논문 2010-05-15

기지국을 이용한 차량간 GPS 정보 교환을 위한 효율적인 인증 프로토콜

(An Efficient Authentication Protocol for GPS Information Exchange between Cars Using the Base Station)

조 국 래*, 손 종 욱, 조 희 섭
(Kookrae Cho, Jong-wuk Son, HuiSup Cho)

Abstract : Inter-vehicle communication is one of the most important parts in Intelligent Vehicle System. Through this communication, drivers can recognize what is happening out of their sights, such as the freezing condition of the street, traffic accidents, and so on. Each car in IVS gives various services to the drivers after analyzing those received information from cars or a base station. If the message is, however, exchanged from car to car directly, the computation cost which is needed for all the car to authenticate the transmitted message between nearby cars is tremendously high. Therefore, one can naturally think that the message communication between cars is performed with the help of the base station to reduce the computation cost. In this case where the base station collects all the information transmitted from cars and broadcasts them nearby, there should be an efficient way both for the base station to authenticate the car message within its communication range and for the car to authenticate the information received from the base station. In this paper, we present a two-way authentication protocol using a hash chain to efficiently exchange GPS information between a car and a base station. This information can be used to provide a driver with the navigation which displays all the moving cars around him in real time. When a car goes into an area of a base station, the car authenticates itself to the base station using its private key of PKI, sends a commitment of a hash chain, then starts to send a message with the hash value for authentication. The message includes GPS information, driver's status and so on. The base station also authenticates itself to the nearby cars using its private key, transmits the commitment of the hash chain, and sends all the messages gathered from cars with authentication information.

Keywords : IVS, Hash chain, Authentication, Inter-vehicle communication

1. 서 론

차량을 운전하다 보면, 운전자의 좌, 우, 후방 및 운전자의 시야가 미치지 못하는 지역의 상황을 제대로 인식하지 못해 여러 가지 긴급한 상황에 부딪치는 경우가 발생한다. 예를 들어, 길가에 차량이 빽빽이 주차되어 있는 골목길의 교차로를 지나가는 상황에서 빠른 속도로 교차하는 다른 차량이 있는 경우, 혹은 비가 심하게 몰아치는 험악한 날씨에서는 백미러나 사이드 미러만 가지고는 근처에서 움직이는 차량들을 제대로 인식할 수가 없으므로 차선 변경 시에 높은 접촉 위험에 노출될 수밖에 없다. 그러므로 어떤 경우라도 차량 주변 환경을 실시간으로 편리하게 확인할 수 있는 방법이 존재한다면, 운전자들에게 운전에 대한 심리적인 불안감을 덜어주고 안전한 차량 운행에 기여할 것으로 생각한다.

현재 사용되고 있는 위성 GPS로 차량의 위치를 단독 측정할 경우 전리층, 천체력, 대류권 등의 오

* 교신저자 (Corresponding Author)
논문접수 : 2010. 01. 29., 수정일: 2010. 07. 28.,
채택확정 : 2010. 08. 17.
조국래, 손종욱, 조희섭 : 대구경북과학기술원

차등으로 8~12m 정도의 수평 오차가 발생한다. 하지만, 보조 장치를 이용하면, DGPS 측위기법의 경우 1m~2m의 수평오차가 생기고, OTF CDGPS는 2cm~4cm의 오차가 발생하다[10]. 여기에, 유럽의 갈릴레오 위성항법장치의 경우 수평 4m의 오차가 발생하고, 특히, 보강 시스템을 이용하면 10cm 이내의 정확한 위치를 제공받을 수 있다[11]. 이러한 오차는 차량의 길이, 움직이는 속도, 그리고 제안하는 알고리즘이 운행의 보조 지표로 활용된다는 점을 고려할 때, 무시될만한 수치라고 볼 수 있다.

운전자가 차량 주변 환경 인식을 위하여 백미러나 사이드 미러를 볼 때, 운전 중 눈이 초점을 맞추고 이미지를 인식하기 위해 최소 2초 정도의 딜레이가 발생하는데, 이러한 시간 딜레이를 감안할 때, 기지국이 각 차량의 GPS를 수신하여 1초마다 이를 전송하는 경우, 사용자는 실시간으로 정보를 제공받는 것과 같이 느낄 수 있을 것으로 기대한다. 그리고 필요하다면, GPS 정보 외에 각 차량의 속도를 포함한다면, 특정 반경 내에 있는 차량들에 대해서는 속도를 고려한 차량의 예상 위치를 표현하여, 실시간성을 보다 높이는 방법을 추구할 수도 있을 것으로 생각한다.

본 논문에서는 운전자가 차량 내에서 주변 환경을 실시간으로 인식할 수 있도록, 기지국 송수신 범위내의 모든 움직이는 차량들의 GPS 정보를 기지국이 수집하여, 이를 조합한 후, 기지국내의 차량 그룹으로 전송하기 위한 효율적인 인증 프로토콜을 제안한다. 각 차량들은 수신한 GPS 정보 및 차량 정보를 네비게이션과 같은 디스플레이 단말기에 표시하여 운전자가 차량 주변 상황을 실시간으로 인식할 수 있도록 도와준다.

본 논문의 진행순서는 다음과 같다. 2장의 '선행기술 조사'에서 송수신되는 메시지 인증을 위해 이전에 제안된 기법들을 살펴보고, 3장의 '배경지식'에서 본 논문 서술에 사용되는 수식 및 용어들을 기술한다. 4장의 '제안하는 프로토콜'에서는, 본 논문에서 차량간 통신 인증을 위해 제시하는 프로토콜을 설명한다. 그 이후의 장에서 프로토콜에 대한 안정성 및 효율성 검증을 작성한 후, 논문을 마무리 한다.

2. 선행 기술 조사

ITS(Intelligent Transportation System)에서는 차량간 혹은 차량과 기지국간에 송수신되는 메시지를 인증하기 위한 여러 가지 기법들이 제안되어 있다. 이들 기법들은 송수신되는 메시지의 특성과 송수신 주기 등에 따라서 다양하게 나누어 질 수 있다.

몇몇 기법들은 공개키 암호 기법[4,5,6]을 사용하여 송수신되는 메시지에 대한 인증을 수행한다. 이들은 차량간 지속적인 메시지의 교환보다는 '앞 차량 주변에서 발생한 교통사고 알림'과 같은 일회성 메시지의 인증을 주된 목적으로 하고 있다. 이들은 RSA, ECDSA, NTRU와 같이 공개키를 기반으로 서명 및 검증을 수행하므로, 다음과 같은 Security를 모두 제공하는 것이 가능하다.

- Authentication and integrity
- Non-Repudiation
- Confidentiality

하지만, 이들은 Modular exponent multiplication을 지속적으로 수행해야 하므로, 메시지 인증을 위해서 수행하는 서명 및 검증에 필요한 처리량이 부담이 되고, 이로 인한 처리 속도 지연은 불가피하다.

이에 반해서 비밀키 암호기법을[7,8]을 사용해서 송수신되는 메시지를 인증하게 되면, 비트 이동과 XOR과 같은 기본 연산만으로 메시지의 인증이 가능하기 때문에 처리 속도 지연에 대한 문제를 해결하고, 짧은 주기 내에 지속적인 메시지를 처리하는 것이 가능하다.

[7]는 교차로에서 차량들의 진행 순서를 정하기 위해서 안전한 통신 채널 확보를 위하여 multipary Diffie-Hellman agreement protocol 등을 사용하는 것을 권고하고 있다. 각 차량들은 공개키를 사용해서 그들 간에 비밀키를 형성한 후, 이를 사용해서 필요한 정보들을 송수신한다.

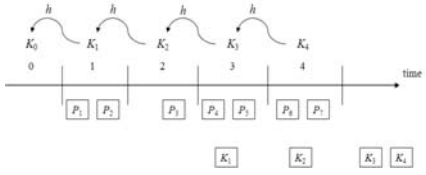

그림 1. TESLA 인증 방법
Fig. 1. TESLA authentication method

TESLA[8]는 메시지와 MAC을 브로드 캐스팅한 후, 아래 그림과 같이 일정 주기 마다, MAC에 대한 키를 전송함으로써, 이전에 수신한 메시지에 대한 인증을 수행한다. 이때, 사용하는 키는 Hash Chain을 이용해서 생성한다.

[6]는 IEEE 1609.2 VANET 표준을 준수하여 ECDSA를 TESLA에 적용하였다. TESLA에 비해 느려졌지만, TESLA에서 보장할 수 없었던 Non-Repudiation을 제공할 수 있게 되었다.

본 논문은 차량과 기지국간의 효율적인 GPS 송수신을 위해서 TESLA와 같이 해쉬 체인을 사용하여 메시지에 대한 인증을 수행한다. 하지만, 네트워크상의 임의의 공격자를 가정하고 있는 TESLA와는 달리 기지국과 이에 속한 모든 차량들은 송수신되는 모든 메시지 패킷들을 관찰 할 수 있다. 그러므로 TESLA와 같이 메시지에 대한 MAC을 생성한 후, Hash Chain의 값을 일정 주기 이후에 전송하는 것이 아니라, 각 차량이나 기지국이 송신하는 메시지에 대한 해쉬 체인의 해쉬 값을 같이 전송하는 양방향 GPS 전송 방법을 본 논문에서 제안한다.

3. 배경 지식

이장에서는 본 논문에서 사용하고 있는 수식 및 용어에 대해서 설명하고, 본 프로토콜의 핵심인 Hash Chain에 대한 간략한 설명을 하고자 한다.

1. 수식 및 용어

- ID : 모든 객체가 자신을 유일하게 표현하는 일련번호
- Hash(·) : 임의의 입력으로부터 160bit ~ 256bit를 생성하는 SHA-1, RIPEMD-160과 같은 Cryptographic Oneway Hash Function
- $h_i = Hash(h_{i+1})$: h_{i+1}에 해쉬 연산 Hash()를 수행해서 h_i을 생성, i는 n에서 시작해서 1씩 감소함
- h_n : 해쉬 체인에서 일련의 해쉬값을 생성하기 위해 입력되는 seed 값
- h_0 : h_n 를 n번 해쉬해서 생성한 해쉬 값
- h_{curr} : 해쉬 체인에서 메시지 인증을 위해 사용되는 해쉬 값으로 curr는 1에서 1씩 n 까지 증가함
- $sig = Sign_{key}(message)$: message를 key로 서명하여 서명문 sig를 생성
- $message = Veri_{key}(sig)$: sig를 key로 검증하여 검증문 message를 생성
- Pri_{object} : 공인 인증기관이 발급한 object의 비공개키
- Pub_{object} : 공인 인증기관이 발급한 object의 공개키
- $ciphertext = Enc_{key}(message)$: Enc는 RSA와 같이 key로 message를 암호화해서 ciphertext를 생성하는 공개키 암호 알고리즘
- $message = Dec_{key}(ciphertext)$: Dec는 RSA와 같이 key로 ciphertext를 복호해서 message 생성하는 공개키 복호 알고리즘
- $ciphertext = E_{key}(message)$: E는 AES와 같이 key로 message를 암호화해서 ciphertext를 생성하는 같은 비밀키 암호 알고리즘
- $message = D_{key}(ciphertext)$: D는 AES와 같이 key로 ciphertext를 복호해서 message 생성하는 비밀키 복호 알고리즘
- GPS_{id} : id의 GPS 정보
- MN_{id} : 차량 및 객체 id에 대한 Model Number
- TCM : 기지국 내의 모든 객체 및 차량에 대한 GPS, MN 등을 포함하고 있는 Traffic Control Message
- TS : 공유되어 사용되는 시간, TimeStamp
- pebble : 해쉬 체인 Traversal에 사용되는 최대 저장 공간 크기
- budget : 한 번의 해쉬 체인 Traversal에 사용되는 최대 Hash() 연산 횟수

2. Hash Chain

임의의 값에 Cryptographic Hash Function을 연속으로 사용해서 생성된 일련의 해쉬값을 가리킨다. 아래 그림과 같이 vn을 시작점으로 해쉬 함수 h를 적용하여 v0까지 연속해서 생성된 <v0, v1, ..., vn>을 해쉬 체인이라고 한다. 이때, v0를 공개하고, vn을 이용해서 그 이하의 모든 해쉬값을 생성한다. vi를 수신하면, i번 해쉬하여 v0와 같은지를 비교한다. 가장 최근에 vi-1이 메시지 인증을 위해 사용되었으면, 그 이전의 해쉬값은 그 효력을 상실한다. 즉, 해쉬 체인의 해쉬값은 노출된 순간에만 유효한 것이다.

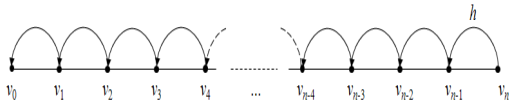

그림 2. 해쉬 체인 Traversal 과정
Fig. 2. Hash chain traversal process

해쉬 체인 생성에 대한 대표적인 논문으로는 [1]가 있고, 각각의 해쉬 체인 생성에 소모되는 메모리 복잡도를 나타내는 pebble은 log2n, 이때 사용되는 계산 복잡도를 표시하는 budget은 log2n이 된다. 이러한 해쉬 체인 Traversal를 향상시킨 논문으로 [2,3]등이 제안되었고, [3]에서는 log2n의 pebble과 0.5log2n의 budget을 사용해서 해쉬 체인내의 모든 해쉬 값을 생성할 수 있다.

4. 제안하는 프로토콜

본 논문에서 차량들은 자신의 GPS 정보를 1초

간격으로 기지국으로 송신하고, 기지국은 자신의 송수신 범위내의 모든 차량들의 GPS를 수집하여 TCM을 생성한 후, 이를 브로드 캐스팅한다. 차량이 GPS 위치정보를 기지국에 전송할 때는, AES와 같은 비밀키 알고리즘으로 암호화해서 전송하므로, 초기단계에서 비밀키를 생성한 후, 기지국에 전달하는 과정이 필요하다.

반면에, 기지국에서 차량으로 메시지를 브로드캐스팅 할 때에는, 송신되는 메시지가 Hash Chain을 통해서 인증되므로, 기지국이 사용할 해쉬 체인 초기 값을 차량에 설정하는 과정이 필요하다. 이러한 비밀키와 해쉬 체인 초기 값 설정과정은 아래 'Key Setup'에서 이루어진다. 이렇게 초기 값이 설정되면, 차량은 GPS 및 기타 정보를 비밀키로 암호화해서 기지국에 전송하고, 기지국은 이를 수집하여 해쉬 체인 인증 정보와 함께 모든 차량에게 브로드 캐스팅한다. 이를 위한 프로토콜은 '기지국과 차량의 GPS 및 TCM 송수신 프로토콜'에서 설명한다.

1. Key Setup

- A.1 *Server(s) Hash Value Setup in Server()*

모든 기지국(s)은 기지국의 일련번호 IDs, 비공개키 Pris, 공개키 Pubs를 보유하고, 아래의 기지국 Setup 과정을 수행하고 해쉬체인의 공개값 h0와 Seed hn을 저장한다.

$Extract\ randomly\ h_n;\ h_0 = h_n;$
$For(i=0; i<n; i++)$
$\ \ h_0 = Hash(h_0);$
$m = n, h_0, ID_s, TS, etc;$
$h' = Hash(m); sig = Sign_{pri_s}(h');$
$Store\ sig, m\ into\ memory;$
$Message\ Type = 1;$
$Broadcast(Message\ Type, sig, m);$

* 해쉬 횟수 n과 해쉬 체인의 최종값인 h0 등 각종 정보를 담고 있는 메시지m을 해쉬하여 h'을 생성하고, 이를 기지국의 비공개키인 pris로 서명하여, m에 대한 서명값인 sig를 생성한다.

- A.2 *Server(s) Hash Value Setup in Car (MessageType, sig, m or i_{curr}, h_{i_curr}, m_2)*

기지국(s)이 송신한 해쉬 체인 초기 설정값을 차량이 수신하여 저장하는 역할을 한다. 기지국이 송신하는 메시지는 두 가지가 있다. 첫 번째는 기지국이 해쉬체인을 처음 설정하고자 할 때, 관련 초기 정보들을 전송하는 경우이고(Message Type 1), 두 번째는 주변차량으로부터 수신한 GPS들을 조합하여 만든 TCM을 송신하는 경우이다(Message Type 0). 후자의 경우, 기지국이 Hash chain setup을 수행한 이후에, 기지국으로 들어온 차량을 위해서 현재 사용 중인 해쉬체인 설정 정보들을 포함하고 있다. 그러므로 이러한 차량들은 의무적으로 Server(s) Hash Value Setup in Car()를 한번 수행해서 인증에 필요한 해쉬 체인 정보들을 제공받아야 한다. 이후에는 Message Type이 1인 경우에만 Server(s) Hash Value Setup in Car() 을 수행한다. Message Type이 1인 경우는 sig,m이 입력되고, 0인 경우에는 icurr, hi_curr, m2가 수신된다.

$if(Message\ Type == 0)\ \{$
$\ \ extract(sig, m)\ from\ m_2;$
$\ \ h' = Hash(m); h'' = Veri_{pub_s}(sig);$
$\ \ if(h' == h'')\ \{$
$\ \ \ \ for(i=1; i \leq i_{curr}; i++)$
$\ \ \ \ \ \ h_{i_{curr}} = Hash(h_{i_{curr}});$
$\ \ \ \ extract\ (n, h_0, ID_s, TS, etc)\ from\ m;$
$\ \ \ \ if(h_i \equiv h_0)\ then\ \{$
$\ \ \ \ \ \ store\ (n, h_0, ID_s, TS, h_{i_{curr}}, etc);$
$\ \ \ \ \}\ else\ reject\ this\ message;$
$\ \ \}\ else\ reject\ this\ message;$
$\}$
$else\ if\ (Message\ Type == 1)\ then\ \{$
$\ \ h' = Hash(m); h'' = Veri_{pub_s}(sig);$
$\ \ if(h' \equiv h'')\ then\ \{$
$\ \ \ \ h_{i_{curr}} = h_0;$
$\ \ \ \ extract\ from\ m\ and\ store$
$\ \ \ \ (n, h_0, ID_s, TS, h_{i_{curr}}, etc);$
$\ \ \}\ else\ reject\ this\ message;$
$\}$

* MessageType이 0인 경우는 (icurr, hi_curr, m2)가 A.8로부터 전송되는데, 이들을 이용해서 해쉬 체인 setup에 필요한 정보들을 생성한다. 먼저 m2에서 sig과 m를 추출하여 m에 대한 sig의 서명이 유효한지를 검증한 후, 통과되면, hi_curr를 icurr번 Hash() 연산을 수행한 후, 결과값이 h0와 같은지를 확인하고, 같다면, 필요한 정보들을 저장하고, 같지 않다면, 다시 수행한다.

* Messagetype이 0이 아닌 경우는 기지국으로부터 (Messagetype, sig, m)가 전송되는데, m에 대한 서명 sig가 유효한지를 검증한 후, 유효하다면, 필요한 정보들을 저장한다. 이때, hi_curr에는 h0를 삽입한다.

- A.3 *Car(c) Key Setup in Car()*

모든 차량 및 객체(c)들은 일련번호 IDc, 비공개키 Pric, 공개키 Pubc를 소유하고 있으며, 아래의

Car Setup 과정을 수행한다. 이때, 차량은 기지국과의 이후 통신에 사용될 비밀키 key를 임의로 생성하여, 전송되는 값들을 공개키로 암호화하고, 이를 전자 서명한 후, 기지국에 전송한다. 이후의 메시지 전송은 이 비밀키로 암호화해서 전송한다.

$Extract\ randomly\ key;$
$m = Enc_{pub_s}(ID_c, key, MN_c, etc);$
$h' = Hash(m);\ sig = Sign_{pri_c}(h');$
$SendToServer(sig, m);$

- A.4 *Car(c) Key Setup in Server(sig, m)*

기지국(s)은 새로운 차량이 자신의 영역에 들어와서 GPS 위치정보 송신에 사용될 비밀키 key를 자신에게 전송하면, 이를 검증한 후, 유효하면 필요한 정보들을 저장한다. 저장되는 정보에는 차량의 모델 넘버인 MNc도 포함되어 있다.

$h' = Hash(m); h'' = Veri_{pub_c}(sig);$
$\text{if}(h' == h'')\ then\{$
$(ID_c, key, MN_c, etc) = Dec_{pri_s}(m);$
$store\ them;$
$\}else\ reject\ this\ message;$

2. 기지국과 차량의 GPS 및 TCM 송수신 프로토콜

본 프로토콜은 GPS를 차량이 생성해서 기지국에 전송하는 Car to Server GPS Transmission, 수신한 GPS와 MN을 추출하여 TCM에 저장하는 GPS Processing in Server, GPS와 MN의 집합으로 되어 있는 TCM을 기지국 영역내에 송신하는 Server to Car TCM Transmission, 수신한 TCM과 저장되어 있는 위성 사진을 mapping하여 이를 디스플레이에 나타내는 TCM Processing in Car로 구성되어 있다.

- A.5 *Car to Server GPS Transmission()*

각 차량은 수신한 GPS를 자신이 속한 기지국으로 비밀키 key를 사용하여 AES와 같은 비밀키 알고리즘 E()로 암호화해서 서버에 전송한다.

$m = ID_c, GPS_c, TS, etc;$
$ciphertext = E_{key}(m)$
$SendToServer(ID_c, ciphertext);$

- A.6 *GPS Processing in Server(ID_c, ciphertext)*

기지국은 차량으로부터 전송된 IDc를 이용해서 관련정보를 메모리로부터 복원하고, ciphertext를 복호하여 차량의 GPS 위치정보를 아래와 같이 추출하여, TCM에 관련 정보들을 삽입한다.

$extract\ key_c, MN_c$
$from\ memory\ in\ Server\ using\ ID_c;$
$m' = D_{key}(ciphertext);$
$extract\ ID_c', GPS_c'\ TS', ..from\ m';$
$\text{if}(ID_c' \equiv ID_c)\ then\ \{$
$\text{if}(TS\ is\ same\ time\ period)\ then\{$
$store\ (GPS_c', MN_C, ..)\ into\ TCM;$
$\}\ else\ reject\ this\ message;$
$\}else reject\ this\ message;$

* IDc를 이용하여 초기에 설정된 비밀키 key를 복원하여, 이를 이용하여 암호문 ciphertext를 복호한다.

* 복호된 IDc'가 IDc와 동일한지 확인한다.

* 수신된 메시지의 TimeStamp TS를 확인하여 이 값이 전체 송신할 TCM 메시지와 같은 시간 주기 안에 들어 있다면, GPSc', MNc를 TCM에 포함시킨다.

- A.7 *Server to Car TCM Transmission()*

Server는 Time을 확인한 후, 일정 주기가 되면, TCM에 저장되어 있는 모든 GPS와 MN을 자신의 영역 내에 브로드 캐스팅한다. 차세대 이동통신 기술인 UMB는 최고 280Mbps의 전송속도를 제공하므로[12], TCM에 들어가는 각 차량의 GPS, MN 등의 정보가 최대 256 바이트 정도라고 가정할 때, 하나의 기지국에서 이론적으로 1초에 100만대의 차량 정보를 전송할 수 있다. 그러므로 차량과 기지국의 송수신 범위를 고려할 때, TCM이 아무리 늘어나더라도 차량과 기지국간에 충분히 처리 가능한 프로세싱 범위에 있을 것으로 생각한다.

$TCM = [GPS_{c1}, MN_{c1}], [GPS_{c2}, MN_{c2}], ..$
$within\ the\ same\ time\ period;$
$h_{i_{curr}} = Hash(h_{i_{curr}+1});$
$TEMP = i_{curr}, h_{i_{curr}}, TS, etc;$
$m_1 = TEMP, TCM;$
$retrieve\ sig, m\ from\ memory;$
$m_2 = sig, m;\ Message\ Type = 0;$
$i_{curr}\ ++;$
$Broadcast(Message\ Type, ID_s, m_1, m_2);$

* 기지국은 이번 주기에 차량으로부터 수신된 모든 GPS와 MN을 포함하고 있는 TCM을 생성한 후, 인증을 위한 해쉬값 hi_curr와 함께 브로드캐스팅한다. hi_curr는 [3]를 사용하면, log2n의 pebble과 0.5log2n의 budget만으로 생성 가능하다.

* 기지국이 Server(s) Hash Value Setup in Server() A.1을 수행한 후, 새로 기지국 내에 진입한 차량들을 위해서 A.1에서 생성한 후, 저장한 sig와 m을 m2에 넣어 전송한다.

- A.8 *TCM Processing in Car*
(MessageType, ID_S, m_1, m_2)

기지국으로부터 전송된 TCM을 차량이 수신한 후, 수신된 메시지의 유효성을 검증한다. 유효하다면, TCM의 모든 GPS와 MN을 추출하여, 차량의 데이터 베이스에 저장되어 있는 위성 사진과 mapping한다. 이렇게 mapping된 사진을 디스플레이에 나타낸다.

$$
\begin{aligned}
&\text{if}(MessageType == 0)\{\\
&\quad extract\ from\ storage\ in\ car\\
&\qquad h_n, i_{curr}, h_{i_{curr}};\\
&\qquad i_{last} = i_{curr}, h_{i_{last}} = h_{i_{curr}};\\
&\quad extract\ from\ m_1\\
&\qquad i_{curr}, h_{i_{curr}}, TCM, etc;\\
&\quad \text{if}(i_{curr} \le i_{last})then\\
&\qquad reject;\\
&\quad i_{temp} = i_{curr};\ h_{i_{temp}} = h_{i_{curr}};\\
&\quad \text{for}(j=0; j<i_{curr}-i_{last}; j++)\\
&\qquad h_{i_{curr}} = Hash(h_{i_{curr}});\\
&\quad \text{if}(h_{i_{curr}} == h_{i_{last}})then\{\\
&\qquad i_{curr} = i_{temp}; h_{i_{curr}} = h_{i_{temp}};\\
&\qquad store\ i_{curr}, h_{i_{curr}};\\
&\qquad \text{if}(time\ is\ same\ time\ period)then\{\\
&\qquad\quad place\ all\ GPS_c, MN_c\ in\ map;\\
&\qquad\quad display\ the\ map\ on\ Navigation;\\
&\qquad \}else\ reject;\\
&\quad \}else\ reject;\\
&\}else\ call\ A.2\ with(i_{curr}, h_{i_{curr}}, m_2);
\end{aligned}
$$

* 기지국에서 전송된 MessageType이 0인 경우는(Messagetype, IDs, m1, m2)가 전송되는데, m1에 TCM이 존재한다.

* icurr가 가장 최근에 수신된 메시지의 인증 해쉬값인 ilast와 작거나 같다면, icurr는 이미 사용되었기 때문에 더 이상 유효하지가 않으므로 수신된 메시지의 인증값으로 사용될 수가 없다. 그러므로 모든 수신된 메시지를 폐기한다.

* 메시지가 유효하다면, TCM내의 모든 GPS와 MN을 차량 내에 저장되어 있는 인공 위성 사진(map)에 mapping하여 이를 네비게이션과 같은 디스플레이 단말기에 표현한다. 위성사진에 mapping 할 때는 자신의 GPS와 MN도 같이 포함한다.

* MessageType이 0가 아니라면 1이므로, 이는 기지국이 해쉬 체인에서 사용되는 기본 해쉬 값들을 새롭게 설정하고자 함을 뜻한다. 그러므로 차량 내에서 서버측 해쉬 값을 새로 설정하기 위한 프로토콜인 'Server(s) Hash Value Setup in Car()'인 A.2를 호출하여 수행한다. 이때, icurr, hi_curr, m2를 함께 파라미터로 호출한다.

3. 제안하는 프로토콜 전체 구조

제안하는 프로토콜에서는 기지국과 차량이 각각 Key Setup 프로시져 A.1~A.4를 먼저 수행한다. 그리고 아래 그림과 같이 인공위성으로부터 GPS를 수신한 자동차가 A.5 'Car to Server GPS Transmission'을 통해서 GPS 및 기타 정보들을 전송한다. 이를 수신한 기지국은 A.6 'GPS Processing in Server'를 통해서 정해진 시간 내에 전송된 모든 GPS 정보를 TCM에 저장한다. 기지국은 이 TCM을 A.7 'Server to Car TCM Transmission'을 통해서 해쉬 체인 인증값과 함께 브로드캐스팅한다. TCM을 수신한 차량은 'A.8 TCM Processing in Car'를 수행하여 네비게이션과 같은 디스플레이에 움직이는 모든 주변차량의 위치를 표시한다.

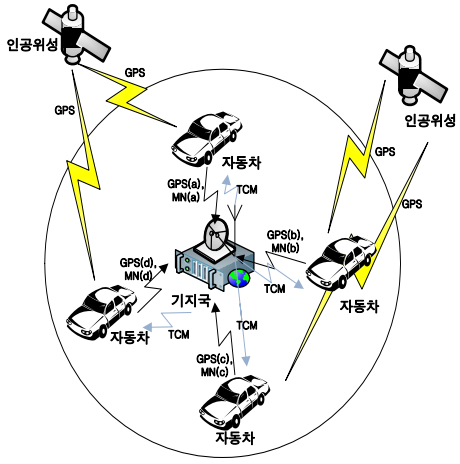

그림 3. 제안하는 프로토콜 전체 구조
Fig. 3. Proposed protocol structure

5. 안정성 및 효율성

1. 제안하는 프로토콜의 Security

1.1 프로토콜의 안전도

제안하는 프로토콜의 안전도는 다음과 같은 가정에 의존한다.

□ 인증용으로 수신된 해쉬 값은 이전 세션에 한번도 사용되지 않았거나, 이전에 수신된 해쉬값으로 생성이 가능해서는 안된다.

□ 본 논문의 해쉬 함수(Hash), 서명(Sign) 및 검증(Veri) 알고리즘은 암호학적으로 안전한 함수 및 알고리즘을 사용한다.

이러한 가정들이 만족되는 한, 수신된 메시지를 성공적으로 인증할 수 있는 메시지를 공격자가 임의로 생성하는 것은 계산적으로 아주 어렵다.

1.2 프라이버시 침해에 대한 안전도

차량의 위치정보를 알려주기에 개인의 프라이버시 문제가 발생할 수 있는데, 이는 휴대 전화기와 기지국간의 프라이버시 문제와 비슷하다고 볼 수 있다. 각 기지국은 통화 효율성을 위하여, 일정주기마다 휴대 전화기가 속해있는 기지국에서 Query-Response를 통해 각 사용자가 존재하는 기지국의 위치를 파악하고 있기 때문이다. 기지국은 모든 휴대전화의 위치정보를 알 수 있으므로, 기지국이 차량의 프라이버시를 침해하는 관점에서 보면, 제안하는 알고리즘은 휴대전화와 동일한 사생활 침해가 발생된다.

본 논문에서는 효율성을 극대화하기 위해 최소의 데이터 송수신 및 처리량을 프로토콜로 작성하였는데, 만약 기지국이 아닌, 제삼의 공격자에 의한 개인의 프라이버시가 문제가 된다면, 새로운 기지국에 접근할 때마다, 초기 설정단계에서 IDc 대신에 Psuedo-identity를 각 차량들이 생성하여, 기지국의 공개키로 암호화한 후, 전송하고, TCM에서 Psuedo-identity를 브로드캐스팅하면, 제삼의 공격자가 운전자의 Identity를 가지고 추적할 방법은 휴대전화기의 보안성과 동일하다.

그러나 공격자가 운전자의 GPS 위치정보를 정확히 알고 있다면, 위치정보 변화를 이용한 차량의 추적은 가능하다. 이러한 경우, 공격하고자 하는 차량의 정확한 위치 정보를 공격자가 직접 그 위치에서 확인해야하고, 추적하고자 하는 차량과 동일한 기지국으로 계속 이동해야 한다는 부담이 공격자에게 지속적으로 작용한다.

2. 제안하는 프로토콜의 Performance

제안하는 프로토콜은 차량의 위치정보를 주변 차량들이 공유하는 것을 목표로 하고 있다. 공격자가 위치정보를 임의로 생성하거나, 다른 차량의 정보를 변경하여 전송하는 것을 막기 위해서 송신 메시지마다 수신자가 정보의 신뢰성을 확인할 수 있는 인증정보를 함께 전송하거나 암호화해서 암호문을 전송하고 있다.

기지국을 이용하지 않고, 차량간 통신 표준인 WAVE를 통해서 위치정보를 개별적으로 송수신하는 경우, 각 차량은 주변 모든 차량과 개별적으로 안전한 채널을 형성해야 하므로, 차량이 R개인 경우 각 차량은 R개의 비밀키를 저장해야 하고, 이를 위해서 R번의 전자서명 및 검증이 소요된다. 그리고 개별사용자의 공개키가 유효한지 혹은 폐기되었는지를 확인해야하는 문제점도 존재하며, R개의 메시지를 각 차량들끼리 지속적으로 송수신하기위해서는 최대 (R-1)!의 통신채널이 필요하고, 이러한 통신 채널 분배를 어떻게 할 것인지에 대한 문제가 발생한다.

본 논문에서는 설치비용은 늘어나지만, 뛰어난 성능을 갖춘 기지국을 활용함으로써, 각 차량이 자신과 기지국의 비밀키 혹은 해쉬체인만을 소유하도록 하였고, 이를 인증하기 위한 전자서명 및 검증은 초기 설정단계에서 1회만 수행하면 된다. 기지국은 자신의 공개키 인증서를 짧은 주기에 지속적으로 받을 수가 있으므로, 기지국 공개키 신뢰성 검증은 높은 신뢰성을 확보할 수 있다. 더욱이 모든 차량이 GPS 위치정보를 기지국에 전송하고, 기지국이 이를 취합하여 브로드캐스팅하므로, R개의 메시지를 지속적으로 전송하기 위해 필요한 통신 채널은 최대 R+1개이므로, 차량보다 성능이 훨씬 뛰어난 기지국은 휴대 전화기의 기지국과 같이 통신 채널의 분배를 효율적으로 처리할 수 있다.

본 논문에서 사용되는 알고리즘들의 Computational Complexity를 살펴보기 위해서, RSA 1024를 이용한 공개키 암복호, 전자서명 및 검증, 해쉬체인에 사용되는 SHA-512 해쉬함수, 비밀키 알고리즘에 사용되는 AES/CTR의 1초당 수행 횟수를 살펴보면 아래 표와 같다[9].

위 정보들을 바탕으로 차량과 기지국간의 GPS 위치정보 전달을 위해서 일반적인 공개키만을 사용하여 구현한 경우와 제안하는 알고리즘을 성능 측면에서 비교하고자 한다. 해쉬체인에 사용되는 해쉬 횟수를 N, 기지국 내의 차량 대수를 R, 차량이 서버에 보내는 GPS 위치정보 메시지 크기를 암복호 단위로 나눈 횟수를 CGPS, 서버가 차량에 전송하는 TCM 메시지의 크기를 암복호 단위로 나눈 횟수를 CTCM이라고 가정한다.

일반적인 공개키 방법에서 데이터 암복호는 고려하지 않더라도, 송수신 데이터 인증을 위해 RSA 전자서명을 사용하는 경우, 메시지 전송마다 각 차량은 송신데이터 전자서명 및 수신데이터 검증을 1회씩 수행하고, 기지국은 R번 수신데이터 검증과 1번의 송신메시지 전자서명을 해야 한다. 그러므로 메시지를 전송할 때마다, 전체적으로 R+1번의 전자

서명과 2R번의 서명 검증을 수행해야 한다. 3R번의 RSA 연산이 매 메시지 전송마다 발생하는 것은 상당히 비효율적이다.

표 1. 함수의 초당 연산횟수
Table 1. Function's operation number per sec.

함수이름	operation/sec
RSA 1024 Encryption	12500
RSA 1024 Decryption	685
RSA 1024 Signature	676
RSA 1024 Verification	14285
SHA-1	8021606
AES/CTR(128-bit)	9109504

표 2. 일반적인 공개키 방법의 계산 복잡도
Table 2. Computational complexity using general public key algorithm

순서	RSA Enc	RSA Dec	RSA Sig	RSA Veri	AES(128) (Enc/Dec)	SHA-1
A.1						
A.2						
A.3	1		1			1
A.4		1xR		1xR		1
A.5					CGPS	
A.6					R	
A.7					CTCMxR or R	
A.8					CTCM or R	

표 3. 제안하는 프로토콜 계산 복잡도
Table 3. Computational complexity using proposed protocol

순서	RSA Enc	RSA Dec	RSA Sig	RSA Veri	AES (Enc/Dec)	SHA-1
A.1			1			N+1
A.2				1		N+1
A.3	1		1			1
A.4		1		1		1
A.5					CGPS	
A.6					R	
A.7						$0.5\log_2 N+1$
A.8						$0.5\log_2 N$

따라서 일반적인 공개키 방법에서는 공개키를 사용해서 기지국과 초기에 비밀키를 생성하고, 차량에서 전송하는 데이터는 본 논문의 방법과 동일하게 처리하고(A.5~A.6), 기지국에서 송신하는 데이터에 대한 인증은 각 차량과 설정한 이 비밀키를 사용하여, Keyed MAC으로 처리하는 것을 가정하였다(A.7~A.8).

이러한 방법을 제안하는 프로토콜의 진행순서(A.1~A.8)에 맞춰서 발생되는 계산 복잡도를 계산하였는데, 표는 기지국과 1대의 차량에서 발생하는 계산량을 보여준다. 아래표에서 A.4를 살펴보면 프로토콜상에서는 1회의 RSA Dec를 수행하지만, 기지국은 자신에게 속해있는 모든 차량 R에 대해서 A.4를 수행해야하므로, 1xR번의 연산을 수행하는 것으로 표기하였다.

표를 살펴보면, 제안하는 프로토콜이 일반적인 공개키 방법보다 전자서명 및 검증 횟수가 1회씩 많은 것을 볼 수 있다. A.1과 A.2는 기지국의 해쉬체인 인증정보 초기화 과정이므로, 해쉬체인 주기 N회에서 1회씩만 수행한다. 하지만, 서버에서 TCM을 인증메시지와 같이 보내는 A.7의 경우, 제안하는 알고리즘에서 자세히 기술하진 않았지만, [3]를 사용해서 (0.5log2N+1)번의 SHA-1 연산을 수행하면 된다. 반면에 일반적인 공개키의 경우에서는 CTCMxR번의 AES연산이 필요하다.

왜냐하면, TCM 메시지에 대한 인증정보로써 각 차량과의 비밀키를 이용한 Keyed MAC(Message Authentication Code)를 생성해야 하는데, 이는 AES를 Block Cipher로 하는 CBC-MAC으로 생성하고, 이곳에서, CTCMxR번의 AES 연산이 사용되기 때문이다. 만약 이것이 부담이 된다면, Security Level은 떨어지더라도, TCM을 해쉬한 후, 그 값에 대해서만 각 차량의 비밀키로 Keyed-MAC을 생성할 수도 있다. 이러한 경우에도 R번의 AES 연산은 필요하다.

여기서 사용되는 GPS 위치정보가 256 바이트, R이 10,000대 인 경우 TCM은 2,560,000바이트가 된다. AES는 한번에 16바이트씩 처리하므로, 이 경우에 CTCM은 160,000번의 AES 연산을 수행한다. 그러므로 A.7의 CTCMxR은 1,600,000,000번의 AES 연산을 수행해야하므로, 제안하는 프로토콜의 성능이 뛰어남을 확인할 수 있다. A.7에서 기지국은 TCM을 해쉬하여 Keyed MAC을 R번만 수행한다고 하더라도, 10,000번의 AES 연산을 실행해야 하므로, N이 212이 된다고 하더라도, 제안하는 알고리즘은 SHA-1 연산 횟수가 7회밖에 되지 않으므로, 10,000에 비해 상당히 효율적임을 알 수가 있다.

더욱이, 각 프로토콜 과정에서 A.1~A.4는 초기 설정 단계이므로, 일정 주기 N마다 1회만 수행되지만, A.5~A.8은 메시지전달 주기마다 수행되어야 하므로, 제안하는 프로토콜이 더욱 효율적임을 알 수가 있다.

6. 결 론

본 논문은 센스 네트워크에 효율적인 TESLA를 기반으로 차량과 기지국이 GPS를 효과적으로 전달 할 수 있는 프로토콜을 제안하였다. 모든 메시지가 relay될 수 있는 것을 전제로 하고 있는 TESLA와는 달리 기지국 내에서 송수신되는 모든 메시지를 차량과 기지국이 관찰 할 수 있는 특성에 효과적이도록 해쉬 체인값을 메시지와 함께 전송해도 메시지 인증을 안전하게 수행할 수 있는 양방향 인증 방법을 제안하고 있다.

제안하는 프로토콜을 사용해서 GPS를 안전하게 전송하면, 주변에서 움직이는 모든 차량에 대한 위치 정보를 운전자가 실시간으로 파악할 수 있는 서비스를 제공할 수 있다. 그러므로 운전자가 백미러나 사이드 미러를 확인하는 횟수를 줄일 수 있으므로, 운전시에 주변환경 인식의 부재로 인해서 발생할 수 있는 여러 위험들을 감소시키고, 운전자의 안전 운행에 대한 효과적인 보조지표를 제공할 수 있을 것으로 기대한다.

참고문헌

[1] M. Jakobsson, "Fractal hash sequence representation and traversal", ISIT, 2002.

[2] M.Jakobsson, "Almost optimal hash sequence traversal", FC'02, LNCS, 2002.

[3] D. H. Yum, J. W. Seo, S. Eom, and P. J. Lee, "Single-layer fractal hash chain traversal with almost optimal complexity", CT-RSA, 2009.

[4] T. Leinmuller, L.Buttyan, J. Hubaux, F. Kargl, R. Kroh, P. Papadimitratos, M. Raya, and E. Schoch, "SEVECOM - Secure vehicle communication", IST Mobile and Wireless Communication Summit, 2006.

[5] G. Calandriello, P. Papadimitratos, J. Hubaux, and A. Lioy, "Efficient robust pseudonymous authentication in VANET", ACM International workshop on Vehicular ad hoc networks, 2007.

[6] A. Studer, F. Bai, B. Bellur, and A. Perrig, "Flexible, extensible, and efficient VANET authentication", ESCAR, 2008.

[7] J. Hubaux, S. Capkun, and J. L. Epfl, "The Security and privacy of smart vehicles", IEEE Computer Society, 2004.

[8] A. Perrig, R. Canetti, D. Song, and D.Tygar, "The TESLA broadcast authentication protocol", RSA Seucirty, 2002.

[9] http://www.cryptopp.com/benchmarks.html

[10] Michael Lee O'Connor "Carrier-phase differential GPS for automatic control of land vehicles", Doctor Thesis, 1997.

[11] "GALILEO, misstion high level definition", European Commission ESA, Sep, 2002.

[12] 장재득, 박형준, "3G LTE 이동통신 시스템 단말 플랫폼 기술 동향과 전망", 전자통신동향분석, 2월, 2008.

저 자 소 개

조 국 래

2004년 : 부경대학교 컴퓨터공학과 학사.
2006년 : POSTECH 정보통신학과 석사.
현재, DGIST 연구원.
관심분야 : 차량통신보안, 임베디드 소프트웨어.
Email : kookrae@postech.ac.kr

손 종 욱

2004년 : 경북대학교 전자전기컴퓨터학과 학사.
2006년 : KAIST 전자과 석사.
현재, DGIST 연구원.
관심분야 : 임베디드소프트웨어, 차량용통신.
Email : jwson@dgist.ac.kr

조 희 섭

1999년 : 경북대학교 전자공학과 학사.
2001년 : 경북대학교 전자공학과 석사.
현재, DGIST 선임연구원.
관심분야 : 임베디드하드웨어, Real-time OS.
Email : mozart73@dgist.ac.kr