

---

# XML 응용을 이용한 생체 정보의 전송 및 처리

김재원\* · 박강령\*\* · 이수연\*

## Transfer of Biometric Information using XML Notations

Jac-Won Kim\* · Kang Ryoung Park\*\* · Soo-Youn Lee\*

---

이 논문은 2004년도 광운대학교 연구비를 지원받았음

---

### 요 약

최근 지문, 얼굴 및 홍채 등과 같은 생체정보를 이용하는 정보 전송 및 처리에 관한 연구가 활발하게 진행되고 있다. 지금까지 연구 및 개발된 결과를 볼 때, 많은 생체 정보 시스템이 생체 정보의 가공과 전송에 있어서 개별적인 포맷과 인터페이스를 사용하고 있는 경향이 많다. 이로 인하여 생체 정보의 상호 공유, 호환성 및 상호운용성 등과 같은 문제점이 발생하고 있다. 이러한 문제점을 해결하기 위하여, 본 논문에서는 서버-클라이언트 환경에서 생체정보의 처리 및 전송을 위하여 XML 기술(XCBF, SOAP 등)을 활용하는 연구를 하였으며, 그 결과에 따른 XML 형식의 생체정보 전송 및 처리 방식을 제안한다.

### ABSTRACT

There have been active researches utilizing Biometric informations such as fingerprints, face, iris and so on. but they have tended to use their own data format and interfaces in processing Biometric informations, which causes the difficulties including sharing of Biometric informations and interoperability. To overcome such problems, we have studied XML applications such as XCBF and SOAP to process Biometric informations in the client-server environment. A method suitable to process and transfer Biometric informations in XML notation is proposed as a result.

### 키워드

XCBF, SOAP, Biometric information, Client/Server Environment

## I. 서론

복잡한 현대 사회에서의 보안에 대한 요구가 점점 많아 짐 따라 다양한 응용분야에서 자문, 홍채 또는 혈관등과 같은 생체 정보(Biometric information)가 널리 사용되고 있다. 생체정보의 응용분야를 예를 들면 건물 등에서의 출입관리(access control), PC 에서 로그인

시 본인 인증, 인터넷 뱅킹, ATM(Automatic Teller Machine)의 접근 제어 및 공항, 항만 등에서의 출입국 신원확인 등을 들 수가 있다.

기존에 소규모의 생체시스템(Biometric system)은 스탠드얼론 응용이 많으며 생체정보의 공유나 상호호환이 중요 시 되지 않았다, 대규모의 생체시스템인 경우에는 인터넷등과 같은 네트워크를 통하여 다양한 입

---

\* 광운대학교 컴퓨터공학부  
\*\* 상명대학교 미디어학부

출력장치가 연결되어 있으며, 등록된 생체정보를 여러 응용에서 공유하는 등과 같은 측면을 포함하는 상호운용성 및 확장성의 문제점들을 갖고 있는 경향이 많다.

생체정보를 이용하여 소정의 목적을 달성하기 위한 생체시스템은 적용 규모에 관계없이 공통적으로 등록(enroll), 인식(identify) 및 검증(verify)과 같은 3가지 기본적인 기능을 필요로 한다. 등록은 새로운 생체 데이터를 저장하는 기능이며, 인식과 검증은 저장된 생체 데이터와 현재 사용자의 생체 데이터를 비교하는 동작으로, 인식의 경우는 여러 사람의 데이터에 관하여 1 : N 매칭(matching)을 제공하며, 검증은 한사람의 데이터와 1 : 1 매칭을 제공하는 기능을 의미한다.

특히 이러한 생체정보를 분산처리 환경 속에서 전송하고 등록하고 인식 및 확인할 뿐만 아니라 다양한 응용에서 정보공유를 위해서는 생체정보의 표현방식이나 전송방식에 관한 표준과 이를 실제로 적용하는 연구가 절실히 요구되고 있다.

이에 따라 BAPI[1] 및 HA-API[2]가 소개되었으며, BioAPI[3]는 앞의 두 가지 표준을 통합하기위한 표준으로 제안되었다. 그런데, 최근 XML(eXtensible Markup Language)[4] 기술의 발전에 따라서 각종 데이터 양식이 XML 형식으로 표현되고 있으며, 앞에서 소개한 생체정보의 표준 방식 역시 XML형태로 나타내는 XCBF[6]가 최근 제안되어 사용되고 있다.

또한 분산처리 환경에서 등록된 생체정보를 다양한 응용에서 이를 공유할 뿐만 아니라 응용(Application)의 통합이나 다른 통신 방식을 가진 응용과의 연동을 용이하게 하기 위하여서는 보다 유연한 통신환경을 필요로 한다. 그 예로서 SOAP(Simple Object Access Protocol)을 들 수 있으며, 이 프로토콜을 통하여 다양한 프로그램 모델이나 프로토콜과의 바인딩(binding)을 쉽도록 하는 특징을 갖고 있다.

본 논문에서는 생체정보처리 시스템을 구현하는 연구의 일부로서, 인터넷 환경의 클라이언트 서버 모델에 따라서 생체정보를 서버에 전송하여 등록하거나 서버에 등록된 정보를 찾아서 클라이언트에 재전송하는 방식을 연구하였으며, 이때 서버에게 전달하거나 클라이언트에 전달하는 생체정보는 XCBF 표준을 채택하여 변환하였으며, 변환된 정보는 SOAP[5] 프로토콜을 이용하여 전송하였다.

## II. 관련연구

### 2.1 XCBF(XML Common Biometric information Format)

XCBF는 생체 정보의 표현 및 처리를 용이하게 하기 위해 제안된 XML 표현이다[6]. 이러한 XCBF는 기존의 존재하던 생체 데이터의 포맷과 처리의 표준인 X9.84와 BioAPI와의 상호운용성을 보장한다.

이러한 XCBF는 그림 1 및 2와 같이 기존의 BioAPI에서 데이터 값으로만 표현되던 BIR(Biometric Identification Record) 데이터 포맷을 처리 가능하고 구조적이며 기계와 인간의 서로 인식 가능한 형태인 XML 형태로 구조화 시킨 것이다. 따라서 BioAPI나 X9.84 에서 처리 과정에 필요한 정보를 가지고 있다.

기존의 생체 데이터 포맷의 경우 단순히 정보만을 담고 있어서 그것을 인식하거나 처리해야 할 경우 일일이 문자열에 대한 처리를 해줘야만 했다.

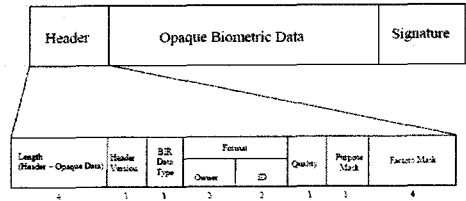


그림 1. BioAPI에서의 BIR 데이터 포맷  
Fig. 1. BIR data format in BioAPI

```

<?xml version="1.0" ?>
<BiometricSyntax:Set>
  <Biometric:Subject>
    <Biometric:Header>
      <version?/ version>
      <record? type>
      <id? / id>
      <record? type>
      <data? type>
      <intermediate? />
      <data? type>
      <purpose?>
      <verify? />
      <purpose?>
      <quality? 100? quality>
      <validity? period>
      <create? before? 2003.11.2? /notBefore>
      <create? after? 2004.11.2.23.59.59? /notAfter>
      <validity? period>
      <format?>
      <format? owner?
      <id? 2.23.42.9.10.4.2? /id>
      <format?>
      <biometric? header?
      <biometric? data? 7080E00C831C1D7F2CFBF84CFB0460260C3307F30073F9F
    </Biometric:Subject>
  </Biometric:Syntax:Set>
  
```

그림 2. XCBF포맷으로 인코딩된 BIR 데이터의 예  
Fig. 2. Example of BIR data encoded by XCBF format

그러나 그림 2와 같이 XCBF 포맷으로 BIR 데이터를 처리할 경우 문자열의 무의미한 처리과정을 줄여 주며 또한 사용자의 측면에서 보아도 상당히 가독성이 뛰어난 정보처리를 할 수 있게 된다.

## 2.2 SOAP(Simple Object Access Protocol)

SOAP은 분산된 환경에서 XML을 이용해서 구조나 정보의 교환에 이용되는 간단하고 프로세스 적으로 가벼운 메커니즘을 제공하는 프로토콜로서, 주로 웹 서비스에서의 정보 전달이나 메시지 전송을 위해 사용된다. 또한 전송 단에서 수신 단으로의 XML 메시지를 보내는 방법을 정의하고 있는 것으로서 확장성과 다양한 프로토콜에서의 동작 그리고 프로그램 모델에서의 독립성을 보장하여준다. 확장성의 경우 SOAP에 있어서 가장 중심이 되는 특징으로서 SOAP이 가지는 특징인 간결함으로 인해서 생겨난다. 또한 다양한 프로토콜과의 바인딩은 각 프로토콜의 환경에 따라 정의가 되어 있다. 다양한 프로토콜마다 적용 가능하지만 최근 HTTP가 가장 많이 쓰인다. SOAP은 흔히 RPC(Remote process call)에서 많이 쓰이지만, 메시지가 사용되는 모든 경우에 적용이 가능한 프로토콜이기 때문에 단 방향의 메시지 전달이나 본 논문에서 사용될 그림 3과 같은 요청/응답(request /response) 모델, Peer to Peer 모델 모두에 사용 가능하다.

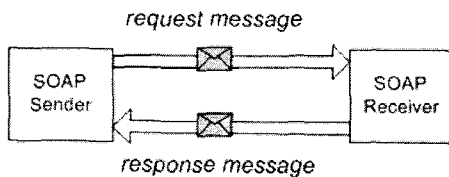


그림3. Request와 이에 대한 처리를 알려주는 SOAP 메시지 통신

Fig. 3. SOAP message communication which informs requests and its transaction

SOAP 메시지는 인벨로프(Envelope) 엘리먼트(element)에 담긴 헤더(Header) 엘리먼트와 바디(Body) 엘리먼트로 구성되어 있다. 헤더 엘리먼트의 정보는 목적지에서 SOAP 메시지가 어떻게 처리 되어야 하는지와 목적지의 위치와 정보를 다루기 위한 부가적인 정보를 가지고 있다. 바디 엘리먼트의 정보는 목적지에 전달되는 정보를 가지고 있는 부분이다. 인벨로프

엘리먼트는 위의 SOAP을 구성하는 부분인 헤더 엘리먼트와 바디 엘리먼트를 담고 있다. 이러한 SOAP 메시지는 각종 프로토콜, 예를 들면 HTTP, SMTP(Simple Mail Transfer Protocol), FTP등의 프로토콜과 바인딩 함으로 전송이 되어 진다. 그렇기 때문에 상당히 다양한 환경에서 프로그램간의 통신에 있어서 장점을 가지고 있다.

## III. 생체정보 처리 서버시스템의 설계

본 연구에서는 그림 4와 같은 클라이언트-서버 환경을 대상으로 하였으나, 생체정보를 추출하거나 서버에 등록된 정보를 전송받아서 신원을 검색하는 인식 및 검증 기능을 갖는 클라이언트의 경우는 본 연구의 범위에 포함되지 않기 때문에 우리는 이를 생체 정보처리 서버시스템으로 칭하기로 한다.

그러나 XCBF포맷으로 저장된 생체정보를 이용하여 서버 측에서 전적으로 인식 및 검증기능을 수행하는 것도 가능하다.

그림 4에서 클라이언트(A)는 생체정보를 그림 1의 BIR 데이터 포맷으로 생성하며, 이를 서버에 전송하고 등록하기 위하여 변환과정(B)을 통하게 하였다. 이 과정에서 BIR 데이터 포맷을 XCBF 포맷으로 변환한다. 그 다음, SOAP 메시지 핸들러 단계(C)를 거치면서 XCBF 포맷을 이용하여 SOAP 메시지 형식으로 변환(packing)한다. SOAP의 특성상 이는 다양한 프로토콜과의 바인딩이 쉽기 때문에 HTTP, SMTP 또는 MIME을 이용하여 서버로 전달된다.

서버로 전달된 SOAP 메시지는 클라이언트의 처리와 역순으로 SOAP 메시지 핸들러에서 XCBF 포맷으로 변환(unpacking)하게 된다. 서버 측에서 등록은 생체정보의 공유를 위하여 XCBF 포맷을 이용하여 생체정보를 등록하거나 또는 BIR 포맷으로 변환(E)한 다음, 등록되게 된다.

서버에서 전적으로 인식 및 검증을 하는 경우, SOAP의 RPC을 이용하여, 클라이언트가 서버측이 제공하는 인터페이스를 직접적으로 접근할 수도 가능하다.

이하 본 연구를 SOAP 메시지 핸들러, 서버 측 및 클라이언트와 같은 세 부분으로 나누어서 설명하겠다.

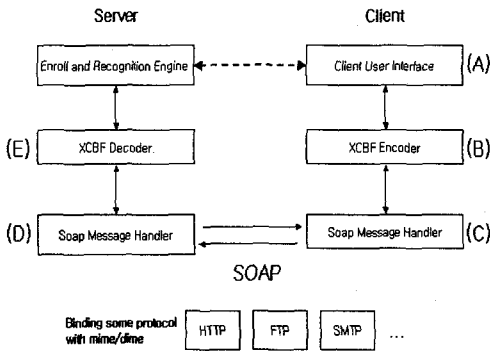


그림 4. 생체 정보 처리를 위한 전체 구조  
Fig. 4. Overall structure for biometric information processing

### 3.1 SOAP 메시지 핸들러

전체적인 메시지의 전달되는 구조는 그림 5에서 보듯이 SOAP의 전달 메커니즘을 가지고 있다. 따라서 HTTP 이나 FTP 등, 다른 프로토콜과 바인딩을 통하여 전송을 하게 된다. 다음 절에서 소개될 서버와 클라이언트 메시지 핸들러는 모두 그림 4와 같은 SOAP 메시지 전달 메커니즘을 가지고 전달하게 된다. 본 논문에서 제시하고자 하는 메시지 핸들러는 그림 5의 SOAP의 메시지 전달 메커니즘에서 SOAP 응용과 메시지 인터페이스에 해당하며, 메시지를 발생시키며 처리하는 서버와 클라이언트 메시지 핸들러는 다음 그림 6 에서와 같이 제안하였다.

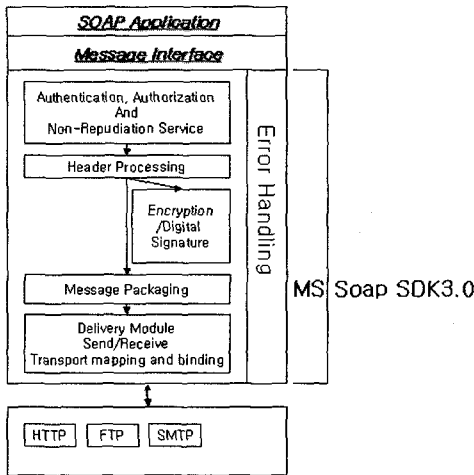


그림 5. SOAP의 메시지 전달 메커니즘(client-server)  
Fig. 5. Transferring mechanism for SOAP message(client-server)

응용은 클라이언트-서버 모델을 사용해서 클라이언트에서는 기존 BIR 데이터를 인코딩 하여 XCBF 데이터를 가공한 후에 가공한 데이터를 서버에게 등록, 인식 및 검증 등의 요청(request) SOAP 메시지를 보내는 역할을 한다. 데이터를 받은 XCBF 파일을 파싱하여 각 파일의 목적에 맞게 처리를 하고 처리한 결과를 응답(response) SOAP 메시지를 통하여 전달하게 된다. 본 논문에서 SOAP 메시지의 가공 및 바인딩은 마이크로소프트(Microsoft)사에서 제공하는 마이크로소프트 SOAP Toolkit 3.0을 사용하였다.

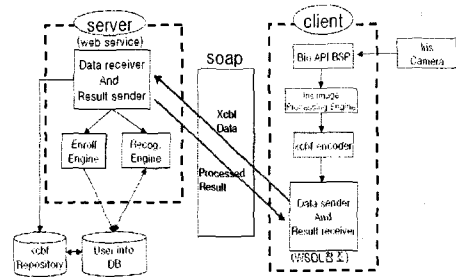


그림 6. SOAP기반 Biometric message handler (client-server)  
Fig. 6. Biometric message handler based on SOAP (client-server)

### 3.2서버의 설계 및 구현

서버는 그림 7에서 보듯이 클라이언트에서 보내진 요청 SOAP 메시지를 통해서 보내진 XCBF 데이터를 파싱하는 부분인 메시지 핸들러 부분과 XCBF 데이터의 파싱을 통해서 메시지의 목적대로 등록, 인식 및 검증 동작을 수행하기 위해서 필요한 정보를 등록 및 인식(검증) 엔진으로 보내주는 디코더와 실제적인 등록과 검증, 인식 등의 동작을 하는 등록 및 인식(검증) 엔진으로 이루어져 있다.

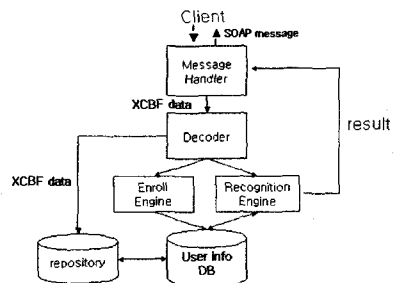


그림7. SOAP 기반 Biometric 메시지 핸들러(서버)  
Fig. 7. Biometric message handler based on SOAP(server)

### 3.2.1 메시지핸들러 (서버)

서버 측의 메시지 핸들러는 클라이언트 측에서 SOAP 메시지를 통해서 보내진 XCBF 데이터와 사용자에 관한 데이터를 받아서 그것을 파싱을 통해서 디코더와 등록 및 인식(검증) 엔진에 전달하게 된다.

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<SOAP-ENV:Envelope xmlns:SOAPENV="http://www.w3.org/2001/XMLSchema"
  xmlns:SOAPENC="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:SOAPSDK="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-
  ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body SOAP-
  ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAPSDK4:xcbfmessagingResponse
    xmlns:SOAPSDK4="http://tempuri.org/xcbfmessaging/message/">
    <Result>You are not verified scramble2, different= 247</Result>
  </SOAPSDK4:xcbfmessagingResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

그림8. 클라이언트에 Response로 전달되는 SOAP Message

Fig. 8. SOAP message which is transferred to client as response

서버는 HTTP 프로토콜과의 바인딩을 통해서 그림 8과 같은 데이터를 받게 된다. 마이크로소프트 SOAP Toolkit을 통해서 메시지의 BODY에 있는 xcbfIn, filename, inId 엘리먼트 내의 데이터를 SOAP 응용에 전달해 주게 된다.

서버는 xcbfIn 엘리먼트에 담겨서 전달된 데이터를 파싱하여 XCBF 데이터 부분을 저장소(repository)에 저장하고, 데이터베이스와 연동을 통하여 XCBF데이터의 목적에 맞추어서 등록 혹은 인식(검증) 동작을 하게 된다. 서버는 데이터베이스를 통한 동작을 마친 후 결과 값을 그림 8과 같은 SOAP 메시지로 가공하여 응답으로 클라이언트에 전송한다. 여기에서도 마이크로소프트 SOAP Toolkit을 통한 HTTP 프로토콜과의 바인딩을 통해서 전달되게 된다.

### 3.2.2 디코더

디코더는 메시지 핸들러를 통해서 전달된 XCBF데이터에서 등록 혹은 인식(검증) 엔진에 필요한 데이터의 부분을 추출하여 전달하게 된다. 이 과정에서의 인코딩 된 XCBF 데이터의 원본적인 BIR 데이터의 복원 작업은 하지 않는다.

### 3.2.3 등록 혹은 인식(검증) 엔진

등록 동작은 데이터베이스에 필요한 정보인 사용자의 ID와 디코더를 통하여 전달된 생체정보데이터를 입력한다. 인식이나 검증 동작은 데이터베이스를 검색해서 해당되는 생체 데이터를 검색하거나 검사를 하게 된다. 인식은 1 : N 매칭으로서 모든 데이터 가운데 대응되는 데이터를 찾게 되며, 검증 같은 경우는 1 : 1 매칭으로서 사용자로부터 입력받은 데이터와 ID 값을 바탕으로 그것을 검증하게 된다. 그리고 그 결과를 클라이언트에 응답 SOAP 메시지를 통해서 전송한다.

## 3.3 클라이언트의 설계 및 구현

클라이언트는 그림 9와 같이 세 부분으로 구성된다. 사용자로부터 데이터를 등록과 인식이 필요한 정보를 받는 클라이언트 사용자 인터페이스 부분과 사용자로부터 선택된 BIR 데이터와 입력 받은 데이터를 바탕으로 서버 측에 전송 및 등록을 하기 위한 XCBF 데이터 포맷으로 BIR 데이터를 변환하는 인코더와 변환된 XCBF 데이터를 SOAP 메시지로 가공해서 서버 측에 전달하는 메시지 핸들러로 구성된다.

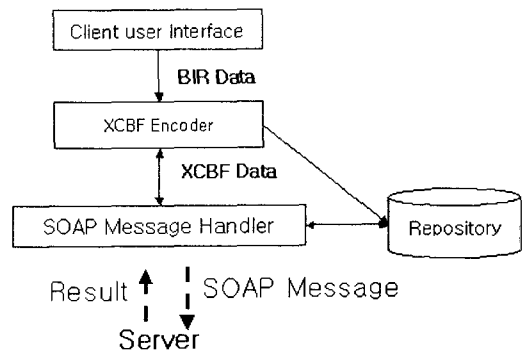


그림9. 클라이언트의 설계

Fig. 9. Design of client

### 3.3.1 클라이언트 사용자 인터페이스

클라이언트 사용자 인터페이스는 사용자로부터 데이터를 가공하거나 인식이나 인증 등록에 필요한 사용자 데이터를 입력 받는 부분이다. BIR 데이터 포맷에는 없으나 XCBF 데이터 포맷에 필요한 데이터의 유효 기간이나, 사용자가 등록해 놓은 BIR 데이터를 선택할 수 있게 하며, 혹은 이미 인코딩 과정을 거친 XCBF 데이터를 선택할 수 있다. 또한 사용자 등록 및 인증

에 필요한 아이디를 입력받는 기능을 담당하는 부분이  
다.

### 3.3.2 XCBF 인코더

사용자가 선택한 BIR 데이터와 입력한 데이터의 유효기간을 바탕으로 SOAP 으로 전송하기 알맞은 형태인 생체정보데이터의 XML 표현인 XCBF 데이터 포맷 스키마(Schema)에 맞추어서 XCBF 데이터 포맷으로 바꾸어 주는 역할을 하며 또한 변환된 XCBF 데이터 포맷의 파일을 클라이언트의 저장소에 저장하게 되어 재사용을 가능하게 한다.

```
528 0 4 65535 65535 100 1 16
112 128 224 012 131 028 029 127 044 251 251 076 251 004 096 038 012 051 007 243
115 249 253 204 205 142 136 199 142 199 239 064 115 064 194 056 008 061 057 159
255 159 142 007 135 039 127 101 103 000 004 144 216 186 131 188 186 252 060 192
124 193 192 003 192 003 003 003 023 126 252 127 252 127 031 126 025 126 104 124
126 121 031 099 065 097 098 001 051 005 005 004 065 030 079 026 026 031 050 015
071 046 015 050 014 048 050 049 051 001 025 004 005 014 037 014 014 059 030 051
051 017 051 000 000 012 012 031 015 007 019 014 022 011 012 003 010 003 003 005
002 010 012 004 015 008 003 012 000 029 016 001 031 023 031 003 002 000 002 072
001 120 092 121 252 227 243 066 195 078 074 126 004 126 095 179 254 185 177 153
145 093 024 012 239 006 102 003 004 003 003 225 131 248 224 254 236 127 063 052
127 240 100 241 145 243 241 134 255 006 015 015 004 125 006 240 123 216 248 120
232 003 130 038 001 118 037 126 254 126 250 094 072 158 196 190 190 188 062 120
172 000 128 002 018 094 119 014 100 014 140 003 015 226 099 192 111 230 223 231
230 238 255 095 207 239 202 008 255 247 227 237 247 063 123 117 126 250 115 239
255 191 239 231 230 219 199 252 075 159 095 255 255 122 223 223 191 191 143 151
127 180 055 140 103 194 221 228 039 255 253 061 118 255 189 255 127 062 255 255
255 239 255 254 215 124 125 125 255 095 063 054 091 061 115 031 045 004 127 047
011 029 127 031 022 063 111 054 027 019 119 019 091 019 021 015 095 015 015 051
015 053 041 023 053 019 055 055 023 061 063 047 063 019 059 061 063 015 057 019
063 027 027 004 013 006 019 002 025 006 005 007 010 013 015 001 012 005 015 000
013 015 003 003 001 030 019 031 001 001 001 001 001 001 001 001 001 001 001 001
255 223 239 190 159 183 103 175 231 054 102 241 127 255 255 127 111 046 063 189
109 255 127 254 255 127 255 206 207 234 239 100 239 174 095 001 135 175 013 254
030 229 175 055 247 015 246 255 223 134 126 255 239 222 189 203 127 255 250
253 182 255 126 255 092 126 100 166 062 118 254 198 250 126 238 158 062 254 204
238 198 255 063 111 127 255 158 127 092 142 221
```

그림 10. BIR Data  
Fig. 10. BIR Data

```
<?xml version="1.0" ?>
- <BiometricSyntaxSets>
- <BiometricObjects>
- <BiometricObject>
- <biometricHeader>
- <version>0</version>
- <recordType>
- <cid>0</cid>
- </recordType>
- <dataType>
- <processed />
- </dataType>
- <purpose>
- <verify />
- </purpose>
- <quality>100</quality>
- <validityPeriod>
- <notBefore>2003.11.2</notBefore>
- <notAfter>2004.11.2.23.59.59</notAfter>
- </validityPeriod>
- <format>
- <formatOwner>
- <oid>2.23.42.9.10.4.2</oid>
- </formatOwner>
- </format>
- </biometricHeader>
- <biometricData>7080E00C831C1D7F2CFBF94CF80462060</biometricData>
- </BiometricObject>
- </BiometricObjects>
</BiometricSyntaxSets>
```

그림 11. XCBF포맷으로 인코딩된 데이터  
Fig. 11. Encoded data by XCBF format

그림 10과 같은 BIR Data의 데이터를 XCBF 인코더를 통해서 인코딩하게 되면 그림 11과 같은 XCBF 데이터 포맷을 가진 데이터로 변환되게 된다. 이 과정에서 16진수로 인코딩된 데이터는 biometricdata 엘리먼트에 들어간다.

### 3.3.3 SOAP 메시지 핸들러

SOAP 메시지 핸들러는 XCBF 인코더를 통하여 만들어진 XCBF 데이터 포맷으로 된 데이터를 SOAP 메시지 안에 포함하여 서버에 그림 12와 같은 응답 메시지로 보낸다. 또한 요청 메시지를 통해서 등록 및 인식(검증) 되어진 결과를 서버로부터 보내진 요청 메시지를 통해서 받게 된다.

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
- <SOAP-ENV:Envelope SOAP-ENV:encodingStyle=""
xmlns:SOAPSDK1="http://www.w3.org/2001/XMLSchema"
xmlns:SOAPSDK2="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAPSDK3="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
- <SOAP-ENV:Body SOAP-ENV:encodingStyle="">
- <SOAPSDK4:xcbfmessaging
xmlns:SOAPSDK4="http://tempuri.org/xcbfmessaging/message/" SOAP-
ENV:encodingStyle="">
- <xcbfIn SOAP-ENV:encodingStyle=""><?xml version="1.0" ?>
- <BiometricSyntaxSets><BiometricObjects><BiometricObject><biometricHe-
- <filename SOAP-ENV:encodingStyle="">scremble_10_verify.xml</filename>
- <inId SOAP-ENV:encodingStyle="">scremble2</inId>
- </SOAPSDK4:xcbfmessaging>
- </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

그림 12. 서버로 전달하기 위하여 가공된 SOAP 메시지  
Fig. 12. Processed SOAP message for transferring to server

## IV. 실험 및 고찰

본 논문에서는 SOAP프로토콜의 대체 가능성을 확인해 보기위해 SOAP과 TCP/IP 소켓(socket)전송과의 속도의 차이를 실험해 보았는데 실험은 각각의 가공된 XCBF 데이터를 바탕으로 중간처리(intermediate), 완전 처리된(processed) 데이터에 대해서 각각 등록, 인식, 검증에 대한 속도 측정을 해보았다. 메시지 핸들러는 Soap 메시지 작성에 WriteXML 메서드를 사용한다. WriteXML로 사용된 메시지는 서버에서 문자열(string) 형식으로 처리되는데 문자열 형식으로 처리 할 수 있

는 메시지의 용량은 최대 65,400 byte까지 처리의 한계를 가지고 있다. 그러므로 홍채 의 날 데이터 (raw data)의 용량은 약 307,200 byte(=640\*480 pixels)로서 문자열이 허용하는 한도인 6.5 kbyte를 훨씬 초과하기 때문에 메시지로 전달하기에 적합하지 못하다. 따라서 메시지 방식으로 적합한 완전 처리된 데이터와 중간 처리된 데이터만을 실험 대상으로 하였다.

중간 처리된 데이터는 이리디안 폴라 이미지 (Iridian Polar Image )기준으로 8\*256 = 2,048 byte, 완전 처리된 512 byte의 홍채 코드(iris code ) [7][9] 이고 실험 환경은 각각 스탠드얼론(Standalone), 내부 랜(Lan) 환경과 외부 네트워크를 통한 전송에 대한 속도를 20 회 정도 측정 한 후에 그것의 평균값을 구하여 보았다. 스탠드얼론 환경에서 사용된 컴퓨터는 CPU-펜티엄-4 2.4GHz , 메모리 - 512M 컴퓨터가 사용되었고, LAN 은 서버 CPU - 펜티엄3 1GHz\*2, 메모리-512M 컴퓨터와 클라이언트 CPU - 펜티엄4 2.4GHz, 메모리-512M 컴퓨터와 LAN 100Mbps 네트워크 환경에서 실험하였다. 외부 환경의 경우 서버 펜티엄 - 3 1GHz\*2 512M 메모리의 컴퓨터, 클라이언트 펜티엄 M 1.5GHz 256M 메모리의 5Mbps 무선 랜 환경에서 진행하였다.

표 1. Standalone 환경에서의 속도

Table 1. Processing speed in standalone environment

메시지의 목적과 종류	TCP/IP	SOAP
intermediate(enroll)	59.3	79.5
intermediate(identification)	136	114
intermediate(verification)	61.8	60.75
processed(enroll)	59.5	60.95
processed(identification)	105	83.6
processed(verification)	57.7	61.8

(단위 ms)

표 2. LAN 환경에서의 속도

Table 2. Processing speed in LAN environment

메시지의 목적과 종류	TCP/IP	SOAP
intermediate(enroll)	75.1	159
intermediate(identification)	218	304
intermediate(verification)	78	165
processed(enroll)	76	177
processed(identification)	147	171
processed(verification)	71.9	175

(단위 ms)

표3. 외부 네트워크 환경에서의 속도  
Table 3. Processing speed in outer network environment

메시지의 목적과 종류	TCP/IP	SOAP
intermediate(enroll)	263.5	205.8
intermediate(identification)	348	355
intermediate(verification)	246.7	208.8
processed(enroll)	141.7	228
processed(identification)	195.8	265.8
processed(verification)	139.3	272.9

(단위 ms)

표1 은 스탠드 얼론 환경에서의 생체 데이터의 전송 및 처리속도를 비교한 것이다. 중간 처리된 데이터와 완전 처리된 데이터에 대해서 각각 등록, 인식, 검증의 동작을 TCP/IP프로토콜을 통한 BIR 데이터의 전송과 SOAP을 통한 XCBF 형식의 데이터의 전송 및 처리속도를 비교한 것이다. 전송의 속도의 차이가 없다고 보고 데이터를 봐도 상관없다. 직관적인 과상을 하는 처리 보다 XML파서를 사용해서 처리해야 하기 때문에 약간의 시간이 더 들 수 있으나 많은 시간의 차이를 보이지 않고 있으며 적은 양의 처리에서는 조금 더 빠른 경향도 있다.

표2 또한 1과 동일한 데이터와 전송 방식을 통해서 LAN 환경에서의 생체데이터의 전송 및 처리속도를 비교한 것이다. 네트워크의 전송 시에 헤더에 대한 처리의 오버헤드로 인해서 스탠드 얼론 환경 보다 더 많은 속도 차이를 보이고 있다. 그러나 모두 0.1초 내의 시간차이라는 것을 알 수 있다.

표3 또한 1과 동일한 데이터와 전송 방식을 통해서 외부 네트워크 환경에서의 생체데이터의 전송 및 처리에 대한 전송 및 처리속도를 비교한 것이다. 2가지 프로토콜 모두 많은 시간을 보이며 아무리 많은 시간차가 나더라도 0.2초 이내의 시간차를 보이는 것을 알 수 있었다.

위의 실험을 분석하여 보면 TCP/IP 소켓을 이용한 전송이 조금 더 빠른 경향이 보이지만, SOAP프로토콜이 객관적으로 느리다고만 할 수 없는 결과가 나왔다. 따라서 속도가 조금은 느리지만 이 기종이나 다른 프로그래밍 모델을 사용한 응용프로그램 간의 통합성을 볼 때, SOAP 메시지의 사용은 더 많은 장점을 가지고 있다고 할 수 있다.

## V. 결 론

본 논문에서는 인증이나 보안 분야에서 각광 받고 있는 생체정보 분야 있어서 그동안의 많은 연구를 된 데이터 가공이나 인식에 있어서 독립적인 스탠드 얼론 환경에서 응용과는 달리 네트워크 환경에서 BIR 데이터의 XML 포맷인 XCBF 데이터와 SOAP 프로토콜을 사용한 전송에 관하여 연구하고 설계 구현하여 보았다. 비록 TCP/IP 소켓에 비하여 전송 속도 면에서 성능이 다소 떨어지는 경향이 있으나 그에 반하여 SOAP이 주는 장점인 플랫폼 독립적인 환경이나 프로그래밍 환경 독립적인 전송을 고려하여 볼 때, 분산 환경이나 네트워크 환경에서의 응용의 간의 통합에 있어서 훨씬 많은 장점과 XCBF 형식을 사용함으로써 데이터의 가공 및 처리의 융합을 가진 응용을 구현 할 수 있었다.

본 논문에서 구현된 시스템에서는 메시징에서의 날 데이터 전송 시의 문자열 데이터 한계를 가지고 있지만 DIME[11] 전송을 통한 전송으로 해결 할 것이며, 정보전송이기 때문에 전송 시의 암호화에 대한 문제는 MD5[12] 모듈이나 혹은 XCBF의 암호화를 통해서 보안에 대하여 강화할 것이다.

## 참고문헌

- [1] Standard in Biometric API - BAPI, <http://www.iosoftware.com/pages/Products/BiometricAPI/index.asp>
- [2] The Human Authentication API (HA-API) Project , <http://www.biometrics.org/meetings/BC10/Tilton/>
- [3] Extensible Markup Language (XML) 1.1 <http://www.w3.org/TR/2004/REC-xml11-20040204/>
- [4] Version 1.1 of BioAPI specification, <http://www.bioapi.org/BIOAPI1.1.pdf>
- [5] SOAP Version 1.2 Part 1: Messaging Framework, <http://www.w3.org/TR/2003/REC-soap12-part1-20030624/>
- [6] XCBF - XML Common Biometric Format [www.oasis-open.org/committees/download.php/3353/Oasis-200305-xcbf-specification-1.1.doc](http://www.oasis-open.org/committees/download.php/3353/Oasis-200305-xcbf-specification-1.1.doc)
- [7] John G. Daugman, "High confidence visual

recognition of personals by a test of statistical independence". IEEE Trans. Pattern Anal. Machine Intell., Vol. 15, No. 11, pp. 1148-1160, 1993

- [8] Tony Mansfield, etc, "Biometric Product Testing Final Report", Draft 0.6, National Physical Laboratory, March 2001
- [9] <http://www.iridiantech.com>
- [10] Biometric Information Management and Security <http://asn-1.com/x984.htm> X9.84:2003
- [11] [www.gotdotnet.com/teany/xml\\_wsspecs/dime/WSDL-Extension-for-DIME.htm](http://www.gotdotnet.com/teany/xml_wsspecs/dime/WSDL-Extension-for-DIME.htm)
- [12] RFC 1321 (rfc1321) - The MD5 Message-Digest Algorithm

## 저자소개



김재원(Jae-Won Kim)

2003년 광운대학교 컴퓨터공학과 졸업(공학사)  
2003.3~현재 광운대학교 컴퓨터공학과 석사과정

※관심분야 : XML, 웹서비스, 시맨틱웹



박강령(Kang-Ryoung Park)

1994년 연세대학교 전자공학과(공학사)  
1996년 연세대학교 전자공학과(공학석사)  
2000년 연세대학교 전기컴퓨터공학과(공학박사)

2003.3~현재 상명대학교 미디어 학부 전임강사

※관심분야 : 영상신호처리, 생체인식



이수연(Soo-Youn Lee)

1977 연세대 전자공학 석사  
1983 일본교토대 정보공학 박사  
1973~현재 광운대 컴퓨터공학과 교수

※관심분야 : SGML/XML, XML, 전자상거래