자바 카드에서 애플릿간 안전한 객체 공유 방안

전동호[1], 김동휘, 최영근, 김순자
경북대학교 정보통신학과[1], 전자공학과
{jdh0692[1], dewwind, ind}@palgong.knu.ac.kr, sjkim@ee.knu.ac.kr

Secure Object Sharing between Applets in Java Card

Dong-ho Jeon[1], Dept. of Information & Communication,
Dong-Hwee Kim, Yeong-Geun Choe, Soon-Ja Kim Dept. of Electronic
Engineering, Kyungpook Nationanl University

요 약

자바카드 플랫폼은 멀티 어플리케이션 환경이다. 여러 벤더로부터 다중 애플릿은 하나의 카드에 공존할 수 있고 카드 제조사로 부터 추가적인 애플릿이 다운로드 되어진다. 애플릿은 전자 화폐, 지문, 개인 키 값 등과 같은 중요한 정보를 저장하고 애플릿들간에 데이터를 공유하는 경우가 발생하는데 악의를 가진 다른 애플릿으로 데이터가 옮겨질 경우 심각한 결과를 초래할 수 있다. 본 논문에서는 공유 인터페이스를 통하여 애플릿들간의 안전한 데이터 공유와 접근 제한을 고려한 설계방법을 제시하고 이에 관해 논의 해보고자 한다.

1. 서 론

자바 카드는 JAVA 언어로 쓰여진 프로그램을 실행 할 수 있는 스마트카드이다. 자바카드를 사용함으로써 객체지향 프로그램의 장점과 플랫폼 비 의존성 그리고 하나의 카드에 여러 개의 애플릿이 동시에 존재할 수 있으며, 추가적으로 애플릿이 다운로드 되어질 수 있다.

현재 연구되고 있는 동향은 자바 카드에서 애플릿을 최적화 시키는 방법과 카드 단말과 카드간의 트랜잭션간 보안등이다. 자바 카드 플랫폼은 멀티 어플리케이션 환경을 지원하기 때문에 종종 중요한 데이터를 서로 공유하는 경우가 발생한다. 악의를 가진 다른 애플릿으로 데이터가 옮겨질 경우 심각한 결과를 초래 할 수 있다[1,2,3].

본 논문에서는 이를 방지하기 위해 자바 카드에서 애플릿간의 공유 인터페이스를 통해 애플릿간 중요한 데이터를 안전하게 공유할 수 있는 방법을 설명하고 이에 따라 설계한다.

본 논문의 구성은 다음과 같다. 2절에서는 자바카드의 구조와 특징을 살펴보고,3절에서는 애플릿 개발과정에 대해서 설명하고,4절에서는 애플릿간 안전한 객체 공유를 위한 설계방법과 분석, 마지막으로 5절에서는 결론과 향후 연구에 대해 설명한다.

2. 자바 카드의 구조와 특징

자바 카드는 Sun과 스마트카드제조 업체가 연합하여 재정한 규격이다. 카드 제조사가 제공하는 COS(Chip OS) 위에 자바카드 API를 이용해 자바 가상머신을 이식한 형태이다. 스마트카드상의 응용 프로그램인 Cardlet(Card Applet)을 자바를 이용하여 설계한다. 한 번 설계된 Cardlet은 스마트 카드의 종류에 무관하게 사용된다. 애플릿 설계와 개발은 자바카드 spec에 따라서 구현된다[4,5,6].

2.1 자바카드 구조

그림1 에서 자바 카드 구조는 하드웨어와 카드 운영체제, 자바카드 가상 머신, API, 네이티브 메소드, 애플릿으로 이루어져 있다. JCRE(Java Card Runtime Environment)는 자바카드 API, 가상머신, 네이티브 메소드 등으로 구성되고 애플릿 관리는 물론 객체와 class 관리, 입출력 관리, 카드 보안, atomicity 관리등

을 수행하는 중요한 일을 담당하는 부분이다[4,5,6].

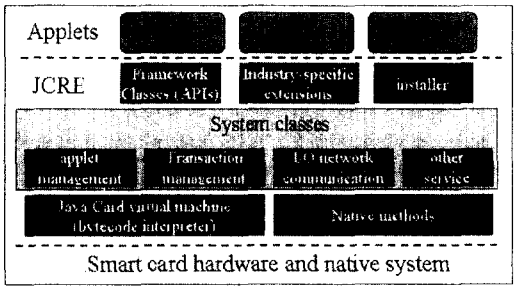

그림 1 자바카드구조

2.2 자바카드 특징

자바 카드는 플랫폼 독립성, 보안성, 객체 지향성 등과 같은 자바의 특성을 가지고 있고 다중 응용 프로그램 지원과 선·후 재 발행 기능과 ISO7816 국제 표준과 호환성을 가진다. 개발자의 측면에서 자바컴파일러와 상용 개발도구를 그대로 사용 할 수 있으며 개방형 API를 제공하는 장점을 가지고 있다. 자바카드 가상머신은 오프 카드부분과 온 카드 부분으로 나누어져 카드의 제한된 리소스를 효율적으로 활용한다[2,3].

3. 자바카드 애플릿

3.1 자바 카드 애플릿

자바 카드 애플릿은 자바 언어로 작성된 응용프로그램은 자바카드 가상머신에 의해서 해석될 수 있는 바이트 코드형태로 개발된다. AID(Application Identifier)에 의해 식별되고 APDU(Application Program Data Unit)교환을 통해 JCRE와 통신을 수행한다. Reactive 응용 프로그램이고 동적으로 다운로드 되어질 수 있다.[4,5,6]

3.2 자바 카드 애플릿 개발 절차

애플릿 개발과정은 패키지를 포함하여 원하는 어플리케이션을 작성한 후, 컴파일해서 바이트 코드로 만들고, 카드 바이너리 파일로 변환 후, 카드에 로딩 및 인스턴스를 생성한다. 단계적으로 보면 먼저 애플릿에 구현하고자하는 기능을 정의하고 각 애플릿 별로 AID값을 할당한다. 다음으로 애플릿에 구현된 메소드를 설계하고 구현된 애플릿 선택을 위한 Select APDU와 Process APDU를 설정하는 단계를 거친다. 애플릿 검증을 위해 시뮬레이션과 에뮬레이션 환경을 거친다.[4,5,6,7]

4. 애플릿간 객체 공유

4.1 애플릿 firewall과 context

그림2 에서 애플릿 firewall은 하나의 애플릿을 정해진 공간에 고립시켜 다른 애플릿에 의해 중요한 데이터가 빠져나가는 것을 방지하며 해킹에 대한 보호를 제공한다. 애플릿 firewall은 context라 불리는 분리 보호되어지는 객체 공간을 나누는 영역이다. 하나의 context와 다른 하나의 context는 firewall에 의해 구분된다.

애플릿이 만들어지면 JCRE는 애플릿을 context에 배정하는데 하나의 패키지에 포함되는 모든 애플릿들은 그룹 context를 공유한다. 같은 애플릿 사이의 객체 접근은 허용되고 그렇지 않은 경우는 거부된다. JCRE는 자신만의 특별한 권한을 가지는 시스템 context인 JCRE context를 가진다.

JCRE context가 활성 상태이면 object는 JCRE에 의해 소유되어진다. 자바카드에서 애플릿이 정적으로 생성되어지고 객체 배열들의 소유권은 패키지의 어떤 애플릿에 할당된다. 객체 배열들의 소유 context는 패키지의 group context이다.

객체가 접근 되어질 때 context가 일치하지 않으면 접근은 거부되고 결과로 SecurityException이 된다. firewall이 제한된 패키지의 애플릿들만의 객체 공유를 가질 수 있다.

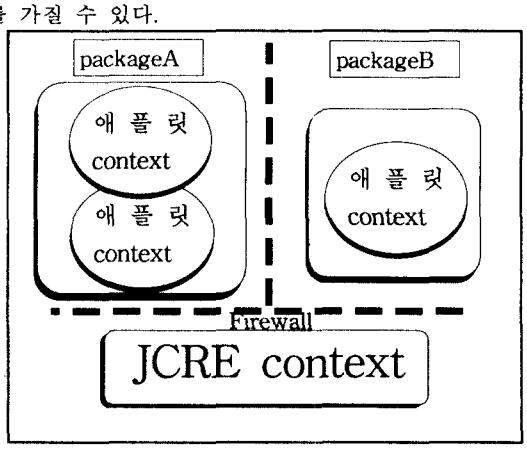

그림 2 애플릿 firewall과 context

4.2 제안하는 안전한 객체공유

애플릿 firewall은 애플릿이 가리키는 context에 제한되고 context를 벗어난 경우는 다음과 같이 제안된 보안된 공유 메카니즘을 통하여 다른 context에 속하는 객체에 접근 할 수 있다. 사용된 대표적인 메소드들을 살펴보면 그림 3과 같다.

```
public static AID lookupAID (byte[ ] buffer, short offset,
                             byte length)
public static Shareable
getAppletShareableInterfaceObject (AID server_aid, byte
                             parameter)
public static Shareable
getAppletShareableInterfaceObject (AID server_aid, byte
                             parameter)
```

그림 3 대표적으로 사용된 메소드들

첫 번째 메소드는 lookupAID 메소드는 JCRE소유의 서버 애플릿의 AID를 반환하거나 null값을 반환한다. 두 번째 메소드는 서버 AID식별을 위한 것이고 파라메터는 보안 토큰으로 사용된다. 세 번째 메소드는 클라이언트 AID식별을 위한 것이고 파라메터는 보안 토큰으로 사용 된다. 각 파라메터는 SIO를 선택하는데 사용된다.

클라이언트 애플릿과 서버 애플릿은 firewall을 경계로 다른 context에 있다. 클라이언트 애플릿이 서비스를 요청하고 서버 애플릿이 SIO를 만든다. 클라이언트 애플릿은 서버 애플릿에게 SIO를 요청하고 SIO 서비스 메소드를 invoke해서 만들어지는 서비스를 제공 받을 수 있다.

그림 4는 다른 패키지에서 애플릿간 객체공유를 요청하는 과정의 요약이다.

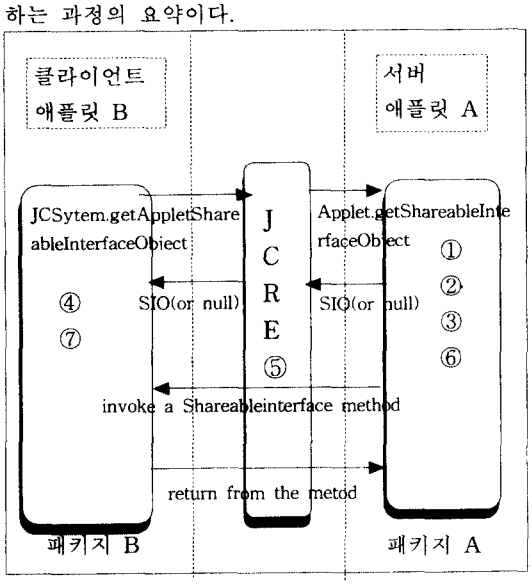

그림 4 다른 패키지에서 애플릿간 객체 공유

다른 패키지에서 보안된 객체 공유의 상세절차는 다음과 같다.

(1) 서버 애플릿 A가 다른 애플릿과 객체 공유를 원하면, 공유 인터페이스 (SI)를 정의한다. javacard.framework.Shareable 인터페이스에서 확장하고 정의된 메소드는 애플릿 A가 다른 애플릿에 접근할 수 있는 서비스를 나타낸다.

(2) 애플릿 A는 공유인터페이스 SI를 수행하는 서비스 제공 클래스 C를 정의한다. C는 SI에서 정의된 메소드를 위한 실제적 동작을 제공하고 애플릿 firewall에 의해 보호되어 진다.

(3) 애플릿 A는 클래스C의 객체 인스턴스 X를 생성한다. X는 애플릿A의 소유이고 Firewall은 A가 X의 필드와 메소드가 접근하는 것을 허락한다.

(4) 만약 클라이언트 B가 애플릿 A의 객체 X에 접근하길 원하면, 애플릿 A로부터 공유인터페이스를 요청하는 JCSystem.getAppletShareableInterface 메소드를 invoke 한다.

(5) JCRE는 애플릿 A를 위한 내부의 애플릿 테이블을 검색하고 발견되면, 애플릿 A의 getshareableinterfaceObject 메소드를 invoke 한다.

(6) 애플릿A는 요청을 받고 애플릿 B와 객체 X의 공유를 원하는지를 결정한다. 애플릿A와 애플릿B와 공유하는 것이 일치하면, A는 X에 대한 참조로서 요청에 응답한다.

(7) 애플릿 B는 애플릿 A로부터 객체 참조를 받아서 SI타입으로 캐스트하고, 객체 참조 SIO에 저장한다. SIO가 A의 객체 X를 참조하지만 SIO는 SI타입이다. 오직 SI에 정의된 공유인터페이스 메소드들 중 하나를 invoke하여 애플릿 A로부터 서비스를 요청 할 수 있다. 서비스를 수행하기 전에, 공유인터페이스 메소드는 서비스를 제공할 것인지 결정하는 클라이언트 B를 인증할 수 있다.

본 논문의 설계와 구현은 펜티엄400환경에서 자바카드 2.1.1Spec을 준수하였고 GemXpresso RAD211 자바 카드 툴킷을 사용하여 검증하였다. 제안된 방법을 이용하면 카드 발행 후 컨텐츠를 제공하는 많은 애플릿들이 다운로드 되어는 것을 신뢰 할 수 있다. 다운 로드된 애플릿은 같은 패키지가 아니더라도 애플릿간의 안전한 객체 공유가 이루어진다.

5.결 론

본 논문에서는 멀티 어플리케이션이 지원되는 자바카드 환경의 firewall을 벗어나 공유 인터페이스를 이용한 자바카드 애플릿간의 안전한 데이터 공유를 위한 설계방법을 제안하고 검증하였다. 향후에는 자바카드플랫폼 보안을 이용해 하드웨어 제한적인 자바카드를 효율적으로 이용할 수 있는 방안에 대해 연구하여야 한다.

6. 참 고 문 헌

[1] M.Oestreicher and K.Ksheeradhi, "*Object Life time in Java Card*", Usenix Workshop Smart Card Technology, 1999.

[2] M.Oestreicher, "*Transaction in Java Card*", Annual Computer Security Applications Conf, IEEE, 1999.

[3] Michael Baentsch, Peter Buhler,,Thomas eirich, Frank horing, and Marcus Oestreicher, "*JavaCard From Hype to reality*", IEEE Concurrency, 1999.

[4] Sun Microsystem, "*,Java Card 2.1 Development kit User's Guide*", 1999.

[5] Sun Microsystem, "*Java Card 2.1 Specification*", 1999.

[6] Zhiqun Chen, "*Java Card Technology for Smart Cards*", Addison-Wesley, 2000.

[7] W.Rankle, W. Effing, *Smart Card Handbook*, John Wiley & Sons, 1996.

[8] *http://web3.javasoft.com/features/2000/20/cartes.print.html*

[9] scott B.Guthery, "*JAVA CARD*", Internet Computing On a Smart Card, IEEE Internet computing, 1997.

[10] ISO 7816 Specification Parts 1-6.