情報保護學會論文誌
第19卷 第6號, 2009. 12

차량용 블랙박스 시스템을 위한 실시간 무결성 보장기법*

김 윤 규,† 김 범 한, 이 동 훈‡
고려대학교 정보경영공학전문대학원

Real-time Integrity for Vehicle Black Box System*

Yungyu Kim,† Bum Han Kim, Dong Hoon Lee‡
Graduate School of Information Management and Security, Korea University

요 약

차량용 블랙박스는 음성, 영상 및 자동차의 여러 운행정보를 저장하는 매체이며, 이를 근거로 사고의 재구성이 가능하기 때문에 최근 자동차 시장에서 주목을 받고 있다. 또한 상업용 차량을 중심으로 블랙박스의 장착이 확산되면서 수년 내에 시장은 더욱 커질 전망이다. 그러나 현재의 블랙박스는 저장된 데이터에 대한 변경을 확인할 수 있는 무결성을 제공하지 못하기 때문에 사고 원인에 대한 법적 증거로 채택되기에는 적합하지 않다. 즉, 블랙박스는 생성한 데이터를 단지 저장만 할 뿐이기 때문에 저장된 데이터는 외부공격자나 내부공격자에 의해 위, 변조될 수가 있다. 무결성을 보호받지 못한 데이터는 신뢰 받을 수 없기 때문에 이점은 자동차 보험회사나 법정에서 큰 이슈이다. 본 논문에서는 이러한 문제를 해결할 수 있는 차량용 블랙박스 시스템을 위한 실시간 데이터의 무결성을 보장하는 기법을 제안하고, 이 기법을 구현한 시뮬레이션 프로그램의 실험 결과를 제시한다.

ABSTRACT

Recently, a great attention has been paid to a vehicle black box device in the auto markets since it provides an accident re-construction based on the data which contains audio, video, and some meaningful driving informations. It is expected that the device will get to promote around commercial vehicles and the market will greatly grow within a few years. Drivers who equips the device in their car believes that it can find the origin of an accident and help an objective judge. Unfortunately, the current one does not provide the integrity of the data stored in the device. That is the data can be forged or modified by outsider or insider adversary because it is just designed to keep the latest data produced by itself. This fact cause a great concern in car insurance and law enforcement, since the unprotected data cannot be trusted. To resolve the problem, in this paper, we propose a novel real-time integrity protection scheme for vehicle black box device. We also present the evaluation results by simulation using our software implementation.

Keywords: Vehicle Black Box, Real-time Integrity, Integrity Protection

접수일(2009년 10월 7일), 수정일(2009년 11월 9일),
게재확정일(2009년 11월 30일)
* 본 연구는 지식경제부 및 한국산업기술평가관리원의 IT산업원천기술개발사업의 일환으로 수행하였음. (2009-F056-01, Car-헬스케어 보안 기술개발)

† 주저자, menbal69@korea.ac.kr
‡ 교신저자, donghlee@korea.ac.kr

I. 서 론

비행기 블랙박스는 비행기가 추락하였을 때에 사고 당시의 상황을 재현하기 위해 비행기의 고도 및 속도, 동작 상태, 조정석 안의 목소리, 관제탑과의 교신 등을 기록하고 사고의 원인 규명을 위한 단서를 찾기 위해 쓰인다.

이와 같은 개념으로 차량용 블랙박스(EDR, Event Data recorder)는 일상적으로 빈번하게 일어나는 교통사고들의 원인들을 파악할 목적으로 차량에 탑재되기 시작했다. 사고가 발생 했을 때에 가해자와 피해자의 주장이 서로 다를 경우에는 사고 당시의 상황을 재현할 수가 없기 때문에 사실 판단에 많은 어려움이 있다. 또한 자동차가 주차 되어 있을 때에 누군가 접촉사고를 내거나 도난 사고를 낸다면 그 범인을 찾는 것은 매우 어렵다. 이러한 환경에서 차량용 블랙박스는 블랙박스 내에 저장된 운전자의 운행정보 및 실시간 동영상정보를 통해서 사고원인을 찾을 수 있는 주요한 매체로 활용될 수 있다. 국내에서는 국토해양부, 기술표준원, 텔레매틱스산업협회가 표준안을 마련 중이며 상업용 차량을 중심으로 차량용 블랙박스를 의무화하는 법안이 2011년에 법제화 될 전망이다. 따라서 그로 인한 관련 시장도 더욱 커질 전망이다. 업계의 차량용 블랙박스 연간 판매실적은 지난해 약 6만 6000대(약 120억 원)이었으며 올해는 약 10만대(약 200억 원) 이상 성장할 것으로 추정되고 있다[1]. 실제로 서울, 경기도 등의 지방자치단체를 중심으로 택시, 버스 등에 블랙박스 의무 장착이 추진되고 있다. 또한 최근의 법원 판결이 운전자의 정확한 사고 입증을 요구하는 추세가 강화되면서 일반 소비자들도 차량용 블랙박스의 탑재를 선호할 것으로 예상된다.

세계적인 동향도 비슷하다. 미국의 연방교통부(DOT, Department of Transportation)에서는 2004년 블랙박스 표준안을, 고속도로안전협회(NHTSA, National Highway Traffic Safety Administration)에서는 2008년 9월부터 자국 수입 경자동차에 블랙박스 장착을 권장하는 권고안을[2], 미국자동차기술자협회(SAE, Society of Automotive Engineers)와 국제전기전자기술자협회(IEEE, Institute of Electrical and Electronics Engineers)에서는 자동차 블랙박스 표준안을 발표하였다[3,4]. 또한 유럽의 국제유럽경제위원회(UNECE, United Nations Economic Commission for Europe)는 강제조항인 상호 수출입 규격에서 차량용 블랙박스의 표준화 제정을 진행하고 있고, 유럽 연합(EU, European Union)에서는 2009년부터 EU 가입 국내 모든 차량 의무 장착 법안을 확정했다. 일본은 2008년부터 일부 차종에 따라 의무 장착을 시작해 택시, 버스, 트럭 등에 이미 장착을 시작하였고 중국은 2008년에 모든 차량의 디지털 주행기록장치 장착을 의무화했다[5].

하지만 아직까지 블랙박스는 수집된 데이터를 저장하는 역할만 할 뿐이기 때문에 차량 소유자나 제삼자에 의한 의도적인 데이터 위, 변조로부터 안전하지 않다. 즉, 현재 상용화되고 있는 블랙박스는 데이터의 위, 변조를 확인하기 위한 무결성을 보장할 수가 없으며 이러한 데이터는 사고의 원인규명을 위한 법적증거자료로서의 효력이 전혀 없게 된다. 또한 차량용 블랙박스에 저장되는 운행정보 및 동영상 데이터는 그 특성상 실시간으로 기록된다. 하지만 데이터의 무결성을 보장하기 위한 기법이 실시간으로 생성되는 데이터를 빠르게 처리할 수 없을 경우에는 큰 문제점이 생긴다. 예를 들어, 사고가 발생했을 때에 물리적인 충격으로 인해 블랙박스 시스템이 멈추는 상황이 발생될 수가 있다. 하지만 무결성을 보장하는 기법의 처리속도가 느리다면 사고시점 이전의 데이터에 대해서만 무결성을 보장하게 될 것이다. 따라서 빠른 처리속도로 실시간 무결성을 보장할 수 있는 기법이 요구된다. 하지만 블랙박스는 자신이 생성한 데이터를 자신이 저장하는 구조이기 때문에 무결성을 보장하는 것이 쉽지만은 않다. 또한 아직까지 이러한 실시간 무결성을 보장하는 기법이 제안되지 않았다. 따라서 본 논문에서는 차량용 블랙박스에서 생성되는 데이터의 실시간 무결성을 보장하는 기법을 제안하고자 한다. 논문의 구성은 다음과 같다. 2장에서는 이와 관련된 기술 및 연구들을 알아보고, 3장에서는 이러한 시스템 환경에서의 가정사항을 정의하고, 4장에서는 우리가 제안하는 실시간 무결성 보장 기법을 자세히 다룬다. 5장에서는 제안한 기법을 분석하고, 6장에서는 우리가 제안한 기법을 실제로 구현하여 실험한 결과를 알아보고, 7장에서는 결론을 맺고 향후 연구 방향에 대해 논한다.

II. 관련기술 및 연구들

이번 장에서는 본 논문에서 제안하는 기법과 관련된 기술들에 대해서 알아보고, 또한 현재까지 차량용 블랙박스와 관련된 연구들에는 어떤 것들이 있는지 알아본다.

2.1 블랙박스

현재 상용화되어 차량에 탑재되고 있는 차량용 블랙박스는 비행기의 블랙박스와는 달리 주로 디지털영상저장장치(DVR, Digital Video Recorder)로서의 기능을 하고 있다. 말 그대로 주행 당시의 상황을 영상으로 기록하는 장치이다. 비행기 블랙박스처럼 여러 기초 데이터 등을 저장 한 뒤 사후 데이터를 분석하는 방법은 아니지만 사고의 원인을 분석하고자하는 목적은 동일하다. 또한 여러 가지 센서와 GPS(Global Positioning System)와의 연계를 통해 차량의 속도, 방향, 브레이크 작동, 안전띠 착용유무, 조향각 등 다양한 정보를 저장한다. 또한 사고의 원인규명 목적 외에도 운전자 스스로 안전한 운전을 하기위한 목적으로도 쓰이고 자신의 운행정보를 기록하는 목적으로도 쓰인다.

차량용 블랙박스는 1990년대 GM, 포드 등에서 자동차의 에어백 ECU(Electronic Control Unit)에 EDR(Event Data Recorder)을 장착하면서 시작되었다[6]. 그 후에는 충격당시 영상만을 저장하는 1세대 제품이 나오게 되었고, 기술의 발전으로 인해 주행 도중의 영상, 각종 운행 정보 등 사고분석에 필요한 데이터들을 저장하는 현재의 2세대 제품들이 나오게 되었다. 현재 2세대 제품들은 다양한 기술들과 결합되어 더욱 발전해 나가고 있다.

2.2 실시간 무결성

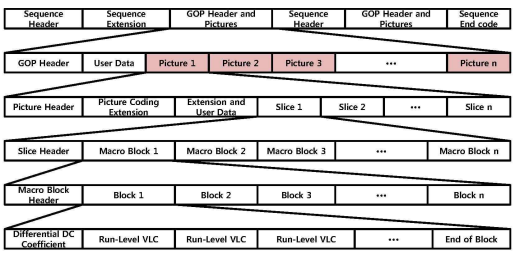
위에서 블랙박스에 대해 알아보았다. 하지만 비행기 블랙박스와 차량용 블랙박스 모두 저장된 데이터의 위, 변조는 고려하지 않고 있다. 다양한 데이터의 수집을 통해 사고의 원인 분석을 한다고 하지만 수집된 데이터가 악의적인 조작에 의해 위, 변조 되었다면 그러한 데이터들을 통해 나온 분석 결과는 신뢰를 받지 못할 것이다. 비행기 블랙박스 같은 경우는 다루는 사람이 한정되어 있고 정부기관과 같은 신뢰된 기관에서 수거와 분석을 담당하기 때문에 대부분 신뢰를 하였지만 차량용 블랙박스와 같은 경우는 일반인 누구나 장착을 할 수가 있고 저장장치에 손을 댈 수가 있기 때문에 데이터의 위, 변조는 심각한 이슈로 떠오를 수밖에 없다. 차량용 블랙박스가 시장에 나온 지 얼마 안 되었고 기술적인 부분에 집중되어 발전을 하고 있기 때문에 이러한 부분이 아직 고려되고 있지 않고 있다.

이러한 데이터의 위, 변조에 관한 문제를 해결하기 위해선 데이터의 무결성을 입증할 수 있는 기법이 고려되어야 한다. 무결성(Integrity)은 말 그대로 데이터가 저장된 그대로 보존되어 변하지 않았다는 것을 나타내는 성질을 말한다. 즉, 데이터를 저장하고 난 뒤에 데이터가 위, 변조 되었는지 알 수 있어야 함을 말한다. 또한 블랙박스에 저장되는 데이터의 경우 실시간으로 계속해서 생성되기 때문에 무결성도 실시간으로 보장되어야 한다. 하지만 앞서도 말했듯이 지금의 블랙박스들은 이러한 점을 고려하지 않고 있다.

2.3 MPEG

차량용 블랙박스에서 현재 가장 많이 다루어지는 데이터는 동영상 데이터이다. 차량이 주행하는 순간부터 주행정보를 영상으로 담기 때문에 따로 분석을 필요로 하지도 않고 이해하기도 쉽다. 또한 동영상은 구조적으로 여러 장의 이미지가 연결되어 구성되어 있기 때문에 이미지로의 변환도 가능하다. 이러한 특성으로 인해 저장된 동영상 데이터에 오류가 생겨 올바르게 재생이 되지 않더라도 우리는 정지영상에 대한 이미지를 추출하여 확인할 수 있다. 그렇기 때문에 정지영상 이미지에 대한 무결성 보장도 필요하게 된다.

보통 동영상 데이터는 촬영과 동시에 실시간으로 압축되어져서 데이터 공간에 저장이 되는데 압축방식은 크게 DV 계열과 MPEG계열로 나뉜다. 그중 우리는 보다 널리 쓰이는 MPEG 구조에 대해 알아보고자 한다. [그림 1]은 MPEG의 비디오 계층의 구조이다[7]. 이 구조에서 색깔이 칠해진 Picture에 해당하는 부분이 하나의 프레임이라고 보면 된다. 프레임은 I, B, P 3가지 종류가 존재하며 I 프레임을 보통 키 프레임이라고도 부른다. 이 I 프레임은 하나의 온전한 이미지 정보를 담고 있어서 이 정보를 가지고 이미지 파일로 변환이 가능하다. 따라서 이러한 특성을 이용하여 정지영상 이미지에 대해 무결성을 검증하는 것이 가능하다.



[그림 1] MPEG 비디오 계층 구조

2.4 관련 연구들

이번에는 차량용 블랙박스와 관련된 현재까지의 연구들을 살펴보고자 한다. 차량용 블랙박스와 관련된 연구도 몇 개 없지만 무결성과 관련된 연구는 거의 없는 실정이다.

2.4.1 Vehicle Black Box System

Abdallah Kassem 등이 제안한 Vehicle Black Box System[8]은 어느 차량에나 장착 될 수 있는 차량용 블랙박스 시스템의 프로토타입을 제안하고 있다. 블랙박스의 목적은 차량을 안전하게 하고, 사고의 발생을 줄이고 보험회사들의 차량사고 조사를 돕기 위함이다. 제안하는 블랙박스 시스템은 여러 개의 센서(Speed Sensor, Water Sensor, Lights Sensor 등)들과 데이터 처리장치로 구성된다. 블랙박스의 데이터들은 EEPROM에 저장되고 후에 분석을 위해서 특정 컴퓨터로 전송된다. 하지만 EEPROM도 롬 라이터를 이용해서 데이터를 쓰고 지우는 것이 가능하기 때문에 단순 저장만으로는 무결성을 보장하지 못한다.

2.4.2 자동차의 블랙박스를 이용한 실시간 포렌식 자료 생성 연구

박대우 등이 제안한 자동차의 블랙박스를 이용한 실시간 포렌식 자료 생성 연구(A Study of Using the Car's Black Box to generate Real-time Forensic Data)[9]에서 차량용 블랙박스는 고유한 IPv6를 부여받고, 시동 시에 운전자의 인증을 받아 작동하며, 블랙박스에 기록된 자료는 암호화되어 도로변의 기지국 센서 네트워크를 통해 교통운영관리센터의 교통기록 데이터베이스에 저장되는 구조를 가지고 있다. 이 논문에서는 블랙박스의 IP주소와 교통기록 데이터베이스에 저장된 IP의 일치함을 통해 데이터의 무결성을 검증한다고 언급하고 있다. 하지만 자료를 암호화할 때 쓰이는 키의 생성이나 분배, 암호 알고리즘에 대한 언급이 전혀 없을뿐더러 블랙박스의 송신 IP와 데이터베이스의 수신 IP 일치로 데이터의 무결성을 검증하는 것이 말이 되지 않는다. IP 주소는 얼마든지 위조가 가능하기 때문에 IP 주소가 일치한다고 해서 데이터의 무결성을 보장할 수는 없다. 또한, 모든 자료가 한 데이터베이스에 저장되기 때문에 저장 공간에 많은 오버헤드가 발생하게 된다.

2.4.3 모바일 장치를 이용한 자동차 영상블랙박스 설계

김진일 등이 제안한 모바일 장치를 이용한 자동차 영상블랙박스 설계(A Design of Car Video Black box on Mobile Device)[10]는 차량용 블랙박스를 GPS 수신 장치가 내장된 모바일 스마트폰으로 사용하는 것을 제안하고 있다. 기존의 차량용 블랙박스는 추가적인 별도의 장치(카메라, DVR 등)들을 구비해야 하기 때문에 추가적인 비용이 발생하고 사고가 일어났을 때 자동으로 사고 접수 및 구조 신고를 하지 못한다고 지적하고 있다. 하지만 GPS가 수신 장치가 내장된 모바일 스마트폰 또한 사용자가 가지고 있지 않다면 추가적인 비용이 드는 것은 마찬가지이고, 자동으로 사고 접수 및 구조 신고를 하는 것은 어플리케이션의 문제라고 생각한다.

이 논문에서 블랙박스에 저장되는 데이터는 외장 메모리에 별도로 저장된다. 하지만 이 논문 역시 데이터의 무결성을 입증하기 위한 과정은 없기 때문에 이 블랙박스에 저장된 데이터는 차후 법적증거자료로서의 기능을 할 수가 없게 된다.

블랙박스에서의 무결성 보장에 관련된 연구는 현재까지는 거의 없다. 대부분 블랙박스와 관련된 연구들은 수집된 데이터를 통한 사고의 재구성에 초점이 맞추어져 있다. 이러한 연구들은 블랙박스에 저장된 데이터가 위, 변조 되는 것을 고려하지 않기 때문에 차후 법적증거자료로서의 기능을 할 수가 없다.

III. 시스템 가정

이번 장에서는 차량용 블랙박스 시스템 환경의 가정을 정의한다. 차량용 블랙박스 시스템을 위협하는 공격자 모델을 정의하고, 이러한 공격자가 행할 수 있는 공격의 유형, 이러한 공격자로부터 안전하기 위해 필요한 보안 요구사항을 정의한다.

3.1 공격자 모델

블랙박스 시스템을 위협하는 공격자의 범위와 능력에 대해서 다음과 같이 정의한다.

1) 블랙박스 시스템을 제외한 모든 개체는 공격자가 될 수 있다. 차량을 소유한 차주를 비롯해서 블랙박스 시스템에 접근할 수 있는 모든 사람 및 시스템이 공격자가 될 수 있다.
2) 공격자는 블랙박스 시스템 데이터 저장 공간의

데이터를 위, 변조할 수 있다. 공격자는 저장되어 있는 데이터를 통째로 바꿔치기 하거나, 중간 데이터 삽입 등의 행위를 할 수 있다.

3) 공격자는 블랙박스 시스템 내부의 메모리를 읽어 데이터를 찾아낼 수 있다. 공격자는 메모리의 값을 읽어 시스템 내부에서 발생하는 연산 값들을 알아낼 수 있다.

3.2 공격 유형

3.2.1 데이터 재생

공격자가 블랙박스 시스템 내부에 저장된 입력 데이터 및 무결성 검증 데이터를 모두 삭제한 뒤 새로운 데이터를 다시 만들어 저장을 하는 공격이다.

3.2.2 데이터 삽입

공격자가 위조한 데이터를 블랙박스 시스템 내부에 저장된 입력 데이터 및 무결성 검증 데이터의 특정 위치에 삽입을 하는 공격이다.

3.2.3 데이터 교체

데이터 삽입과는 달리 공격자가 위조한 데이터를 블랙박스 시스템 내부에 저장된 입력 데이터 및 무결성 검증 데이터의 특정 위치에 있는 데이터와 교체하는 공격이다.

3.2.4 데이터 순서 교체

이 공격은 공격자가 위조된 데이터를 생성하지 않고 블랙박스 시스템 내부에 저장된 입력 데이터 및 무결성 검증 데이터의 순서를 교체함으로서 데이터를 변조하는 공격이다.

3.2.5 중간 데이터 삭제

이 공격은 공격자가 블랙박스 시스템 내부에 저장된 입력 데이터 및 무결성 검증 데이터의 중간 부분을 삭제하는 공격이다.

3.2.6 후위 데이터 삭제

이 공격은 공격자가 블랙박스 시스템 내부에 저장된 입력 데이터 및 무결성 검증 데이터의 특정 시점 이후부터 끝까지 전부 삭제하는 공격이다.

3.2.7 데이터 부인

공격자가 블랙박스 내에서 생성된 데이터가 자신의 차량에서 생성된 데이터가 아니라고 부인하는 공격이다.

3.3 보안 요구사항

3.3.1 무결성

블랙박스 시스템은 저장된 데이터가 위, 변조되었음을 알아낼 수 있어야 한다. 공격자에 의해 저장된 데이터가 바뀌거나 변조 되었을지라도 원본 데이터를 통해 계산된 의미 있는 값(예: 해시 값, 전자서명 값)과의 비교를 통해 해당 데이터가 위, 변조되었음을 알 수 있어야 한다.

3.3.2 부인방지

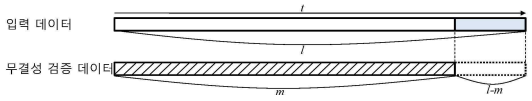
블랙박스 시스템은 생성된 데이터가 유일한 사용자로부터 생성되었음을 입증할 수 있어야 한다. 공격자가 불이익을 당하는 것을 피하기 위해 자신의 블랙박스 시스템에서 생성된 데이터를 자신의 것이 아니라고 부인할 수 있는 경우를 방지하기 위해 블랙박스 시스템에서 생성되는 모든 데이터는 유일한 사용자로부터 생성되었음을 나타낼 수 있어야 한다.

3.3.3 오류복구

블랙박스 시스템은 저장된 데이터에 오류가 발생하더라도 복구할 수 있어야 한다. 데이터가 저장된 뒤에 외부의 충격(예: 전기적 신호 오류, 전파장애 등)으로 인해 데이터에 오류가 발생하더라도 이를 알아내고 무결성 검증에 차질이 없도록 일정부분 복구할 수 있어야 한다.

3.3.4 빠른 계산속도

앞서 언급했듯이, 차량에서 실시간으로 생성되는 데이터에 대해 무결성을 보장하는 기법이 빠른 계산속도를 가지지 못한다면 문제가 될 수 있다. 생성되는 무결성 검증 데이터의 크기가 해당 입력 데이터의 크



[그림 2] 데이터 생성과정

기와 동일하다고 가정하자. [그림 2]와 같이 시간 t동안 입력 데이터는 l만큼 생성되었는데 무결성 검증 데이터는 l보다 작은 길이인 m만큼 생성된다면, 즉, 입력 데이터의 m에 해당하는 만큼만 무결성 검증 데이터가 생성되었다면 $(l-m)$의 길이에 해당하는 입력 데이터는 무결성을 검증받지 못하게 된다. 따라서 블랙박스 시스템이 사고에 의해 동작을 멈추더라도 최대한 많은 입력 데이터에 대한 무결성을 보장하기 위해 무결성 보장 기법은 빠른 계산 속도를 내야한다.

IV. 제안하는 실시간 무결성 보장 기법

이번 장에서는 우리가 제안하는 차량용 블랙박스 시스템을 위한 실시간 무결성 보장 기법에 대해 말한다. 이 기법에는 몇 가지 가정이 존재하며 우리는 이 가정에 기반을 둔 시스템 모델과 앞서 정의한 보안 요구 사항을 만족하는 실시간 무결성 보장 기법을 제안한다. 또한 본 논문에서 쓰이는 표기법은 [표 1]과 같다.

[표 1] 표기법

표기	의미
TTP	제3 신뢰기관
sk	서명키
pk	서명 검증키
$Sign_k()$	키 k를 사용한 전자서명
$Verify_k()$	키 k를 사용한 서명검증
i	데이터의 인덱스
IAD	초기 인증 데이터
BD	블록 데이터
BD_i	i번째 블록 데이터
ts_i	i번째 타임스탬프
IVD	무결성 검증 데이터
IVD_i	i번째 무결성 검증 데이터
$h()$	해시 함수
h_i	i번째 제1 해시값
h'_i	i번째 제2 해시값

4.1 가정

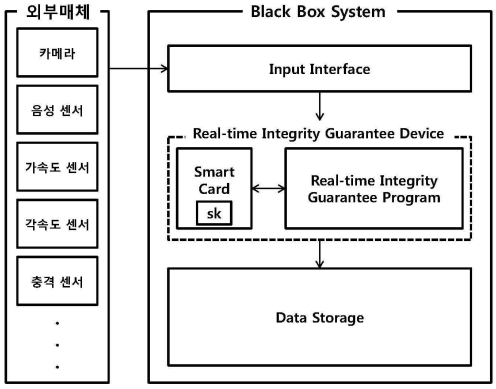
우리가 제안하는 기법은 다음과 같은 가정 사항을 두고 이루어진다.

1) 사용자는 블랙박스 구입 시에 혹은 차량 등록 시에 제3 신뢰기관(TTP, Third Trust Party)(예: 정부기관)으로부터 블랙박스 시스템에 들어가는 스마트카드를 발급받는다.
2) TTP로부터 스마트카드를 발급 받을 때에 블랙박스마다 유일한 서명키 sk를 발급받으며 이는 스마트카드에 저장된다.
3) 스마트카드는 자체적으로 전자서명이 가능하며, 임의의 메시지 M에 대해 전자서명값 $Sign_{sk}(M)$을 출력한다.
4) 스마트카드에 저장되는 sk와 전자서명값은 TTP만이 불러올 수 있다.
5) 스마트카드를 통해 출력 되는 전자서명값은 실시간 무결성 보장 프로그램 내에서만 얻을 수 있으며 사용자가 임의로 전자서명값을 얻어낼 수는 없다.
6) 서명 검증을 하는 공개키 pk는 TTP만이 소유하기 때문에 무결성 검증은 TTP만이 할 수 있다.

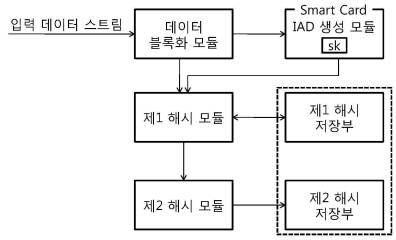
4.2 시스템 모델

본 논문에서 제안하는 시스템 모델은 [그림 3]과 같다.

1) 블랙박스 시스템은 입력 인터페이스, 실시간 무결성 보장 장치, 데이터 저장 공간으로 이루어져 있다.
2) 블랙박스 시스템의 입력은 외부매체(예: 카메라, 센서 등)로부터 실시간으로 수집되는 영상, 음성 등 다양한 실시간 데이터 스트림이다.



[그림 3] 시스템 모델

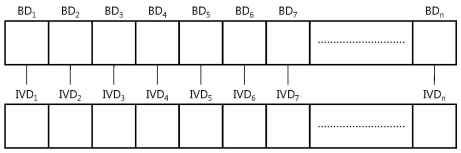


[그림 4] 실시간 무결성 보장 장치

3) 블랙박스 시스템은 외부매체와 유선으로 연결되어 있다.
4) 입력 데이터는 입력 인터페이스를 거친 후 실시간 무결성 보장 장치의 입력으로 쓰이며, 데이터 저장 공간에 저장된다.
5) 실시간 무결성 보장 장치는 스마트카드와 실시간 무결성 보장 프로그램으로 구성되며 [그림 4]와 같다.
6) 실시간 무결성 보장 장치의 출력값은 입력 데이터값과 별도로 데이터 저장 공간에 저장된다.
7) 데이터 저장 공간에 더 이상 데이터를 저장할 수 있는 공간이 없으면 처음부터 다시 덮어 쓴다.
8) 스마트카드는 초기 인증 데이터(Initial Authentication Data)인 *IAD*값을 생성하는 *IAD* 생성 모듈을 포함하고 있다.
9) 제2 해시모듈의 출력은 제2 해시 저장부에 저장되며 이는 실시간 무결성 보장 장치의 출력값이 된다.
10) 제1 해시 저장부와 제2 해시 저장부는 논리적으로는 다르지만 물리적으로는 하나로 구현될 수 있다.

4.3 실시간 무결성 보장 기법

블랙박스의 입력 데이터 스트림은 [그림 5]와 같이 일정한 크기의 블록으로 나누어져서 처리가 되며 BD_i는 i번째 블록 데이터(Block Data)를 의미한다. 또한



[그림 5] 입력 데이터 스트림

IVD_i는 BD_i를 처리하여 생성된 무결성 검증 데이터(Integrity Verification Data)를 의미한다. IVD_i의 개수는 BD_i의 개수와 동일하며, IVD_i의 크기는 BD_i의 크기에 독립적이다.

4.3.1 기본적인 실시간 무결성 보장 기법

기본적인 실시간 무결성 보장 기법은 [표 2]와 같이 매우 간단하다. 방법 1은 각 블록 데이터 BD_i마다 해시를 방법 2는 전자서명을 취하는 것이다. 방법 1은 무결성과 부인방지 둘 다 보장하지 못한다. 무결성을 보장하지 못하는 이유는 데이터 재생공격과 교체공격, 후위 데이터 삭제공격이 가능하기 때문이다. 방법 2는 부인방지는 보장하지만 무결성은 보장하지 못한다. 무결성을 보장하지 못하는 이유는 후위 데이터 삭제공격이 가능하기 때문이다. 따라서 우리는 이러한 공격들에 대해 안전하면서도 효율적인 실시간 무결성 보장 기법을 아래와 같이 제안한다.

4.3.2 제안하는 실시간 무결성 보장 기법

우리가 제안하는 실시간 무결성 보장 기법은 [표 3]과 같다. 알고리즘의 구체적인 이해를 위해 실시간 무결성 보장 장치를 나타낸 [그림 4]를 참조하도록 한다.

1) 데이터 블록화 모듈은 입력 데이터 스트림을 미리 결정된 일정한 크기의 블록 데이터로 분할하여 *IAD* 생성 모듈 및 제1 해시 모듈의 입력으로 전달한다.
2) 최초의 블록 데이터값인 BD_1과 타임스탬프 ts_1,

[표 2] 기본적인 실시간 무결성 보장 기법

방법 1	방법 2
$IVD_1 = h(BD_1 \| ts_1 \| 1) \| ts_1$	$IVD_1 = Sign_{sk}(BD_1 \| ts_1 \| 1) \| ts_1$
$IVD_2 = h(BD_2 \| ts_2 \| 2) \| ts_2$	$IVD_2 = Sign_{sk}(BD_2 \| ts_2 \| 2) \| ts_2$
$IVD_3 = h(BD_3 \| ts_3 \| 3) \| ts_3$	$IVD_3 = Sign_{sk}(BD_3 \| ts_3 \| 3) \| ts_3$
⋮	⋮
$IVD_n = h(BD_n \| ts_n \| n) \| ts_n$	$IVD_n = Sign_{sk}(BD_n \| ts_n \| n) \| ts_n$

[표 3] 제안하는 실시간 무결성 보장 기법

입력 : $BD_i(i \in 1,\ldots,n)$

1: for $i \leftarrow 1$ to n
2: if $i = 1$ then
3: $IAD \leftarrow Sign_{sk}(BD_i \| ts_i \| i)$
4: $h_i \leftarrow IAD$
5: else
6: $h_i \leftarrow h(h_{i-1} \| BD_i \| ts_i \| i)$
7: delete h_{i-1}
8: $IVD_{i-1} \leftarrow h(h'_{i-1} \| h_i) \| ts_{i-1}$
9: end if
10: $h'_i \leftarrow h(h_i)$
11: $IVD_i \leftarrow h'_i \| ts_i$
12: end for

인덱스 1의 연결값은 IAD 생성 모듈의 입력으로 들어가 서명키 sk로 전자서명이 되어 출력된다. 이 전자서명값이 초기인증데이터(Initial Authentication Data) IAD값이 된다.

3) IAD값은 스마트카드에 저장된다.
4) 출력된 IAD값은 제1 해시 모듈의 입력으로 들어가지만 해시되지는 않고 제1 해시 저장부에 제1 해시값 h_1으로 저장되고 제2 해시 모듈의 입력으로 들어간다.
5) 제2 해시 모듈은 입력된 제1 해시값 h_1을 해시하여 제2 해시값 h'_1을 생성한 뒤 제2 해시 저장부에 저장하고, h'_1과 ts_1은 블랙박스 시스템의 데이터 저장 공간에 블록 데이터 BD_1에 해당하는 무결성 검증 데이터 IVD_1으로 저장된다. 즉, $h(IAD) \| ts_1$가 IVD_1에 저장된다.
6) 두 번째 블록 데이터값인 BD_2는 제1 해시 모듈의 입력으로 들어가 직전 블록데이터 BD_1의 제1 해시값 h_1과 타임스탬프 ts_2, 인덱스 2와 연결되어 해시된 후 제1 해시 저장부에 제1 해시값 $h_2(=h(h_1 \| BD_2 \| ts_2 \| 2))$로 저장된다.
7) 직전 블록의 제1 해시값 h_1은 h_2를 생성하자마자 제1 해시 저장부에서 삭제된다. 즉, IAD값이 삭제된다.
8) 직전 블록의 무결성 검증 데이터 IVD_1의 앞부분인 h'_1을 h_2와 연결하여 해시한 뒤 다시 IVD_1의 앞부분에 저장한다. 즉 $h(h'_1 \| h_2) \| ts_1$이 IVD_1에 저장된다.
9) h_2는 제2 해시모듈의 입력으로 들어가 해시된 후 제2 해시저장부에 h'_2으로 저장된 후 ts_2와 함께 데이터 저장 공간에 IVD_2로 저장된다. 즉, $h(h_2) \| ts_2$값이 IVD_2에 저장된다.
10) (6)~(9)와 같은 방법으로 이후 BD_i의 무결성 검증 데이터 IVD_i를 계속 생성 및 저장해 나간다. 생성결과는 [표 4]와 같다.

차후에 TTP가 생성된 데이터들의 무결성을 검증하는 알고리즘은 [표 5]와 같다. TTP는 스마트카드에서 IAD값을 불러온 뒤 sk에 대응하는 pk로 서명검증을 하고, 서명검증이 통과되면 IVD_i 생성과 유사한 방법으로 IVD'_i값을 생성한 후 IVD_i값과 IVD'_i값이 일치하는지 비교하여 무결성 검증을 수행한다.

4.3.3 부가적인 활용

부가적으로 입력 데이터가 저장된 후 특정 BD_i에 오류가 발생한 경우라도 데이터가 동영상일 경우에는 부분적인 오류복구를 지원하기 위해 동영상의 각 I 프레임(Intra frame)을 BD_i로 삼아 위의 기법을 적용한다(2.3 참조). 이렇게 하여 나온 출력값은 기존의 IVD_i와 별도로 저장을 한다. 각 I 프레임은 추후 디코딩을 하여 이미지파일로 변환할 수 있기 때문에 동영상에 오류가 나더라도 매 초의 정지영상 이미지에 대해 무결성을 검증할 수 있게 된다. 매 초의 정지영상 이미지라 할지라도 사람의 눈으로 보았을 때 움직임의 차이가 미미하므로 사고의 원인을 파악하는 자료로서는 충

[표 4] IVD_i 생성결과

$h_1 = IAD = Sign_{sk}(BD_1 \| ts_1 \| 1)$	$h'_1 = h(h_1)$	$IVD_1 = h'_1 \| ts_1 \searrow$	
$h_2 = h(h_1 \| BD_2 \| ts_2 \| 2)$	$h'_2 = h(h_2)$	$IVD_2 = h'_2 \| ts_2 \searrow$	$IVD_1 = h(h'_1 \| h_2) \| ts_1$
$h_3 = h(h_2 \| BD_3 \| ts_3 \| 3)$	$h'_3 = h(h_3)$	$IVD_3 = h'_3 \| ts_3 \searrow$	$IVD_2 = h(h'_2 \| h_3) \| ts_2$
$h_4 = h(h_3 \| BD_4 \| ts_4 \| 4)$	$h'_4 = h(h_4)$	$IVD_4 = h'_4 \| ts_4 \searrow$	$IVD_3 = h(h'_3 \| h_4) \| ts_3$
⋮	⋮	⋮	$IVD_4 = h(h'_4 \| h_5) \| ts_4$
⋮	⋮	⋮	⋮
$h_n = h(h_{n-1} \| BD_n \| ts_n \| n)$	$h'_n = h(h_n)$	$IVD_n = h'_n \| ts_n$	$IVD_{n-1} = h(h'_{n-1} \| h_n) \| ts_{n-1}$

[표 5] 무결성 검증 알고리즘

입력 : $BD_i(i \in 1,...,n)$, $IVD_i(i \in 1,...,n)$, IAD
출력 : True or False

```
1:  Verify_pk(IAD)
2:  if 서명검증에 성공했는가? then
3:    for i←1 to n
4:      if i = 1 then
5:        h_i ← IAD
6:      else
7:        h_i ← h(h_{i-1} ‖ BD_i ‖ ts_i ‖ i)
8:        IVD'_{i-1} ← h(h'_{i-1} ‖ h_i) ‖ ts_{i-1}
9:      end if
10:     h'_i ← h(h_i)
11:     IVD'_i ← h'_i ‖ ts_i
12:     if (i > 1) and (IVD_{i-1} ≠ IVD'_{i-1}) then
13:       return False
14:     end if
15:     if (i = n) and (IVD_i ≠ IVD'_i) then
16:       return False
17:     end if
18:   end for
19:   return True
20: else
21:   return False
22: end if
```

분하다. 따라서 이러한 우리가 제안한 기법의 이러한 활용도 필요하게 된다.

V. 분 석

이번 장에서는 우리가 제안한 실시간 무결성 보장 기법이 앞서 정의한 보안요구사항을 만족하는지의 여부와 이 기법의 계산량을 분석해본다. 본 논문에서 제안한 실시간 무결성 보장 기법은 스마트카드에 저장된 sk와 IAD값이 외부에 노출되지 않는다는 것에 기반하여 무결성과 부인방지를 제공하며, 오류복구 또한 지원한다.

5.1 무결성 보장

우리가 제안한 기법은 다음과 같은 원리로 인해 공격자가 정당한 무결성 검증 데이터를 생성하지 못하기 때문에 무결성을 보장한다.

1) IVD_1은 sk를 이용해 만들어진 IAD를 해시한값이고 공격자는 스마트카드에 저장되어 있는 sk와 IAD값을 알 수 없기 때문에 IVD_1을 생성할 수가 없다. IAD값은 오직 TTP만이 불러올 수 있다.
2) IAD, 즉 h_1을 생성할 수 없기 때문에 h_2를 생성할 수 없고 h'_2 역시 생성할 수 없다. 이것은 IVD_2를 생성할 수 없음을 의미한다.
3) h_2를 생성할 수 없기 때문에 h_3를 생성할 수 없고 h'_3 역시 생성할 수 없다. 이것은 IVD_3를 생성할 수 없음을 의미한다.
4) 같은 원리로 이후의 IVD_i도 생성할 수가 없게 된다.

정당한 IVD_i를 생성할 수 없기 때문에 데이터 재생공격을 할 수가 없다. 또한 인덱스 값으로 인해 데이터 삽입공격과 순서 교체공격, 중간 데이터 삭제공격을 할 수 없고, 해시 함수의 특성으로 인해 h'_i의 프리이미지(Pre-image)값인 h_i값을 알 수 없기 때문에 데이터 교체공격도 불가능하다.

IVD_1 이후부터 해시를 두 번 씩 취한 이유는 [표 6]과 같이 h_2를 IVD_2에 저장할 경우 IVD_2까지는 공격자가 생성할 수 없지만 IVD_3부터는 h_{i-1}값이 노출 되고 공격자가 정당한 IVD_i값을 생성할 수 있게 되어 위조가 가능해 안전하지 않기 때문이다.

IVD_2 이후부터 IVD_{i-1}의 값에 h_i값을 넣어 해시를 취하여 값을 다시 저장하는 이유는 후위 데이터 삭제공격을 막기 위함이다. 맨 마지막 IVD_i값은 h_{i+1}값이 존재하지 않기 때문에 무결성 검증을 할 때에 마지막 IVD_i값이 삭제되었는지 안 되었는지 알 수가 있게 된다.

[표 6] 해시를 한 번 취 했을 때의 IVD_i의 위조

$h_1 = IAD = Sign_{sk}(BD_1 \| ts_1 \| 1)$	$IVD_1 = h(h_1) \| ts_1$ ↘	
$h_2 = h(h_1 \| BD_2 \| ts_2 \| 2)$	$IVD_2 = h_2 \| ts_2$ ↘	$IVD_1 = h(h(h_1) \| h_2) \| ts_1$
$h_3 = h(h_2 \| BD_3 \| ts_3 \| 3)$	$IVD_3 = h_3 \| ts_3$ ↘	$IVD_2 = h(h_2 \| h_3) \| ts_2$
$h_4 = h(h_3 \| BD_4 \| ts_4 \| 4)$	$IVD_4 = h_4 \| ts_4$ ↘	$IVD_3 = h(h_3 \| h_4) \| ts_3$
⋮	⋮	⋮
$h_n = h(h_{n-1} \| BD_n \| ts_n \| n)$	$IVD_n = h_n \| ts_n$	$IVD_{n-1} = h(h_{n-1} \| h_n) \| ts_{n-1}$

또한 h_2 이후의 h_i값을 생성할 때 BD_i와 IAD를 연결하면 해시를 한 번만 취하고도 무결성을 보장할 수가 있지만 메모리상에 계속 IAD값을 노출시킴으로써 공격자로 하여금 메모리 해킹을 통해 IVD_i값을 위조할 수 있는 가능성을 열어두게 된다.

5.2 부인방지 지원

각 블랙박스마다 유일한 서명키 sk를 소유하게 되므로 부인방지를 제공한다. 가정에도 나와 있듯이 서명키는 블랙박스마다 유일하게 하나씩 지급된다. 또한 서명키는 스마트카드에 저장되어 외부에 노출되지 않기 때문에 안전하다.

5.3 오류복구 지원

5.3.1 BD에 발생한 오류

입력 데이터가 동영상인 경우 4.3.3에서 서술한 방법으로 부분적인 오류 복구를 지원한다. 하지만 일반적인 데이터인 경우 BD_i가 저장된 후에 오류가 발생한다면 무결성 검증에서 실패를 하게 되고 오류 복구는 지원하지 못한다.

5.3.2 IVD에 발생한 오류

IVD_i값에 오류가 발생해 BD_i의 무결성 검증이 되지 않을지라도 IAD값을 알 수 있기 때문에 오류가 발생하지 않은 나머지 모든 블록의 무결성 검증이 가능하다. IAD만 알아도 h_i를 구할 수 있기 때문에 h'_i 또한 구할 수 있게 된다. 따라서 오류복구를 지원하게 된다.

5.4 계산량

각 암호연산에 대한 표기법은 [표 7]과 같다.

1) 블랙박스가 n개의 블록 데이터에 대한 무결성 검증 데이터를 생성할 때의 계산량

[표 7] 암호연산 표기법

표기	의미
$Hash^n$	n번의 해시 연산
$Sign^n$	n번의 전자서명 연산
$Verify^n$	n번의 서명검증 연산

[표 8] 기본적인 기법과의 비교

구분	방법1	방법2	제안한 기법
데이터 재생 방지	×	●	●
데이터 삽입 방지	●	●	●
데이터 교체 방지	×	●	●
데이터 순서 교체 방지	●	●	●
데이터 부인 방지	×	●	●
중간 데이터 삭제 방지	●	●	●
후위 데이터 삭제 방지	×	×	●
BD 오류 복구	●	●	▲
IVD 오류 복구	●	●	●
블랙박스의 계산량	$Hash^n$	$Sign^n$	$Sign^1 + Hash^{3n-2}$
TTP의 계산량	$Hash^n$	$Verify^n$	$Verify^1 + Hash^{3n-2}$
스마트카드 이용 횟수	0	n	1

$$Sign^1 + Hash^1 + (n-1) \cdot Hash^3 = Sign^1 + Hash^{3n-2} \quad (1)$$

2) TTP가 n개의 무결성 검증 데이터에 대한 무결성 검증을 할 때의 계산량

$$Verify^1 + Hash^1 + (n-1) \cdot Hash^3 = Verify^1 + Hash^{3n-2} \quad (2)$$

n개의 데이터에 대해 위와 같은 정도의 계산량이 요구되며 이는 시스템에 전혀 무리를 주지 않는다. 또한 무결성 검증 데이터를 생성할 때와 검증할 때의 계산량의 차이는 서명생성과 서명검증의 차이만 있을 뿐 동일하다. 그리고 첫 블록을 서명할 때만 스마트카드의 도움을 받으므로 계산 속도에 미치는 영향은 매우 작다.

5.5 기본적인 기법과의 비교

우리가 제안한 기법을 앞서 제안한 기본적인 실시간 무결성 보장 기법과 비교한 결과는 [표 8]과 같다. ×는 지원하지 않음을 의미하고, ●는 지원함을 의미하며, ▲는 부분적인 지원을 의미한다. 이 결과를 통해 우리가 제안한 기법이 기본적인 방법들보다 안전하면서도 매우 효율적임을 알 수 있다.

VI. 실 험

이번 장에서는 우리가 제안한 기법을 실제로 구현한 뒤 여러 환경과 조건에서 실험을 한 결과를 제시한

[표 9] 시스템 오버헤드 측정

단위 : 초

시스템 환경	CPU	Pentium D 3.0G	Pentium D 2.8G	Core2Duo 2.0G	Core2Quad 2.4G
	RAM	2G	2G	2G	4G
	OS	Windows XP Professional SP2	Windows XP Professional SP3	Windows XP Professional SP3	Windows Vista Enterprise K SP2
측정 결과		30.02	31.09	72.48	37.50

다. 실험은 시스템에 미치는 오버헤드 측정과 데이터 생성시간 측정, 블랙박스의 입력 데이터 스트림 블록 크기에 따른 효율성 비교, 무결성 검증시간 측정 이렇게 4가지로 하였다. 실험에 쓰인 프로그램의 전자서명 알고리즘에는 RSA 1024bits-PSS Encoding을 해시 알고리즘에는 SHA-1을 사용하였다.

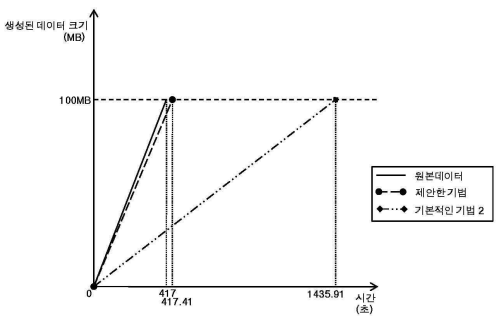
6.1 시스템 오버헤드 측정

시스템의 오버헤드 측정은 자체적으로 720MB의 바이너리 데이터를 생성하면서 실시간으로 *IVD*를 생성할 때에 걸리는 시간을 측정하였다. 블록의 크기는 100KB로 하였다. 실험은 다양한 시스템 환경에서 진행하였고 측정결과는 [표 9]와 같다.

720MB는 보통 1시간 분량의 동영상을 나타낼 수 있는 크기이다. 1시간 분량의 데이터에 대한 *IVD*의 생성시간은 시스템별로 차이는 있지만 모두 다 시스템에 무리를 주지는 않는 것으로 나타났다.

6.2 데이터 생성시간 측정

이번에는 100MB의 데이터를 실시간으로 생성하면서 각 데이터의 생성시간을 측정해 비교하였다. 원본데이터, 우리가 제안한 기법으로 생성된 무결성 검증 데이터, 기본적인 기법 중 방법 2를 적용해 생성된 무결성 검증 데이터 이렇게 3가지 데이터를 대상으로 하였다. 기본적인 기법 중 방법 1을 실험에서 제외한 것은 앞 장에서의 분석결과 보안요구사항을 많이 만족시키지 못하기 때문이다. 결과는 [그림 6]과 같다.



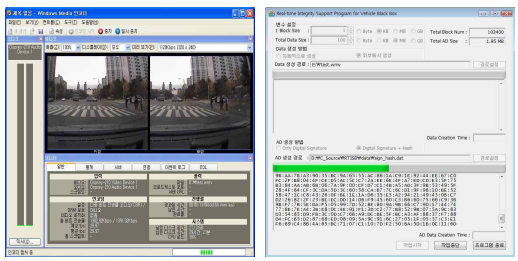
[그림 6] 데이터 생성시간 측정

우리가 제안한 기법의 무결성 검증 데이터 생성시간은 원본 데이터의 생성시간과 거의 유사함을 보임으로서 우리가 제안한 기법이 실시간 데이터를 처리함에 있어서 매우 효율적임을 알 수가 있었다.

6.3 효율적인 블록 크기 측정

이번에는 블랙박스의 입력 데이터 스트림을 어느 정도 크기의 블록으로 나누어 처리했을 때 더 효율적인지 알아보는 실험을 하였다. 이번 실험은 실제로 동영상을 촬영하면서 100MB의 데이터에 대해 실시간으로 *IVD*를 생성할 때에 걸리는 시간을 측정하였다. [그림 7]은 동영상 촬영에 쓰인 프로그램(a)과 우리가 구현한 무결성 검증 프로그램(b)의 동작화면을 캡처한 것이다. 사용한 시스템 환경은 CPU는 Pentium D 3.0GHz이고, RAM은 2G이고, OS는 Windows XP Professional SP2 이며, 동영상을 촬영하는데 쓰인 비디오 카드는 Osprey-210 AVStream Video Device이다. 측정결과는 [표 10]과 같다.

우리가 제안한 기법은 블록의 크기에 상관없이 평균적인 수치를 나타냄으로서 블록의 크기는 성능에 크게 영향을 미치지 않는 것으로 나타났다.



(a) (b)

[그림 7] (a) 동영상 촬영에 쓰인 프로그램
(b) 무결성 검증 프로그램

[표 10] 효율적인 블록 크기 측정

단위 : 초

변수	블록 크기	1KB	2KB	4KB	8KB	16KB	32KB	64KB	128KB
	블록 개수	102400개	51200개	25600개	12800개	6400개	3200개	1600개	800개
측정 결과		152.51	153.73	153.64	153.02	152.41	153.39	153.13	153.76

[표 11] 무결성 검증시간 측정

단위 : 초

변수	블록 크기	1KB	2KB	4KB	8KB	16KB	32KB	64KB	128KB
	블록 개수	102400개	51200개	25600개	12800개	6400개	3200개	1600개	800개
측정 결과		6.89	6.65	5.90	6.03	6.01	5.95	5.71	5.63

6.4 무결성 검증시간 측정

이번에는 위에서 생성된 *IVD*에 대해서 무결성 검증을 하는 데에 걸리는 시간을 측정하였다. 측정결과는 [표 11]과 같다. 위의 실험결과와 수치는 다르지만 블록의 크기에 상관없이 평균적인 수치를 나타냄으로서 무결성 검증시간 역시 블록의 크기에 영향을 받지 않는 것으로 나타났다.

또한 무결성을 검증하는 시간이 무결성 검증 데이터를 실시간으로 생성할 때보다 약 25배 빠른 것으로 나타났다. 계산량은 무결성 검증 데이터 생성시간과 검증시간이 비슷하지만 무결성을 검증하는 시점에는 이미 데이터가 완성되어 있는 상태이기 때문에 이 같은 결과가 나타났다. 이것은 TTP가 무결성 검증을 빠르게 할 수 있음을 의미한다.

6.5 기타

실험에서는 직접 스마트카드에 데이터를 넣어 전자서명값을 얻지는 못했다. 하지만 우리가 제안한 기법은 첫 번째 블록만 스마트카드의 도움을 받기 때문에 수치에 미치는 영향은 미미할 것으로 본다.

VII. 결 론

차량용 블랙박스 시스템에 저장된 데이터가 차후 법적증거로서의 효력을 발휘하려면 무결성의 보장이 필수로 되어야 하고 블랙박스 시스템의 특성상 데이터가 실시간으로 생성되기 때문에 무결성 또한 실시간으로 보장해야 한다. 하지만 아직까지 그런 연구가 진행되지 않고 있는 실정이다. 따라서 본 논문에서는 차량용 블랙박스 시스템을 위한 실시간 무결성 보장 기법을 제안하고 실제로 구현한 뒤에 성능을 테스트 해보았다.

차량에서 실시간으로 생성되는 데이터는 아무런 연산을 거치지 않기 때문에 생성되는 시간이 매우 빠르다. 그렇기 때문에 무결성 검증 데이터 또한 최대한 빠르게 생성되어야하고 공격자의 다양한 공격으로부터도 안전해야한다. 만약 사고가 발생했을 때 물리적인 충격으로 인해 블랙박스가 작동을 멈춘다고 해도 최대한 많은 무결성 검증 데이터가 생성되어야 할 것이다. 하지만 무결성 검증 데이터의 생성시간이 느리다면 블랙박스는 정작 사고 당시의 데이터의 무결성은 입증하지 못하고 그 이전의 데이터에 대해서만 무결성을 입증하게 될 것이다. 우리가 제안한 기법은 실험결과 이러한 문제를 최소화 할 수 있는 매우 높은 성능을 나타내었고, 보안 분석결과에서도 공격자의 다양한 공격으로부터 안전한 것으로 나타났다.

아직까지는 이 분야의 연구가 진행되지 않았고 시작하는 단계인 만큼 추후에는 더 나은 기법을 고안하여 현재 제안한 기법과 비교 분석할 예정이다.

참 고 문 헌

[1] http://thumb.paoin.com/paoweb/common/flash/ArticleViewer02.swf?CNo=60204159

[2] NHTSA EDR Working Group, “Event Data Recorders Summary of Findings,” NHTSA, USDOT, May 2002.

[3] IEEE 1616 Working Group, “Motor Vehicle Event Data Recorders,” IEEE, Sep. 2004.

[4] VEDI Technical Committee, “SAE J1698: Vehicle Event Data Interface-Vehicular Output Data

Definition," SAE, Feb. 2005.
[5] 한인환, "차량용 블랙박스 기술 특허분석 및 표준화 방안," 대한교통학회지, 25(3), pp. 29-43, 2007년 6월.
[6] 김형민, "자신을 보호하기 위한 새로운 선택 - 차량용 영상 블랙박스," 모터매거진, 카오디오, 1호, p. 73, 2009년 1월.
[7] http://www.mpeg.org/
[8] A. Kassem, R. Jabr, G. Salamouni, and Z.K. Maalouf, "Vehicle Black Box System," IEEE, SysCon-IEEE International Systems Conference, pp. 1-6, Apr. 2008.
[9] 박대우, 서정만, "자동차의 블랙박스를 이용한 실시간 포렌식 자료 생성 연구," 한국컴퓨터정보학회논문지, 13(1), pp. 253-260, 2008년 1월.
[10] 김진일, 윤장혁, 김진수, "모바일 장치를 이용한 자동차 영상블랙박스 설계," 한국정보기술학회 하계학술대회논문집, pp. 364-367, 2009년 6월.

<著者紹介>

김 윤 규 (Yungyu Kim) 학생회원
2008년 2월: 명지대학교 컴퓨터공학과 졸업
2008년 3월 ~ 현재: 고려대학교 정보경영공학전문대학원 석사과정
<관심분야> 암호프로토콜, VANET, 정보보호 응용, 암호시스템

김 범 한 (Bum Han Kim) 학생회원
2004년 2월: 숭실대학교 수학과 졸업
2006년 2월: 고려대학교 정보경영공학전문대학원 석사 졸업
2008년 3월 ~ 현재: 고려대학교 정보경영공학전문대학원 박사과정
<관심분야> 암호프로토콜, VANET, USIM 보안, 애드 혹 네트워크, 응용암호

이 동 훈 (Dong Hoon Lee) 종신회원
1983년 8월: 고려대학교 경제학사 졸업
1987년 12월: Oklahoma University 전산학과 석사 졸업
1992년 5월: Oklahoma University 전산학과 박사 졸업
1993년 3월 ~ 1997년 2월: 고려대학교 전산학과 조교수
1997년 3월 ~ 2001년 2월: 고려대학교 전산학과 부교수
2001년 3월 ~ 현재: 고려대학교 정보경영공학전문대학원 교수
<관심분야> 암호프로토콜, 암호이론, USN이론, 키 교환, 익명성 연구, PET 기술