情報保護學會論文誌
第20卷 第1號, 2010. 2

VANET를 위한 효율적인 서명 일괄 확인 시스템*

임 지 환,[1†] 오 희 국,[1] 김 상 진[2‡]
[1]한양대학교, [2]한국기술교육대학교

An Efficient Signature Batch Verification System for VANET*

Jihwan Lim,[1†] Heekuck Oh,[1] Sangjin Kim[2‡]
[1]Hanyang University, [2]Korea University of Technology and Education

요 약

VANET(Vehicular Ad hoc NETwork)에서 차량들은 일괄 확인(batch verification) 기법을 이용해 많은 수의 서명 메시지를 효율적으로 검증할 수 있다. 하지만 각 차량에서의 개별적인 일괄 확인은 네트워크 전체적으로 볼 때 불필요한 중복 검증을 발생시킨다. 이 문제를 해결하기 위해 RSU(Road Side Unit)가 노드를 대신해서 일괄 확인을 수행 할 수 있지만, 이 방법은 일괄 확인이 실패했을 경우 유효하지 않은 서명을 효율적으로 찾을 수 있는 방법이 추가적으로 필요하다. 본 논문에서는 분산 일괄 확인 시스템을 설계하기 위해 고려되어야하는 몇 가지 방법론에 대해서 분석하고 참여 차량이 작은 크기의 서명 집합을 분산해서 일괄 확인하는 효율적인 분산 일괄 확인 시스템을 제안한다. 제안하는 시스템에서 각 노드는 RSU에게 단순 일괄 확인 결과만을 보고하거나 식별한 유효하지 않은 서명들을 보고할 수 있으며 이를 수신한 RSU는 노드의 이 일괄 검증 결과 리포트를 이용하여 효율적으로 유효하지 않은 서명을 식별하여 배제할 수 있다.

ABSTRACT

In VANET (Vehicular Ad hoc NETwork), vehicles can efficiently verify a large number of signatures efficiently using batch verification techniques. However, batch verification performed independently in each vehicle raises many redundant verification cost. Although, an RSU (Road Side Unit) can perform the batch verification as a proxy to reduce this cost, it additionally requires an efficient method to identify invalid signatures when the batch verification fails. In this paper, we analyze several ways of constructing a distributed batch verification system, and propose an efficient distributed batch verification system in which participating vehicles perform batch verification in a distributive manner for a small size signature set. In our proposed system, each node can report the batch verification result or the identified invalid signatures list and the RSU who received these reports can identify the invalid signatures and efficiently exclude them.

Keywords: VANET, batch verification

I. 서 론

컴퓨팅과 통신기술의 비약적인 발전과, 전자부품 비용의 감소로 인한 다양한 정보기술의 발달은 ITS(Intelligent Transport System) 연구를 촉진 시키고 있으며 그 연구 결과들을 현실화하고 있다. VANET은 이러한 무선 통신 기능을 갖춘 지능형 차량들로 구성된 큰 규모의 애드혹(ad hoc) 망으로 정의할 수 있다. VANET의 응용은 크게 교통안전 서비스, 교통 정보 서비스, 위치 기반 서비스, 자동 비용 징수 서비스, 일반 통신 서비스 등으로 분류 할 수 있

접수일(2009년 7월 1일), 게재확정일(2009년 10월 23일)
* 본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT 연구센터 지원사업의 연구결과로 수행되었음. (NIPA-2009-C1090-0902-0035)
* 이 논문은 2008년도 정부(교육과학기술부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임. (KRF-2008-313-D01024)
† 주저자, jihwan.lim@gmail.com
‡ 교신저자, sangjin@kut.ac.kr

다. 특히 뒷 차량에게 위험신호를 전달하여 주는 충돌 회피(collision avoidance) 서비스나 신호가 없는 교차로나 고속도로 진입로 등에서 활용될 수 있는 협력운전(cooperative driving) 서비스와 같은 교통 안전 서비스는 향후 VANET의 중요한 응용 서비스가 될 것으로 간주되고 있다.

차량 간 교환되는 메시지를 통해 안전한 차량 운행 시스템을 구축하려고 할 때 가장 중요한 부분은 차량이 어떻게 수신한 정보를 신뢰할 수 있게 하는가 하는 문제이다. 악의적인 차량의 허위 정보는 차량 운행을 방해하고 운행의 안전성에 심각한 피해를 일을 킬 수 있기 때문이다. 따라서 VANET의 안전한 서비스 제공을 위해서는 악의적인 행동을 한 차량을 식별하거나 사고가 발생했을 때 책임자를 선별할 수 있는 방법이 필요하다. 현재 VANET에서는 메시지의 인증을 위해 각 차량이 생성하는 메시지를 전자 서명하여 전달하는 방법에 대한 연구가 진행되고 있으며, 사용자 프라이버시 보호를 위해 익명 식별자와 익명 인증서를 사용하여 제한적 익명성을 제공하는 조건부 익명 인증 시스템에 관한 연구가 활발하게 진행되고 있다[1-7].

이와 같은 조건부 익명 인증시스템이 VANET 환경에 도입되기 위해선 효과적인 전자서명 검증 시스템에 관한 연구가 선결되어야한다. 각 차량이 100 ms-300ms당 한 번의 안전 관련 메시지를 서명하여 전송한다고 할 때 다수의 차량에 의한 많은 양의 서명 메시지를 하나씩 검증하는 것은 차량의 입장에서도, RSU의 입장에서도 시간적 측면이나 연산량적 측면에서 매우 큰 부담이 되기 때문이다. 특히 다수의 서명을 빠른 시간에 검증하여 그 유효성을 확인할 수 있도록 하는 것은 안전한 VANET 환경을 구축하기위한 매우 중요한 요구사항이다. VANET에서 일괄 확인 기법은 이러한 요구사항을 해결할 수 있는 효과적인 방법이다[6,7].

일괄 확인을 이용하면 VANET의 차량들은 수신한 전자 서명들을 하나씩 개별적으로 확인하는 대신 서명 메시지 전체를 한 번에 검증 할 수 있다. 하지만 네트워크 측면에서 볼 때 많은 개별 차량의 일괄 확인은 중복된 서명 확인이 되어 매우 큰 자원 낭비가 된다. 이를 해소하기 위해 RSU가 참여 차량을 대신하여 자신의 구역에서 발생하는 모든 서명에 대한 일괄 확인을 수행하는 방법을 사용할 수 있으나 이 방법은 RSU가 일괄 확인에서 실패했을 때 배치(batch)된 서명 중 어떤 서명이 잘못되어서 일괄 확인이 실패했는지 효율적으로 확인할 수 있는 방법을 제공하지 못한다. 즉 N개의 메시지에 대한 일괄 확인 결과가 성공이라면 N개의 메시지가 모두 올바른 서명임을 빠르게 검증할 수 있는 장점이 있지만 일괄 확인이 실패했을 경우 다시 N개의 메시지를 모두 일일이 재검증 해보아야 유효하지 않은 서명을 구분해 낼 수 있다. 본 논문에서는 이와 같은 문제를 해결하기위해 참여하는 노드가 일부 메시지만을 선택해 분산된 형태로 일괄 확인을 수행하도록 하며 이를 통해 노드 및 RSU의 검증 비용 부담을 줄이고 일괄 확인 결과가 실패로 나왔을 때 유효하지 않는 서명을 효율적으로 추출하고 해당 공격자를 효과적으로 배제할 수 있는 분산 일괄 확인 시스템을 제안하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 일괄 확인의 개요 관련 연구 결과들을 살펴보고 3장에서 제안하는 일괄 확인 시스템을 소개한다. 4장에서 제안된 기법을 분석하고, 5장에서 결론을 맺는다.

II. 관련 연구

2.1 일괄 확인 기법

일괄 확인 기법은 여러 값을 개별적으로 확인하는 것이 아니라 모아서 한 번에 확인할 수 있도록 하는 것으로 전자서명의 경우 다수의 서명을 하나의 서명 확인 비용으로 검증할 수 있도록 한다. 일괄 확인의 개념은 1989년에 Fiat[8]에 의해서 처음 소개되었으며 이후 Beller 등[9]과 M'Raihi 등[10]에 의해서도 연구 되었지만 체계화되고 형식화된 것은 Bellare 등[11]에 의해서 이다.

Bellare 등은 법 지수승에 대한 일괄 확인 기법으로 단순 기법(naive method)에서부터 이를 확률적으로 확장하여 검증의 효율성 및 안전성을 높인 임의 부분집합 검사(random subset test), 작은 지수 활용 검사(small exponents test), 버켓 검사(bucket test)등을 제안한다. g가 군 $\mathbf{G}$의 생성자이고 $x_i \in \mathbf{Z_p}$, $y_i \in \mathbf{G}$인 n개의 법 지수승 쌍 $(x_1,y_1),\cdots,(x_n,y_n)$이 주어졌을 때, 시스템 보안 파라미터 l에 대한 각각의 테스트 방법은 다음과 같다.

- Naive method: $x=\sum_{i=1}^{n} x_i$, $y=\prod_{i=1}^{n} y_i$ 를 계산한 후 $y = g^x$가 만족하는지 확인한다.
- Random Subset Test: 다음 과정을 l번 반복
  - $S \subset_R \{1,2,\cdots,n\}$를 선택하고

- $x=\sum_{i\in S}x_i$, $y=\prod_{i\in S}y_i$를 계산한 후 $y=g^x$를 만족하는지 확인한다.

• Small Exponents Test

- s_1부터 s_n까지를 $\{0,1\}^l$에서 랜덤하게 선택하고,
- $x=\sum_{i=1}^{n}s_ix_i$, $y=\prod_{i=1}^{n}y_i^{s_i}$를 계산한 후 $y=g^x$를 만족하는지 확인

• Bucket Test: $M=2^m(m\geq 2)$인 M에 대해서 다음 과정을 $\lceil l/(m-1)\rceil$번 수행

- 각 $i=1,\cdots,n$에 대해 $t_i\in\{1,\cdots,M\}$를 랜덤하게 선택하고,
- 각 $j=1,\cdots,M$에 대해 $B_j=\{i:t_i=j\}$를 계산하여
- $c_j=\sum_{i\in B_j}x_i$와 $d_j=\prod_{i\in B_j}y^i$를 생성하고
- 모든 $(c_1,d_1),\cdots,(c_M,d_M)$에 대해 Small Exponents Test를 수행한다.

Bellare 등은 이 논문에서 DSA 서명을 어떻게 일괄 확인할 수 있는지 보였고 일괄 확인과 스크니링(screening)의 차이점에 대해서 명확히 구분하였다.

Camenisch 등[12]은 이전에 발표된 여러 서명 알고리즘에 대한 일괄 확인 기법을 소개하였고 단일 서명자가 아닌 다수의 서명자에 대한 일괄 확인 기법을 처음으로 제안한다. Camenisch 등이 제안한 다수 서명자에 대한 일괄 확인 기법은 다음과 같다.

n개의 메시지 $m_1,\cdots,m_n$에 대한 n개의 서명 $\sigma_1,\cdots,\sigma_n$과 시스템 상수 ϕ가 주어졌을 때 일괄 확인 알고리즘은 먼저,

• 모든 서명이 군 **G**의 원소인지 확인하고 $X=((m_1,\sigma_1),\cdots,(m_n,\sigma_n))$, $w_i=H_3(m_i\|\phi)$로 설정한다.

• $s_i\in_R\{0,1\}^l$를 선택하고 $\hat{e}(a,\prod_{i=1}^{n}X_i^{s_i})\hat{e}(b,\prod_{i=1}^{n}X_i^{ws_i})=\hat{e}(\prod_{i=1}^{n}\sigma_i^{s_i},g)$ 인지를 검사한다.

여기서 $a=H_1(\phi)$, $b=H_2(\phi)$이고 $H_1,H_2:\Phi\rightarrow\mathbf{G}$인 해쉬함수이고 H_3는 메시지 공간 $M=\{0,1\}^*$에 대해 $H_3:M\times\Phi\rightarrow\mathbf{Z}_q$를 만족하는 해쉬함수이다.

2.2 VANET에서의 일괄 확인

Zhang 등[6]은 VANET에서 발생하는 많은 수의 서명 메시지를 효율적으로 검증하기위해 처음으로 일괄 확인 기법을 VANET에 적용하고자 하였다. Zhang 등은 많은 수의 서명 메시지를 각 차량이 확인해야하는 문제점과 개별 노드의 서명 중복 검증 문제를 개선하기위해 RSU의 도움을 받는 기법을 제안하였다. Zhang 등이 제안하는 기법에서는 RSU와 각 차량이 키 확립 프로토콜을 통해 세션키를 공유하고 있으며 차량과 RSU는 같은 메시지 집합을 수집할 수 있다고 가정한다. 따라서 RSU는 검증해야할 전체 서명 메시지에 대한 서명 확인을 개별 차량에 대신해서 일괄 확인해 줄 수 있다. 하지만 이 방법은 교통량이 많을 경우에 RSU에 많은 부하를 발생시키게 되고 일괄 확인 결과가 실패로 나올 경우 어떤 서명 때문에 일괄 확인이 실패했는지 확인할 수 있는 방법을 제공하지 못한다. 유효하지 않는 서명을 추출하기 위해서는 RSU가 모든 서명에 대해 개별적으로 서명 검증을 시도해 보아야하고 이는 RSU에게 심각한 부하를 발생 시키며 긴 검증 시간 지연을 발생 시킨다.

Zhang 등은 같은 논문에서 RSU의 도움 없이 각 차량이 확률적으로 서명을 검증하는 COMET이라는 협력 인증시스템을 제안한다. 이 기법은 일괄 확인 기법은 아니지만 노드가 확률적으로 메시지를 선택하여 검증하는 방법으로 서명 검증 부담을 줄이고자 했다. 서로 한 홉 이웃인 차량 V_i, V_j, V_k가 있을 때 COMET의 동작은 다음과 같다.

- V_i가 V_j가 전송한 메시지와 서명쌍 $<M_j\|\sigma_j>$를 수신하게 되면, V_i는 시스템에서 정해진 확률 p로 이 메시지를 검증할지 여부를 결정한다.
- V_i가 $<M_j\|\sigma_j>$를 검증하기로 결정하였고 그 검증결과가 올바르다면 아무 메시지도 전송하지 않고 가만히 있는다.
- 만약 서명의 검증 결과가 잘못 되었다면 V_i는 이웃 노드로 서명 검증 결과가 틀렸음을 방송한다.
- V_j의 서명이 틀렸음을 보고 받은 이웃 노드는 $<M_j\|\sigma_j>$를 재검증하여 이를 확인하고, 만약 일정 시간동안 아무런 보고를 받지 못했다면 이웃 노드들은 전송된 모든 서명이 올바른 서명이라고 간주한다.

COMET의 아이디어는 매우 간단하다. 개별 노드가 주어진 확률 p로 도착하는 메시지를 검증할 경우, 어떤 메시지가 서명을 생성한 차량의 주행방향 앞쪽과 뒤쪽에 있는 2대 이상의 차량에 의해서 매우 높은 확률로 검증 된다는 것이다. n대의 차량이 있을 때, 확률 p에 대해 이 확률 P는 $P=1+(1-p)^n-2(1-\frac{p}{2})^n$으로 계산되고, $n=120$, $p=15\%$인 경우를 예로 들면, 어떤 메시지는 2대의 참여 차량에 의해 약 99.98%로

검증된다는 것이다.

III. 제안하는 일괄 확인 시스템

3.1 시스템 가정

제안하는 VANET 분산 일괄 확인 시스템은 일괄 확인에 포함된 유효하지 않은 서명을 효과적으로 찾아내고 배제하는데 목적이 있다. 여기서 유효하지 않은 서명이란 시스템에서 사용하는 서명 알고리즘을 정상적으로 통과하지 못하는 서명으로 공격자가 재전송한 유효기간이 만료된 서명이나 메시지 포맷만 맞춘 쓰레기(garbage) 값이 될 수 있다. 본 논문에서는 다음의 시스템 환경을 가정한다.

- **시스템 가정 1** - 차량은 기 제안된 다양한 종류의 익명 식별자 모델을 통해 익명 공개키 쌍을 생성할 수 있다. 차량은 익명 공개키 쌍을 통해 원하는 메시지를 서명하여 전송할 수 있으며 이를 수신한 차량은 해당 서명을 검증하여 메시지를 신뢰할 지 여부를 판단한다. 여기서 가정하는 익명 공개키 쌍은 조건부 익명성이 보장되는 차량의 익명 식별자를 기반으로 생성되었다고 가정하며, 따라서 차량의 익명 식별자는 필요시 차량의 실제 신원으로 복원될 수 있다.
- **시스템 가정 2** - 네트워크에 참여하는 차량 및 RSU는 100ms~300ms 마다 안전 운행에 관련된 비콘(beacon) 주기적으로 생성하며 이를 서명하여 방송한다. VANET에서는 GPS 등으로 정확한 시간동기화가 가능하므로 비콘에는 비콘이 생성되어 방송된 시간의 타임스탬프(timestamp)와 차량의 익명 아이디가 포함된다. 따라서 RSU와 참여 차량은 타임스탬프를 이용하여 비콘을 필터링할 수 있으며 유효하지 않은 타임스탬프를 가진 비콘은 검증 대상 메시지에서 필터링 된다.
- **시스템 가정 3** - 같은 RSU 영역에 있는 차량들을 하나의 그룹으로 묶는다고 할 때 그룹의 크기는 그룹원의 참여 이탈로 인해 동적으로 변할 수 있으나 일괄 확인이 이루어지는 특정 시점의 그룹은 정적인 그룹으로 간주한다. 또한 같은 그룹의 차량들은 모두 같은 메시지를 수신할 수 있으며 따라서 일괄 확인을 위해 일정 시간동안 수집된 전체 메시지 집합도 모두 같다고 가정한다.
- **시스템 가정 4** - RSU는 차량의 정보(차량의 아이디, 익명 아이디 등)를 이용하여 자신의 영역에 있는 차량의 순번(*NO: Node Order*)을 결정할 수 있다.
- **시스템 가정 5** - RSU나 참여 차량이 전송하는 메시지는 무선 채널을 이용하기 때문에 전송된 시점 이후 메시지를 수정하거나 조작 할 수 없다고 가정한다. 즉 공격자가 정상적인 메시지를 확보한 후, 원 메시지의 확산을 차단하고 조작한 메시지를 네트워크에 다시 유입시키는 행위는 무선 채널을 이용하는 VANET 환경에서 불가능하다고 가정한다.

3.2 제안하는 VANET 일괄 확인 시스템

제안하는 VANET 일괄 확인 시스템은 다음과 같은 단계로 진행된다.

단계 1. RSU는 자신의 관할 지역에서 발생하는 차량의 메시지들을 일괄 검증하기 위해 일정 시간동안 수집한다.

단계 2. RSU는 수집된 메시지에 대한 일괄 확인을 수행한다.

2-1 일괄 확인 결과가 성공이라면 수집된 모든 메시지가 올바른 메시지이므로 개별 서명의 검증은 필요 없다.

2-2 일괄 확인의 결과가 실패로 나왔다면 RSU는 개별적 서명 검증을 통해 유효하지 않은 서명을 추출해야 한다. RSU는 스스로 개별적 서명 검증을 시도하는 대신 자신의 지역에 존재하는 노드들에게 분산 일괄 확인을 요청한다.

단계 3. RSU로부터 분산 일괄 확인 요청을 수신한 참여 노드들은 전체 메시지 중 일정수의 메시지를 선택하여 분산 일괄 확인을 수행한다.

단계 4. 개별 노드는 일괄 확인 결과에 대한 리포트를 생성하고 서명하여 RSU에게 전송한다.

단계 5. 개별 노드의 일괄 확인 결과 리포트를 수신한 RSU는 리포트 생성자의 서명을 검증하고 이후 보고된 리포트 내용을 재검증한다.

단계 6. 검증 결과를 자신의 영역에 있는 차량 및 이웃 RSU에게 전파한다.

위의 과정에서 가장 핵심적인 부분은 단계 3의 '개별 참여 노드에게 어떻게 메시지를 분배하는가?' 하는

문제와 단계 4에서 일괄 검증이 실패 하였을 경우 '실패한 일괄 확인 메시지에 대한 개별 검증을 누가 수행하는가?' 하는 문제이다. 먼저 단계 3의 메시지 분배에서는 개별 노드가 독립적으로 랜덤한 메시지 집합을 선택하는 방법과 사전 동의된 메시지 선택 알고리즘을 이용하여 노드별로 지정된 메시지를 선택하는 방법을 고려해 볼 수 있다. 단계 4의 리포트 생성 방법은 실패한 일괄 확인에 대한 메시지 재검증을 노드 자신이 수행해서 개별 검증 결과를 리포트로 생성하는 방법과 일괄확인의 성공 여부와 관련 메시지 번호만을 기록하여 리포트를 생성하는 방법을 고려해 볼 수 있다.

3.2.1 분산 일괄 확인을 위한 메시지 분배

이 절에서는 RSU가 일괄 확인에 실패하여 참여 차량에게 분산 일괄 확인을 요청할 경우, 각 차량이 일괄 확인해야하는 메시지를 어떤 방법으로 분배받을 수 있는가에 대해서 서술한다. 먼저 메시지 분배에 있어 가장 간단하면서도 직관적으로 생각할 수 있는 방법은 개별 노드들이 독립적으로 메시지를 선택하도록 하는 확률적인 분배 방법이다. 이 경우 참여 차량은 메시지 선택에 대한 사전 동의가 필요하지 않으며 독립적으로 메시지를 선택할 수 있다는 장점이 있다. 하지만 개별 차량의 독립적 메시지 선택으로 인해 특정 메시지가 여러 대 차량에 중복되어 선택될 수도, 어떤 차량에게도 선택되지 않을 수도 있다. 따라서 메시지의 확률적 분배에서는 서명 검증에 누락되는 메시지가 없도록 고려되어야 하며 이를 위해 전체 메시지가 임의 참여 차량에게 적어도 한번 이상은 선택되어야 한다는 추가적인 요구사항을 만족 시켜야 한다. 이 문제는 다음과 같이 확률적으로 풀어볼 수 있다.

- N개의 노드에 의해서 M개의 메시지가 모두 검증되어야 한다. 이때 N개의 노드는 각각 독립적으로 K개씩의 메시지를 랜덤하게 선택한다. M개의 메시지가 모두 검증될 확률 P_1은?

이 문제의 답은 다음과 같이 계산된다.

$$P_1 = \left(1-\left(1-\left(\frac{K}{M}\right)\right)^N\right)^M \qquad (1)$$

예를 들어 30개 차량이 각각 300ms 당 한 번의 비콘을 생성하여 1초 동안 약 100개의 서명 메시지를 생성했다고 했을 때(M:100, N:30) 분산 일괄 확인을 위해서 노드는 30개(K:30) 이상의 메시지를 선택해야 약 99.80%의 확률로 모든 메시지가 임의 노드에 의해 한번 이상은 검증된다. 따라서 위 검증 성공 확률 P_1이 시스템 변수로 주어졌을 때 차량이 선택해야할 메시지 수 K는 다음과 같이 결정된다.

$$K = M\left(1-\left(1-P_1^{\frac{1}{M}}\right)^{\frac{1}{N}}\right) \qquad (2)$$

시스템 환경에 따라 P_1을 조정할 수 있다면 개별 차량의 부담이 될 수 있는 K의 크기 역시 조절할 수 있다. 즉 M:100, N:30일 때 P_1이 0.9999라면 K는 약 40이 되고 0.9900이라면 K는 약 26까지 줄어든다.

위 확률적 메시지 선택 방법을 이용하면 개별 차량이 선택해야하는 메시지수가 많이 줄어들기는 하지만 효과적인 일괄 확인과 유효하지 않은 서명의 추출을 위해서 K를 더 줄일 필요가 있다. K를 더 줄이기 위해선 참여 차량이 다른 차량이 이미 선택한 메시지는 다시 선택하지 않게 하여 전체 메시지가 고르게 선택될 수 있도록 해야 한다. 이것은 각 차량이 확인해야 하는 메시지를 시스템에서 지정해주면 쉽게 해결되지만 RSU가 개별 통신을 통해 참여 차량에게 메시지 집합을 넘겨주는 방법은 시간적으로나 통신량 측면에서 매우 비효율적이다.

앞서 언급한 바와 같이 개별 차량이 RSU와 통신 없이 스스로 약속된 메시지를 선택하기 위해서는 메시지를 선택해야하는 차량들 사이에 사전에 약속된 규칙이 있어야 한다. 한 가지 방법으로 같은 RSU의 관할 지역에 있는 차량들이 해당 지역에 있는 전체 차량들 중 자신이 몇 번째 차량인지 그 순번 NO를 결정할 수 있다면, RSU와 참여 차량이 메시지 일련 번호(sequence number)나 타임스탬프와 같은 공통된 기준으로 메시지를 정렬할 수 있는 환경에서 쉽게 개별 차량이 선택해야할 메시지를 지정해 줄 수 있다. 즉 K개의 메시지를 선택한다고 할 때, 개별 차량은 선택할 메시지의 시작 번호를 자신의 순번 NO와 선택해야할 메시지 수의 곱, $M_{start} = NO \times K$로 계산하여 그 뒤로부터 K개의 메시지를 선택하면 각 차량은 중복 없이 전체 M개의 메시지를 분할해서 나눠 가질 수 있다.

하지만 메시지 검증에 참여할 전체 노드 집합은 VANET의 특성상 그룹원의 참여와 이탈이 빈번한 동적 그룹으로 구성된다. 이러한 동적 그룹에서 개별 노드가 자신의 순번을 RSU와 통신 없이 알게 하는 방법으로 본 논문에서는 다음과 같은 방법을 사용한다.

1. 개별 차량은 새로운 RSU 지역에 들어가게 되면 RSU와 상호인증하게 되는데 이때 RSU로부터 자신의 순번을 함께 확인 받는다.
2. 특정 차량이 자신의 영역을 벗어나게 되면 RSU는 주기적으로 전송하고 있는 비콘 신호에 해당 차량의 순번을 넣어 해당 차량이 그룹에서 이탈했음을 방송한다.
3. RSU의 비콘을 수신한 개별 차량은 다음과 같이 자신의 순번을 개별적으로 조절한다.
   - 수신한 비콘의 순번이 자신의 순번보다 작다면 자신의 순번을 하나 감소시킨다.
   - 수신한 비콘의 순번이 자신의 순번보다 크다면 자신의 순번을 변경하지 않는다.

이와 같은 방법으로 노드가 자신이 확인해야할 메시지를 다른 차량과 중복 없이 결정적으로 선택하면 K를 최대 $\lceil \frac{M}{N} \rceil$ 까지 줄일 수 있다는 장점이 있고 이때 유효하지 않은 서명의 추출 효율성도 최대로 만들 수 있다.

3.2.2 성공 리포트 vs. 실패 리포트

개별 노드는 일괄 확인 결과에 따라 일괄 확인 성공 리포트나 일괄 확인 실패 리포트를 생성할 수 있다. 개별 노드는 생성한 리포트가 성공 리포트인지 실패 리포트인지에 상관없이 RSU에게 무조건 리포트를 전송할 수 있겠지만 RSU는 수신한 리포트의 서명을 확인하여야하기 때문에 많은 수의 리포트 전송은 메시지 전송량 측면에서나 서명의 검증 비용, 시간적 측면에서나 비효율적이다. 특히 결정적 메시지 분배의 경우 RSU는 특정 노드가 어떤 메시지를 선택하여 일괄 확인하였는지 알 수 있으므로 성공 리포트를 생성한 차량들만 또는 실패 리포트를 생성한 차량들만 RSU에게 리포트를 전송하도록 하여도 RSU는 개별 노드의 분산 일괄 확인 결과를 확인 할 수 있다. 실패 리포트만 전송하도록 약속하였다면 RSU는 리포트를 전송하지 않은 차량들을 모두 일괄 확인을 성공한 것으로 간주하면 되고 반대의 경우는 모두 실패한 것으로 간주하면 되기 때문이다. 일반적으로 유효하지 않은 서명의 수가 유효한 서명의 수보다 적은 수라고 볼 수 있기 때문에 유효하지 않은 서명을 포함한 일괄 확인을 수행한 노드의 수, 즉 일괄 확인 실패 리포트를 생성하는 노드 수가 상대적으로 적다고 볼 수 있다. 따라서 결정적 메시지 분배 방법일 경우 일괄 확인 실패 리포트를 생성한 노드들만 RSU로 리포트를 전송하는 것이 효율적이다.

하지만 확률적 메시지 분배 방법을 사용할 경우 일괄 확인 성공 리포트를 생성한 노드들이 RSU로 리포트를 보고하는 것이 효율적이다. 앞선 예에서처럼 30대의 차량에서 생성한 약 100개의 메시지 중 5개의 유효하지 않은 서명이 있으며 개별 노드가 적어도 30개 이상의 메시지를 선택해야 하는 환경이라면 임의 노드가 선택한 30개의 메시지 중 유효하지 않은 서명이 1개 이상 포함될 확률, 즉 노드가 일괄 확인 실패 리포트를 생성할 확률은 약 84% 이상이다. 따라서 실패 리포트를 생성한 차량이 RSU에게 리포트를 전송한다면 확률적으로 30대의 차량중 약 25대의 차량이 실패 리포트를 서명하여 전송하게 되지만 성공 리포트를 생성한 차량은 약 5대 정도가 되기 때문에 성공 리포트를 전송하는 것이 전송 비용 및 계산 비용 측면에서 효율적으로 일괄 확인 결과를 전달할 수 있다.

3.2.3 유효하지 않은 서명의 추출

참여 차량이 분산 일괄 확인에서 실패했을 경우 앞서 언급한 것처럼 실패한 일괄 확인 메시지에 대한 개별 검증을 누가 수행하는가에 대해서 고려해 보아야한다. 이 장에서는 RSU가 노드로 부터의 리포트를 수신하여 적은 수의 유효하지 않은 서명 후보(ISC: Invalid Signature Candidate) 리스트를 생성하고 이 리스트에 포함된 서명을 하나씩 재검증하는 방법과 분산 일괄 확인에서 실패한 개별 노드가 자신이 선택한 메시지를 각자 재검증하고 재검증 결과를 실패 리포트로 RSU에게 전송하는 방법에 대해 고려해본다. 따라서 앞서 서술한 메시지 분배 방법과 리포팅 방법을 고려하면 다음의 4가지 경우에 대해 생각해 보아야한다.

- Case 1: 결정적 메시지 분배 - 개별 노드에 의한 재검증 - 실패 리포팅
- Case 2: 결정적 메시지 분배 - 실패 리포팅 - RSU에 의한 재검증
- Case 3: 확률적 메시지 분배 - 개별 노드에 의한 재검증 - 실패 리포팅
- Case 4: 확률적 메시지 분배 - 성공 리포팅 - RSU에 의한 재검증

Case 1의 경우, 즉 결정적으로 메시지를 분배하여 개별 차량이 각각 $\lceil \frac{M}{N} \rceil$ 개 씩의 메시지를 선택한 경

우, 실패 한 분산 일괄 확인에 대해 각 차량은 $\left\lceil \frac{M}{N} \right\rceil$ 개의 서명만 재검증하여 리포트를 생성하면 된다. 즉 M: 100, N: 30인 경우 개별 차량은 4개의 서명만 재검증해보면 유효하지 않은 서명을 추출해 낼 수 있다. Case 2의 경우 즉, RSU가 각 차량으로 부터의 리포트를 수신하여 개별 메시지를 재검증하는 경우, RSU는 다음과 같은 방법으로 전체 메시지 크기에 비해 상대적으로 적은 수의 메시지를 포함한 ISC 리스트를 생성할 수 있다.

RSU는 비어있는 ISC 리스트를 만든다.

노드로부터 실패 리포트를 수신한 RSU는 실패 리포트의 서명을 확인한다.

리포트의 서명이 이상 없다면 실패 리포트를 전송한 노드가 일괄 확인했던 서명들을 ISC 리스트에 추가한다.

위와 같은 방법으로 생성된 ISC 리스트는 유효하지 않은 서명의 수를 S_F라 할 때 약 $\left\lceil \frac{M}{N} \right\rceil \times S_F$개의 서명 메시지를 포함하고 있기 때문에 RSU는 $\left\lceil \frac{M}{N} \right\rceil \times S_F$ 개의 메시지만 재검증하면 모든 유효하지 않은 서명을 추출할 수 있다.

Case 3의 경우 3.1.1절에서 언급한 것처럼 대다수의 차량이 분산 일괄 확인 실패 리포트를 생성하기 때문에 위와 같은 실패 리포트에 의한 ISC 리스트를 생성할 수 없다. 따라서 개별 차량이 자신이 선택한 메시지를 재검증하는 방법을 사용해야 하나 결정적 메시지 분배와 달리 차량이 선택해야하는 메시지 수가 많기 때문에 각 차량에서 유효하지 않은 서명을 추출하여 RSU로 보고하는 것은 많은 수의 서명 중복 검증을 야기한다. Case 4의 경우 RSU는 개별 차량의 성공 리포트를 수신하여 다음과 같은 방법으로 ISC 리스트를 생성할 수 있다.

- RSU는 M개의 전체 메시지를 ISC 리스트에 포함시킨다.
- 노드로부터 성공 리포트를 수신한 RSU는 성공 리포트의 서명을 확인한다.
- 확인된 성공 리포트에 포함된 서명들을 ISC 리스트에서 제외시킨다.

성공 리포트를 전송한 차량이 R_S 대이고 각각 K개씩의 메시지를 분산 일괄 확인 했을 경우, 임의 두 차량의 메시지 중복 확률 d에 대해 RSU는 약 $R_S \times K \times d$개의 올바른 서명을 추출할 수 있다. 따라서 성공 리포트로부터 생성할 수 있는 ISC 리스트는 $M-(R_S \times K \times d)$개의 유효하지 않은 서명 후보를 포함하고 있기 때문에 RSU는 $M-(R_S \times K \times d)$개의 메시지만 재검증하면 모든 유효하지 않은 서명을 추출할 수 있다.

3.2.4 공격자와 페널티 시스템

제안하는 시스템에서는 유효하지 않는 서명을 네트워크로 유입시키는 공격자를 외부 공격자로, 제안하는 분산 일괄 확인 시스템에 참여하여 서명 검증과 검증 결과의 리포팅을 수행하지만 경우에 따라 비정상정인 행위를 할 수 있는 공격자를 내부 공격자로 정의한다. 외부 공격자는 정상적인 VANET 익명 아이디 시스템을 사용하지 않은 채 쓰레기 메시지를 네트워크에 유입 시킬 수 있으며, 단순히 다른 차량의 아이디를 포함한 과거의 메시지를 재전송 할 수도 있기 때문에 이 공격자를 찾아내고 제재하는 것은 본 논문의 범위를 벗어난다. 하지만 내부 공격자의 경우 제안하는 분산 일괄 확인 과정에 참여하기 위해 정상적인 VANET 익명 아이디를 사용하여야 하며 따라서 내부 공격자가 시스템의 성능 저하를 목적으로 이상행동을 하면 시스템은 이를 식별하고 배제할 수 있어야 한다. 제안하는 시스템에서 고려할 수 있는 내부 공격자의 유형은 다음의 2가지로 분류할 수 있다.

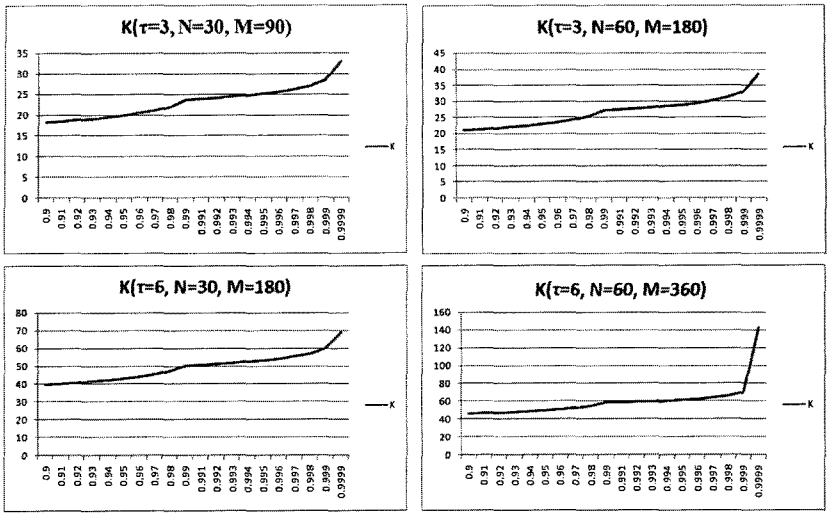
- 이기적인 노드: 자신에게 이득이 될 때에만 시스템 룰에 참여하며 그렇지 않은 경우에는 시스템 룰을 따르지 않아 서명 검증하지 않거나 검증 결과 리포팅을 하지 않는 노드
- 악의적인 노드: 시스템의 서비스를 저해하기 위한 목적으로 서명검증 결과를 거짓으로 보고하는 노드

이 내부 공격자는 3.2절의 단계 5 과정을 다음과 같이 확장하여 효과적으로 식별할 수 있다.

단계 5. 개별 노드의 일괄 확인 결과 리포트를 수신한 RSU는 리포트 생성자의 서명을 검증하고 노드의 개별 검증 결과 또는 ISC 리스트로부터 추출한 유효하지 않은 서명들을 제외한 나머지 서명들을 다시 일괄 확인한다.

5-1. 일괄 확인 결과가 성공이라면 내부 공격자가 없다고 판단한다.

5-2. 일괄 확인 결과가 실패라면 일괄 확인한 메시지를 하나씩 재검증해 거짓 리포트를 보고하였거나 보고해야할 리포트를 전송



〔그림 1〕 τ와 차량 수 N 의 변화에 따른 K 변화

하지 않은 차량을 추출한다.

3.2.3절의 Case 3, 4의 경우 리포트를 전송하지 않은 노드가 어떤 메시지를 선택했는지 알 수 없기 때문에 내부 공격자를 식별해 낼 수 없다. 하지만 Case 1, 2의 경우 RSU는 각 차량이 어떤 메시지를 검증하는지 알 수 있기 때문에 이기적으로 리포팅을 하지 않는 노드나 거짓 리포팅을 한 노드를 위 5-2의 과정에서 찾아낼 수 있다.

일반적으로 VANET은 행정부 소속의 차량등록소나 사법부 소속의 공개키 등록/철회 기관을 가정하며 이러한 이유로 MANET(Mobile Ad-hoc Network)과는 달리 구조화된 네트워크의 구성이 가능하다. 차량은 네트워크에 참여하기 전에 차량등록소에 등록되어야하며 이때 앞으로 사용할 공개키와 인증서, 익명 식별자 등을 발급받을 수 있다. 이러한 특성 때문에 VANET에서는 현재 교통법규 위반 차량에 대한 무인 카메라 감시 적발 및 벌점 제도처럼 차량의 악의적인 행동에 대한 강한 페널티 시스템을 도입할 수 있다. 따라서 위 단계 5에서 식별된 내부 공격자에게 페널티 시스템을 도입한다면 내부 공격자의 비정상적인 행동을 억제할 수 있고 궁극적으로는 식별된 내부 공격자를 시스템에서 완전히 배제할 수 있다.

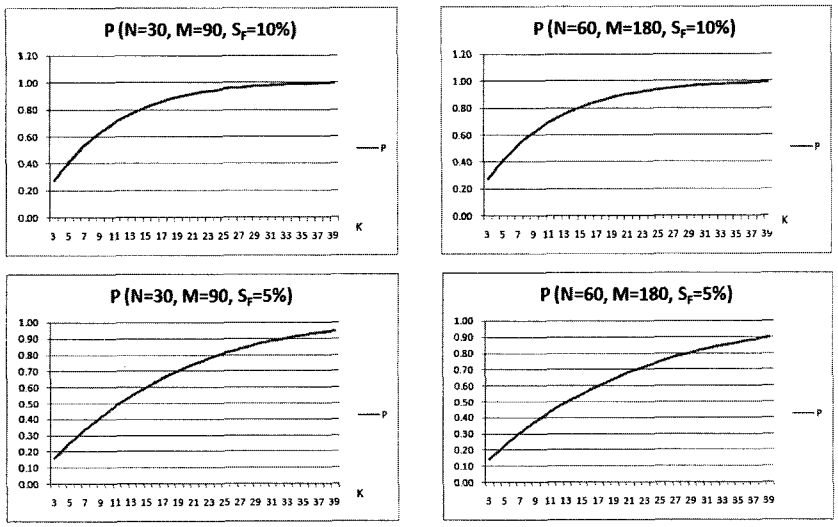
IV. 분석

4.1 메시지 분배와 유효하지 않은 서명 후보 리스트 생성

이 장에서는 제안하는 일괄 확인 시스템의 정확성과 효율성에 대해 분석한다. 서명 메시지의 확률적 분배의 경우 전체 메시지가 1번 이상 검증되어야 하기 때문에 결정적 분배 방식에 비해 노드가 선택해야 하는 메시지 수가 많아지게 된다. 전체 메시지 수는 참여 노드의 수와 비례관계에 있으므로 임의 비례 상수 τ에 대해 $M=\tau N$의 관계가 있다. 여기서 τ는 일괄 확인을 얼마나 자주하느냐에 따라 결정되는 시스템 파라미터다. 개별 노드가 300ms당 1회씩 비콘 메시지를 전송한다고 했을 때 제안하는 시스템에서는 안전한 시스템의 운용을 위해 1초($\tau=3$)~2초($\tau=6$) 정도 메시지를 수집하고 이를 일괄 확인하는 환경을 가정한다. 따라서 단일 RSU 지역에서 메시지 인증에 참여하는 노드수가 N개이고 노드의 검증확률 P_1이 식 1과 같이 주어질 때 식 2의 노드가 선택해야할 메시지 수 K는 〔그림 1〕과 같이 분석된다.

〔그림 1〕에서 보듯이 모든 메시지가 검증될 확률을 0.9999로 고정하면 참여 노드수가 30개일 때 각 노드는 90개, 180개의 메시지 중에서 약 33개, 68개의 메시지를 선택하여야한다. 이때 노드가 일괄 확인에 실패할 확률, 즉 K개의 메시지중 적어도 하나 이상의 유효하지 않은 서명을 포함할 확률 P_2는 식 3과 같이 계산되고 〔그림 2〕에서 확인 할 수 있다.

$$P_2 = 1 - \frac{\binom{M-S_F}{K}}{\binom{M}{K}} \qquad (3)$$



〔그림 2〕 개별 차량의 부분 일괄 확인이 실패하여 RSU로 보고할 확률

〔그림 2〕에서 보는 것처럼 노드가 선택하는 메시지 수 K가 일정 수 이상으로 커지게 되면 각 노드는 유효하지 않은 서명을 포함한 일괄 확인을 수행하게 되어 거의 모든 노드가 RSU로 일괄 확인 실패를 보고하게 된다. 이렇게 되면 ISC 리스트의 크기가 원 메시지와 비슷하게 되고 3.2.2절에서 언급한 것처럼 RSU가 대신해서 유효하지 않은 서명을 추출하기 힘들게 된다. 하지만 결정적 메시지 분배 방식을 사용하면 노드가 선택해야할 메시지를 최소한으로 줄일 수 있어 노드 스스로 실패한 일괄 확인 서명들을 재검증할 수도 있고, 일괄 확인 리포트만 전송한다고 해도 RSU는 작은 크기의 ISC 리스트를 생성할 수 있다. M개의 메시지가 참여 노드에 나뉘어 모두 할당되면 되기 때문에 $K=\left\lceil \frac{M}{N} \right\rceil$개의 메시지만 각 노드에 중복 없이 할당하면 전체 메시지는 1회 이상 씩 검증되게 된다. 노드수가 30개, 60개 일 때 90개, 180개의 메시지에 대해서 각 노드가 3개, 6개의 메시지만 선택하여 검증하면 모든 서명을 확인할 수 있으며 〔그림 2〕에서처럼 $K=3(M=90, S_F=9)$일 때 개별 노드는 27%로 일괄 확인 실패를 보고하게 된다. 따라서 전체 90개의 서명이 30개의 노드에 의해 검증된다면 약 8대의 차량에서 일괄 확인이 실패하게 되는 것이며 개별 차량에서 유효하지 않은 서명을 추출할 시 각 차량은 3개의 서명에 대해 재검증을 수행해보면 되는 것이며, RSU는 개별 노드의 보고를 통해 약 24개 정도의 ISC 리스트를 생성할 수 있게 된다.

4.2 유효하지 않은 서명 추출 비용

본 절에서는 RSU, 참여 노드, 시스템 레벨에서 제안하는 일괄 확인 기법을 이용한 경우와 그렇지 않은 경우의 유효하지 않은 서명 추출 비용을 계산적 측면과 시간적 측면에서 분석한다. RSU가 개별 노드에게 전송하는 분산 일괄 확인 요청 메시지나 개별 노드가 RSU에게 전송하는 성공/실패 리포트는 VANET 환경에서 주기적으로 전송되는 비콘을 이용하여 전송될 수 있기 때문에 별도의 통신 비용이나 통신 지연시간이 발생하지 않는 것으로 가정한다. 정량적 비용 분석을 위해 다음의 표기법을 사용한다.

- M: 일괄 확인될 전체 메시지 수
- N: 분산 일괄 확인에 참여할 전체 차량 수
- K: 분산 일괄 확인을 위해 개별 노드가 선택하는 메시지의 수, 결정적 메시지 분배의 경우 $K=\left\lceil \frac{M}{N} \right\rceil$, 확률적 메시지 분배 방법의 경우 3.1.1절의 식 (2)에 의해서 결정
- S_F: 유효하지 않은 서명의 수
- C_{verify}: 1개의 서명을 검증하는 공개키 연산 비용
- T_{verify}: 1개의 서명을 검증하는 공개키 연산 시간
- $C_{aggre}(m)$: m개의 서명을 하나의 일괄 확인 인스턴스로 결합하는 비용
- $T_{aggre}(m)$: m개의 서명을 하나의 일괄 확인 인스턴스로 결합하는 시간
- $C_{batch}(m)$: m개의 서명을 일괄 확인하는 비용,

$C_{aggre}(m)+C_{verify}$

- $T_{batch}(m)$: m개의 서명을 일괄 확인하는 시간, $T_{aggre}(m)+T_{verify}$
- R_F: 실패 리포트를 생성한 노드 수

위 표기법을 이용하여 RSU가 모든 차량을 대신해서 일괄 확인 및 유효하지 않은 서명의 추출을 수행하는 경우와 개별 차량이 독립적으로 일괄 확인 및 유효하지 않은 서명의 추출을 수행하는 경우, 그리고 제안하는 시스템의 결정적 메시지 분배 방법을 사용하는 2가지 분산 일괄 확인 기법에 대해서 비용적, 시간적 효율성을 분석한다. 확률적 메시지 분배 방법의 경우 결정적 메시지 분배 방법에 비해 비효율적임을 직관적으로 분석할 수 있으므로 분석에서 제외한다.

• RSU에 의한 일괄 확인 및 유효하지 않은 서명 추출

RSU에 의한 일괄 확인 과정은 간단하다. 전체 M개의 메시지를 일괄 확인해본 후 실패한 일괄 확인에 대해 개별 메시지를 하나씩 재검증해보면 된다. 따라서 검증 비용과 검증 시간은 다음과 같이 분석할 수 있다.

〔검증 비용 - Naive〕

- RSU: $C_{batch}(M) + M \times C_{verify}$
- Node: None
- System: $C_{batch}(M) + M \times C_{verify}$

〔검증 시간 - Naive〕

- System: $T_{batch}(M) + M \times T_{verify}$

위 RSU에 의한 단순 일괄 확인 방법은 제안하는 시스템에서 사용하는 분산 일괄확인 기법을 변형해서 적용할 수 있다. 즉 RSU는 M개의 전체 메시지 집합을 α개의 배타적인 부분 집합으로 나누고 이 부분 집합에 대한 부분 일괄 확인 인스턴스를 생성제안하보다 효율체 메시지 집합을 할 수 있다. S_F개의 유효하지 않은 서명이 모두 다른 부분 집합에 포함되는 최악집합에 포검증 비용과 검증 시간은 다음과 같이 분석할 수 있다.

〔검증 비용 - Subset〕

- RSU: $\alpha C_{batch}(M) + S_F \times \frac{M}{\alpha} \times C_{verify}$
- Node: None
- System: $\alpha C_{batch}(M) + S_F \times \frac{M}{\alpha} \times C_{verify}$

〔검증 시간 - Subset〕

- System: $\alpha T_{batch}(M) + S_F \times \frac{M}{\alpha} \times T_{verify}$

• 개별 노드에 의한 일괄 확인 및 유효하지 않은 서명 추출

개별 노드는 전체 M개의 메시지를 독립적으로 일괄 확인해본 후 실패한 일괄 확인에 대해 개별 메시지를 하나씩 재검증한다. 검증 시간은 개별 노드에 의해 병렬로 수행된다고 가정하여 1대의 차량이 일괄 확인을 수행하는 시간과 N대의 차량이 일괄 확인하는 시간이 같다고 가정한다. 이 기법의 검증 비용과 검증 시간은 다음과 같이 분석할 수 있다.

〔검증 비용 - Naive〕

- RSU: *None*
- Node: $C_{batch}(M) + M \times C_{verify}$,
- System: $N(C_{batch}(M) + M \times C_{verify})$

〔검증 시간 - Naive〕

- System: $T_{batch}(M) + M \times T_{verify}$

개별 노드에 의한 일괄 확인 역시 부분 일괄 확인 방법을 이용하면 보다 효율적으로 일괄 확인할 수 있다.

〔검증 비용 - Subset〕

- RSU: *None*
- Node: $\alpha C_{batch}(M) + S_F \times \frac{M}{\alpha} \times C_{verify}$
- System: $N(\alpha C_{batch}(M) + S_F \times \frac{M}{\alpha} \times C_{verify})$

〔검증 시간 - Subset〕

- System: $\alpha T_{batch}(M) + S_F \times \frac{M}{\alpha} \times T_{verify}$

제안하는 시스템 Case 1 (결정적 메시지 분배 - 개별 노드에 의한 재검증 - 실패 리포팅)

RSU는 전체 메시지를 일괄 확인해 본 후 실패한 일괄 확인에 대해 분산 일괄 확인을 개별 노드에게 요청하고 개별 노드는 K개의 메시지를 선택하여 분산 일괄 확인을 수행한다. 분산 일괄 확인에 실패한 개별 노드들은 K개의 메시지에 대해 개별 재검증을 수행한 후 실패 리포트를 RSU에게 전송한다. RSU는 수신한 R_F개의 실패 리포트에 대해 리포트 생성자의 서명을 검증하고 유효한 서명일 경우 리포트에 포함된 S_F개의 유효하지 않은 서명을 재검증한다. 이후 RSU는 확인된 유효하지 않은 서명을 제외한 전체 메시지를 일괄 확인하여 그 결과를 네트워크에 방송한다.

〔검증 비용〕

- RSU: $C_{batch}(M) + (R_F \times C_{verify}) + (S_F \times C_{verify}) + C_{batch}(M - S_F)$
- Node: $C_{batch}(K) + (K \times C_{verify})$

- System: $C_{batch}(M)+N\times(C_{batch}(K) + K \times C_{verify}) + (R_F\times C_{verify}) + (S_F\times C_{verify}) + C_{batch}(M-S_F)$

〔검증 시간〕

- System: $T_{batch}(M)+T_{batch}(K)+(K\times T_{verify})+(R_F\times T_{verify})+(S_F\times T_{verify})+T_{batch}(M-S_F)$

• 제안하는 시스템 Case 2 (결정적 메시지 분배 - 실패 리포팅 - RSU에 의한 재검증)

RSU는 전체 메시지를 일괄 확인해 본 후 실패한 일괄 확인에 대해 분산 일괄 확인을 개별 노드에게 요청하고 개별 노드는 K개의 메시지를 선택하여 분산 일괄 확인을 수행한다. 분산 일괄 확인에 실패한 개별 노드들은 별도의 재검증 없이 실패 리포트를 생성하고 RSU에게 전송한다. RSU는 수신한 R_F개의 실패 리포트에 대해 리포트 생성자의 서명을 검증하고 유효한 서명일 경우 유효하지 않은 서명 후보 리스트를 구성한다. RSU는 유효하지 않은 서명 후보 리스트에 포함된 $K\times R_F$개의 유효하지 않은 서명 후보들을 재검증하고 이후 RSU는 확인된 유효하지 않은 서명을 제외한 전체 메시지를 일괄 확인하여 그 결과를 네트워크에 방송한다.

〔검증 비용〕

- RSU: $C_{batch}(M)+R_F\times C_{verify}+R_F\times K\times C_{verify}+C_{batch}(M-S_F)$
- Node: $C_{batch}(K)$
- System: $C_{batch}(M)+N\times C_{batch}(K)+(R_F\times C_{verify})$

〔표 1〕 검증 비용과 검증 시간의 근사화

검증 연산 비용	
RSU(Naive)	$C_{batch}(M) + M\times C_{verify}\le (M+2)C_{verify}$
RSU(Subset)	$\alpha C_{batch}(M) + S_F\times \frac{M}{\alpha}\times C_{verify}\le (2\alpha+S_F\times\frac{M}{\alpha})C_{verify}$
개별 노드(Naive)	$N(C_{batch}(M) + M\times C_{verify})\le N(M+2)C_{verify}$
개별 노드(Subset)	$N(\alpha C_{batch}(M) + S_F\times \frac{M}{\alpha}\times C_{verify})\le N(2\alpha+S_F\times\frac{M}{\alpha})C_{verify}$
Case 1	$C_{batch}(M)+N\times(C_{batch}(K)+K\times C_{verify})+(R_F\times C_{verify})+(S_F\times C_{verify})+C_{batch}(M-S_F)$ $\le 2C_{verify}+N(K+2)C_{verify}+(R_F+S_F)C_{verify}+2C_{verify}$ $=(4+N(K+2)+(R_F+S_F))C_{verify}$
Case 2	$C_{batch}(M)+(N\times C_{batch}(K))+(R_F\times C_{verify})+(R_F\times K\times C_{verify})+C_{batch}(M-S_F)$ $\le 2C_{verify}+2NC_{verify}+R_F(1+K)C_{verify}+2C_{verify}$ $=(4+2N+R_F(1+K))C_{verify}$
검증 시간	
RSU(Naive)	$T_{batch}(M)+M\times T_{verify}\le (2+M)T_{verify}$
RSU(Subset)	$\alpha T_{batch}(M) + S_F\times \frac{M}{\alpha}\times T_{verify}\le (2\alpha+S_F\times\frac{M}{\alpha})T_{verify}$
개별 노드(Naive)	$T_{batch}(M)+M\times T_{verify}\le (2+M)T_{verify}$
개별 노드(Subset)	$\alpha T_{batch}(M) + S_F\times \frac{M}{\alpha}\times T_{verify}\le (2\alpha+S_F\times\frac{M}{\alpha})T_{verify}$
Case 1	$T_{batch}(M)+T_{batch}(K)+(K\times T_{verify})+(R_F\times T_{verify})+(S_F\times T_{verify})+T_{batch}(M-S_F)$ $\le 6T_{verify}+(K+R_F+S_F)T_{verify}$
Case 2	$T_{batch}(M)+T_{batch}(K)+(R_F\times T_{verify})+(R_F\times K\times T_{verify})+T_{batch}(M-S_F)$ $\le 6T_{verify}+R_F(1+K)T_{verify}$

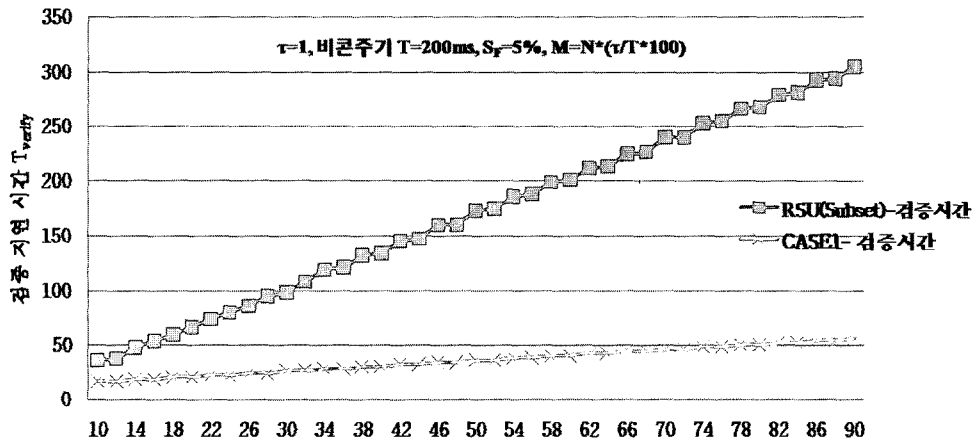
$+(R_F \times K \times C_{verify}) + C_{batch}(M - S_F)$

〔검증 시간〕

- System: $T_{batch}(M) + T_{batch}(K) + (R_F \times T_{verify}) + (R_F \times K \times T_{verify}) + T_{batch}(M - S_F)$

이상의 4가지 방법에 대해 정량적 비교 분석을 위해 다음과 같이 근삿값(approximation)을 구한다. 먼저 $C_{batch}(M)$는 M개의 메시지를 일괄 확인하는 비용으로 M개의 메시지에 대한 일괄 확인 인스턴스를 생성하는 비용 $C_{agree}(M)$과 생성된 인스턴스를 검증하는 비용 C_{verify}로 정의된다. $C_{agree}(M)$는 어떤 서명 알고리즘을 사용하느냐에 따라서 연산 비용이 달라지는데, RSA, ElGamal, DSA 등의 지수연산 기반 공개키 서명의 경우 M개의 곱셈연산이, 타원곡선 상의 신원 기반 서명 알고리즘이라면 M개의 덧셈 연산이 필요하다. 일반적으로 작은 수의 M에 대해서 곱셈 연산이나 타원곡선 상의 덧셈 연산이 각각 지수연산이나 페어링 연산보다 저렴하다 할 수 있으므로 본 논문에서는 $C_{batch}(M) \leq 2C_{verify}$, $T_{batch}(M) \leq 2T_{verify}$라 가정하고 $C_{batch}(M)$, $T_{batch}(M)$을 각각 $2C_{verify}$, $2T_{verify}$로 근사화한다. 따라서 위 6가지 일괄 확인 방법의 검증 비용 및 검증 시간은 〔표 1〕과 같이 근사화된다.

〔표 2〕는 메시지 수가 M, 노드 수가 N, 유효하지 않은 서명의 수가 S_F인 환경 $Env(M,N,S_F)$에서 앞서 분석한 6가지 일괄 확인 기법의 연산 비용과 연산 시간을 비교한 결과를 보여준다. VANET에서 일괄 확인해야하는 메시지 수에 영향을 미치는 요소는 크게 3가지로 개별 차량의 비콘 전송 주기와 메시지 수집 시간, 그리고 참여 차량의 수이다. 비콘의 전송 주기와 메시지 수집 시간은 검증 대상 메시지 수에 따른 비용 비교를 위해 고려되어야할 요소로서 전송 주기가 짧아지거나 메시지 수집 시간이 늘어날수록 전체 메시지 수 M이 증가하고 반대의 경우 메시지 수가 줄어든다. 다른 요소를 고정 시킨 채 참여 차량의 수를 증가시켜도 메시지 수는 늘어나지만 제안하는 Case 1의 경우 늘어나는 메시지는 참여 차량에 의해 고르게 분배되기 때문에 검증 비용에 미치는 영향은 각기 다르게 나타난다. 〔표 2〕의 $Env(100,20,5)$, $Env(200,20,10)$는 본 논문에서 보이고자하는 기본 시나리오다. 20대의 차량이 매 200ms와 100ms 마다 각각 서명 메시지를 방송하고 RSU는 1초간 메시지를 수집하며 수집된 메시지에는 5%의 유효하지 않은 서명이 포함되어 있다. 또한 네트워크 노드 밀도, 즉 네트워크에 유입되는 차량의 수에 따른 효율성을 분석 하기위해 참여 차량이 40대로 늘어난 경우를 실험하는 $Env(200,40,10)$, $Env(400,40,20)$ 시나리오를 추가한다. 〔표 2〕의 결과를 보면 6가지 일괄 검증 방법 중 서명 검증을 위한 연산량 측면에서는 RSU가 스스로 부분 일괄 확인 인스턴스를 만들어 검증하는 방법이 가장 효율적이지만 이 방법은 RSU에게 많은 부하를 발생시키는 방법이고 VANET 일괄 확인 환경에서는 서명 검증 연산량을 줄이는 것보다는 검증 시간을 최소화하는 것이 더욱 중요하다는 측면을 볼 때 제안하는 Case 1, 2에 비해 비효율적이다. 제안하는 Case 1의 경우 RSU가 분산 일괄 확인 하는 방법에 비해 $Env(100,20,5)$, $Env(200,20,10)$, $Env(200,40,10)$, $Env(400,40,20)$ 환경에서 각각 약 67%, 72%, 76%, 79%씩 검증 시간을 줄일 수 있다. 〔그림 3〕은 비콘 전송 주기 및 메시지 수집 시간 τ를 고정한 상태에서 노드수가 증가됨에



〔그림 3〕 참여 노드수의 증가에 따른 검증 지연 시간의 비교

〔표 2〕 연산량 및 연산 시간 근삿값 예

$Env(M, N, S_F)$		근삿값			비고
		연산 (C_{verify})		시간 (T_{verify})	
RSU(Naive)		$M+2$		$M+2$	
RSU	each node	$M+2$	0		
RSU(Subset)		$2\alpha+S_F\times\frac{M}{\alpha}$		$2\alpha+S_F\times\frac{M}{\alpha}$	
RSU	each node	$2\alpha+S_F\times\frac{M}{\alpha}$	0		
개별노드(Naive)		$N(M+2)$		$M+2$	
RSU	each node	0	$M+2$		
개별노드(Subset)		$N(2\alpha+S_F\times\frac{M}{\alpha})$		$2\alpha+S_F\times\frac{M}{\alpha}$	
RSU	each node	0	$2\alpha+S_F\times\frac{M}{\alpha}$		
Case 1		$4+N(K+2)+(R_F+S_F)$		$6+(K+R_F+S_F)$	
RSU	each node	$4+(R_F+S_F)$	$K+2$		
Case 2		$4+2N+R_F(1+K)$		$6+R_F(1+K)$	
RSU	each node	$4+R_F(1+K)$	2		

$Env(M, N, S_F)$		$Env(100,20,5)$			$Env(200,20,10)$			비고
		연산 (C_{verify})		시간 (T_{verify})	연산 (C_{verify})		시간 (T_{verify})	
RSU(Naive)		102		102	202		202	
RSU	each node	102	0		202	0		
RSU(Subset)		65		65	132		132	$\left(\begin{matrix}(100,20,5):\alpha=15\\(200,20,10):\alpha=31\end{matrix}\right)$ *
RSU	each node	65	0		132	0		
개별노드(Naive)		2040		102	4040		202	
RSU	each node	0	2040		0	4040		
개별노드(Subset)		1300		65	2640		132	$\left(\begin{matrix}(100,20,5):\alpha=15\\(200,20,10):\alpha=31\end{matrix}\right)$
RSU	each node	0	1300		0	2640		
Case 1		154		21	264		36	$\left(\begin{matrix}(100,20,5):K=5,R_F=5\\(200,20,10):K=10,R_F=10\end{matrix}\right)$
RSU	each node	14	7		24	12		
Case 2		74		36	154		116	$\left(\begin{matrix}(100,20,5):K=5,R_F=5\\(200,20,10):K=10,R_F=10\end{matrix}\right)$
RSU	each node	34	2		114	2		

$Env(M, N, S_F)$		$Env(200,40,10)$			$Env(400,40,20)$			비고
		연산 (C_{verify})		시간 (T_{verify})	연산 (C_{verify})		시간 (T_{verify})	
RSU(Naive)		202		202	402		402	
RSU	each node	202	0		402	0		
RSU(Subset)		132		132	266		266	$\left(\begin{matrix}(200,40,10):\alpha=31\\(400,40,20):\alpha=63\end{matrix}\right)$ *
RSU	each node	132	0		266	0		
개별노드(Naive)		8080		202	16080		402	
RSU	each node	0	8080		0	16080		
개별노드(Subset)		5280		132	10640		266	$\left(\begin{matrix}(200,40,10):\alpha=31\\(400,40,20):\alpha=63\end{matrix}\right)$
RSU	each node	0	5280		0	10640		
Case 1		304		31	524		56	$\left(\begin{matrix}(200,40,10):K=5,R_F=10\\(400,40,20):K=10,R_F=20\end{matrix}\right)$
RSU	each node	24	7		44	12		
Case 2		89		31	304		226	$\left(\begin{matrix}(200,40,10):K=5,R_F=10\\(400,40,20):K=10,R_F=20\end{matrix}\right)$
RSU	each node	29	2		224	2		

*: $2\alpha+S_F\times\frac{M}{\alpha}$의 최소값은 $\alpha=\sqrt{\frac{S_FM}{2}}$ 일 때

따라 RSU(Subset) 방법과 제안하는 Case 1 방법의 서명 검증 지연시간을 그래프로 나타낸 것이다. RSU(Subset) 방법은 메시지 검증에 소요되는 시간이 메시지 수에 의해서 결정되기 때문에 참여 노드수가 늘어날수록 그에 비례하여 검증 소요 시간도 증가하는 것을 알 수 있다. 하지만 제안하는 Case 1의 경우 분산 일괄 확인에 참여하는 노드 수만큼 전체 메시지가 분할되어 분배되기 때문에 검증 시간의 증가 폭이 매우 완만한 것을 알 수 있다.

V. 결 론

VANET에서는 차량 간 많은 수의 메시지가 교환되며, 교환되는 메시지를 인증하기위해 전자서명을 사용하고 있다. 일괄 확인 기법을 사용하면 많은 수의 서명 메시지를 효과적으로 검증 할 수 있지만 VANET 환경에 그대로 적용하기에는 몇 가지 어려움이 있다. 본 논문에서는 VANET 환경에서 일괄 확인 기법을 도입할 때 문제가 되는 서명 검증의 비효율성과 유효하지 않은 서명 추출의 어려움 문제를 해결하였다. 참여 차량은 전체 메시지 중 일부의 메시지만 선택하여 분산 일괄 확인하면 되고 RSU는 참여 차량의 분산 일괄 검증 결과로부터 또는 보고받은 리포트로부터 생성한 ISC 리스트를 이용하여 적은 수의 서명 재검증을 통해 유효하지 않은 서명을 추출해낼 수 있다.

참 고 문 헌

[1] M. Raya and J. Hubaux, "Securing Vehicular Ad hoc Networks," J. of Computer Security, vol. 15, no. 1, pp. 39-68, Jan. 2007.

[2] X. Lin, X. Sun, P. Ho, and X. Shen, "GSIS: A Secure and Privacy Preserving Protocol for Vehicular Communications," IEEE Trans. on Vehicular Technology, vol. 56, no. 6, pp. 3442-3456, Nov. 2007.

[3] G. Calandriello, P. Papadimitratos, J. Hubaux, and A. Lioy, "Efficient and Robust Pseudonymous Authentication in VANET," Proc. of the 4th ACM Int. Workshop on Vehicular Ad Hoc Networks, pp. 19-28, Sep. 2007.

[4] R. Lu, X. Lin, H. Zhu, P. Ho, and X. Shen, "ECPP: Efficient Conditional Privacy Preservation Protocol for Secure Vehicular Communications," Proc. of the IEEE INFOCOM 2008, pp. 1229-1237, Apr. 2008.

[5] X. Lin, C. Zhang, X. Sun, P. Ho, and X. Shen, "TSVC: Efficient and Secure Vehicular Communications with Privacy Preserving," IEEE Trans. on Wireless Communications, vol. 7, no. 12, pp. 4987-4998, Dec. 2008.

[6] C. Zhang, R. Lu, X. Lin, P. Ho, and X. Shen, "An Efficient Identity-based Batch Verification Scheme for Vehicular Sensor Networks," Proc. of the IEEE INFOCOM 2008, pp. 246-350, Apr. 2008.

[7] C. Zhang, X. Lin, R. Lu, P. Ho, and X. Shen, "An Efficient Message Authentication Scheme for Vehicular Communications," IEEE Trans. on Vehicular Technology, vol. 57, no. 6, pp. 3357-3368, Nov. 2008.

[8] A. Fiat, "Batch RSA," Journal of Cryptogly, vol. 10, no. 2, pp. 75 - 85, Mar. 1997.

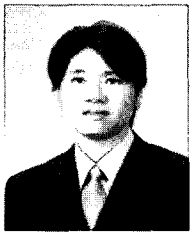
[9] M. Beller and Y. Yacobi, "Batch Diffie-Hellman key agreement systems and their application to portable communications," Advances in Cryptology, Eurocrypt 1992, LNCS 658, pp. 208-220, 1992.

[10] D. M'RAHI and D. NACCACHE, "Batch exponentiation - A fast DLP based signature generation strategy," Proc. of the 3rd ACM Conference on Computer and Communications Security, pp. 58-61, Mar. 1996.

[11] M. Bellare, J.A. Garay, and T. Rabin, "Fast Batch Verification for Modular Exponentiation and Digital Signatures," Advances in Cryptology, Eurocrypt 1998, LNCS 1403, pp. 236-250, 1998.

[12] J. Camenisch, S. Hohenberer, and M. Pedersen, "Batch Verification of Short Signatures," Advances in Cryptology, EUROCRYPT 2007, LNCS 4515, pp. 246-263, 2007.

〈著者紹介〉



임 지 환 (Jihwan Lim) 학생회원
2005년 2월: 한양대학교 전자컴퓨터공학부(학사)
2007년 2월: 한양대학교 컴퓨터공학과(석사)
2007년 3월~현재: 한양대학교 컴퓨터공학과 (박사과정)
〈관심분야〉 네트워크 보안
URL: http://infosec.hanyang.ac.kr/jhlim/

김 상 진 (Sangjin Kim) 종신회원
1995년 2월: 한양대학교 전자계산학과(학사)
1997년 2월: 한양대학교 전자계산학과(석사)
2002년 8월: 한양대학교 전자계산학과(박사)
2003년 3월~현재: 한국기술교육대학교 인터넷미디어공학부 부교수
〈관심분야〉 암호기술 응용
URL: http://infosec.kut.ac.kr/sangjin/

오 희 국 (Heekuck Oh) 종신회원
1983년: 한양대학교 전자공학과(학사)
1989년: 아이오와주립대학 전자계산학과(석사)
1992년: 아이오와주립대학 전자계산학과(박사)
1993년~1994년: 한국전자통신연구원 선임연구원
1995년 3월~현재: 한양대학교 컴퓨터공학과 교수
〈관심분야〉 암호프로토콜, 네트워크 보안
URL: http://infosec.hanyang.ac.kr/~hkoh/