지상파 UHD 방송 기반 재난경보메시지 서명 및 검증 기능 구현

조숙희, 배병준, *오성흔, *이명호
한국전자통신연구원, *㈜디지캡
{shee, 1080i}@etri.re.kr, *{shoh, mhlee}@digicaps.com

Terrestrial UHD broadcasting based signing and verification of emergency alerting message

Sukhee Cho, Byungjun Bae, *Sungheun Oh, and *Myoungho Lee
ETRI, *DigiCAP

요 약

공익을 목적으로 제공되는 재난경보 서비스에서 누군가의 해킹으로 잘못된 재난경보 메시지가 전달되면 사회적으로 큰 혼란이 발생할 수 있다. 본 논문에서는 지상파 UHD 방송을 통한 재난경보 메시지가 해킹으로부터 안전하게 보호될 수 있도록 ATSC 3.0 표준에 정의된 서명 방식을 기반으로 하여 보안 기능을 구현하는 방안을 제안한다.

1. 서론

지진, 화재, 미세먼지, 바이러스 등 여러 종류의 재난재해가 증가됨에 따라 방송통신망을 이용한 재난경보서비스가 보편적 서비스로 중요한 역할을 담당하고 있다. 방송망을 통한 재난경보 서비스는 무료 보편적 사회안전 서비스로서 공공영역에서의 재난정보 전달 효과를 기대할 수 있다. 이에, 우리나라는 2019 년 9 월 23 일부터 수도권을 대상으로 지상파 UHD 방송을 이용한 재난경보 시범서비스를 시작하였다. 또한, 2021 년에는 광역시, 2022 년에는 전국 시·군으로 단계적으로 확대할 계획이며, 옥외전광판과 버스, 지하철, 장애인 등 공공장소 및 취약계층 거주 시설에 수신기를 우선적으로 도입할 예정이다.

지상파 UHD 재난경보 서비스는 행정안전부, 기상청 및 지자체 등 재난경보발령기관에서 발령한 재난정보를 수신하여 UHD 방송 신호로 변환하여 송출하고, 옥외 전광판, 지하철, 버스 등에 설치된 UHDTV 또는 재난경보 전용수신기가 UHD 방송신호를 수신하여 재난정보를 표출하게 된다.

미디어 매체가 다양해짐에 따라 콘텐츠 보호의 중요성이 더욱 부각되고 있으며, 특히 공익을 목적으로 제공되는 재난경보 서비스에서 누군가의 해킹으로 잘못된 재난경보 메시지가 전달되면 사회적으로 큰 혼란이 발생할 수 있다. ATSC A/360 표준은 MPEG-DASH 콘텐츠 전달을 위한 콘텐츠 보호, ATSC 3.0 애플리케이션 코드 인증과 방송 시그널링 메시지 인증 등 콘텐츠 및 데이터 흐름을 보호하기 위한 규격을 정의하고 있다[1]. 이에 본 논문에서는 지상파 UHD 방송을 통한 재난경보 메시지가 해킹으로부터 안전하게 보호될 수 있도록 ATSC 3.0 표준에 정의된 서명 방식을 기반으로 하여 보안 기능을 구현하는 방안을 제안한다.

본 논문의 구성은 다음과 같다. 2 절에서는 ATSC 3.0 표준에서 정의하는 AEAT 보안과 관련된 표준을 살펴본 후, 3 절에서는 이러한 표준을 기반으로 지상파 UHD 재난방송 송수신 시스템에서 AEAT 서명 및 검증을 위한 기능 설계를 제안하고, 4 절에서는 제안한 기능 설계를 기반으로 구현한 결과를 확인한다.

마지막으로 5 절에서는 본 논문에 대한 결론을 맺는다.

2. ATSC 3.0 기반 AEAT 보호

지상파 UHD 방송에서 재난과 관련한 메시지 및 미디어 정보는 AEAT (Advanced Emergency Alert Table)에 의해 전달되므로, AEAT 를 보호하는 방안을 검토한다. 현재까지 국내 UHD 방송에서는 ATSC 3.0 보안과 관련한 기술을 국내 표준으로 도입하지 않고 있다. ATSC A/360 표준에서는 ATSC 3.0 애플리케이션 코드와 방송 시그널링 메시지에 대한 보안을 위해 PKI 기반의 디지털 서명 방식을 이용한다. 이에 따라, 지상파 UHD 재난방송에서는 그림 1 에 나타낸 바와 같이 AEAT 는 저레벨 시그널링(LLS)의 일종이므로 디지털 서명 방식을 이용하여 보안을 유지할 수 있다[2].

디지털 서명에 대한 정보는 그림 1 에 나타낸 바와 같이 LLS 테이블에 포함되는 SignedMultiTable(LLS_table_id OxFE)와 CertificationData(LLS_table_id 0x06)에 의해 전달된다. 이들은 각각 A/331 과 A/360 표준에 정의되어 있다[1][3].

Syntax	No. of Bits	Format
LLS_table() {		
LLS_table_id	8	uimsbf
LLS_group_id	8	uimsbf
group_count_minus1	8	uimsbf
LLS_table_version	8	uimsbf
switch (LLS_table_id) {		
case 0x01:		
SLT	var	
break;		
case 0x02:		
RRT	var	
break;		
case 0x03:		
SystemTime	var	
case 0x04:		
AEAT	var	
break;		
case 0x05:		
OnscreenMessageNofification	var	
break;		
case 0x06:		
CertificationData	Var	
break;		
case 0x80:		
VIT	Var	
break;		
case 0xFE:		
SignedMultiTable	Var	
break;		
default:		
Reserved	var	
}		
}		

<그림 1> LLS 비트스트림 구조

SignedMultiTable 은 그림 2 에 나타낸 바와 같이 LLS 테이블 중 하나로 구성된다. SLT, RRT 등 LLS 테이블에 포함된 각각의 시그널링에 대해 signature() 필드를 통하여 디지털 서명된 서명값을 삽입할 수 있다. signature () 필드는 A/360 표준에서 정의하는 CMS Signed Data 로 채워진다.

CMSSignedData 는 RFC5652 규격에 준수하여, 서명자정보(SignerInfo), 사용자 ID(SubjectKeyIdentifier), 서명값과 Hash 값을 생성하는 Hash 함수를 나타내는 Message Digest Algorithm 정보가 포함된다[4].

CertificationData 는 SignedMultiTable 을 검증하기 위해 필요한 X.509 인증서들, OCSP(online certificate status protocol; 인증서 유효성 정보) 응답, CertificationData 자체에 대한 서명 값을 포함하는 LLS 테이블(XML 포맷)이다[1].

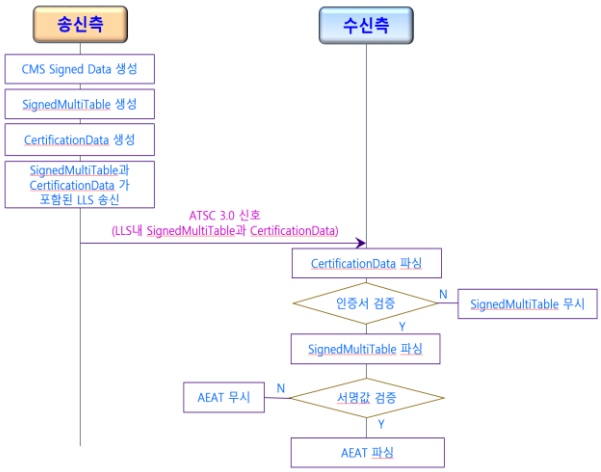
Syntax	No. of Bits	Format
SignedMultitable() {		
LLS_payload_count	8	uimsbf
for (i=0; i<LLS payload count; i++) {		
LLS_payload_id	8	uimsbf
LLS_payload_version	8	uimsbf
LLS_payload_length	16	uimsbf
LLS_payload()	var	
}		
signature_length	16	uimsbf
signature()	var	uimsbf
}		

<그림 2> SignedMultiTable 비트스트림 구조

3. AEAT 서명 및 검증 기능 설계

ATSC 3.0 에서 정의하는 시그널링 메시지에 대한 서명/검증은 PKI(Public Key Infrastructure) 기반의 비대칭형 암호화시스템을 기반으로 한다. 즉, 송신측이 평문 데이터의 Hash value 를 송신측의 개인키로 서명한 후, 공개키를 포함하는 인증서, 서명값을 평문 데이터와 함께 수신측에게 전달한다. 수신측은 평문 데이터에서 자체 계산한 Hash value 와 송신측의 공개키를 이용하여 전달된 서명값에서 구한 Hash value 가 동일한지를 검증하여 진위여부를 판별한다. 따라서, 본 논문은 지상파 UHD 재난방송 신호의 진위 여부를 판별하기 위한 AEAT 의 디지털 서명 및 검증 기능을 설계하고 구현한다.

그림 3 은 지상파 UHD 재난방송 시스템에서의 AEAT 디지털 서명 및 검증 절차를 나타낸다.



<그림 3> AEAT 디지털 서명 및 검증 절차

지상파 UHD 재난방송 송신시스템에서의 AEAT 디지털 서명 기능을 수행하는 모듈은 아래와 같이 CMS Signed Data 생성, SignedMultiTable 생성, CertificationData 생성 및 LLS 업데이트 기능을 순차적으로 수행한다.

① CMS Signed Data 생성

우선 LLS 를 입력받아 AEAT 를 추출한다. 추출된 AEAT 에 대해 Message Digest Algorithm 을 이용하여 Hash value 를 생성한다. 생성된 Hash value 를 A/360 규격 5.2.2.1 절에 기술된 서명 알고리즘을 사용하여 사전에 저장되어 있는 개인키로 서명하여 서명값을 생성한다. 서명이 이루어진 시간정보(SigningTime), 서명자 식별자(SubjectKeyIdentifier), 서명값 및 서명에 사용된 알고리즘 정보를 포함하는 CMS Signed Data(RFC 5652)를 생성한다.

② SignedMultiTable 생성

AEAT 를 SignedMultiTable 의 LLS_payload()에 삽입하고, 생성한 CMS Signed Data 를 signature()에 삽입한다. 그리고 그림 2 에 나타낸 SignedMultiTable 규격 내 다른 필드들에 대한 값을 생성한다. 그림 4 는 LLS 내 AEAT 에 대해서만 서명이 있는 경우에 대한 SignedMultiTable 생성 예제를 나타낸다.

③ CertificationData 생성

A/360 규격 5.2.2.2 에 명시된 CertificationData 규격에 따라 PKI 기반 X.509 인증서, 인증서에 대한 상태정보(OCSPResponse) 등의 필드값을 생성한다. PKI 기반의 X.509 인증서에는 공개키, 공개키 주인, 공개키 보증인, 유효기간 등의 정보가 포함되어 있다. PKI 기반의 인증서와 개인키는 송신시스템 내의 키 저장소에 저장되어 있는 것을 사용한다. 그리고 CertificationData 자체에 대한 메시지 인증을 위한 서명값이 CertificationData 의 CMSSignedData 에 포함된다. 이 서명값을 검증하기 위한 인증서도 같이 포함된다.

④ LLS 업데이트

입력받은 LLS 내에 SignedMultiTable 과 CertificationData 가 포함되도록 업데이트하여 출력한다.

✓LLS_payload_count는 1로 설정
✓LLS_payload_id는 LLS_table_id의 값, 0x04(AEAT 값)로 설정
✓LLS_payload_version은 LLS_table_version 값으로 설정
✓LLS_payload_length는 LLS_payload() 길이를 bytes로 표시
✓**LLS_payload()에 AEAT 삽입**
✓Signature_length는 signature() 길이를 bytes로 표시
✓**Signature()는 CMS Signed Data (A/360 5.2.2.3)**

<그림 4> SignedMultiTable 생성 예제

지상파 UHD 재난방송 수신시스템에서의 AEAT 검증 기능을 수행하는 모듈은 아래와 같이 1) LLS 내 CertificationData 와 SignedMultiTable 파싱 기능과 서명 검증 기능을 수행한다.

① LLS 내 CertificationData 와 SignedMultiTable 파싱

CertificationData 내에 있는 PKI 기반 X.509 인증서를 추출하고, Root CA 인증서를 이용하여 CertificationData 내에 있는 PKI 기반 X.509 인증서들에 대해 체인 검증을 수행한다. Root CA 인증서는 사전에 수신기에 저장되어 있으며, CertificationData 내 인증서들이 유효한지를 검증한다. 검증이 성공적으로 이루어지면 SignedMultiTable 을 파싱하고, 반대로 검증에 실패하면 SignedMultiTable 을 무시한다. 검증이 성공한 경우는 SignedMultiTable 내의 signature()에 포함되어 있는 시간정보, 서명자 식별자, 서명값 등 필드값을 추출하고, AEAT 서명에 사용된 서명자 식별자와 동일한 값을 갖는 PKI 기반 X.509 인증서를 CertificationData 내에서 찾는다.

② 서명 검증

CertificationData 내에서 찾은 PKI 기반 X.509 인증서 내의 공개키를 이용하여 서명값으로부터 Hash value 를 복원하고, LLS_payload()에 포함되어 있는 AEAT 에 대해 hash 함수를 이용하여 Hash value 를 생성한다. 복원된 Hash value 와 생성한 Hash value 가 동일하면 서명 검증에 성공하는 경우로 AEAT 를 추출하고, 그렇지 않을 경우는 서명 검증에 실패하는 경우로 AEAT 를 버린다.

4. AEAT 서명 및 검증 기능 구현 결과

지상파 UHD 재난방송 신호의 보안을 위하여 송신시스템에서 AEAT 신호에 대한 디지털 서명 기능과 수신시스템에서 AEAT 신호를 검증하는 기능을 구현하여 기존 지상파 UHD 방송시스템과 연동하여 필드테스트를 수행하여 기능 검증을 수행하였다.

그림 5 는 실험을 위하여 임의로 생성한 지진발생 재난에 대한 AEAT 내용을 나타내며, 이 신호를 gzip 으로 압축하여 서명값을 생성하였다. 그림 6 은 그림 5 의 AEAT 에 대하여 생성한 CMS Signed Data 내용을 나타낸다.

```
<?xml version="1.0" encoding="utf-8"?>
<AEAT xmlns="tag:atsc.org,2016:XMLSchemas/ATSC3/Delivery/AEAT/1.0/">
  <AEA xmlns="tag:atsc.org,2016:XMLSchemas/ATSC3/Delivery/AEAT/1.0/"
    xmlns:ns2="http://www.digicaps.com/eas/" aeaId="KR.TI0402_2019001736_EQI_1"
    issuer="EDBS" audience="public" aeaType="alert" priority="4" wakeup="false">
    <Header effective="2019-07-23T16:14:07+09:00" expires="2019-07-23T16:20:07+09:00">
      <EventCode type="EDBSAEAS">EQI</EventCode>
      <EventDesc xml:lang="ko">지진정보</EventDesc>
      <EventDesc xml:lang="en">Earthquake information</EventDesc>
      <EventDesc xml:lang="zh">地震信息</EventDesc>
      <Location type="juso">2600000000, 5000000000, 1100000000, 2800000000, 2700000000,
        4400000000, 2900000000, 4500000000, 3100000000, 3000000000, 4600000000,
        4700000000, 4300000000, 4800000000, 4200000000, 4100000000, 3600000000</Location>
      <Location type="circle">37.5,126.5 999.0</Location>
    </Header>
    <AEAText xml:lang="ko">오늘 11시 6분 인천 중구 서북서쪽 11km 지역에서 규모 5.0 지진 발생</AEAText>
    <AEAText xml:lang="en">(Earthquake information) 2019-07-29 14:06:23(KST), M 5.0,
      11km WNW of Jung-gu, Incheon, Korea (onshore)</AEAText>
    <AEAText xml:lang="ja">地震情報</AEAText>
    <AEAText xml:lang="zh">(地震信息) 2019-07-29 14:06:23(KST), 震級 5.0, 韓国 仁川 中区
      西北西 11km (地区)</AEAText>
    <Media xml:lang="ko" mediaDesc="진앙지도이미지" url="Map_2019009762_102_1_2.jpg"
      alternateUrl="http://203.247.79.201:8091/noti/IMG/2019/Map_2019009762_102_1.jpg"
      contentType="image/jpeg"/>
  </AEA>
</AEAT>
```

<그림 5> 실험에 사용한 AEAT 내용

```
CMS_ContentInfo:
  contentType: pkcs7-signedData (1.2.840.113549.1.7.2)
  d.signedData:
    version: 3
    digestAlgorithms:
        algorithm: sha256 (2.16.840.1.101.3.4.2.1)
        parameter: <ABSENT>
    encapContentInfo:
      eContentType: pkcs7-data (1.2.840.113549.1.7.1)
      eContent: <ABSENT>
    certificates:
      <ABSENT>
    crls:
      <ABSENT>
    signerInfos:
        version: 3
        d.subjectKeyIdentifier:
          0000 - a2 00 e7 24 17 36 3f 87-a9 f7 d9 79 f2 2d dc   ...$.6?....y.-.
          000f - 00 2e 12 4b 32                                 ...K2
        digestAlgorithm:
          algorithm: sha256 (2.16.840.1.101.3.4.2.1)
          parameter: <ABSENT>
        signedAttrs:
            object: contentType (1.2.840.113549.1.9.3)
            set:
              OBJECT:pkcs7-data (1.2.840.113549.1.7.1)

            object: signingTime (1.2.840.113549.1.9.5)
            set:
              UTCTIME:Jul 23 09:10:41 2019 GMT

            object: messageDigest (1.2.840.113549.1.9.4)
            set:
              OCTET STRING:
                0000 - cf 5c 67 af db 13 a1 8b-ac 1c e7 f6 35   .\g.........5
                000d - 4c 42 c1 2f c2 53 fb 37-f2 4d f2 58 89   LB./.S.7.M.X.
                001a - 99 10 31 4e a6 8f                        ..1N..
        signatureAlgorithm:
          algorithm: sha256WithRSAEncryption (1.2.840.113549.1.1.11)
          parameter: NULL
        signature:
          0000 - 34 52 8f 9b a5 71 9d 5f-e1 c1 52 a8 4b a3 52   4R...q._..R.K.R
          000f - 14 06 fe 4f f2 92 08 47-8f c6 dc ba a3 cb 8f   ...O...G.......
          001e - 8a 31 49 91 cf aa 5e a4-5a d5 6f 67 a6 6a 55   .1I...^.Z.og.jU
          002d - 2c 61 74 ab 14 a9 47 0b-92 33 a5 b9 33 86 89   ,at...G..3..3..
          003c - 38 44 9d 19 7c 49 bc 43-9b 14 b3 90 30 6c 16   8D..|I.C....0l.
          004b - 95 2e c6 f5 db c3 3d 8d-7e e6 dd 16 d4 c2 7e   ......=.~.....~
          005a - 57 b1 e5 7b 3b 6e e9 e9-ff 20 b5 a4 6f d9 97   W..{;n... ..o..
          0069 - 95 31 ac 20 7c 2e 10 78-3e 1b 86 66 18 a6 86   .1. |..x>..f...
          0078 - 37 bb 49 98 0f 64 84 cd-c9 41 26 6e 05 9d 83   7.I..d...A&n...
          0087 - 58 d9 d5 d1 1d 71 33 da-ed 60 cf a6 b2 15 72   X....q3..`....r
          0096 - b5 51 52 8e 0c 70 3b 37-9d 17 d8 8e 0e ff 9b   .QR..p;7.......
          00a5 - f8 9d 40 78 e8 42 1f 71-e3 03 f8 81 fc 48 4b   ..@x.B.q.....HK
          00b4 - e8 ce 81 5d 48 95 0e fd-8e de b2 58 1e 52 a8   ...]H......X.R.
          00c3 - 0a 35 b9 7c a6 40 d4 3c-39 24 80 75 25 16 04   .5.|.@.<9$.u%..
          00d2 - d1 da b0 80 08 95 5e e4-4f 4b d8 3b 26 21 a8   ......^.OK.;&!.
          00e1 - 09 e2 af f1 3b 8f 34 d7-b2 cc 40 e6 c8 2a e5   ....;.4...@..*.
          00f0 - 30 63 48 2f 5c 2c a2 a8-ef 95 20 5a 2b 6a de   0cH/\,.... Z+j.
          00ff - 16                                             .
        unsignedAttrs:
          <ABSENT>
```

<그림 6> CMS Signed Data 결과

5. 결론

본 논문에서는 지상파 UHD 방송을 통한 재난경보 메시지가 불법적인 공격으로부터 안전하게 보호될 수 있도록 ATSC 3.0 표준에 정의된 방안 중에서 LLS 내에서 전달되는 AEAT 에 대한 서명 및 검증 방안을 기술하였다. 또한 이에 대한 기능 설계 및 구현 결과를 제시하였다. 향후에는 AEAT 뿐만 아니라 SLS(service level signaling), NRT 보호를 통한 재난 미디어 보안 방안에 대하여 검토해 나갈 계획이다.

Acknowledgement

이 논문은 2020 년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (2018-0-01364, 재난피해 저감을 위한 지상파 UHD 기반 재난방송 서비스)

참고문헌

[1] A/360: ATSC 3.0 Security and Service Protection, 2019. 08.

[2] TTA.KO-07.0127/R4, 지상파 UHDTV 방송 송수신 정합, 2019. 12.

[3] A/331: Signaling, Delivery, Synchronization, and Error Protection, 2018. 08.

[4] RFC 5652: Cryptographic Message Syntax (CMS), 2015. 10.