 측면에서 자유롭다. 또한, 전자금융거래 시 항상 Online 상태로 진행이 되기 때문에 다양한 로그 저장기법의 적용이 가능하다.

4. 대칭키 기반의 Forward Secure 무결성 기법

4.1 Key Evolution 방식



(그림 3) 대칭키 기반의 Forward Secure 무결성 기법

(그림 3)은 3가지 유형의 대칭키 기반의 Forward Secure 무결성 기법 [6, 7, 8]을 보여주고 있다. 여기에 적용되는 Key Evolution 방식은 공통적으로 One-way hash function $H(.)$을 이전 키 K_{A_i} 및 K_{B_i}에 적용하여 새로운 키 $K_{A_{i+1}} = H(K_{A_i})$와 $K_{B_{i+1}} = H(K_{B_i})$를 생성한다. 다음 로그항목 L_{i+1}에 해당되는 인증태그 $(Z_{i+1}, I_{i+1}, V_{i+1}, T_{i+1})$를 생성하기 위해 적용할 새로운 키가 생성되면 이전 키는 시스템에서 안전하게 삭제된다.

4.2 인증태그의 생성 및 검증

개별검증을 위한 개별인증태그 (I_i)는 로그항목 L_i, $(i = 1, 2, \cdots, j)$에 MAC 함수를 적용하여 생성되며, 누적검증을 위한 누적검증태그 (Z_i, V_i, T_i)는 (그림 3) (a), (b), (c)와 같이 해시체인 방식을 통해서 생성된다. 검증을 위해 로그분석장치 V는 온라인 Trusted 서버 T에게 해당 로그항목들과 인증태그들을 보내어 간접적으로 검증하는 방식과 직접 무결성 대칭키를 다운받아서 검증하는 방식으로 구분된다. (그림 3) (a)는 전자, 그리고 (그림 3) (b)와 (c)는 후자에 해당한다. (그림 3) (a)의 경우, 로그분석장치 V는 누적검증을 위해서 $\{(L_1, Y_1), (L_2, Y_2), \cdots, (L_j, Y_j)\}$를 $Y_i = H(Y_{i-1}, L_i)$를 통해 자체적으로 검증하고, 최종적으로 (Y_j, Z_j)를 T에게 전송하여 검증 받게 된다. 직접 Z_i를 생성하지 않고 Y_i를 생성하는 이유는 무결성 대칭키를 다운받지 않고 직접 검증 받을 최소한의 정보를 전송하기 위한 목적이다. 따라서 (그림 3) (a)는 대칭키 방식임에도 불구하고 공개검증을 가능하게 한다 [6]. 내부검증 목적의 (그림 3) (b)와 (c)의 차이점은 로그분석장치 V의 내부적인 신뢰관계에 따른 분류이다. V에게 직접 무결성 대칭키 K_{A_1}을 전달할 경우에 후속 키들을 도출하여 개별검증은 물론 누적검증도 가능하다. 하지만, 이 경우에 V가 기존 로그항목들에 대한 조작이 가능하기 때문에 (그림 3) (b)의 경우 [7, 8]는 내부적으로 절대적인 신뢰관계가 구축된 V의 경우에 가능하다. 그렇지 않다면 (그림 3) (c)와 같이 2개의 무결성 대칭키 K_{A_i}, K_{B_i}를 통해서 2개의 누적검증태그를 생성하고 검증을 위해서는 K_{A_1}만을 제공하는 것이다 [7]. 따라서 만약 V가 기존 로그항목들을 조작한다면 V_i에 대해서는 가능하지만 T_i에 대해서는 불가능하게 된다.

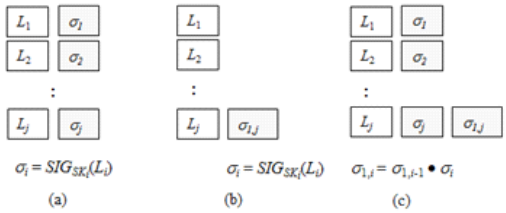
4.3 대칭키 기반의 무결성 기법분석

(그림 3) (a)는 누적검증 및 공개검증을 가능하게 하는 온라인 검증기법이다. 단순 변조에 대한 무결성이 보장되지만 절단공격에는 취약하다. Y_i 그리고 누적인증태그 Zi가 생성되는 방식의 특성상 그리고 매 로그항목마다 Y_i와 Z_i가 존재하기 때문에 마지막 로그항목과 해당 인증태그의 삭제가 행해져도 이에 대한 검출은 불가능하게 된다. 반면에 (그림 3) (b)와 (c)의 경우는 마지막 로그항목에 대해서만 누적인증태그가 존재하기 때문에 이들에 대한 삭제의 경우 그 이전 로그항목까지의 누적인증태그의 도출이 불가능하기에 절단공격에는 안전하다. (그림 3) (a), (b), (c) 모두 해시체인방식에 기반을 두고 있기 때문에 공통적으로 순서변경공격에는 안전하다.

5. 공개키 기반의 Forward Secure 무결성 기법

5.1 Key Evolution 방식

서명용 개인키 SK 그리고 검증용 공개키 PK가 사용되는 공개키 기반의 Forward Secure 무결성 기법에서, Key Evolution 방식은 2가지 유형이 존재한다. 첫째, j개의 로그항목을 서명하기 위해서 j개의 공개키 / 개인키 쌍을 독립적으로 생성하는 방식이다 [9]. 즉, $\{(PK_1,SK_1),(PK_2,SK_2),\cdots(PK_j, SK_j)\}$. 둘째, 현재 서명용 개인키 SK_i에 One-way hash function $H(.)$을 적용하여 새로운 개인키 $SK_{i+1}= H(SK_i)$를 생성하며, 이에 대응되는 공개키의 생성은 서명기법에 따라 상이하게 된다 [7, 10, 12, 13]. 겹선형 사상(Bilinear Map)에 기반을 둔 BLS (Boneh-Lynn-Schacham) 서명 [11]을 사용할 경우에는 $PK_i = g_2^{SK_i}$. 이때, g_2는 겹선형 사상 $e: G_1 \times G_2 \rightarrow G_T$에서 G_2의 생성자(Generator)이다. ECDLP (Elliptic Curve Discrete Logarithm Problem)에 기반을 둔 서명 [14]에서는 $SK_i = (a_i, b_i)$, $PK_i = (A_i=a_iG, B_i = b_iG)$. 이때, G는 Prime Field F_p상의 타원곡선 $E(F_p)$에서 정의되는 군(group) G의 생성자(Generator)이며, $a_i, b_i \in F_p$이다.



(그림 4) 공개키 기반의 Forward Secure 무결성 기법

5.2 인증태그의 생성 및 검증

(그림 4) (a) 방식은 가장 j개의 로그항목들을 $\{(PK_1,SK_1),(PK_2,SK_2),\cdots(PK_j, SK_j)\}$을 이용해 개별인증태그 σ_i, $i = 1, 2, \cdots, j$를 생성하고 개별검증하는 가장 간단한 방식이다. 이 경우에 서명용 개인키의 생성은 상호 독립적으로 생성이 된다 [9]. (그림 4) (b) 방식 [12, 13]은 ECDLP에 기반을 둔 서명에 기반을 두고 있다. j개의 로그항목들에 대한 누적인증태그 $\sigma_{1,j}$는 j개의 서명용 개인키 $SK_i = (a_i, b_i)$, $i = 1, 2, \cdots, j$로 다음과 같이 생성된다. $\sigma_{1,j} = \sigma_{1,j-1} + \sigma_j$, where $\sigma_j = a_jH_1(L_j \,//\, j) + b_j$. 이에 대한 누적검증은 공개키 $PK_i = (A_i=a_iG, B_i = b_iG)$, $i = 1, 2, \cdots, j$가 이용된다. (그림 4) (c) 방식은 개별 로그항목들은 BLS 서명을 통해 개별인증태그 σ_i을 생성하고, 이들을 통합하여 누적인증태그 $\sigma_{1,i}$를 구성한다 [7, 10]. 즉, 겹선형 사상 $e: G_1 \times G_2 \rightarrow G_T$에서 소수 q 그리고 생성자 $g_2 \in G_2$에 대해 서명용 개인키 $SK \in Z_q$에 대한 공개키는 $PK = g_2^{SK} \in G_2$. 로그항목 L_i에 대한 개별인증태그 σ_i은 full-domain hash function $H_1:\{0,1\}^* \rightarrow G_1$을 적용하여 계산된다. (L_i, σ_i), where $\sigma_i = h_i^{SK_i}$ and $h_i = H_1(L_i)$. 개별검증을 위해서는 먼저 $h_i = H_1(L_i)$을 구한 후에 $e(\sigma_i, g_2) = e(h_i,PK_i)$임을 확인하게 된다. j개의 로그항목 $(L_1,L_2,\cdots L_j)$에 대한 누적인증태그 $\sigma_{1,j}$은 다음과 같이 정의된다. $\sigma_{1,j} = \sigma_{1,j-1} \bullet \sigma_j$, where $\sigma_j = h_j^{SK_j}$, $h_j = H_1(j, L_j)$ and $\sigma_{1,0} = 1$. 누적검증은 공개키 $(PK_1= g_2^{SK_1}, PK_2=g_2^{SK_2}, \cdots, PK_j= g_2^{SK_j})$를 이용하여 $e(\sigma_1, g_2)e(\sigma_2, g_2)\cdots e(\sigma_j, g_2)=e(h_1,PK_1)\ e(h_2,PK_2)\cdots e(h_j, PK_j)$을 확인한다.

<표 1> 보안요구사항 및 기능 비교분석

	(그림 3) (a)	(그림 3) (b)	(그림 3) (c)	(그림 4) (a)	(그림 4) (b)	(그림 4) (c)
기본 변조공격	O	O	O	O	O	O
순서변경공격	O	O	O	O	O	O
절단공격	X	O	O	X	O	O
개별/누적검증	누적	O	O	개별	누적	O
공개검증	O	X	O	O	O	O
온라인/오프라인 검증	Online	Offline	Offline	Offline	Offline	Offline
온라인 Trusted 서버/ 인증기관	Trusted Server	Trusted Server	Trusted Server	인증기관	인증기관	인증기관

5.3 공개키 기반의 무결성 기법분석

(그림 4) (a) 방식은 태그생성 및 검증에 있어서 가장 간단하지만, 안전성 및 효율적인 측면에서는 유효하지 않다. 비록 독립적인 키 생성방식이지만 Key Evolution 조건을 만족하고 기본적인 무결성이 제공되지만, 개별인증태그 간의 연관관계가 없기에 절단공격에는 안전하지가 않다. 반면에 (그림 4) (b)와 같이 누적인증태그만 유지할 경우 메모리측면에서는 긍정적이지만, 개별 로그항목에 대한 검증에는 비효율적이 된다. (그림 4) (c)의 경우는 겹선형 사상 기반의 암호를 사용하기에 태그생성 및 검증과정에서 타 기법에 비해 비교적 많은 지연이 초래된다.

6. 보안요구사항 비교분석

모든 구조에서 로그 저장 기법에서 기본적으로 요구하는 변조 공격과 순서 변경 공격에 저항하는 전방 안전성은 보장하고 있다. (그림 3) (a)의 경우, Y_i 그리고 누적인증태그 Z_i가 생성되는 방식의 특성상 그리고 매 로그항목마다 Y_i와 Z_i가 존재하기 때문에 절단 공격에 취약하다. 또한, 누적검증을 위한 누적인증태그는 생성하지만 개별검증을 위한 개별인증태그는 생성하지 않기 때문에 누적검증만 보장하게 된다. (그림 3) (a)는 대칭키 방식의 기법이지만 (Y_i,Z_i)를 T에 전송하여 최종 검증을 하기 때문에 공개검증이 가능하고, 이러한 검증 과정을 거쳐야 하기 때문에 온라인 TTP가 필요하게 된다.

(그림 3) (b)와 (그림 3) (c)의 경우, 개별인증태그와 누적인증태그를 생성하기 때문에 개별검증과 누적검증을 모드 보장하고 절단공격에 안전하다. (그림 3) (c)에서는 두 개의 누적인증태그를 생성하기 때문에 추가적으로 공개검증이 가능하다.

(그림 4) (a)의 경우, 기본적인 전방 안전성은 보장하지만 개별인증태그 간의 연관관계가 없기 때문에 절단공격에 취약하고 개별인증태그만 생성하여 누적검증은 불가능하다. 하지만 공개키 기반의 무결성 기법으로 공개검증이 가능하다.

(그림 4) (b)와 (그림 4) (c)의 경우, 모두 누적인증태그를 생성하기 때문에 누적검증이 가능하고 절단공격에 안전하고 공개키 기반의 무결성 기법으로 공개검증이 가능하다. (그림 4) (c)에서는 개별인증태그를 생성하여 저장하면서 추가적으로 개별검증도 보장하게 된다.

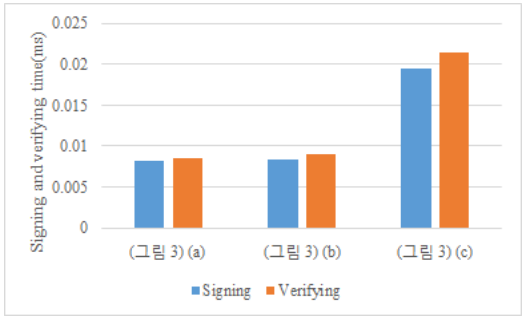
위에서 설명한 방화벽과 같은 환경에서는 수많은 로그가 생성되고 저장된다. 만약 방화벽에서 누적검증만이 가능한 로그 저장 기법을 적용했다면, 특정 부분의 로그들의 무결성을 확인하기 위해 모든 저장된 로그에 대한 무결성을 확인해야만 한다. 수많은 로그가 저장되는 방화벽에서는 이러한 검증 방식은 효율적이지 못하기 때문에 개별검증이 필수적인 보안요구사항 된다. 추가적으로 내부 관리자를 신뢰할 수 없는 경우에는 공개검증이 가능한 기법을 적용하는 것이 적합할 것이다. 체내 이식형 의료기기의 경우, 의료 사고나 개인정보 보호 문제로 저장된 로그의 무결성을 확인해야 하는 경우가 대부분이다. 따라서 검증 과정이 공개적으로 이루어질 수 있는 공개검증이 필수적인 보안요구

사항이 된다. 전자금융거래의 경우 또한, 개인 사용자와 금융회사의 거래사실에 대한 부인을 방지하기 위해 공개 검증이 필수적인 보안요구사항이 된다.

7. 성능분석 및 비교평가

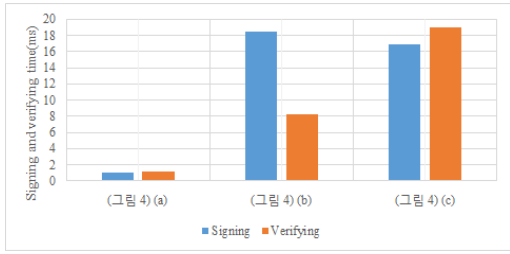
앞에서 소개한 기법들은 구조와 사용 함수들에 따라 많은 성능 차이를 보인다. 이러한 기법들의 처리 시간 및 메모리 효율을 비교하기 위해 구현을 통해 실환경에서의 처리 시간과 메모리 요구사항을 비교한다. 실구현은 Pentium 2 30GHz 프로세서, 4GB RAM의 데스크탑에서 진행하였다.

7.1 성능분석



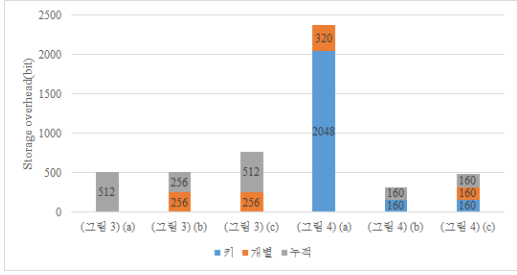
(그림 5) 단일 로그항목의 암호화 및 검증 소요시간

대칭키 기반의 Forward Secure 무결성 기법들은 대칭키의 특성상 공개키 기반의 Forward Secure 무결성 기법들에 비해 적은 소요시간을 보인다. 각 기법들은 로그항목 무결성 검증에 사용되는 인증태그들의 생성 구조에 따라 암호화 및 검증에 사용되는 소요시간의 차이를 보인다. (그림 3) (a) 암호화의 경우, Key evolution과 Y_i 생성을 위한 One-way hash function $H(.)$와 Z_i 생성을 위한 MAC 함수를 사용한다. (그림 3) (b) 암호화의 경우, 또한 구조는 다르지만 적용되는 함수의 종류와 수가 같기 때문에 (그림 3) (a)의 소요시간과 차이를 보이지 않는다. 하지만, (그림 3) (c)의 경우, (그림 3) (b)의 구조에서 추가적으로 T_i를 생성하기 때문에 다른 대칭키 기법들에 비해 많은 암호화 소요시간을 보인다. 무결성 검증도 암호화와 같은 이유로 (그림 3) (a)와 (그림 3) (b)는 비슷한 소요시간을 보이고, (그림 3) (c)는 다른 대칭키 기법들에 비해 많은 검증 소요시간을 보인다.



(그림 6) 단일 로그항목 암호화 및 검증 소요시간

공개키 기반의 Forward Secure 무결성 기법들은 사용되는 암호화 함수들은 대칭키 암호화에 비해 많은 소요시간을 갖기 때문에 대칭키 기반의 Forward Secure 무결성 기법들에 비해 상당히 많은 소요시간을 보인다. (그림 4) (a)가 (그림 4) (b)와 (그림 4) (c)에 비해 적은 소요시간을 보이는 것은 사용되는 공개키 함수가 다르기 때문이다. (그림 4) (a)는 DSA 서명과 검증 함수를 사용하고 (그림 4) (b)와 (그림 4) (c)는 겹선형 사상에 기반을 둔 BLS 서명과 검증 함수를 사용하기 때문에 소요시간에 큰 차이를 보이게 된다. (그림 4) (b)와 (그림 4) (c)는 같은 함수를 사용하여 암호화와 검증을 하기 때문에 비슷한 소요시간을 보인다. (그림 4) (c)의 경우, (그림 4) (b)와는 다르게 개별검증과 누적검증을 동시에 진행하기 때문에 무결성 검증에서 더 많은 소요시간을 보이게 된다.



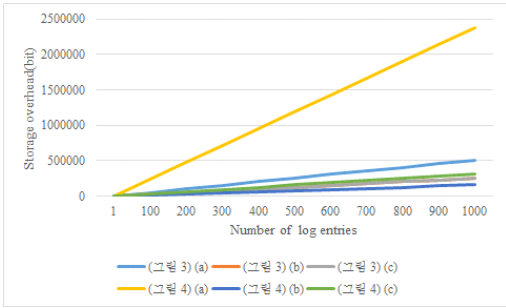
(그림 7) 단일 로그항목의 메모리 요구사항

대칭키 기반의 Forward Secure 무결성 기법과 공개키 기반의 Forward Secure 무결성 기법 간의 가장

큰 차이는 검증에 필요한 키의 저장 여부에 있다. 대칭키 기반의 기법들의 경우, Seed만 안전하게 저장한 후 검증에 사용되는 키는 One-way hash function $H(.)$를 사용해 도출한 후 검증 과정을 진행한다. 공개키 기반의 기법들의 경우, 검증에 사용되는 공개키를 도출해 낼 수 없기 때문에 검증 태그에 해당하는 공개키를 저장해야 한다. 따라서 공개키 기반의 기법들은 대칭키 기반의 기법들에 비해 검증에 사용되는 공개키를 저장하는 추가적인 메모리가 요구된다.

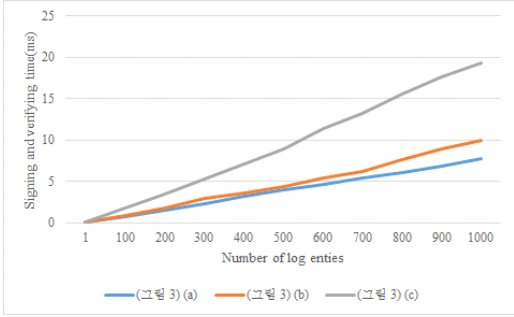
(그림 3) (a)의 경우, 누적 검증을 위한 Y_i와 최종 검증을 위한 Z_i를 모두 저장하는 구조이다. 만약 n -bit의 암호화 함수를 사용한다면, j 개의 로그항목이 저장된 로그파일의 무결성 검증을 위해 추가적인 메모리는 $2jn$ bits를 요구하게 된다. (그림 3) (b)의 경우, 로그항목의 개별검증을 위한 I_i와 누적검증을 위한 하나의 V_i를 저장하는 구조이다. 따라서 n-bit의 암호화 함수를 사용한다면, j 개의 로그항목이 저장된 로그파일의 무결성 검증을 위해 $n(j+1)$ bits를 추가적으로 요구하게 된다. (그림 3) (c)의 경우는 (그림 3) (b)와 같은 구조를 갖고 추가적으로 하나의 T_i를 저장하는 구조이다. 따라서 (그림 3) (b)의 요구 비트 수에 n bits가 추가적으로 필요하게 되므로 총 $n(j+2)$ bits를 요구하게 된다. 메모리적인 측면에서 (그림 3) (a)가 (그림 3) (b)와 (그림 3) (c)에 비해 약 두 배의 메모리를 요구하게 된다.

공개키 기반의 Forward Secure 무결성 기법에서는 대칭키 기반의 Forward Secure 무결성 기법과는 다르게 서명 생성에 사용된 개인키를 삭제하기 때문에 그에 대응하는 공개키를 저장해 두어야 검증이 가능하다. 따라서 공개키 기반의 기법들을 사용하려면 로그의 무결성을 보장하기 위한 태그와 추가적으로 공개키가 저장되어야 하므로 대칭키 기반의 기법들에 비해 추가적인 메모리가 요구된다. 특히, (그림 4) (a)의 경우 DSA 서명을 사용하기 때문에 검증을 위한 공개키의 길이가 개별 검증을 위한 태그의 길이보다 더 많은 메모리를 차지하게 된다. (그림 4) (b)와 (그림 4) (c)의 경우 BLS 서명을 사용하기 때문에 저장해야 하는 공개키와 개별 및 누적 검증을 위한 태그의 길이 또한 DSA 서명을 사용하는 (그림 4) (a)보다 적은 메모리를 요구하게 된다.



(그림 8) 로그항목 수의 증가에 따른 메모리 요구사항

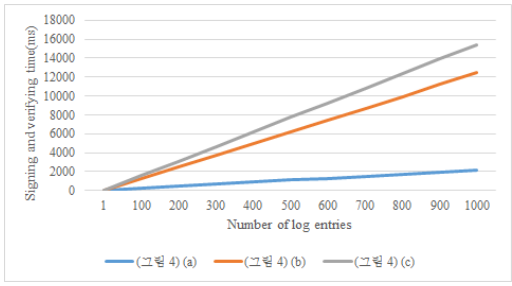
대칭키 기반의 기법인 (그림 3) (a)의 경우에는 대칭키 기반의 기법이지만 다른 (그림 3) (b)와 (그림 3) (c)의 기법들에 비해 로그항목당 두 개의 태그를 저장하기 때문에 많은 메모리를 요구하게 된다. (그림 3) (b)와 (그림 3) (c)의 경우에는 누적 인증 태그의 수가 다르다는 것을 제외하면 같은 구조를 사용하기 때문에 메모리 요구사항은 큰 차이를 보이지 않는다. 공개키 기반의 기법인 (그림 4) (a)의 경우에는 개별 인증 태그와 같이 저장해야 하는 공개키의 길이가 길기 때문에 다른 기법들에 비해 가장 많은 메모리를 요구한다. (그림 4) (b)와 (그림 4) (c)의 경우에는 공개키 기반의 기법이지만 (그림 4) (a)에 비해 키와 태그의 길이가 짧기 때문에 대칭키 수준의 메모리를 요구한다.



(그림 9) 로그항목 수의 증가에 따른 암호화 및 검증 시간

대칭키 기반의 Forward Secure 무결성 기법들의 경우, 공개키 기반의 기법들보다 비교적 적은 시간을 보이지만 기법들간의 구조의 차이로 암호화 및 검증

시간에 차이를 보이게 된다. (그림 3) (a)는 로그항목의 누적 검증을 위해 Y_i를 확인하고 최종 검증을 위해 Z_i를 확인하는 구조이다. 다른 기법들에 비해 비교적 간단한 구조를 갖고 있어 빠른 처리 속도를 보인다. 또한, 많은 로그항목의 암호화 및 검증 시간에서도 처리 시간의 유리함을 보인다. (그림 3) (b)는 로그의 개별 검증을 위해 I_i를 확인하고 누적검증을 위한 V_i를 확인하는 구조이다. (그림 3) (a)와 비슷한 처리 속도를 보이며 많은 로그항목에 대해서도 적은 시간으로 효율적임을 보인다. (그림 3) (c)의 경우는 (그림 3) (b)에서 추가된 하나의 누적 인증 태그 T_i를 검증하는 구조이며, 추가된 누적 검증을 하는 만큼 다른 대칭키 기법들에 비해 많은 소요시간을 보인다. 하지만, 이러한 대칭키 기법들은 공개키 기법들에 비해 매우 적은 시간으로 암호화와 검증을 수행한다.



(그림 10) 로그항목 수의 증가에 따른 암호화 및 검증 시간

공개키 기반의 Forward Secure 무결성 기법들은 비대칭 암호화 함수인 DSA 서명 함수와 BLS 서명 함수를 사용하면서 많은 소요시간을 보이게 된다. 공개키 기반의 기법들 중에서도 DSA 서명 및 검증 함수를 사용하는 (그림 4) (a)와 BLS 서명 및 검증 함수를 사용하는 (그림 4) (b)와 (그림 4) (c)의 서명 및 검증시간이 많은 차이를 보인다. (그림 4) (a)의 경우에는 개별 검증을 위해 로그항목 마다 다른 개인키를 사용하여 DSA 서명함수로 서명하고, 그에 대응되는 공개키로 개별 인증 태그를 확인하는 구조이다. 공개키 기반의 기법들 중 간단한 구조이며 DSA 서명 함수를 사용해 비교적 빠른 처리 속도를 보인다. 따라서 많은 로그항목의 암호화와 검증을 처리함에 있어서 유리함을 보인다. (그림 4) (b)의 경우 BLS 서명 함수를 사용하여 누적 검증을 위한 $\sigma_{I,i}$를 생성하고 이를 검증하는 구조이다. 누적 인증 태그를 생성하고 검증하는 과정은 간단하지만 BLS 서명을 사용하면서 DSA 서명을 사용하는 (그림 4) (a)의 구조에 비해 많은 시간을 요구한다. (그림 4) (c)는 (그림 4) (b)의 구조에 개별 검증을 위한 si도 같이 저장하는 구조이다. 개별 검증 과정을 거치기 때문에 (그림 4) (b)의 암호화 및 검증에 소요되는 시간이 더 많게 나타난다.

7.2 비교평가

방화벽에서는 수많은 로그의 생성으로 로그의 Forward Secure 무결성 보장을 위한 인증태그 생성에 많은 자원을 할당하기 어렵다. 또한, 저장된 수많은 로그들의 효율적인 검증을 위해 개별검증은 필수적인 보안요구사항이다. 많은 계산량을 요구하는 공개키 기반의 Forward Secure 무결성 기법들과 누적검증만이 가능한 (그림 3) (a)는 적용하기 어렵다. 따라서 방화벽과 같은 환경에서는 (그림 3) (b)가 적용하기 가장 적합한 기법이고, 추가적으로 공개검증이 필요한 환경에서는 (그림 3) (c)의 기법을 적용하는 것이 적합할 것으로 보인다.

체내 이식형 의료기기와 같은 경우에는 생성되는 로그의 수는 적지만 계산 시간 및 에너지 소모 측면에서 많은 비용이 드는 기법을 적용하기 어렵다. 또한, 의료 분쟁과 개인정보 보호 문제로 저장된 로그의 무결성을 검증해야 하기 때문에 공개검증이 필수적인 보안요구사항이 된다. 많은 계산량을 요구하는 공개키 기반의 Forward Secure 무결성 기법들과 공개검증이 불가능한 (그림 3) (b)는 체내 이식형 의료기기에는 적용하기 어렵다. 따라서 (그림 3) (c)가 체내 이식형 의료기기에 적용하기 적합한 기법이다.

전자금융거래의 경우 발생하는 로그의 수가 적어 계산량과 에너지 소모 측면에서 다양한 로그 저장기법을 적용할 수 있다. 하지만, 다양한 전자 기기에서 전자금융거래가 진행되기 때문에 메모리 사용 측면에서 제한적이라고 할 수 있다. 또한, 개인 사용자와 금융회사 간의 거래 사실 부인방지를 위한 공개검증이 가능해야 하기 때문에 (그림 3) (b)와, 공개검증이 가능하지만 메모리를 많이 요구하고 절단공격에 취약한 (그

림 4) (a)의 적용은 바람직하지 않다. 따라서 전자금융 거래의 경우 메모리 적게 요구하고 공개검증이 가능한 (그림 4) (b)의 적용이 가장 적합할 것이며, 추가적으로 개별검증이 필요한 환경이라면 (그림 3) (c)와 (그림 4) (c)의 적용이 적합할 것이다.

8. 결 론

다양한 IT 시스템으로부터 발생되는 로그 데이터는 그 활용범위가 다양하다. 수집된 로그 데이터를 통해서 해당 IT 시스템 사용내역에 대한 감사 및 개선 등이 가능하기 때문에 해당 로그 데이터에 대한 무결성 확보가 중요하다. 본 논문에서는 기존에 제안된 Forward Secure 무결성이 보장되는 로그 데이터에 대한 이론적인 기법들을 분석하였다. 다양한 요구사항과 응용환경이 존재하기 때문에 "One-size fits all" 개념의 기법설계는 어렵다는 것을 기존기법들에 대한 분석을 통해서 확인하였다. 기존연구는 공격자의 침입이전의 로그 데이터에 대한 보안만을 대상으로 하였으나, 향후에는 침입 이후의 로그 데이터 보안에 대한 활발한 연구가 필요할 것으로 판단된다.

참고문헌

[1] A. A. Chuvakin, K. J. Schmidt and C. Phillips, Logging and Log Management, Elsevier, 2013.

[2] K. Malasri and L. Wang, "Securing Wireless Implantable Devices for Healthcare: Ideas and Challenges," IEEE Communications Magazine, vol. 47, no. 7, pp. 74-80, July 2009.

[3] Y. Wang, and Y. Zheng, "Fast and Secure Magnetic Worm Storage Systems," In Proc. of the 2ndIEEEInternationalSecurityinStorageWorkshop (SISW'03), pp. 11 - 25, Oct. 31, 2003.

[4] D. Halperin, T. S. Heydt-Benjamin, K. Fu, T. Kohno, and W. H. Maisel, "Security and Privacy for Implantable Medical Devices," IEEE Computer Society, Vol. 7, No. 1, January-March 2008.

[5] S. Gollakota, H. Hassanieh, B. Ransford, D. Katabi, K. fu, "They Can Hear Your Heartbeats: Non-Invasive Security for Implantable Medical Devices," ACM SIGCOMM 2011, Toronto, Ontario, Canada, 2011.

[6] J. Kelsey and B. Schneier "Secure Audit Logs to Support Computer Forensics," ACM Transactions on Information and System Security, vol.2, no.2, pp. 159-176, 1999.

[7] D. Ma and G. Tsudik, "A New Approach to Secure Logging," ACM Transactions on Storage, vol.5, Issue 1, pp. 2:1-2:21, Mar. 2009.

[8] R. Accorsi, "BBox : A Distributed Secure Log Architecture," Public Key Infrastructures, Services and Applications, LNCS, vol. 6711, pp 109-124, 2011.

[9]] J. E. Holt, "Logcrypt: Forward Security and Public Verification for Secure Audit Logs," In Proc. of the Australasian Workshops on Grid Computing and E-research, vol.54, pp. 203-211, Tasmania, Jan. 2006.

[10] D. Ma and G. Tsudik, "Forward-Secure Sequential Aggregate Authentication," " In Proc. of the IEEE Symposium on Security and Privacy, pp. 86-91, Berkeley, May 2007

[11] D. Ma, "Practical Forward Secure Sequential Aggregate Signatures," In Proc. of the ACM Symposium on Information, Computer and Communications Security, pp.341-352, Tokyo, Mar. 2008

[12] A. A. Yavuz and P. Ning, "BAF: An Efficient Publicly Verifiable Secure Audit Logging Scheme for Distributed System," In Proc. of the Annual Computer Security Applications Conference, pp. 219-218, Honolulu, Dec. 2009.

[13] A. A. Yavuz, P. Ning, and M. Reiter, "BAF and FI-BAF: Efficient and Publicly Verifiable

Cryptographic Schemes for Secure Logging in Resource-Constrained Systems," ACM Transactions on Information and System Security, vol. 15, Issue 2, pp. 9:1-9:28, July 2012.

[14] D. Boneh, "The Decision Diffie-Hellman Problem," In Proc. of the Third Algorithmic Number Theory Symposium, LNCS, vol. 1423, pp. 48 - 63, 1998.

[15] E. Mykletun, M. Narasimha, and G. Tsudik, "Signature Bouquets: Immutability for Aggregated/Condensed Signatures," In Proc. of the European Symposium on Research in Computer Security (ESORICS), pp. 160-176, France, Sep. 2004.

[저 자 소 개]

강 석 규 (Seok-Gyu Kang)

2014년 2월 단국대학교
컴퓨터학과 학사
2015년 3월~현재 단국대학교
컴퓨터학과 석사과정

email : ksg890528@naver.com

박 창 섭 (Chang-Seop Park)

1983년 2월 연세대학교
경제학과 학사
1987년 2월 Lehigh University
컴퓨터과학과 석사
1990년 2월 Lehigh University
컴퓨터과학과 박사
1990년 3월~현재 단국대학교
컴퓨터학과 교수

email : csp0@dankook.ac.kr