情 報 保 護 學 會 誌
第 11 券 第 4 號, 2001. 8

XML 암호화 표준 동향

김 주 한*, 나 중 찬**, 문 기 영***

요 약

XML (eXtensible Markup Language) 암호화는 XML 문서 내용을 작성한 사용자에게만 구별 가능하고, 그 외의 사람들에게는 알아보기 힘들게 XML 문서를 변경하는 방법을 의미한다. W3C (World Wide Web Consortium)의 XML 암호화 표준화 그룹은 XML 문서와 그 일부분을 포함한 디지털 콘텐츠를 암호화 및 복호화하는 프로세스를 개발하고, 의도된 사용자만이 복호화할 수 있는 정보와 암호화된 내용을 표시하기 위한 XML 구문을 만드는 데 그 목적이 있다. 따라서 본 고에서는 XML 암호화에 대한 소개와 W3C's XML 암호화 표준화 동향을 살펴본다.

Ⅰ. 서 론

기존의 HTML(HyperText Markup Language) 한계를 극복하기 위해 만들어진 XML은 최근에 가장 각광 받는 언어 중에 한가지가 되었다. XML은 의미 없는 태그를 사용하던 HTML 한계를 넘어, 태그에 의미 있는 이름을 사용하게 됨으로써 문서의 분류 및 검색, 문서 내의 데이터 활용에 큰 혁신을 가져왔다. 또한 기존의 전자상거래 시스템은 각기 다른 종류의 문서를 사용함으로써 데이터의 이식성 문제로 인해 걸림돌이 되어왔다. 그러나 최근에는 XML 문서가 전자상거래의 표준 문서로 자리잡음으로써 그 영역과 활용이 점차 증가되고 있으며, 각각의 다른 전자상거래의 문서 형식들도 서로 호환 가능하게 됨으로써 활성화에 기여하고 있다.

그러나 여전히 전자상거래의 온라인 구매 방식 상 개인 정보의 유출을 걱정하는 사용자들이 증가하면서 보안 문제점이 크게 부각되어 왔다.

이러한 문제점을 해결하기 위해 최근에 등장한 것이 XML 문서에 대한 보안 서비스 제공을 고려하기 시작하였으며, 이를 위해 W3C's XML Digital Signature 표준화 그룹과 W3C's XML Encryption 표준화 그룹에서 XML 문서의 전자서명과 암호에 관한 표준화 작업을 진행하고 있다.

XML 암호화는 XML 문서의 내용이 의도된 사용자에게만 구별 가능하고, 그 외의 사람들에게는 알기 힘들게 XML문서를 변경하는 방법을 의미한다. W3C's XML 암호화 작업 그룹은 XML 문서와 그 일부분을 포함한 디지털 콘텐츠를 암호화/복호화하는 프로세스를 개발하고, 의도된 사용자만이 복호화할 수 있도록 정보들과 암호화된 내용을 표시하는 데 사용하는 XML 구문을 만드는 데 그 목적이 있다[1].

본 논문에서는 XML Encryption 표준화 그룹에서 진행하고 있는 XML 암호화에 관한 표준화 동향에 대해 소개한다. 제Ⅱ장에서는 W3C's XML Encryption 표준화 그룹에서 발표한 XML 암호화 요구 사항[2]과 2001년 6월에 발표한 XML Encryption Syntax and Processing[3]에서 나오는 XML 암호화 관련 표준에 대해 소개하며, 제Ⅲ장에서 결론을 맺는다.

Ⅱ. XML 암호화

W3C's XML Encryption 표준화 그룹의 실질적인 활동은 2000년도에 시작되었으며, 근래에 와서 XML 암호화 요구 사항과 XML 암호화 구문과 처리를 최근에 발표하는 단계이다. XML 암호화 요

* 한국전자통신연구원 차세대보안응용연구부 EC정보보호연구팀(juhankim@etri.re.kr)
** 한국전자통신연구원 네트워크보안연구부 능동프로토콜연구팀(njc@etri.re.kr)
*** 한국전자통신연구원 차세대보안응용연구부 EC정보보호연구팀(kymoon@etri.re.kr)
**** 본 연구는 정보통신부의 과제로 지원되었습니다.

구 사항은 XML 암호화 시스템 설계 및 구현 시에 요구되는 혹은 고려되어야 될 사항들을 정의한 것이다. XML 암호화 구문과 처리는 XML 암호화를 하기 위해 필요한 절차와 암호화 절차를 따라 문서와 비밀키 등을 각각 암호화를 하고 그 결과 및 암호화에 사용된 알고리즘이나 키에 대한 정보 등을 저장하기 위한 XML 형태에 대한 표준을 정의해 놓은 것이다.

이 장에서는 최근에 발표된 XML 암호화 요구 사항과 XML 암호화 구문 및 처리 등에 대해 설명한다.

1. XML 암호화 요구 사항

W3C's XML Encryption 표준화 그룹은 XML 암호화 표준을 설계 시 여러 가지 요구 사항들을 정의하고 있으며, 그 설계 원칙과 범위는 다음과 같다.

1) 암호화된 자원들에 대한 XML 표현은 참조 가능한 형태가 되어야 하고 별도의 엘리먼트형으로 나타내야 한다.

(1) 하나의 XML 문서에 대해 부분 혹은 전체를 암호화할 수 있어야 한다.

(2) 암호화된 데이터와 암호화에 관련된 정보를 분리해야 하고, 암호화된 데이터 영역을 암호화 정보에서 참조할 수 있는 (혹은 반대의 경우에도) 참조 메커니즘들을 제공해야 한다.

(3) 어떤 엘리먼트들이 이미 암호화되어 있는 XML 내용을 암호화하는 중복 암호화(super-encrypted data)를 허용해야 하며, 다른 암호화 데이터와 동일한 구문과 의미를 가져야 한다.

2) 암호 키 정보를 수신자에 전달하는 유연한 체계를 제공해야 한다.

(1) 대칭 암호 방식 혹은 비대칭 암호 방식으로 암호화되어 있는 키 값을 수신자에게 전달

(2) 알려진 키를 참조하기 위한 이름 및 URI 제공해야 한다. 이때 키 정보는 암호화된 데이터 표현형식의 일부분으로 포함하거나 혹은 외부로 참조할 수 있어야 하며, 또한 키들로 암호화한 데이터를 확인할 수 있어야 한다.

3) 디지털 콘텐츠 및 XML 문서에 대한 암호화 체계는 간단해야 한다.

(1) 어떻게 데이터가 암호화되어 있는지를 설명하는 어떠한 속성의 존재를 강제해서는 안되며, 복호화를 위해 필요한 정보만 제공되어야 한다. 암호 방식과 키 체계에 대해 미리 약속한 경우에는 암호화된 데이터와 그와 관계된 정보에 관한 효율적인 인코딩을 허용한다.

XML 암호화 표준의 요구 사항들은 위에서 설명한 설계 원칙 및 범위 중심으로 암호화 데이터 모델과 구문, 오브젝트, 절차, 알고리즘과 구조, 그리고 보안 등의 관점에서 제안되고 있다. 암호화 데이터 모델과 구문 부분은 암호화된 데이터를 확인하고 표현할 때 사용하는 XML 데이터 모델 및 구문에 대한 사항을, 오브젝트 부분은 암호화되는 데이터의 따른 사항을, 절차 부분은 암호화 및 복호화 절차에 따른 사항을, 알고리즘과 구조 부분에서는 XML 암호화 및 복호화에서 사용되는 인코딩, 압축, 암호 및 키 관련 알고리즘 등에 대한 사항을, 마지막으로 보안 부분에서는 XML 암호화가 어플리케이션에서 사용될 때 나오는 보안 문제에 대한 사항들을 언급하고 있다. 이러한 내용들은 위에서 설명한 설계의 원칙과 범위와 내용이 중첩되는 것들이 많기 때문에 본 논문에서는 따로 기술하지 않는다.

```
<EncryptedData (Id='')? (Type='')?>
  <EncryptionMethod/>?
  <ds:KeyInfo>
    <EncryptedKey/>?
    <ds:*/>?
  </ds:KeyInfo>?
  <CipherData>
    <CipherValue>
      (encrypted character data)
    </CipherValue>?
    <CipherReference URI=''/>?
  </CipherData>
</EncryptedData>
```

〔그림 1〕 EncryptedData 엘리먼트

2. XML 암호화 구문

암호 대상의 데이터는 XML 문서 자체, 하나의 XML 엘리먼트, 한 엘리먼트의 내용이 될 수 있다. XML 암호화 구문은 암호화한 결과를 포함하거나 URI(Uniform Resource Identifier)로 참조하고 있는 *EncryptedData* 엘리먼트로 표현된다. XML 엘리먼트나 엘리먼트의 내용을 암호화 경우는 암호화 대상 엘리먼트 혹은 엘리먼트 내용과 *EncryptedData* 엘리먼트와 교체되며, XML 문

서 전체를 암호화 경우에는 새로운 문서의 루트 엘리먼트가 되며, 이외의 데이터를 암호화하는 경우에는 새로운 문서의 루트 엘리먼트가 되거나 어플리케이션이 선택한 XML 문서의 하위 엘리먼트가 된다.

그림 1은 *EncryptedData* 엘리먼트를 간단한 형태로 나타낸 것이다.

CipherData 엘리먼트는 암호화된 데이터를 포함하거나 참조한다. 암호화된 데이터를 포함하는 경우는 이 데이터가 *CipherValue* 엘리먼트의 내용으로 들어간다. 그리고 암호화된 데이터를 참조하는 경우는 *CipherReference*의 URI 속성의 값을 기술한다.

그림 2는 암호화된 데이터를 포함하는 경우의 예제를 나타낸다.

신원 확인 정보 및 지급 방법에 관한 내용을 기술한 지급 정보에 대한 엘리먼트에 대한 구문을 나타내고 있다. John Smith는 $5,000USD의 한도를 가

```
<?xml version='1.0'?>
<PaymentInfo xmlns='http://example.org
                              /paymentv2'>
  <Name>John Smith<Name/>
  <CreditCard Limit='5,000' Currency
                                  ='USD'>
    <Number>
      4019 2445 0277  5567
    </Number>
    <Issuer>
      Bank of the Internet
    </Issuer>
    <Expiration>04/02</Expiration>
 </CreditCard>
</PaymentInfo>
```

〔그림 2〕 XML 구문 예제

진 신용카드를 사용하는 것을 알 수 있다. 그러나 Smith의 신용카드 번호는 민감한 정보이므로, 응용에서 *CreditCard* 엘리먼트를 암호화함으로써 그 정보를 비밀로 할 수 있다.

따라서 *CreditCard* 엘리먼트 시작과 끝 태그를 포함하여 전체를 암호화함으로써, CreditCard 엘리먼트에 대한 암호문을 생성하였으며, *CipherData* 엘리먼트에는 *CreditCard* 엘리먼트를 암호화한 값이 들어간다.

그림 3은 엘리먼트 전체를 암호화 예제이다.

그러나 경우에 따라서는 엘리먼트 내용에 대해 암호화가 필요하다. 예를 들어, 중개인의 경우에는 비록 번호, 유효 기간 그리고 발행 기관을 숨기더라도 신용카드의 사용 여부와 그 한도를 알 필요가 있다.

```
<?xml version='1.0'?>
<PaymentInfo xmlns='http://example.org/
                                paymentv2'>
  <Name>John Smith<Name/>
  <EncryptedData Type='http://www.w3.
   org/2001/04/xmlenc#Element'  xmlns=
   'http://www.w3.org/2001/04/xmlenc#'>
    <CipherData>
      <CipherValue>
        A23B45C56
      </CipherValue>
    </CipherData>
  </EncryptedData>
</PaymentInfo>
```

〔그림 3〕 엘리먼트 전체를 암호화

```
<?xml version='1.0'?>
<PaymentInfo xmlns='http://example.org/
                                paymentv2'>
  <Name>John Smith<Name/>
  <CreditCard Limit='5,000' Currency
                                  ='USD'>
    <EncryptedData Type='http://www.w3.
     org/2001/04/xmlenc#Element'  xmlns=
     'http://www.w3.org/2001/04/xmlenc#'>
      <CipherData>
        <CipherValue>
          A23B45C56
        </CipherValue>
      </CipherData>
    </EncryptedData>
  </CreditCard>
</PaymentInfo>
```

〔그림 4〕 엘리먼트 내용을 암호화

그림 4는 *CreditCard* 엘리먼트의 내용(문자열 혹은 엘리먼트)만을 암호화하는 예제를 나타내고 있다. 한 엘리먼트의 내용은 문자열 혹은 엘리먼트들이 될 수가 있으나, 암호화 방법에 있어서는 차이가 없다.

임의의 데이터나 XML 문서들을 암호화할 경우에는 암호화된 내용은 8비트 셋으로 표현된다. 또한 암호화하려는 데이터의 종류에 대한 정보도 필요하다.

그림 5는 *EncryptedData* 엘리먼트의 Type 속성이 xml이며, XML 문서를 암호화한 것을 보여준다. 임의의 데이터를 암호화하는 경우에는 Type 값은 그 데이터의 포맷으로 표시된다. 이것은 복호화할 때 복호화된 데이터가 어떤 종류인지를 알 수

있게 한다.

XML 문서는 0 개 이상의 *EncryptedData* 엘리먼트를 포함할 수 있다. 그러나 *EncryptedData* 엘리먼트는 *EncryptedData* 엘리먼트의 상위 혹은 하위 엘리먼트가 될 수 없다. 그러나 실제적으로는 *EncryptedData* 또는 *EncryptedKey* 엘리먼트를 포함한 어떤 데이터라도 암호화는 가능하나, 이때 엘리먼트 전체를 암호화해야 한다.

```
<?xml version='1.0'?>
<EncryptedData xmlns='http://www.w3.org
 /2001/04/xmlenc#' Type='http://www.isi.
 edu/in-notes/iana/assignments/media-
 types/text/xml'>
  <CipherData>
    <CipherValue>A23B45C56</CipherValue>
  </CipherData>
</EncryptedData>
```

〔그림 5〕 XML 문서를 암호화

```
[00] <EncryptedData xmlns='http://www.w3.
     org/2001/04/xmlenc#' Type='http://
     www.w3.org/2001/04/xmlenc#Element '/>
[01]   <EncryptionMethod Algorithm='http:
        //www.w3.org/2001/04/xmlenc#3des-
                                    cbc '/>
[02]   <ds:KeyInfo xmlns:ds='http://www.
              w3.org/2000/09/xmldsig#'>
[03]     <ds:KeyName>
            John Smith
         </ds:KeyName>
[04]   </ds:KeyInfo>
[05]   <CipherData>
         <CipherValue>DEADBEEF</CipherValue>
       </CipherData>
[06] </EncryptedData>
```

〔그림 6〕 대칭키를 사용한 암호화

EncryptedData 또는 *EncryptedKey*의 하위 엘리먼트만을 암호화하는 것은 현재로써는 유효하지 않다.

그림 6은 대칭키를 사용한 예이다. 〔01〕은 대칭키 암호 방식으로 3-DES를 사용하고 모드는 CBC라는 것을 나타낸다. 〔03〕은 대칭키의 이름이 John Smith 것을 의미한다.

그림 7은 키를 참조하는 예제를 나타낸 것이다.

그림 7의 *EncryptedData* 구조는 키를 참조하고 있는 *ds:RetrievalMethod* 엘리먼트를 제외하면 그림 6과 매우 비슷하다. 〔t02〕의 *EncryptionMethod* 엘리먼트는 AES-128-CBC 대칭 키 암호 방식을 사용함을 의미하고, 〔t03〕의 *ds:KeyInfo* 엘리먼트는 AES 키가 '#EK'에 있다는 것을 뜻한다.

〔t04〕의 *ds:RetrievalMethod* 엘리먼트는 *&enc:EncryptedKey* 타입의 키가 있는 위치를 나타낼 때 사용된다. 〔t05〕의 ds:KeyName은 CipherData 엘리먼트를 복호화하기 위해 필요한 키를 나타낸다. *KeyName* 엘리먼트와 *KeyRetrival Method* 엘리먼트는 키를 식별하기 위해 사용된다. 〔t09〕 *EncryptedKey* 엘리먼트는 키 값을 암호화한다는 것을 제외하고는 *EncryptedData* 엘리먼트와 유사하다. 여기서 *CarredKeyName* 속성은 *ds:KeyInfo* 안에 있는 *KeyName* 엘리먼트에 의해 참조되는 암호화된 키 값을 식별하는 데 사용된다. 〔t11〕의 *EncryptedMethod* 엘리먼트는 공개키 알고리즘으로 RSA로 지정되어 있는 것을 알 수 있다. 〔t13〕의 *CipherData* 엘리먼트는 RSA를 사용하여 복호화하기 위해서 "John Smith"라는 키를 소유해야 함을 알 수 있다. 〔t16-18〕의 *ReferenceList* 엘리먼트는 이 키로 암호화된 개체(*DataReference*와 *KeyReference*)를 나타낸다.

```
[t01] <EncryptedData Id='ED' xmlns:enc=
      'http://www.w3.org/2001/04/xmlenc#'>
[t02]   <EncryptionMethod Algorithm=
          'http://www.w3.org/2001/04/xmlenc
          #aes128-cbc'/>
[t03]   <ds:KeyInfo xmlns:ds='http://www.w3
          .org/2000/09/xmldsig#'>
[t04]     <ds:RetrievalMethod URI='#EK'/
          Type="http://www.w3.org/2001/
          04/xmlenc#EncryptedKey">
[t05]     <ds:KeyName>John Doe<ds:KeyName>
[t06]   </ds:KeyInfo>
[t07]   <CipherData>
          <CipherValue>
            DEADBEEF
          </CipherValue>
        </CipherData>
[t08] </EncryptedData>
[t09] <EncryptedKey Id='EK' CarriedKeyName
                              ="John Doe"
[t10]  xmlns='http://www.w3.org/2001/04/
                                  xmlenc#'>
[t11]   <EncryptionMethod Algorithm="http
          ://www.w3.org/2001/04/xmlenc#rsa-
          1_5"/>
[t12]   <ds:KeyInfo xmlns:ds='http://www.w3
          .org/2000/09/xmldsig#'>
[t13]     <ds:KeyName>John Smith</ds:KeyName>
[t14]   </ds:KeyInfo>
[t15]   <CipherData>
          <CipherValue>
            xyzabc
          </CipherValue>
        </CipherData>
[t16]   <ReferenceList>
[t17]     <DataReference URI='#ED'/>
[t18]   </ReferenceList>
[t19] </EncryptedKey>
```

〔그림 7〕 키를 참조하는 암호화 구문

ReferenceList 엘리먼트는 이 구조 안에 있는 대칭 키로 암호화된 데이터들을 참조하는 리스트를 갖는다.

3. XML 암호화 처리

이 절에서는 XML 암호화 및 복호화에 필요한 각각의 요소들에 대한 처리를 소개한다.

XML 암호화 처리는 다음과 같다.

1) 암호화에 사용될 알고리즘과 파라미터를 선택한다.
2) 사용될 키를 생성하거나 가져온다.
3) 암호화 대상을 8비트 배열로 만든다.
4) 암호화를 위해 XML 구조를 만든다.

상기의 3번 단계에서 8비트 배열을 생성할 때, 암호화 대상이 XML 엘리먼트나 엘리먼트의 내용이면 UTF-8으로 인코딩된 8비트 배열의 문자열 형태로 각각 나타낸다. UTF-8로 인코딩된 8비트 배열은 2 단계에서 얻은 키로 암호화한다. 암호화 어플리케이션은 *EncryptedData* 엘리먼트에서 선택적으로 사용되는 Type 속성을 사용하는 것이 좋다. 여기서 암호화 대상의 Type을 정의해 놓으면, 복호화 시에 자동적으로 저장될 수 있게 때문이다.

상기의 4번째 단계에서 암호화되는 대상이 XML 엘리먼트나 엘리먼트의 내용이면, 암호화되지 않은 데이터는 제거되거나 새로 만들어진 XML 구조로 바뀐다. 암호화되는 대상이 외부의 8비트 배열이라면, 암호화된 데이터를 포함하거나 참조하는 *EncryptedData* 구조를 생성한다. 그리고 그 구조를 새로운 XML 문서의 루트 엘리먼트로 사용한다. 어플리케이션에 따라서는 *EncryptedData* 구조를 다른 XML 문서에 넣기도 한다.

다음은 복호화에 필요한 절차로써, *Encrypteddata* 엘리먼트 또는 *Encryptedkey* 엘리먼트를 복호화하는데 사용된다.

1) 사용될 알고리즘, 파라미터 및 키를 결정하기 위해 문서를 파싱한다.
2) 키가 암호화되어 있으면, 복호화 할 수 있는 키를 찾는다.
3) *CipherData* 엘리먼트를 복호화한다.

상기의 3번째 단계에서 복호화 시에, 데이터가 XML 문서이면 복호화 결과는 엘리먼트 혹은 엘리먼트 내용은 UTF-8로 인코딩된 문자열로 표시된다. 또한 *CipherData*가 *CipherValue*를 포함하고 있으면, 그 내용은 base64로 디코딩하여 얻을 수 있다. *CipherData* 엘리먼트가 *CipherReference* 엘리먼트를 포함하고 있다면, 8비트 스트림을 얻기 위해 URI 속성 값에 참조된 특정 변환을 적용한다.

상기의 3번째 단계가 완료된 후에, *EncryptedData* 구조의 Type 속성의 값이 "Element" 혹은 "Content"일 경우, 복호화된 결과가 부모 XML 문서를 인코딩하고 있는 EncryptedData와 교체된다. Type 속성의 값이 "Element"와 "Content"가 아닐 경우, 최종 결과 값은 8비트 배열이 된다.

4. 알고리즘

XML 암호화에 사용되는 알고리즘들은 다음과 같다. XML 암호화에 사용되는 알고리즘은 Encryptio*nMethod* 엘리먼트의 *Algorithm* 속성의 값으로 구별할 수 있다.

표 1에 열거된 알고리즘들은 그것들의 역할에 따라 묵시적인 파라미터들을 가지고 있다. 예를 들면, 암호화/복호화 될 데이터, 키 매체 그리고 암호화 알고리즘들의 동작 방향(암호화/복호화) 등이 묵시적인 파라미터에 해당한다. 그 외의 모든 명시적인 파라미터들은 엘리먼트 안에 하위 엘리먼트로 표시된다.

〔표 1〕 XML 암호화에서 사용되는 알고리즘

Block Encryption	3DES	REQUIRED
	AES-128	REQUIRED
	AES-256	REQUIRED
	AES-192	OPTIONAL
Stream Encryption	ARCFOUR	OPTIONAL
Key Transport	RSA-v1.5	REQUIRED
	RSA-OAEP	REQUIRED
Key Agreement	Diffie-Hellman	OPTIONAL
Symmetric Key Wrap	3DES	REQUIRED
	AES-128	REQUIRED
	AES-256	REQUIRED
	AES-196	OPTIONAL
Message Digest	Digest	REQUIRED
	SHA256	RECOMMENDED
	SHA512	OPTIONAL
	RIPEMD-160	OPTIONAL
Message Authentication	XML Digital Signatur	RECOMMENDED
Canonicalization	Canonical XML with Comments	RECOMMENDED
	Canonical XML (omits comments)	OPTIONAL
Encoding	base64	REQUIRED

Ⅲ. 결 론

본 논문에서는 W3C의 작업 그룹 중 하나인 XML 암호화 그룹에서 발표한 XML 암호화 요구사항과 XML 암호화 구문과 처리에 대한 초안을 중심으로 한 XML 암호화에 대한 내용을 소개하였다.

현재 W3C의 XML에 대한 보안 분야에 대한 활동을 보면, 아직까지는 전자 서명에 비해 표준 제정 정도나 연구 개발 정도가 미진하다. 그렇지만, 최근엔 활발한 활동과 논의가 이루어지고 표준화 진행 정도에 있어서도 가속도가 붙고 있다. 그러나, 국내의 경우에는 XML 암호화는 이제 시작 단계이며 이렇다 할 성과가 아직 없는 것이 현실이며, 최근에 와서야 XML 문서 보안에 대한 국내 표준화 논의가 되고 있을 뿐이다.

마지막으로, XML 암호화는 XML 문서를 포함한 전자 문서에 대한 암호화/복호화에 기능에 관한 것이다. 또한 그 표준은 W3C's XML 암호화 그룹에서 담당하고 있으며, 표준화에 대한 활발한 활동과 논의가 진행되고 있다.

참 고 문 헌

[1] W3C XML Encryption WG, XML Encryption Charter, http://www.w3.org, 2001.

[2] XML Encryption requirements, http://www.w3.org /TR/xml-encryption-req

[3] Takeshi Imamura, Blair Dillaway, Jim Schaad, Ed Simon, "XML Encryption Syntax and Processing", http://www.w3c.org/TR/2001/WD-xmlenc-core-20010626/

〈著 者 紹 介〉

김 주 한 (Ju-Han Kim) 정회원

1997년 2월 : 충남대학교 컴퓨터과학과 졸업

1999년 2월 : 충남대학교 컴퓨터과학과 석사

2000년 9월~현재 : 한국전자통신연구원 EC정보보호연구팀 연구원

관심분야 : XML, 정보보호, 워터마킹

나 중 찬 (Jung-Chan Na) 정회원

1986년 2월 : 충남대학교 계산통계학과 졸업

1989년 2월 : 숭실대학교 전산학과 석사

1989년~현재 : 한국전자통신연구원 능동프로토콜팀장

관심분야 : 실시간시스템, 정보보호, 액티브네트워

문 기 영 (Ki-Young mun) 정회원

1986년 2월 : 경북대학교 전자공학과 졸업

1989년 2월 : 경북대학과 전자공학과 석사

1992년 1월~1994년 3월 : (주)대우정보시스템 기술연구소 대리

1994년 3월~현재 : 한국전자통신연구원 EC정보보호팀장

관심분야 : 분산시스템, 정보보호, 트랜잭션