Journal of The Korea Society of Computer and Information
Vol. 23 No. 12, pp. 73-80, December 2018

www.ksci.re.kr
https://doi.org/10.9708/jksci.2018.23.12.073

# A CRL Distribution Scheme Minimizing the Time for CRL Processing of Vehicles on Vehicular Communications

Hyun-Gon Kim*

## Abstract

Certification revocation list(CRL) is needed for excluding compromised, faulty, illegitimate vehicle nodes and preventing the use of compromised cryptographic materials in vehicular communications. It should be distributed to vehicles resource-efficiently and CRL computational load of vehicles should not impact on life-critical applications with delay sensitive nature such as the pre-crash sensing that affords under 50msec latency. However, in the existing scheme, when a vehicle receives CRL, the vehicle calculates linkage values from linkage seeds, which results in heavy computational load. This paper proposes, a new CRL distribution scheme is proposed, which minimizes the time for CRL processing of vehicles. In the proposed scheme, the linkage value calculation procedure is performed by road-side unit(RSU) instead of the vehicle, and then the extracted linkage values are relayed to the vehicle transparently. The simulation results show that the proposed scheme reduces the CRL computational load dramatically, which would minimize impact on life-critical applications' operations with low latency.

▸Keyword: Certificate Revocation List, linkage seed, linkage value, Linkage Authority, road side unit

## I. Introduction

차량 통신은 운전자의 안전과 편의성을 높이고 교통 시스템이나 혼잡 상황에 대한 정보를 실시간으로 제공함으로써 운전의 효율성을 높인다. 최근에는 차량통신에 이동통신 액세스 기술을 접목하는 표준화 활동이 활발히 진행되고 있다. 이와 관련하여 5G기반의 자동차 연합회(5GAA)는 차량통신에 5G를 적용한 차량통신 응용들을 개발하고 있다. 이 연합회에는 아우디, BMW 그룹, 다임러, 포드 외에도 인텔,퀄컴, 에릭슨, 노키아, 화웨이 등이 참여하고 있다.

한편, 차량통신은 네트워크를 형성하는 차량 노드가 고속으로 이동하므로 링크 연결 시간이 짧고, 패킷 손실률이 높아져 네트워크 연결 상태가 전반적으로 불안하기 때문에 통신 서비스 품질이 낮아질 수 있다. 차량의 응용들은 제공하는 기능에 따라 허용 가능한 최대 지연(latency) 시간이 정의되어 있으며 대부분 응용들은 지연에 매우 민감하다[1].

예를 들어, 교차로 충돌 경고(intersection collision warning)」 응용은 교차로에 진입할 때, 코너에 있는 건물들로 인하여 운전자의 시야에 보이지 않는 잠재적인 측면 충돌 가능성을 판단하여 운전자에게 알려주는 응용으로, 허용 가능한 최대 지연시간은 100msec이다. 또 다른 예로 「충돌 사전 감지(pre-crash sensing)」 응용은 운전자에게 피할 수 없는 충돌을 미리 감지하고 알리는 응용으로, 허용 가능한 최대 지연시간은 50msec이다. 최대 지연시간이란 응용 계층 간에 단 대 단(end-to-end) 지연시간을 의미한다. 따라서 보안 기능을 설계하는 단계에서는 최대 지연시간을 반드시 고려하여야 한다.

한편, 차량통신에서는 차량의 익명성을 보장하기 위하여 공개키 방식의 단기 익명인증서(pseudonyms)를 사용한다. 차량내에는 하드웨어 보안 모듈(HSM; hardware security module)을 통해 서명 키의 안전한 저장과 익명인증서를 사용하여 전자서명과 검증을 수행한다. 그러나 HSM은 CPU에 비해 낮은 클럭 스피드와 버스 딜레이, 다수의 응용들이 동시 후킹할 경우 등의 요소로 빠른 전자서명과 서명검증이 느려져 지연이 발생한다.

차량의 비정상 행위, 차량 오너 변경, 장비 고장, 개인키 및 인증서 손상(compromised) 등으로 인해 인증서를 취소해야 할 경우에는 인증서 취소목록(CRL; Certificate Revocation List)을

• First Author: Hyun-Gon Kim, Corresponding Author: Hyun-Gon Kim
*Hyun-Gon Kim (hyungon@mokpo.ac.kr), Dept. of Information Security, Mokpo National University
• Received: 2018. 09. 18, Revised: 2018. 11. 10, Accepted: 2018. 11. 19.

이용한다[2]. CRL은 발급기관에 의해 서명되고 서명 결과 값이 첨부되어 주기적으로 배포되며, 차량은 CRL을 수신하면 서명을 검증하고 검증에 성공했을 경우에만 CRL 정보를 신뢰한다. 차량통신에서 정의한 CRL은 기존의 유선 X.509[3]의 CRL에 비해 배포기간이 빈번하고, 사이즈가 크고, 무선 채널을 사용하기 때문에 무선자원을 효율적으로 사용해야 하고, 일시적인 통신 장애에도 정확히 제 시간에 CRL을 배포해야 한다.

만약, CRL 사이즈가 크거나 CRL 처리 절차가 복잡하다면 이를 처리하기 위해 차량 OBU(OBU; On-Board Unit)의 CPU 점유율이 비례적으로 높아지게 된다. 이로 인해, 지연에 민감한 다른 응용들의 성능에 부정적인 영향을 미칠 수 있다. 따라서 지연에 민감한 응용들에게 독립적인 성능과 처리의 신뢰성을 제공해 주기 위해서는 차량 OBU의 CRL 처리 부하를 감소시켜줄 필요가 있다.

이와 관련하여 본 논문에서는 차량 OBU의 CRL 처리 부하를 감소시켜서 지연에 민감한 다른 응용들에 미치는 영향을 경감시킬 수 있는 CRL 처리 기법을 제안하였다. 그리고 제안한 기법을 소프트웨어로 구현하고 차량 OBU의 실행시간을 측정한 다음, 기존 기법과 제안한 기법의 성능을 비교분석 하였다. 본 논문은 다음과 같이 구성된다. 서론에 이어 2장에서는 기존 연구에서 제시하는 CRL 배포기법들과 표준에서 정의한 보안 인증관리 시스템을 소개하고[4], 이 시스템에서 CRL을 배포하는 절차를 자세히 살펴본다. 3장에서는 본 논문에서 제안한 CRL 처리 기법을 설계하였다. 4장에서는 기존의 CRL 배포 기법과 제안한 기법의 지연시간을 비교분석하고, 5장에서 결론을 맺는다.

## II. Related Works

### 1. CRL Distribution Schemes

CRL 배포와 관련하여 차량의 주변 환경, 네트워크 환경, 차량 인프라 등의 요소들을 고려하여 CRL 배포를 효율적으로 배포하기 위해 필터나 압축 기법을 사용하여 CRL 사이즈를 줄이거나, 지역별 CRL을 별도로 만들어 배포하거나, 차량통신 채널 외에 이동통신 등과 같은 다른 채널을 이용해 CRL을 배포하는 기법 등이 연구되고 있다.

P. Rapadimitrators[5]는 CRL을 효율적으로 배포하는 기법을 제안하였다. 차량통신 전체 지역을 다수의 영역으로 나누고 지역별로 CA(Certificate Authority)를 둔 다음, 각 지역별 CA가 CRL을 배포하는 방식이다. 차량이 이전 영역을 벗어나면 그 영역의 CA로부터 외부 인증서를 발급받아 사용한다. 이 때 이전 영역의 CA는 그 차량이 자신의 영역을 벗어났으므로 CRL에 등록하여 배포하여 통신을 차단한다. 즉, 영역별로 CRL을 관리하여 효율성을 기한다.

K. Laberteaux[6]가 제안한 기법에서는 CRL을 다수의 조각으로 나눈 다음, 초기에는 기지국(RSU; Road Side Unit)을 통해 배포하고 그 이후부터는 차량과 차량간의 통신에 의해 조각들을 점진적으로 배포한다. Lin X[7]가 제안한 기법에서는 RSU가 CRL을 주도적으로 관리한다. RSU는 자신의 영역에서 송수신되는 모든 메시지의 인증서 상태를 체크하고, 만약 특정 차량의 인증서를 취소해야겠다고 판단하면 자신의 영역에 속해있는 모든 차량들에게 취소 정보를 방송한다. 즉, RSU가 CRL을 관리하여 효율성을 기한다.

최근 Noureddine Lasla[8]는 차량통신에 블록체인 기법을 접목하여 차량 OBU가 수신한 메시지에 대해 전자서명을 검증하는 계산량과 CRL 처리에 필요한 계산량을 감소시키는 기법을 제안하였다. RSU들간에 검증된 공개키 정보를 블록체인 네트워크에 공유한다. RSU는 검증된 공개키를 자신의 영역의 차량들에게 배포한다. 차량 OBU는 이를 저장하고, 수신 메시지에 대해 전자서명과 검증 대신에 저장된 공캐키를 룩업하여 차량을 인증한다. CRL을 처리하는데 있어서도 유사한 동작을 한다. RSU들간에 취소된 공개키를 RSU들로 구성된 블록체인 네트워크에 공유하고, 각 RSU는 취소된 공개키 정보를 자신의 영역의 차량들에게 배포한다. 차량 OBU는 이를 저장하고, CRL을 수신하면 전자서명과 검증 대신에 저장된 취소된 공캐키를 룩업한다. 만약 수신한 공개키가 데이터베이스에 저장된 취소된 공개키와 동일하면 해당 차량과의 통신을 차단한다.

이 외에 관련 연구로, 참고문헌[9]는 기존의 CRL 배포 기법들의 타입, 확장성, 프라이버시 지원 여부, 알고리즘의 사전이나 또는 사후 동작 여부, 기타 특징들을 상세히 비교 분석하였다.

### 2. Security Credential Management System (SCMS) for V2X Communications

본 논문의 기초가 되는 보안 인증관리 시스템(SCMS)은 IEEE 1609.2와 CAMP VSC3에서 정의한 차량용 공개키 기반 구조(PKI)이다[4]. Fig 1은 익명인증서와 CRL 관련 동작만을 설명하기 위해서 SCMS 전체 구조를 나타내지 않고, 연관된 엔티티들만을 추출하여 나타내었다. 아래의 설명도 익명인증서 생성 절차의 일부와 CRL 처리와 관련된 기능만을 서술하였다.

□ OBU : 차량에 탑재되어 통신기능을 수행하며, HSM을 통해 중요 키의 보관과 보안 프리미티브를 제공한다. 익명인증서를 사용하여 송수신되는 메시지에 전자서명과 서명검증을 하여 안전한 통신을 이룬다. CRL을 수신하여 취소된 인증서 목록을 관리하고, 주변으로부터 취소된 인증서를 가진 메시지를 수신하면 그 차량과의 통신을 차단한다.

□ RSU : 차량통신 네트워크와 자신의 영역에 속한 차량 OBU들과의 중계역할을 하는 무선 기지국이다.

□ RA(Registration Authority): OBU가 요청한 익명인증서 발급 요청이나 리필 요청에 대한 유효성을 검증하고, PCA에게 익명인증서 발급을 요청한다. 이 때 하나의 익명인증서 발급 요청에 대해 버터플라이 키 확장을 통해 다수의 익명인증서 발급 요청을 만들어 PCA에게 전달한다. 그리고 다수의 OBU로부터의 익명인증서 발급 요청을 섞어서 PCA가 특정 OBU로부터의 익명인증서 발급 요청인지를 식별할 수 없도록 한다.

□ LA(Linkage Authority) : 사전 링크 값(pre-linkage value, $plv$)을 생성한다. LA는 반드시 두 개의 인스턴스(LA1,

LA2)를 둔다. 두 LA는 $plv$를 각각 생성하고 암호화시켜 RA를 거쳐 PCA에게 전달한다. 암호화된 $plv$는 PCA가 복호화하여 링크 값(linkage value, $lv = plv_1 \oplus plv_2$)을 계산하는데 사용한다. 두 개의 LA를 구성한 이유는 하나의 LA가 생성한 $plv$가 노출되더라도 완전한 $lv$를 계산할 수 없도록 하는 것이다. 즉, 두 개의 LA로부터 생성된 두 개의 $plv$를 XOR해야 $lv$를 계산할 수 있다.

□ PCA(Pseudonym Certificate Authority): OBU가 요청한 익명인증서들을 생성하고 RA를 통해 차량에게 배포한다. 익명인증서 발급 요청 메시지의 해시 값과 $lv$를 매핑시켜 자신의 데이터베이스에 저장해둔다.

□ MA(Misbehavior Authority): 비정상 행위를 하는 차량을 관리하는 엔티티이다. 특정 차량이 비정상 행위를 한다고 보고되면 먼저, 비정상 행위 탐지 알고리즘을 실행하여 요청한 차량의 익명인증서가 비정상 행위를 하는지 판단한다. 만약 비정상이라 판단되면, 인증서 취소목록에 포함시킨다. MA는 주기적으로 취소된 인증서들을 취합하여 CRL을 만들고 배포한다. CRL을 배포하는 방법은 RSU를 통해 주기적으로 방송하거나 차량이 주기적으로 CRL 저장소로부터 CRL을 다운로드 한다.

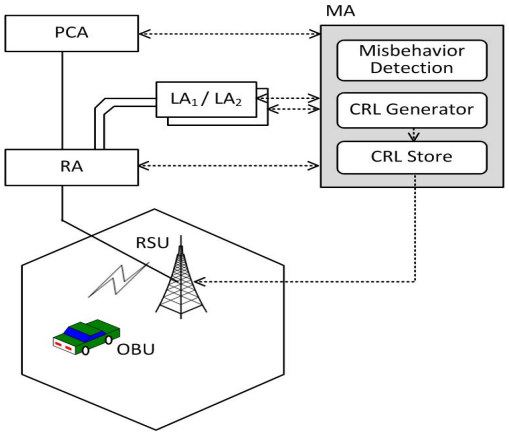


Fig. 1. SCMS architecture with related to the pseudonyms and CRL

## 3. Misbehavior Detection and CRL assembling

이 절에서는 차량통신 네트워크에서 비정상 행위의 인증서가 MA에게 보고되는 단계부터 그 인증서가 CRL에 포함되어 배포되는 과정을 살펴본다.

유선의 X.509v3에는 인증서의 주체가 누구인지를 나타내는 식별자(subject name)가 포함되어 있다. 그러나 차량통신에서는 차량의 익명성을 제공하기 위해서 단기 익명인증서에 주체를 포함시키지 않는다. 주체 대신 익명인증서를 식별할 수 있는 $lv$를포함시킨다. 즉, 인증서에 포함된 $lv$ 값을 추적하면 해당하는 차량을 특정할 수 있다. $lv$는 익명인증서에 대한 식별자로 사용되기도 하지만 인증서를 취소하는 데에 필요한 식별자로도 사용된다. CRL에는 $lv$를 직접 포함시키는 대신에 $lv$를 계산할 수 있는 두 개의 링크 시드(linkage seed, $ls$), 링크 값을 생성하는 두 개의 $la\_id$, 그리고 익명 인증서 발행시간인 $i$값이 포함된다. 따라서 CRL을 수신하는 차량 OBU는 이들 5개의 데이터를 입력으로 하여 $lv$를 계산해야 한다.

$plv$를 계산하는 과정을 살펴보자. Fig. 2에 나타낸 바와 같이, LA1과 LA2는 링크 시드를 기초로 하여 $plv$를 각각 생성한다. 생성된 두 $plv$는 PCA로 전달되며, PCA에서 조합되어 $lv$가 계산된다($lv = plv1 \oplus plv2$). 여기서 $i$ 값은 $plv$를 생성한 시간이며 $j$는 인스턴스이다. $j_{\max}$ 값은 $i$ 시점에 생성하는 인스턴스의 최대 수이며, 이 값은 차량의 익명성을 제공하기 위해 $i$ 시점에 발급된 익명인증서의 최대 개수를 의미한다. 만약 $i$가 한 주이고 $j_{\max}$가 40이라면 한 주에 40개의 $plv$가 생성된다. 따라서 $plv$로부터 $lv$가 생성되므로 40개의 익명인증서($lv$)가 생성되는 것이다.

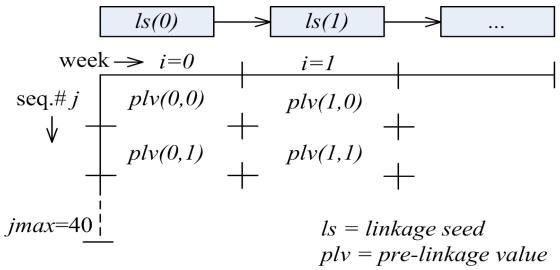


Fig. 2. Calculation of pre-linkage values

특정 $lv$가 비정상 행위로 보고되었을 때부터 그 $lv$가 CRL에 포함되는 과정을 단계별로 Fig. 3에 나타내었다.

□ 단계 1 : 비정상 행위를 하는 차량의 인증서를 수신한 주변 차량이나 기지국은 그 차량의 익명인증서와 비정상으로 판단한 사유를 MA에게 보고한다.

□ 단계 2 : MA는 비정상 행위 검출 알고리즘을 실행하여 해당 인증서를 취소할지를 판단한다.

□ 단계 3a : MA가 취소를 결정하면 PCA에게 $lv(i,j)$을 전달하고 이에 해당하는 *RA-to-PCA pseudonym certificate request*의 해시 값을 요청한다. 이 해시 값은 이전에 OBU가 PCA에게 익명인증서를 요청할 때, RA가 보내는 요청 메시지의 해시 값이며 PCA가 저장하고 있다.

□ 단계 3b : PCA는 *RA-to-PCA pseudonym certificate request*의 해시 값과 이 요청을 처리한 RA 정보를 리턴한다.

□ 단계 4a : MA는 수신한 *RA-to-PCA* pseudonym certificate request의 해시 값을 RA에게 전달한다.

□ 단계 4b : RA는 해시 값에 일치하는 인증서를 블랙리스트에 추가시킨다. 그리고 자신의 데이터베이스에서 해시 값에 일치하는 암호화된 $lci$(Linkage Chain Identifier)들 즉, $lci_1$과 $lci_2$를 추출한다. $lci$는 LA가 초기 링크 시드 값을 자신의 공개 키로 암호화 한 것으로 LA만 복호화할 수 있다.

□ 단계 4c : RA는 MA에게 $lci_1$, $lci_2$와 연관된 $la\_id_1$, $la_id_2$를 리턴한다.

□ 단계 5a : MA는 암호화된 $lci_1$, $lci_2$를 각 LA들에게 전달하고 이에 해당하는 링크 시드 값들을 요청한다.

□ 단계 5b : LA들은 $lci_1$, $lci_2$를 각각 복호화하고 이에 해당하는 $i$ 값, 링크 시드 값 $ls_1(i)$, $ls_2(i)$과 해당 LA들 ID ($la\_id_1$, $la\_id_2$)를 리턴한다.

□ 단계 6 : MA는 수신한 $i$, $ls_1(i)$, $ls_2(i)$, $la\_id_1$, $la_id_2$를 CRL에 포함시킨다. 취소된 인증서들의 목록을 취합하여 CRL로 어셈블링되고 전자 서명되어 최종적으로 RSU를 통해 차량들에게 배포된다.

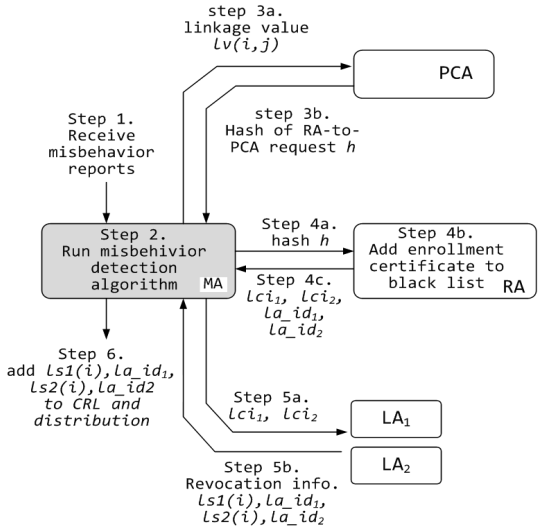


Fig. 3. Misbehavior detection and pseudonym certificate revocation

# III. The Proposed Scheme

## 1. Design Principle

SCMS에서는 CRL에 차량의 인증서 식별자인 $lv$를 사용하지 않고 $ls$를 사용한다. 그러나 차량 OBU 관점에서는 $ls$로부터 $lv$를 계산하기 위해 많은 연산을 해야 하므로 계산량이 커져 일시적으로 OBU의 CPU 자원을 고갈시킬 수 있고, 지연이 커질 수 있다. 또한, CRL내에 포함된 $ls$의 수가 많아지면 이에 비례하여 계산량도 늘어나게 된다. 이로 인해 지연에 민감한 다른 응용들이 정상적으로 동작하는 데에 부정적인 영향을 미칠 수 있다. 따라서 지연에 민감한 응용들에게 독립적인 성능과 처리의 신뢰성을 보장해 주기 위해서는 차량 OBU의 CRL 처리 부하를 감소시켜줄 필요가 있다.

제안한 기법의 주요 아이디어는 차량 OBU가 CRL을 처리하는데 있어서 기존 $lv$를 계산하는 절차를 기지국인 RSU가 대신 실행하도록 한다. 그리고 RSU는 계산된 $lv$를 차량 OBU에게 그대로 전달하여 차량 OBU가 바로 사용할 수 있도록 한다. 이때 $lv$를 텍스트 형태로 CRL에 포함시키면 중간에 노출되어 익명성이 보장되지 않으므로 이를 방지하기 위해 RSU와 차량 OBU간에 암호화와 전자서명을 적용한다.

## 2. RSU Functionalities

이 절에서는 제안한 기법을 위해 RSU가 처리해야 되는 절차를 설계하였다. 기존의 RSU는 CRL을 수신하면 별도의 처리 과정을 거치지 않고 그대로 자신의 영역에 속하는 OBU들에게 전달하나, 제안한 기법에서는 Fig. 4, Fig. 5, Fig. 7에 나타낸 추가적인 절차를 실행해야 한다. 즉, RSU가 $lv$를 계산하므로 RSU 영역에 속한 모든 차량 RSU들은 $lv$ 계산 절차를 모두 생략할 수 있다. 절차에 사용된 용어를 Table 1에 나타내었다.

Table 1. Algorithm Notation

| | |
|---|---|
| $r, s$ | components of a signature |
| $x, y$ | coordinates of a public key |
| $n$ | the multiplicative order of the point $G$ |
| $G$ | elliptic curve base point |
| $ls$ | linkage seed |
| $lv$ | linkage value |
| $iCert$ | an indication of the time period that applies to the certificate |
| $msg\_hash$ | hash_of the message to be verified using SHA-256 |

1) CRL 파싱 및 서명 검증

RSU가 CRL을 수신하면 1609.2 메시지 포맷을 분해하여 CRL을 추출한 다음, ECIES_NISTp256 알고리즘으로 복호화 한다[4]. 그 결과를 ECDSA_NISTp224 with SHA-224 또는 ECDSA_NISTp256 with SHA-256 알고리즘으로 서명을 검증한다. 서명 검증에 성공하면 인증서 취소 엔트리에 포함된 $ls_1$, $la\_id_1$, $ls_2$, $la\_id_2$, $i$를 추출한다. 동작을 Fig. 4에 나타내었다.

```
1) CRL parsing and verify
 Decrypt CRL using ECIES_NISTp256
 Split signature into its two components r and s
 Compute ((s-1)mod n), store the result in z
 Convert message_hash into an integer, store the
     result in u1
 Compute ((z * u1)mod n), store the result in u1
 Compute ((z * r)mod n), store the result in u2
 Compute ((G + public key), store the result in sum
 Compute ((u1 * G) + (u1 * public_key), store the
     result in (xR + yR)
 Compute((xR mod n) and store the result in xR
 if r is equal to x
     For 0 to iMax-1 do
         Extract ls1, la_id1, ls2, la_id2, I, store them
     end for
    return true
 else return false
 endif
```

Fig. 4. Procedure of CRL parsing and verify

2) 링크 값 계산 및 CRL 추가

위의 절차 추출한 $ls_1$, $la\_id_1$, $ls_2$, $la\_id_2$, $i$을 입력으로 하여 $i$부터 $i_{Max}$까지 해시 링크 체인 연산을 통해 $lv$ 값들을 추출한다. 추출한 $lv$ 값들은 CRL에 추가된다. 동작을 Fig. 5와 Fig. 6에 나타내었다.

2) Linkage value calculation and add to CRL
While *i* value < *iCert*
Set $ls_1$ = LSB of 16 octets of [SHA256($la\_id_1$) || $ls_1$ || $0^{112}$)], where $0^{112}$ is 112 0 bits
Set $ls_2$ = LBS of 16 octets of [SHA256($la\_id_2$) || $ls_2$ || $0^{112}$)]
For *j*=0 to $j_{Max}$-1
Set *data* = $la\_id_1$ || Unit32(*j*) || $0^{80}$, where Unit32(*j*) indicates j represented as a 4-octet integer
Set $plv_1(j)$ = AES(key=$ls_1$, data=*data*) XOR(data)
Set $plv_2(j)$ = AES(key=$ls_2$, data=[$0^{10}$ || $la\_id_2$ || *Uint32(j)*]) XOR [$0^{10}$ || $la\_id_2$ || *Uint32(j)*]
Set *lv(j)* = *LSB of 9 bytes of* $plv_1$ *XOR* $plv_2$
Add *lv(j)* to the CRL

Fig. 5. Procedure of linkage value calculation

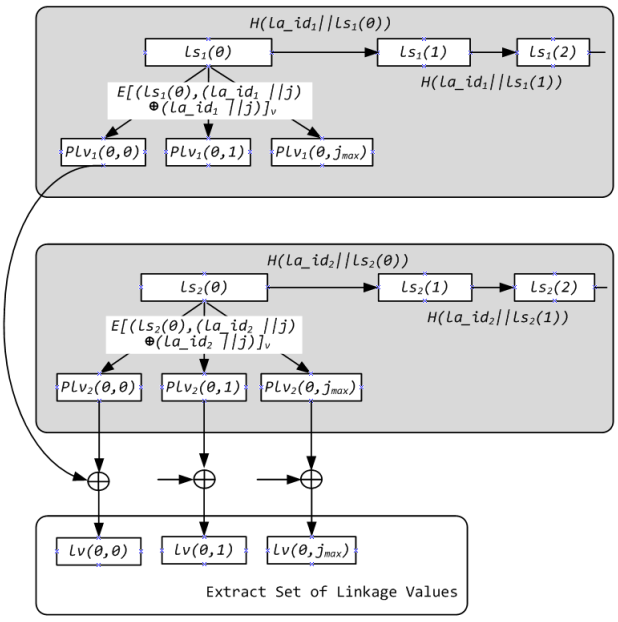


Fig. 6. Linkage value calculation

3) CRL 재조립 및 재서명

계산된 $lv$ 값들을 CRL내에 있는 linkage value entries에 포함시킨다. CRL을 ECDSA_NISTp224 with SHA-224 또는 ECDSA_NISTp256 with SHA-256 알고리즘으로 전자서명 한다. 그 결과를 ECIES_NISTp256 알고리즘으로 암호화 한다. RSU는 완성된 CRL을 자신의 영역에 속한 차량 OBU들에게 배포한다. 동작을 Fig. 7에 나타내었다.

3) CRL reassemble and re-sign
Insert $lv$s to linkage value entries in CRL
For 0 to 64 do
Generate two random numbers *k*, *tmp*
Compute (*k* * *G*) and store the result in *p*
Compute ((*k* * *tmp*)mod n), store the result in *k*
Compute (($k^{-1}$)mod n), store the result in *k*
Compute ((*k* * *tmp*)mod n), store the result in *k*
Set the first 32bytes of *signature* equal to *p*
Compute((*private_key***p*)mod n), store the result in *s*
Convert *message_hash* into an integer and store the result in *tmp*
Compute((*k* * (*s* + *tmp*))mod n), store the result in *s*
if *s* has more than 256bits then break and try again
endif
Set bytes 33 to 64 of *signature* to *s*
Return true
end for
Encrypt CRL using ECIES_NISTp256

Fig. 7. Procedure of CRL reassemble and re-sign

## 3. OBU Functionalities

선술한 바와 같이, 기존에는 $ls$로부터 $lv$를 계산하는 기능을 차량 OBU가 처리했으나 제안한 기법에서는 RSU가 수행한다. RSU가 계산된 $lv$ 값들은 차량 OBU에게 전송하므로 차량 OBU는 별도의 계산 없이 $lv$ 값들을 자신의 데이터베이스에 저장하여 활용한다. 차량 OBU가 RSU나 주변 차량 OBU와 통신을 할 때, 자신의 데이터베이스를 룩업하여 수신한 인증서가 정상적인 인증서인지 아니면 취소된 인증서인지를 확인한다. 즉, 링크 값들을 계산하는 대신에 데이터베이스에 저장된 링크 값을 룩업하는 연산만 수행하므로 계산량도 줄어들고 빠르게 특정 링크 값을 찾아내므로 실행시간을 줄일 수 있다.

# IV. Experimental Comparison

제안한 기법의 성능을 실험을 통해 확인하였다. 실험은 기존 SCMS에서 계산하는 절차와 제안한 기법의 절차로 구분하였다. 제안한 기법에서는 차량 OBU는 $lv$ 값을 계산하지 않으므로 기존 기법과 직접적인 비교가 어렵다. 기존 기법은 차량 OBU가 $ls$로부터 $lv$를 계산하는 시간을 측정하였다. 제안한 기법에서는 차량 OBU는 $lv$ 값들을 자신의 데이터베이스에 저장하고, 필요 시 $lv$ 값을 룩업해서 사용하므로, 기존 기법과 비교를 위해 차량 OBU가 $lv$ 값을 룩업하는 시간을 측정하였다. 두 기법은 소프트웨어로 구현하고 성능을 비교분석하였다. 실험 환경과 사용한 알고리즘은 다음과 같다.

- CPU : Intel Core i5-6500 3.20GHz
- OS : Linux Debian 4.15.11-1kali
- Language : Python 2.7.15
- Library : openssl 1.0.0a

- Database : levelDB
- Sign and verity : ECDSA_NISTp256
- Encryption and Decryption : ECIES
- 해시 알고리즘 : SHA-256

## 1. Experimental Performance of OBU

기존의 기법에서 OBU가 $ls$로부터 $lv$ 값을 계산하는데 소요되는 실행시간을 측정하여 Fig. 8에 나타내었다. 실험 시, $lv$ 개수를 0에서 10,000개 까지 변화시키고, $J_{max}$는 5에서 80까지 변화시켜 결과를 도출하였다. $ls$ 개수가 증가하면 비례적으로 실행시간이 늘어난다. $J_{max}$가 40이고 $ls$의 개수가 6,000개일 때 실행 시간은 28.5sec로 측정되었다.

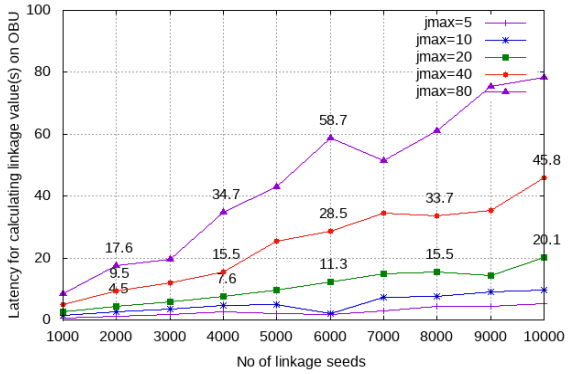


Fig. 8. OBU execution time for calculating linkage values

제안한 기법에서 데이터베이스는 대량의 고정된 패턴의 데이터($lv$ 값)를 저장하고 빠르게 룩업할 수 있는 특징을 가진 구글의 NoSQL 형태의 레벨 DB를 사용하였다. 레벨 DB는 키(key)와 값(value)을 바이트 문자로 저장하고 인덱싱을 사용할 경우 더욱 빠른 룩업이 가능하다. 실험에서는 $lv$ 값이 고정된 길이이므로 인덱싱 기능은 사용하지 않았다.

실험 시, $lv$ 개수를 0에서 10,000개 까지 변화시키고, $J_{max}$는 5에서 80까지 변화시켜 실험 결과를 도출하였다. $lv$ 개수가 증가하면 어느 정도는 불규칙하게 실행시간이 늘어난다. $J_{max}$가 20이고, $lv$의 개수가 6,000개일 때, 실행시간은 0.285ms로 측정되었다.

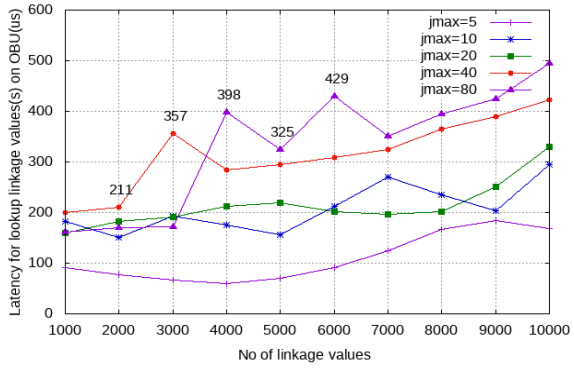


Fig. 9. Lookup time for searching linkage values

## 2. Experimental Performance of RSU

이전 3.2절에 서술한 RSU 동작들 즉, CRL 파싱 및 서명 검증, 링크 값 계산 및 CRL 추가, CRL 재조립 및 재서명 동작들을 을 구현하고 실행시간을 측정하였다. 실험 결과를 Table 2와 Fig. 10에 나타내었다. 실험 결과, CRL의 사이즈가 커질수록 지수적으로 실행시간이 늘어난다. CRL의 개수가 1개이고 CRL의 크기가 2,048KB일 때, 실행시간이 156.5ms로 나타났다. 그러나 이 수치들은 RSU의 플랫폼에 의존하여 많이 달라질 수 있으므로 의미가 크지는 않다.

Table 2. Time for CRL re-assembling on RSU(ms)

| CRL size(kB) | 8 | 16 | 32 | 64 | 128 | 256 | 512 | 1024 | 2048 |
|---|---|---|---|---|---|---|---|---|---|
| #of CRL = 1 | 221 | 20.1 | 20.5 | 21.7 | 22.4 | 43.7 | 63.8 | 103.8 | 156.5 |
| #of CRL = 2 | 38.8 | 40.7 | 45.9 | 47.7 | 52.6 | 77.4 | 104.4 | 161.7 | 280.8 |
| #of CRL = 3 | 58.6 | 52.0 | 62.7 | 78.3 | 86.8 | 102.7 | 140.2 | 201.6 | 291.4 |
| #of CRL = 4 | 68.1 | 69.2 | 71.9 | 85.0 | 115.2 | 129.3 | 162.3 | 225.8 | 509.4 |
| #of CRL = 5 | 91.7 | 92.2 | 85.6 | 103.7 | 112.5 | 164.2 | 211.9 | 334.8 | 650.8 |

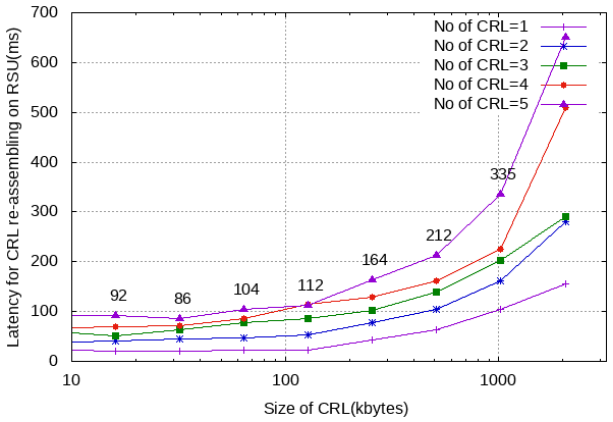


Fig. 10. Time for CRL re-assembling on RSU(ms)

## 3. Performance Evaluation

차량 OBU는 어떤 프로세서를 탑재하느냐에 따라 성능에 크게 영향을 받는다. 위 4.1절에 서술한 성능 측정결과에 의하면, 중간 값 즉, jmx가 40이고 $lv$가 6,000을 기준으로 하였을 때, 기존의 기법은 28.5sec가 소요되고, 제안한 기법은 0.285msec가 소요되므로 제안한 기법이 약 100,000배 빠르게 실행된다. 제안한 기법에서는 RSU의 계산 부하는 급격하게 늘어나지만, 반대로 차량 OBU의 계산 부하가 급격하게 줄어들어 나타난 결과이다.

만약, 차량 OBU가 50msec의 최대 지연시간을 가진 「충돌 사전 감지」 응용이 동작하는 중에 CRL을 수신하였다고 가정해 보자. 기존 기법과 같이 $lv$ 값을 계산하는데 25.8sec가 소요된다면, 차량 OBU는 CRL을 처리하느라 CPU 자원을 공유할 것이므로 「충돌 사전 감지」 응용이 최대 지연시간내에 동작하기 어려울 수 있다. 이에 비해 제안한 기법과 같이 차량 OBU의

$lv$ 계산 절차를 없앤다면 지연에 민감한 다른 응용들이 정상적으로 동작할 수 있을 것이다.

한편, 제안한 기법의 단점으로 RSU가 차량 OBU에게 $ls$ 값들을 전송하는 것에 비해, 계산된 $lv$ 값들을 전달하는 것이 전송해야 할 데이터가 많아지므로 CRL 사이즈가 다소 커지게 된다. 이로 인해 무선자원을 비효율적으로 사용할 수 있다. 즉, 제안한 기법은 차량 OBU의 계산량을 줄이는 데는 효과가 높지만 통신 부하가 다소 증가한다는 점이 있는 것이다. 제안한 기법에 무선 대역이 넓은 5G와 같은 이동통신을 적용할 경우 이러한 단점을 해소할 수 있다.

## V. Conclusions

본 논문에서는 차량 OBU의 CRL 처리 부하를 감소시켜서 지연에 민감한 다른 응용들에 미치는 영향을 경감시킬 수 있는 CRL 처리 기법을 제안하였다. 그리고 제안한 기법을 소프트웨어로 구현하고 차량 OBU의 실행시간을 측정한 다음, 기존 기법과 제안한 기법의 성능을 비교분석 하였다. 제안한 기법에서는 차량 OBU가 CRL을 처리하는데 있어서 링크 값($lv$)을 계산하는 절차를 기지국인 RSU가 대신 실행하도록 한다. 성능 실험 결과, RSU의 계산 부하는 늘어나지만, 차량 OBU가 처리해야 하는 계산 부하가 약 100,000배로 급격하게 줄어드는 것으로 나타났다. 이로 인해, 제안한 기법은 차량 OBU 내에 지연에 민감한 다른 응용들의 성능에 영향을 주지 않고 처리의 신뢰성을 제공해 줄 수 있다.

제안한 기법의 단점으로 RSU가 차량 OBU에게 링크 시드 값($ls$)들을 전송하는 것에 비해, 계산된 $lv$ 값들을 전달하므로 CRL 사이즈가 다소 커지게 된다. 이로 인해 무선자원을 비효율적으로 사용할 수 있다. 즉, 제안한 기법은 차량 OBU의 계산량을 줄이는 데는 매우 효과가 높지만 통신 부하가 다소 증가한다. 제안한 기법은 무선 대역이 넓은 5G 기반의 차량통신에 활용할 수 있다. 향후에는 5G 기반의 차량통신에서 CRL을 효율적으로 배포할 수 있는 배포 영역에 대해 연구할 계획이다.

## REFERENCES

[1] Pano Papadimitratos, et al., "Vehicular Communication Systems: Enabling Technologies, Applications, and Future Outlook on Intellignet Transportation," IEEE Communications Magazine, pp. 84-90, Nov. 2009.

[2] IEEE 1609.2-2016, "IEEE Standard for Wireless Access in Vehicular Environments-Security Services for Applications and Management Messages," IEEE Standard, March 2016.

[3] Bassham, L., Polk, W., and R. Housley, "Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3279, April 2002.

[4] Benedikt Brecht, Dean Therriault, et al,. "A Security Credential Management System for V2X Communications," IEEE Transactions on Intelligent Transportation Systems, pp. 1-25, Feb. 2018.

[5] P. Papadimitratos et al, "Certificate revocation list distribution in vehicular communication systems," Proc. Fifth ACM international workshop on Vehicular Inter-networking, pp. 86-87, 2008.

[6] K. Laberteaux et al, "Security certificate revocation list distribution for vanet," Proc. Fifth ACM international workshop on Vehicular Internetworking, pp. 88-89, Sept. 2008.

[7] Lin X, Lu R, Zhang C, et al., "Security in vehicular ad hoc networks," IEEE Communications Magazine pp. 88-95, 2008.

[8] Noureddine Lasla, Mohamed Younis, et al., "Efficien Distributed Admission and Revocation using Blockchain for Cooperative ITS," 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS) pp. 88-89, Feb., 2018.

[9] Taimur Khan, Naveed Ahmad, et al, "Certificate revocation in vehicular ad hoc networks techniques and protocols: a servery," Science China Press and Springer-Verlag Berlin Heidelberg, 2017.

## Authors

Hyun-Gon Kim received the B.S. and M.S. degrees at the department of Electrical Engineering of Kumoh National University and the Ph.D degree at the department of Computer Science of Chungnam National University, Korea, in 1992, 1994, and 2003 respectively. He worked at the division of Information Security of ETRI from 1994 to 2005 as a senior engineer. He has been a visiting professor at the department of Computer and Information Sciences, University of Delaware, United States from 2011 to 2013. He is a professor at the department of Information Security of Mokpo National University currently. His research interests include security of vehicular communications and security of mobile communications.