생체인식 정보를 이용한 자바카드 인증시스템 구현에 관한 연구

배성환[2], 배지혜[2], 조한신[1], 박윤용[1], 전성익[3]
[1]선문대학교 컴퓨터정보학부
[2]선문대학교 전자계산학과
[3]한국전자통신연구원
e-mail: cosbi@sunmoon.ac.kr

A Study on the Implementation of Java Card Verification Systems Using the Biometric Information

Seong-Hwan Bae[2], Ji-Hyae Bae[2], Han-Shin Jo[1],
Yoon-Young Park[1], Seong-Ik Jeon[3]
Dept. of Computer Science, Sunmoon University

요약(Abstract)

최근 자바카드 기술이 기존의 스마트카드 기술의 문제점들을 해결할 수 있는 새로운 대안으로 등장하고 있고, 카드 소유자의 사용자 인증은 자바카드의 활용을 위해 필수적인 부분이다. 이에 본 논문에서는 스마트카드 상에서 자바카드 실행 환경을 위한 바이오메트릭 API의 구현을 통해 사용자 인증 시스템 구축 방법에 관하여 기술하였다. 본 논문은 먼저 자바카드에 관한 기술 동향 및 분석 등을 기술하였고, 자바카드 바이오메트릭 API에 관하여 설명하였다. 또한, 구현된 바이오메트릭 API를 사용하여 기존의 스마트카드에서 사용되었던 다양한 인증 수단 외에 다양한 생체 정보들을 이용하여 사용자 인증을 제공할 수 있도록 하였다.

1. 서론

본 논문의 목적은 자바카드 BioAPI(Bio-metric API) 기술 분석 및 스마트 카드 상에서 자바카드 실행 환경을 위한 BioAPI를 구현하는 것이다. 최근 자바카드 기술이 기존의 스마트카드의 문제점들을 해결할 수 있는 새로운 대안으로 등장하고 있고, 자바카드 기술은 플랫폼에 독립적인 실행 특성, 통일되고 유연한 개발 환경, 스마트카드 애플리케이션 개발에 유용한 클래스 라이브러리들을 제공함으로써 스마트카드의 활성화를 가속시킬 것으로 기대되고 있다.

자바카드란 자바 언어로 작성된 애플리케이션을 실행시킬 수 있는 스마트카드의 한 종류라고 할 수 있다. 자바카드 기술은 지금까지의 스마트카드들이 하드웨어 환경에 따라 서로 다른 애플리케이션을 사용하고, 이러한 애플리케이션의 개발이 각각의 스마트카드 하드웨어에 맞추어져 이루어지던 것의 자바의 플랫폼에 독립적인 실행 특성을 도입함으로써, 각각의 스마트카드 하드웨어에 구애 받지 않는 통일된 개발 환경을 구축하고, 스마트카드 애플리케이션 개발에 있어서 가장 효과적인 방법으로 자리 잡게 되었다. 이렇게 스마트카드에 자바 기술이 도입된 자바카드 기술은 구체적으로 다음과 같은 장점들을 가지게 된다.[1]

자바카드 기술은 하드웨어 플랫폼에 독립적이고 자바카드 애플릿 방화벽에 의해 서로 보호되는 보안성을 가지며, 개발 시간 및 비용의 절감을 가져오고 다중 애플리케이션의 탑재를 가능케 하여 작업을 보다 안정하고, 쉽도록 함으로써 스마트카드가 더욱 다양한 분야에서 사용될 수 있도록 하고 있다.

본 논문의 2장에서는 스마트카드의 개요에 대하여 기술하고, 3장에서는 자바 카드 인증시스템을 구현하기 위한 설계부분을 기술하고 4장에서는 자바 카드 인증시스템의 실제적인 구현에 대해 기술하며, 그리고 마지막으로 5장에서 결론을 기술하였다.

2. 자바카드에 관한 기술 분석

본 장에서는 자비카드의 기술적인 부분에 대하여 기술하였다. 자바카드 출현의 가장 큰 요인 중 하나는 카드가 가지는 보안제공능력 때문이라고 할 수 있는데 이러한 자바카드의 보안 특성은 보안 메커니즘, 보안 블록과 개인 식별 번호를 통한 메모리 부분에 대한 접근을 차별화 할 수 있으며 파일 접근 제어를 통해 각종 정보를 포함하고 있는 각각의 파일들에 대한 정보의 중요성에 따라 보안정도를 접근 조건을 차별화 할 수 있으며, 자바카드에서 제공되는 보안 메커니즘을 통해서 기밀성을 강화하고 양방향 통신을 통한 인증으로 참여자를 위한 적절한 보안 메커니즘을 가지게 된다. 또한 송신자로부터 전송 받은 데이터가 전

송 상에서 어떠한 변경이 발생하지 않았음을 보장하는 무결성을 제공한다. 또 이 보안 메커니즘은 송신자 자신의 정보를 정확하게 상대방에게 전송하였다고 할지라도 수신자가 이를 부인하거나, 수신측이 정확한 정보를 받았음에도 불구하고 송신측이 자신이 보낸 정보가 아니라고 주장하는 것을 막는 부인봉쇄 능력을 제공할 수 있다.

2.1 보안 블록과 개인 식별 번호

스마트카드는 특정 메모리 부분에 대한 접근을 차별할 수 있도록 하기 위해 스마트카드 제조자, 카드 발행자, 서비스 제공자, 카드 소지자가 서로 독립된 비밀코드를 가질 수 있도록 보안 블록(security block) 기능을 제공한다. 보안 블록은 스마트카드의 메모리 내에 위치하며 카드운영체제(COS)에 의해서만 접근되며 각 디렉터리에 하나의 보안 블록을 두어 디렉터리내의 파일 접근 제어를 수행한다. 보안 블록은 비밀코드(secret code)부분과 비밀코드의 암호화여부, 카드의 초기화 여부, 코드의 제한 회수, 그리고 비밀코드와 관련한 명령어의 종류 등을 위한 접근 조건을 규정하고 있는 비밀코드 기술자(descriptor)로 구성된다. 또한, 스마트카드는 카드 소지자의 정당성을 인증할 수 있는 메커니즘을 제공하는데 이것이 PIN(Personal Identification Number)이다. 카드소지자는 서비스를 제공 받기 위하여 카드리더에 카드를 삽입한 후 보통 8 바이트 길이의 PIN을 입력함으로써 단말기로부터 자신이 정당한 카드 소지자임을 인증 받는다. 그러나, 규격에서 정하는 일정 회수만큼 PIN이 잘못 입력되면 스마트카드는 스스로 잠금 기능을 수행하여 카드 발행자가 재허가 할 때까지 사용하지 못하게 된다.[10]

2.2 파일 접근 제어

메모리에 저장된 각 파일들은 고유의 식별자와 파일 형식, 보안변수, 파일에 대한 접근 조건에 대한 내용을 포함한 각종 정보를 갖는다. 각각의 파일들을 저장하고 있는 정보의 보안정도에 따라 접근 조건이 차별화 되며, 보안변수는 파일이 요구하는 전자서명의 사용여부, 검증을 위한 보안모드 값 등을 갖는다.

2.3 보안 메커니즘

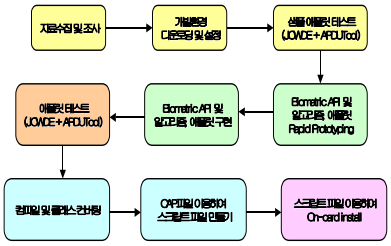
스마트카드에서 제공되는 보안 메커니즘에는 기밀성, 인증, 무결성, 그리고 무인봉쇄가 있다. 기밀성(Confidentiality)은 데이터가 허가 받지 않은 사람(또는 장치)에게 누설되거나 공개되지 않도록 하는 것을 말한다. 인증(authentication)은 수신자이 적법한 지를 확인하거나 수신자에게 자신이 적법한 송신자임을 인식시키는 것이며, 스마트카드에서는 카드 소지자 인증, 카드리더와 스마트 카드간의 인증이 기본적으로 수행되고, 스마트카드를 이용한 서비스 시스템의 참여자 인증을 위한 적절한 보안 정보와 메커니즘을 가지게 된다. 카드소지자의 인증에는 PIN을 이용하는 방법이 가장 보편적이며, 생물학적 특징(지문, 손금, 음성, 망막의 상과 같은 유일한 신체적 특징)을 이용한 인증도 연구개발 되고 있다.

무결성(integrity)이란 송신자로부터 전송 받은 데이터가 전송 상에서 어떠한 변경(삭제, 추가, 또는 데이터의 재배열)이 발생하지 않았음을 보장하는 성질을 말하는데, ISO/IEC 9797에서 정의된 MAC(Message Authentication Code) 무결성 기법(integrity mechanism)을 주로 사용하며, 해쉉 알고리즘, 전자서명 등을 사용할 수 있다.

부인봉쇄(non-repudiation)란 송신자가 자신의 정보를 정확하게 상대방에게 전송하였다고 할지라도 수신자가 이를 부인하거나, 수신측이 정확한 정보를 받았음에도 불구하고 송신측이 자신이 보낸 정보가 아니라고 주장하는 것을 방지하는 것을 말하며, 특히 전자상거래(Electronic Commerce)에서 중요한 역할을 하게 된다. 스마트카드는 이러한 기능을 지원하기 위해서 전자서명, 암호화 알고리즘, 데이터 무결성, 공증 메커니즘 등을 응용한다.

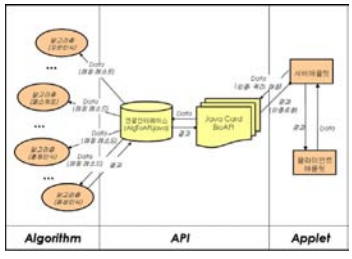
3. 자바카드 생체 인증시스템의 설계

본 장에서는 자바카드 인증시스템을 구현하기까지의 설계적인 측면을 기술하였다. 자바카드 바이오메트릭 인증시스템을 개발하기 위한 과정을 5단계로 나누어 설계하였다. (그림 1)이 각 단계별로 도식화 한 것이고 각 단계별로 내용은 다음과 같다.[2]



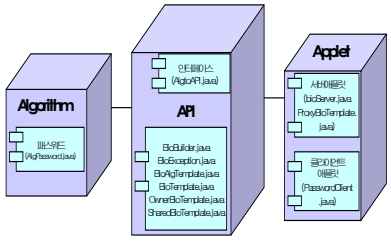
(그림 1) 자바카드 인증시스템의 구현 단계

개발환경 설정 단계, BioAPI 구현 단계, 응용프로그램 구현 단계, 컴파일 및 클래스 변환단계, 카드 설치 단계들을 거쳐 구현된 BioAPI와 자바 응용 프로그램들이 자바카드 상에 인스톨하여 사용되게 된다.[3] (그림 2)는 자바카드 인증시스템의 데이터 흐름도를 도시화한 것이다.



(그림 2) 자바카드 인증시스템의 데이터 흐름도

아래의 (그림 3)은 이들 인증시스템에 대한 컴포넌트를 구성한 것이다.



(그림 3) 자바카드 인증 시스템의 컴포넌트 다이어그램

4. 자바카드 인증시스템 구현

본 장에서는 Sun사에서 제공하는 자바카드 Framework와 JCF의 Biometric API를 이용하여 자바카드 Biometric API를 구현하기 위한 시스템 구조를 기술하였다.

4.1 시스템 구조

자바카드 Biometric API는 4개의 인터페이스와 2개의 클래스로 구성된다.

(1) BioBuilder 클래스 : 다양한 Biometric type을 선언해주고 새로운 OwnerBioTemplate 객체를 만들어 낸다.

(2) BioException 클래스 : Biometric에 지정된 예외 타입들을 제공한다.

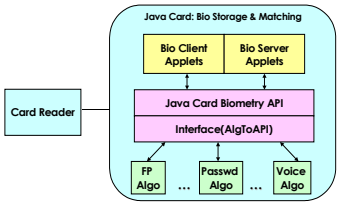
(3) BioAlgTemplate : 알고리즘을 만들기 위한 인터페이스로써 매칭과 데이터의 크기를 지정하기 위해 사용된다.

(4) BioTemplate interface : query, match, enroll에 대한 메소드를 포함하고 있다. 카드상의 서버와 클라이언트 애플릿이 이용할 수 있도록 두개의 sub-interface(OwnerBioTemplate, SharedBioTemplate)의 구성요소를 가진다.

(5) OwnerBioTemplate : 등록을 위한 인터페이스를 포함한다.

(6) SharedBioTemplate : 공유가 가능하도록 BioTemplate를 확장함으로써 클라이언트 애플릿은 바이오메트릭 매니저나 서버 애플릿에 의해 등록된 바이오메트릭을 이용할 수 있다.

Java Card Biometric API와 클라이언트 애플릿, 서버 애플릿, AlgToAPI 인터페이스, 실제 생체정보 매칭 알고리즘과의 관계가 (그림 4)에 도식화되어 있다.[5][6][7]



(그림 4) API, 알고리즘, 애플릿사이의 Interaction과 Java Card와의 연결 관계

4.2 개발단계

Biometric API를 이용한 자바카드 응용 프로그램 개발단계는 아래와 같다.

- Java Card Biometric API 구현 단계
- Interface 구현 단계
- 알고리즘 구현 단계
- Applet 구현 단계

본 논문에서는 API와 Interface 구현단계를 설명하였다.

(1) Biometric API 구현 단계

스마트카드 애플리케이션이 다양한 자바카드에서 동작하고 다양한 Biometric 기술들을 사용할 수 있도록 하기 위해 On-Card Biometric API가 제안되었다. 자바카드는 바이너리와 API의 양쪽에서 상호 운용성이라는 이점을 가지고 있고 Biometric 기술은 On-Card API를 토대로 이루어진다. 특히, 이 API는 안전한 생체인식의 Match-on-Card를 지원함으로써 민감한 바이오메트릭 데이터가 카드 외부로 노출되는 일이 없도록 하였고 보안과 기능성을 제공하는 현재의 자바카드 API 설계를 토대로 만들어 졌다.

(2) 인터페이스 구현단계

인터페이스 AlgToAPI.java는 각 알고리즘과 API와의 연결 인터페이스로서, 매칭 세션을 시작하는 initMatch(), 추가 인증 대상 데이터에 대한 매칭을 진행하는 match(), 등록 세션를 시작하는 init(), 추가가 생체 정보 데이터를 등록하는 update(), 그리고 등록 세션을 마무리하고 인증을 위해 사용할 수 있도록 하는 doFinal() 등의 실제 구현 클래스이다.

다음은 AlgToAPI에서 사용되는 메소드들을 간략히 설명한 것이다.

- initMatch() : 바이오메트릭 매칭 세션을 초기화 또는 재초기화한다. 매칭 스코어가 리턴되는 데, 실제 값은 구현-의존적이다. 예를 들어, 이 스코어 값이 매칭 신뢰도를 나타낼 수 있다. 해당 인스턴스가 봉쇄되지 않는다면 매칭 세션이 시작되고, 다른 처리가 일어나기 전에 유효 플래그가 리셋된다. 또한 재시도 횟수는 감소하고 그것이 0에 도달하면 해당 인스턴스는 봉쇄된다.
- match() : biometric 매칭 세션을 진행한다. 매칭 스코어가 리턴되는 데, 실제 값은 구현-의존적이다. 예를 들어, 이 스코어 값이 매칭 신뢰도를 나타낼 수 있다. 만약 매칭 세션이 진행 중이라면, 이 메소드는 그 세션을 만든다.
- init() : 참조 템플릿의 등록 세션을 초기화한다. 또한 참조 템플릿의 갱신을 위해 사용된다. 유효 플래그를 리셋하고, 이미 갱신 중이라면 이전 참조를 초기화 상태를 해제한다.
- update() : 참조 템플릿의 등록 절차를 진행한다. 이 메소드는 init() 메소드에서 요구되는 모든 입력 데이터가 하나의 바이트 배열로 전달하는 것이 불가능할 때만 사용된다. update()는 여러 번 호출될 수 있다.

• doFinal() : 참조 템플릿의 등록 절차를 완료한다. 등록에 대한 마지막 과정으로, 인증을 위해 사용될 수 있도록 템플릿을 완성한다. 이 루틴은 또한, 참조 템플릿의 인증을 위해 사용되기 전에 필요한 몇몇 오류 검사를 포함 한다.[8][9]

(3) 실행 및 테스트 환경

지금까지 간략히 소개한 구현단계를 근거로 실행 및 테스트를 하였다. 카드 상에 설치하기 위해 거쳐야 할 단계가 있는데 각 단계는 다음과 같다.

- 단계 1: 컴파일
- 단계 2: 컨버트
- 단계 3: 인스톨

javac 컴파일러는 -g 옵션과 함께 사용된다. 이는 디버깅 정보를 생성하기 위해 컴파일러로 알려주는 역할을 한다.

5. 결론

본 논문은 자바카드 BioAPI 기술 분석을 통해 스마트카드 상에서 자바카드 실행 환경을 위한 BioAPI를 구현하는 것을 목적으로 했다. 자바카드 기술은 자바 언어를 이용해 스마트카드를 위한 개방형 애플리케이션 개발 구조를 제공한다. 따라서, 스마트카드 애플리케이션 개발자는 스마트카드 하드웨어에 독립적으로 프로그램을 개발할 수 있다. 생체 인식 기술들은 생체 정보 중립적인 상위 수준의 On-Card API를 토대로 이루어진다. 이 API는 안전한 생체인식의 Match-on-Card를 지원함으로써 민감한 바이오메트릭 데이터가 카드 외부로 노출되는 일이 없도록 한다. 이 API는 보안과 기능성 제공하는 현재의 자바카드 API 설계 위에서 만들어진다. 자바카드가 오늘날 가장 작은 표준화된 컴퓨팅 플랫폼으로 성장하면서, 개발자들은 자바카드와 많은 생체 인식 기술들의 상호 운용성을 보장할 것을 기대하고 있다. 이러한 필요에도 불구하고 Java Card Forum에 의한 JC(Java Card) BioAPI 정의만 존재할 뿐, 아직 구현이 되지 않아 실제로 스마트카드에 적용이 되지 못했다. 따라서, 본 논문에서는 이러한 요구를 충족시키기 위한 자바카드에 내장되는 JC(Java Card) BioAPI를 구현하고, 구현된 API를 이용하여 애플리케이션을 개발하여 생체 인식 기술이 스마트카드에 실제로 활용될 수 있는 기반을 제공하였다. 또한, 본 논문은 여러 종류의 생체 정보를 이용한 스마트카드 인증을 가능하게 함으로써 다양한 생체 인식 기술을 추가할 수 있도록 유연한 구조로 설계되었다. 본 논문의 결과는 차후 스마트카드에 생체 인식 기술을 접목시키는 데 커다란 기여를 했다고 할 수 있다.

참고문헌

[1] 스마트카드 기술 개발 동향,
http://www.kisa.or.kr/technology/sub2/current_smartcard.htm

[2] Java Card™ 2.1 Application Programming Interface, Sun microsystems,
http://java.sun.com/products/javacard/dev_kit.html

[3] Java Card Applet Developer's Guide, Sun microsystems,
http://java.sun.com/products/javacard/dev_kit.html

[4] Java Card 2.0 Reference Implementation User's Guide, Sun microsystems,
http://java.sun.com/products/javacard/dev_kit.html

[5] Java Card™ 2.1.2 Development kit User's Guide, Sun microsystems,
http://java.sun.com/products/javacard/dev_kit.html

[6] Java Card™ Biometric API White Paper(Working Document), http://www.javacardforum.org

[7] Developing a Java Card Applet,
http://wireless.java.sun.com/javacard/articles/applet/

[8] Writing a Java Card Applet,
http://wireless.java.sun.com/javacard/articles/intro/

[9] Writing a Java Card Applet, Part 2,
http://wireless.java.sun.com/javacard/articles/intro/index2.html

[10] Zhiqun Chen, Java Card Technology for Smart Cards, pp. 4-356