SyncML 인증(Authentication) 기능 구현

김창회[0], 류수희, 최 훈
충남대학교 컴퓨터공학과 분산시스템 연구실
{kchoe, shryu, hchoi}@ce.cnu.ac.kr

Implementation of SyncML Authentication

Chang-Hoe Kim[0] ,Soo-Hee Ryu, Hoon Choi
Dept. of Computer Engineering, Chungnam National Univ.

요 약

모바일 통신에 대한 보안 문제가 모바일 시대로 가는 중요한 요소가 되었다. SyncML protocol[1] 역시 보안에 예외일 수는 없다. 이 논문은 다양한 인증 스키마를 적용할 수 있는 SyncML Authentication을 소개하고 현재 SyncML Specification 1.0에서 요구하는 Basic Authentication과 MD5 digest access authentication[3] 의 구현방안과 그 구현 사례를 제시한다.

1. 서 론

무선 기술이 빠르게 발전함에 따라 우리의 생활 및 업무 방식도 급속하게 변신하고 있다. 첨단 무선통신 기술의 발전으로 모바일 비즈니스가 때와 장소를 가리지 않고 모든 장비에서 수행될 것이다.

이제 포스트 PC 시대의 도래로 PDA, 양방향 호출기, 태블릿(tablet), 스마트폰, WAP폰 등 다양한 휴대용 단말기들이 쏟아져 나오고 있다. 최근 지원 제품의 모습까지 드러내고 있는 근거리 무선통신 기술인 블루투스, 초고속 무선데이터통신기술(HDR), IEEE 802.1x 등을 비롯한 다양한 무선 기술들이 1Mbps에 달하는 대역폭을 통해서 모든 장치들이 인터넷이나 인트라넷에 하루 24시간 연결되는 시대가 올 것이다.

그러나 다양한 환경의 장치들은 상호 호환성이 없고, 환경에 종속적이기 때문에 우선 이를 극복할 방법이 요구되어 왔다. 이런 필요성에 부응해 등장한 SyncML은 이기종 환경에서의 데이터 상호호환성을 해결하는 대안으로 인식되고 있다. 이기종 단말기간 동기 문제 뿐만 아니라, 휴대용 장비를 사용해서 기업의 중요한 정보나 기밀 정보에 액세스하거나 비즈니스 트랜잭션을 수행할 경우의 보안 문제가 이슈로 떠오르고 있다 .

그래서 이 논문에서는 SyncML 프로토콜에 정의된 기본 인증을 포함하여 다양한 인증 스키마를 적용할 수 있는 프레임웍을 구현하고, 향후 보다 강력한 인증이 필요할 경우 이 프레임웍을 이용해서 다른 인증 메커니즘을 적용할 수 있는 Authentication Handler 를 소개한다.

2. SyncML에서의 인증절차

SyncML은 데이터 동기화를 위한 프레임웍을 제공할 뿐 새로운 보안 스키마는 정의하지 않는다. 대신 신청에 의한 인증(*challenge authentication*), 인증(authentication), 권한부여(authorization)를 위한 프레임웍을 제공한다. 그래서 송신자와 수신자는 다양한 보안 메커니즘을 추가해서 사용할 수 있다. 본 연구팀에서는 이러한 프레임웍을 제공하기 위하여 SyncML 기능 모듈이외에 "Authentication Handler"라는 별도의 모듈로 두어, 다른 모듈과 독립적으로 추가될 수 있도록 하였다. 이 프레임웍을 테스트하기 위하여 SyncML specification에서 기본으로 제공하는 "Basic Authentication"과 "MD5 digest access authentication" 을 구현하였다.

[redacted]

2.1 기본 인증(Basic Authentication)

SyncML에서 Basic Authentication이란 "송신자의 Userid :password"의 형태의 문자열을 Base64 character encoding 을 사용해서 메시지 인증으로 사용하는 방법이다. Base64는 바이너리 데이터를 아스키 텍스트로 변환하거나, 그 반대로 변환하는 인코딩 방법으로서, MIME에 의해 사용되는 방법들 중 하나이다. Base64는 4개의 7 비트 아스키 문자로서 표현되도록 원래의 데이터에서 각 3 바이트씩을 4개의 6 비트 단위로 나눈다. 이것은 파일 크기를 대체로 원래보다 약 1/3 정도 증가시킨다. 구현이 간단한 대신 네트웍 공격에는 취약하여 추가적인 보안이 고려되어야 한다.

2.2 MD5 인증(MD5 Digest Access Authentication)

MD5 알고리즘은 임의 길이의 입력을 받아 이를 128bit의 해쉬 결과값으로 변환하는 해쉬함수 기반의 암호 알고리즘이다. SyncML에서는 송신자의 userid:password:nonce string의 형태의 문자열을 MD5에 사용한다. Nonce string은 현재 세션에만 유효한 것으로 다음 세션에서는 바뀌어야 한다. SyncML에서의 특징은 신청에 의한 인증기능(challenge authentication)을 지원하는 것이다.

MD5 또한 네트웍 공격의 우려가 있으므로, 더욱 견고한 해쉬함수(hash function)을 필요로 하는 디지털 사인등의 응용에서는 MD5가 더 이상 쓰이지 않을 것이라는 견해가 지배적이며 이의 대용으로 SHA-1[5]이나 RIPEMD-160[6]등의 알고리즘이 제안되고 있다. 따라서 추후 연구는 이를 지원하는 방향이 될 것이다.

3. SyncML에서 Authentication 기능 구현

SyncML specification을 따르는 구현을 위해서는 "Basic"과 "MD5 Digest"는 반드시 지원을 해야 한다. 따라서 본 연구팀은 SyncML specification을 따르면서 추가로 어플리케이션 독립적인 기능을 수행하는 Authentication Handler 모듈을 작성하였다. 간단한 동작과정은 다음과 같다.

커맨드나 메시지가 응답으로 "Unauthorized"나 "Authentication *required*"를 받으면 인증이 필요하다. 이럴 경우상태(*Status*)커맨드는 반드시 인증 신청을 하는 Chal element 를 포함해야 한다. 각각의 경우는 다음과 같다.

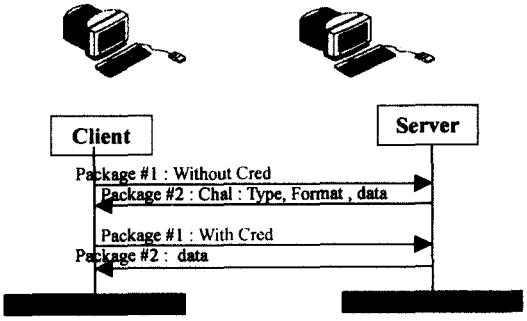
Basic일 경우: 타입, 포맷 MD5일 경우: 타입, 포맷 , nextnonce

장치는 이때 인증을 위한 송신자의 인증 정보를 가지는 Cred element를 포함해서 재 전송한다

3.1 동작 과정

3.1.1 인증신청 기능이 있는 기본인증

그림 1은 Basic scheme의 과정을 보여준다.



< 그림 1 인증 신청기능이 있는 기본인증 >

Package #1은 클라이언트가 송신자의 인증 정보를 가지는 Cred커맨드없이 서버에 접근하려 하는 예제이다. Package #1은 아무런 인증 정보가 없기 때문에 서버와 곧장 동기화를 할 수는 없다.

```
- <SyncML>
    <SyncHdr>.....</SyncHdr>
    <SyncBody>...</SyncBody>
  </SyncML>
```

<그림 2 기본인증에서 인증정보 없는 패키지 1>

인증정보가 없는 메시지를 받았기 때문에 서버는 인증을 위한 시도를 위하여 Package #2에서 Chal을 보낸다. 이때 서버는 'Chal'태그의 메타정보로 타입: 'syncml:auth-basic', 포맷 :'b64'를 설정해서 Chal을 보낸다.

```
- <SyncML>
    <SyncHdr>.....</SyncHdr>
  - <SyncBody>
    - <Status>
      - <Chal>
        - <Meta>
            <Type xmlns="syncml:metinf">syncml:auth-basic</Type>
            <Format xmlns="syncml:metinf">b64</Format>
          </Meta>
        </Chal>
        <Data>407</Data>
        <!-- Credentials missing -->
      </Status>
      ...
    </SyncBody>
  </SyncML>
```

<그림 3 기본인증에서 인증정보 없는 패키지 2>

```
- <SyncML>
  - <SyncHdr>
      .....
    - <Cred>
      - <Meta>
          <Type xmlns="syncml:metinf">syncml:auth-basic</Type>
        </Meta>
        <Data>QnJ1Y2UyOk9oQmVoYXZl</Data>
        <!-- base64 formatting of "userid:password" -->
      </Cred>
    </SyncHdr>
    <SyncBody>...</SyncBody>
  </SyncML>
```

<그림 4 기본인증에서 인증정보를 갖는 패키지 1>

클라이언트는 반드시 메타 타입이 "syncml:auth-basic"인 Cred와 인코드된 데이타를 Package #1으로 재전송 해서 다시 인증을 한다.

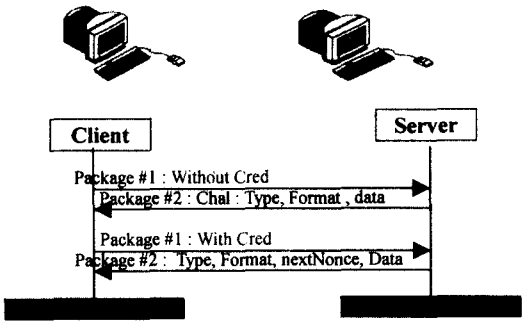
```
- <SyncML>
    <SyncHdr>.....</SyncHdr>
  - <SyncBody>
    - <Status>
        .....
        <Data>212</Data>
        <!-- Authenticated for session -->
      </Status>
      ...
    </SyncBody>
  </SyncML>
```

<그림 5 기본인증에서 인증정보를 갖는 패키지 2>

이때 서버는 송신자의 인증 정보를 가지는 Cred를 받아들이고 남은 세션에서 더 이상 인증이 필요하지 않음을 알리는 '212'코드를 갖는 status 명령어를 내보내게 된다.

3.1.2 인증신청 기능이 있는 MD5인증

그림 6 는 인증 신청이 수반되는 MD5 authentication 의 동작을 보여준다.



< 그림 6 인증 신청기능이 있는 MD5 인증 >

Package #1은 클라이언트가 송신자의 인증 정보를 가지는 Cred없이 서버에 접근하려 하는 예제이다. Package #1은 아무런 인증 정보가 없기 때문에 서버와 곧바로 동기화를 할 수는 없다.

```
- <SyncML>
    <SyncHdr>.....</SyncHdr>
  - <SyncBody>
    - <Status>
      - <Chal>
        - <Meta>
            <Type xmlns="syncml:metinf">syncml:auth-md5</Type>
            <Format xmlns="syncml:metinf">b64</Format>
            <NextNonce
              xmlns="syncml:metinf">Tm9uY2U=</NextNonce>
          </Meta>
        </Chal>
        <Data>407</Data>
        <!-- Credentials missing -->
      </Status>
      ...
    </SyncBody>
  </SyncML>
```

<그림 7 MD5인증에서 인증정보 없는 패키지 2>

인증정보가 없는 메시지를 받았기 때문에 서버는 인증을 위한 시도를 위하여 Package #2에서 인증 정보가 빠졌음을 알리는 Chal 명령어를 보낸다. 이때 서버는 'Chal'태그의 메타정보로 타입: 'syncml:auth-md5', 포맷 :'b64' , 임시 문자열(nextnonce): 'Tm9uY2u'를 설정해서 Chal을 보낸다.

클라이언트는 반드시 타입 태그에 "syncml:auth-md5"를 넣고 데이터에는 MD5로 인코드된 데이터를 입력한 Cred를 갖는 Package #1을 재전송 해야 한다.

```
- <SyncML>
  - <SyncHdr>
      .....
    - <Cred>
      - <Meta>
          <Type xmlns='syncml:metinf'>syncml:auth-md5</Type>
        </Meta>
          <Data>NTI2OTJhMDAwNjYxODkwYmQ3NWUxN2RhN2ZmYmJlMzk=</Data>
        <!-- Base64 coded MD5 digest of "Bruce2:ObBabaye:Nonce"  -->
      </Cred>
    </SyncHdr>
    <SyncBody>...</SyncBody>
  </SyncML>
```

<그림 8 MD5인증에서 인증정보를 갖는 패키지 1>

서버는 Cred를 받아들이고 세션을 인증하게 된다. 이때 서버는 "Basic"과는 다르게 Nextnonce값을 보내기 위해서 Chal 명령어를 포함한 Status 명령어를 보내야 한다.

```
- <SyncML>
    <SyncHdr>.....</SyncHdr>
  - <SyncBody>
    - <Status>
        .....
      - <Chal>
        - <Meta>
            <Type xmlns="syncml:metinf">syncml:auth-md5</Type>
            <Format xmlns="syncml:metinf">b64</Format>
            <NextNonce
              xmlns="syncml:metinf">LG3iZQhhdmKNHg==</NextNonce>
            <!-- this nonce is used at the next session.  -->
          </Meta>
        </Chal>
        <Data>212</Data>
        <!-- authenticated for session  -->
      </Status>
      ...
    </SyncBody>
  </SyncML>
```

<그림 9 MD5인증에서 인증정보를 갖는 패키지 2>

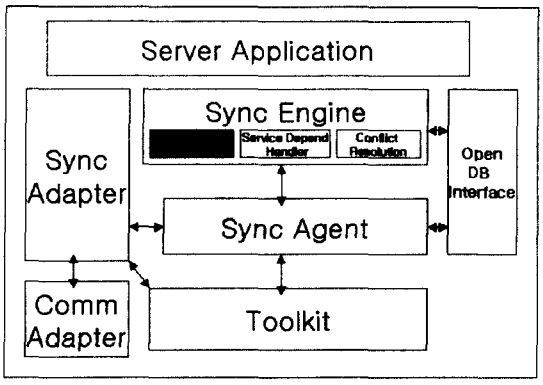
Status 의 코드로 '212'를 보내기 때문에 현재 세션에서 더 이상의 인증을 필요로 하지 않음을 의미한다.

3.2 Authentication Handler

어플리케이션 독립적인 모듈인 'SyncEngine'에 속하는 모듈로 전체 SyncML구조에서 독립적인 모듈이라고 할 수 있는 모듈이다. 본 논문에는 SyncML 표준 인증 타입만 구현을 했지만, 다른 인증 방법도 지원할 수 있는 프레임웍 역할을 한다.

Authentication Handler의 기능은 크게 세가지로 나눌 수 있다. 실제 구현은 C++ 클래스로 작성 되었는데 각각의 기능은 함수로 정의 되었다. 인증타입의 개수와 관계없이 세가지 함수에서 모든 인증을 처리하게 된다. 다른 모듈에서는 인증이 필요할 때 다음의 세 함수중 필요한 것을 사용하면 된다.

- 클라이언트에 대한 인증 : 다른 장치들과 동기화를 할 때 클라이언트가 적합한지를 체크하는 모듈이다. 이 논문에서는 'Auth_Client()'로 구현 되었다.
- DB에 대한 인증 : 클라이언트에 대한 인증 후에 DB에서, 접근할 때 쓰이는 인증이다. 일종의 보조 인증으로 현재는 인증정보가 없을 때는 인증성공으로 하고 있다. 이 기능은 'Auth_DB()'함수에 구현되어 있다.
- 인증정보 빌드(Build) : 클라이언트나 서버에 인증정보를 보낼 때 각 인증타입에 맞게 인증정보를 가지는 'Cred' 를 구성한다. 이 기능은 'Auth_Build()'에 구현되어 있다.



< 그림 10 SyncML 프레임웍>

현재 위 세개의 함수에는 "Basic"가 "MD5"에 대해서 구현되어있다. 이 세가지 기능외에 다른 기능이 필요할 때에는 클래스에 새로운 함수를 추가하고, 새로운 인증 타입이 추가될 때에는 위의 함수를 수정하면 된다. 이를 통해서 인증에 대한 프레임웍을 제공하는 것이다. 예를 들어 써드파티의 인증기관과 인증하려고 하면 위의 세개의 함수를 그대로 호출하면 된다. 다른 모듈은 기존함수를 그대로 호출하면 된다.

4. 결론 및 향후 연구과제

다양한 장치들의 동기화를 위해 사용되는 SyncML에서 서로 다른 장치들간의 인증은 필수가 되었다. SyncML은 specification 1.00에 나타난 기본적인 인증타입인 "Basic"과 "MD5"를 포함하여 다른 스키마를 가진 인증의 지원이 용이하다. 이는 인증 모듈을 따로 분리했기 때문에 가능한데, 지금까지 쓰여온 인증타입과 새로 추가될 인증타입을 볼 때 모듈 분리는 필수적이다. 본 논문에 사용된 모듈 분리 방법은 "Authentication Handler"의 모듈을 구현한 것이다. 이 모듈이 인증을 위한 프레임웍 역할을 한다.

본 논문에 소개된 "Basic"과 "MD5"는 현재 버전의 SyncML에서 표준으로 사용되는 인증타입이다. 다양한 장치들과 동기화를 한다고 할 때 공통으로 사용되는 인증타입이다. 앞으로 버전이 올라갈수록 더욱 많은 스키마를 제공할 것이다. 이때에는 단지 프레임웍에 추가함으로써 이를 지원할 수 있다.

추후연구에서는 보다 다양한 인증타입을 지원하는 "Authentication Handler"를 구현하고 현재 구현된 버전에 대한 문제점을 보안해야 할 것이다.

5. 참고문헌

[1]SyncML Initiative, SyncML Sync Protocol version 1.0.1, 2001.5.30.
[2]SyncML Initiative, SyncML Representation Protocol Version 1.0.1, 2001.5.30.
[3]Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321,April 1992.
[4]N. Borenstein ,"Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message odies",RFC1341, June 1992.
[5]ISO/IEC 10118-3:1998, "Information technology -Security techniques - Hash-functions - Part 3: Dedicated hash-functions," International Organization for Standardization, Geneva, Switzerland, 1998.
[6]U.S. Department of Commerce, "Secure Hash Standard", FIPS PUB 180-1, 17 Apr 1995.