간편 공인인증서 인터페이스 가이드라인

Interface Guideline for HTML5 Based PKI Implementation

v1.00

2016년 08월

KISA 한국인터넷진흥원
KOREA INTERNET & SECURITY AGENCY

목 차

간편 공인인증서 인터페이스 가이드라인
Interface Guideline for HTML5 Based PKI Implementation

1. 개요

스마트폰과 같은 모바일 기기의 USIM, ICCard, MicroSD, 트러스트존 등 보안토큰을 모바일 통신을 통해 PC와 같은 타기기에 연결하여 전자서명하는 경우 가입자 PC에 각 보안토큰과 통신하는데 필요한 별도의 전용 구동프로그램(PKCS#11)들을 필수적으로 설치해야하는 제약이 있다. 이러한 제약으로 인해 가입자소프트웨어 등이 순수 웹 표준 기술로 구현되어 있는 경우에도 모바일기기를 활용하여 전자서명을 할 경우 가입자가 해당 구동프로그램을 PC에 설치를 해야 하는 불편함이 있다.

따라서, 데스크톱 환경에서 각각의 보안토큰 전용 구동프로그램을 설치하지 아니하고 유저인터페이스 측면에서 보안토큰 이용방식 특성에 따라 상이할 수 있는 사용자경험이 되지 않는 일관되고 범용적인 경험을 할 수 있도록 한 것이 간편 공인인증서 인터페이스 가이드라인의 목적이다.

본 가이드라인에서는 브라우저를 이용하는 사용자 측면에서 PC내에 별도의 전용 프로그램 설치 없이 모바일 디바이스를 활용하여 공인인증서를 안전하고 편리하게 이용할 수 있도록 순수 웹 환경 하에서 전자서명이 요구되는 가입자소프트웨어 등과 간편 공인인증서 서비스 제공자가 갖추어야 할 인터페이스 관련 요구사항을 기술한다.

2. 구성 및 범위

본 가이드라인에서는 간편 공인인증서(이하 간편인증) 모델을 제시하고 가입자 측면에서 일관되고 범용적인 사용자 경험을 보장하기 위해 가입자소프트웨어 및 서비스 제공자에게 요구되는 최소한의 기능 및 인터페이스에 대해 아래와 같이 명시하고 있다.

- -. 가입자 소프트웨어 내 간편인증 저장매체 추가 요구사항
- -. 간편인증 UI/UX 인터페이스 최소 요구사항
- -. 이종·유사 간편인증 서비스 도메인 간 상호연계를 위한 인터페이스 요구사항
- -. 모바일 디바이스에서 전자서명을 위한 인터페이스 요구사항
- -. 기타 보안 요구사항(Security Consideration)

3. 관련 표준

3.1 국외 표준 및 규격

[PKCS11] RSA Laboratories PKCS#11, Cryptographic Token Interface Standard v2.11, 2001

3.2 국내 표준 및 규격

[KCAC.TS.HSMUI] KISA, *스마트인증 인터페이스 가이드라인 v2.00, 2013.11*

[KCAC.TS.HSMU] KISA, *보안토큰 기반의 공인인증서 이용기술 규격 v2.10*, 2012.11

[KCAC.TS.CM] KISA, *무선단말기에서의 공인인증서 저장 및 이용 기술규격 v1.30*, 2012.11

[KCAC.TG.DGH] KISA, *보안토큰 구동프로그램 배포 가이드라인 v1.0*, 2012

3.3 기타

[PC/SC] PC/SC Workgroup, *PC/SC Workgroup Specifications 2.01.3*, http://www.pcscworkgroup.com/specifications/overview.php

[USB 2.0] USB Implementers Forum, Inc., *Universal Serial Bus Revision 2.0 Specifications,* http://www.usb.org/developers/docs/

[ISO7816] ISO/IEC 7816, *Identification Cards – Integrated Circuit(s) cards with contacts Part 1 to 10*

4. 정의

4.1 전자서명법 용어 정의

본 가이드라인에서 사용된 다음의 용어들은 법률 제6585호 및 동법 시행령에 정의되어 있다.

가) 공인인증서

나) 가입자

다) 가입자 소프트웨어

4.2 용어의 정의

가) 간편인증 : PC와 같은 기기에서 전자서명이 요구될 때 해당 기기에 별도의 구동프로그램 설치 없이, 모바일기기와 연동하여 키생성 및 전자서명생성을 처리하는 기술의 총칭

나) 다중 간편인증 서비스 연계 : 간편인증 서비스 호출 후 타 간편인증 사용자일 경우 해당 서비스 제공자의 간편인증 서비스로 연계하는 방식의 총칭

4.3 용어의 효력

본 규격에서 사용된 다음의 용어들은 [RFC2119]를 준용하며 다음과 같은 의미를 지닌다.

가) 해야한다, 필수이다, 강제한다 (기호 : M)
반드시 준수해야 한다.

나) 권고한다 (기호 : R)
보안성 및 상호연동을 고려하여 준수할 것을 권장한다.

다) 할 수 있다, 쓸 수 있다 (기호 : O)
주어진 상황을 고려하여 필요한 경우에 한해 선택적으로 사용할 수 있다.

라) 권고하지 않는다 (기호 : NR)
보안성 및 상호연동을 고려하여 사용하지 말 것을 권장한다.

마) 금지한다, 허용하지 않는다 (기호 : X)
반드시 사용하지 않아야 한다.

바) 언급하지 않는다, 정의하지 않는다 (기호 : -)
준수 여부에 대해 기술하지 않는다.

5. 약어

본 가이드라인에서는 다음의 약어가 이용된다.

가) IC Card : Integrated Circuit Card, 집적회로 카드

나) microSD : Micro Secure Digital Card

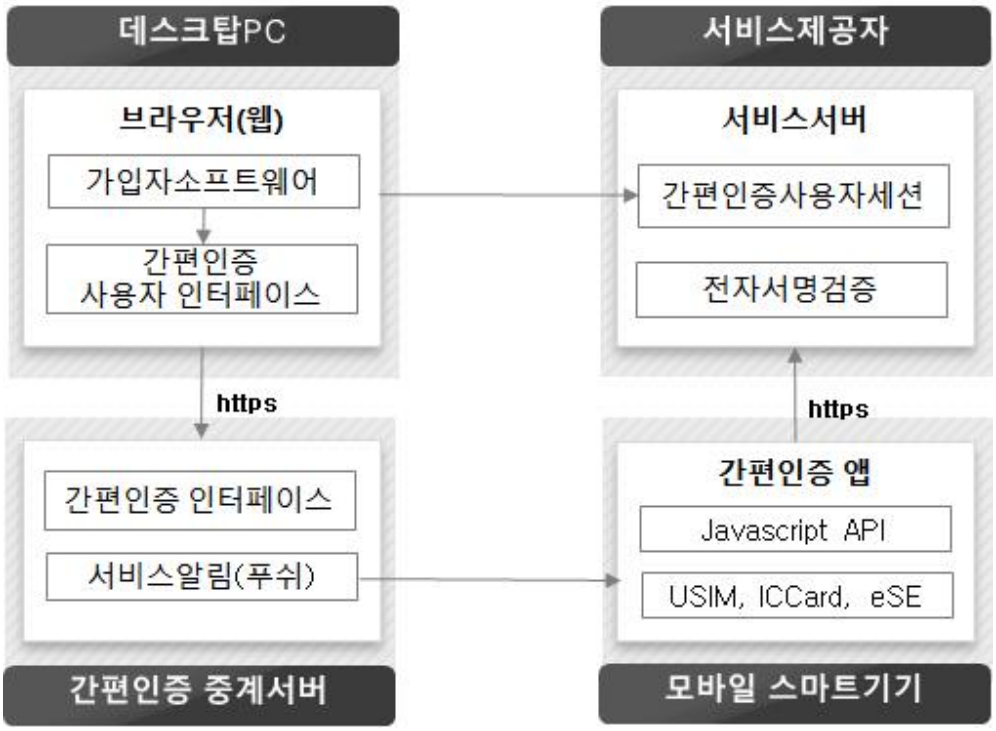
다) PIN : Personal Identification Number, 개인식별번호

라) USIM : Universal Subscriber Identity Module, 범용사용자식별모듈

6. 간편인증 인터페이스

6.1 인터페이스 모델

데스크톱환경의 가입자소프트웨어에서 간편인증 처리를 위한 간편인증 연계, 모바일기기에서 전자서명을 위한 간편인증 앱과 서버스 서버 제공자 간의 일반적인 인터페이스 모델은 [그림 1]과 같다.



[그림 1] 간편인증 인터페이스 모델

모바일 기기에 전자서명을 요청하기 위해서 가입자 소프트웨어는 간편인증 사용자 인터페이스를 간편인증 중계서버를 통해 제공받을 수 있으며, 이 때 모든 간편인증 중계서버는 별도의 설치가 요구되지 않고 순수 웹 표준 기술로 구현된 사용자 인터페이스를 제공하여야한다.

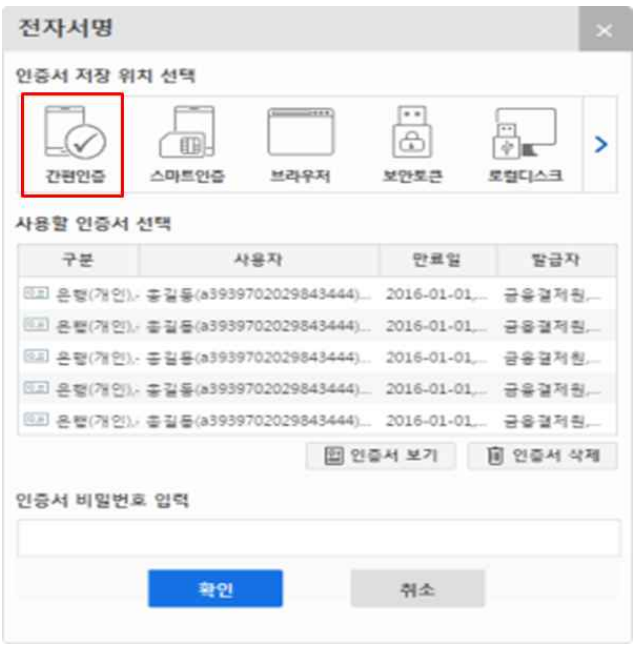
가입자 소프트웨어와 간편인증 중계서버는 웹 프로토콜(HTTPS)을 이용하기 때문에 데스크톱PC와 간편인증 앱 간의 종단(End-To-End) 보안에 취약할 수 있다. 따라서 사용자 인터페이스를 통해 사용자가 요청한 모바일기기의 간편인증 앱을 호출한 이후 간편인증 앱은 전자서명을 위한 원문정보수신 및 제출을 데스크톱 브라우저의 가입자 소프트웨어, 간편인증 중계서버를 통해 송수신하지 않고

서비스 서버와 직접통신을 하여 처리할 것을 권고한다.

6.2 간편인증 이용

모바일 기기의 간편인증 앱과의 연결을 위해서는 [그림 2]과 같이 전자서명이 필요한 웹 페이지 내에 [간편인증] 버튼 메뉴를 추가하거나, 가입자 소프트웨어 내에 저장매체-[간편인증] 메뉴를 아래 [그림 3]과 같이 추가하여 이용할 수 있다.

[그림 2] 간편인증 서비스 직접호출 예시



[그림 3] 순수 웹 기반(HTML) 가입자 소프트웨어 예시

모바일 기기 내의 간편인증 앱을 통해 전자서명을 위해서 간편인증 사용자 인터페이스를 [그림 4]과 같이 가입자에게 제공되도록 해야 하며, 위 제시된 2가지 방식의 [간편인증] 메뉴 선택 시 간편인증 중계서버를 통해 간편인증 사용자 인터페이스를 요청하여 가입자에게 제공하는 것을 권고한다.

USIM, ICCard, 트러스트존 등의 매체를 이용하여 간편인증 서비스를 제공하는 각각의 간편인증 중계서버는 [그림 4]과 같이 일관된 사용자 인터페이스로 구현하여 가입자에게 동일한 사용자경험과 다양한 해상도를 가진 디바이스에서 동작될 수 있는 반응형으로 제공할 것을 권고한다.

간편인증 사용자 인터페이스에는 각각의 간편인증 앱에서 제공하는 보안매체를 추가해야하며, 매체별 연결방식에 따라 모바일 기기와 연결을 위한 휴대폰번호와 같은 추가적인 정보를 요구할 수 있다.

[그림 4] 순수 웹 기반(HTML) 간편인증 사용자인터페이스 예시

모바일기기와 연결을 위한 추가적인 정보입력 시 휴대폰번호와 같은 개인정보는 타인이 재사용 하지 않도록 보안을 고려하여 입력을 요구한 정보 전체를 임시저장(Cache)하지 않는 방식으로 제공할 것을 권고한다.

7 기본요구사항

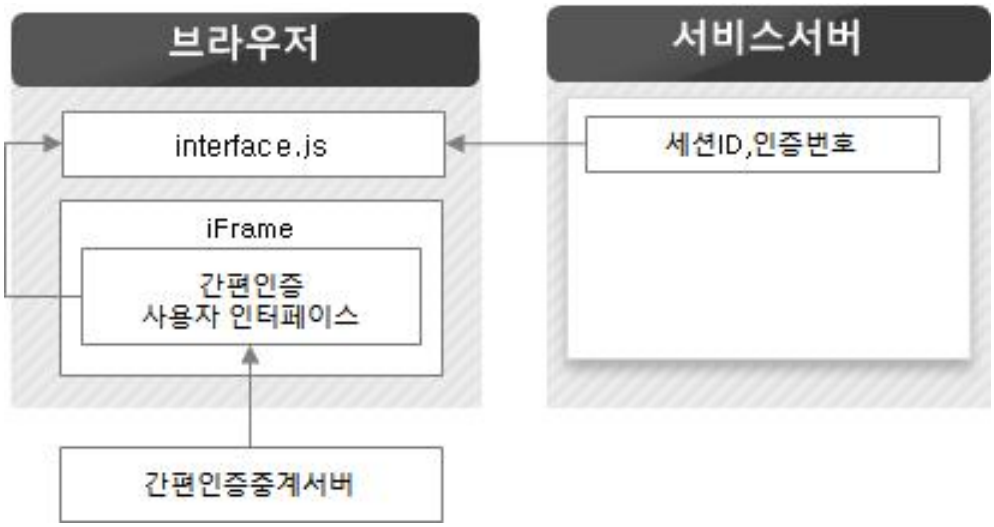
7.1 간편인증 서비스 호출

간편인증 서비스 이용을 위해서는 스마트폰 내 간편인증 앱과 연결을 위한 간편인증 사용자 인터페이스가 사용자에게 제공되어야 하며, 이 인터페이스는 서비스 서버 또는 간편인증 중계서버에서 제공받을 수 있다.

간편인증 중계서버로부터 제공받는 경우, 최초 접속한 CP 서비스 도메인과 간편인증 중계서버 도메인이 다를 수 있으며, [그림 5]과 같이 간편인증 중계서버는 크로스 도메인 환경에서 서비스 이용될 수 있도록 관련 기능을 서비스 서버에 제공해야한다.

간편인증 중계서버로 전송하는 정보 중 서비스 서버에서 생성하는 정보(세션, 인증번호 등)는 간편인증 사용자 인터페이스에서 사용자 정보입력이 완료된 이후 서비스 서버로 부터 전달받아 간편인증 중계서버로 전달하는 방식으로 구성하여 서비스 서버의 자원이 낭비되지 않도록 효율적으로 구성할 것을 권고한다.

[그림 5]의 예시는 간편인증 사용자 인터페이스를 호출하기 위한 웹 페이지 내에 iFrame을 구성하고, iFrame 내에 간편인증 사용자 인터페이스를 출력시키며, 사용자에게 요구된 입력이 완료된 이후 서비스 서버로부터 동적정보(세션ID 등)를 얻기 위해 웹 스크립트 인터페이스를 통해 정보를 획득하는 과정이다.[부록 1] 참조



[그림 5] 순수 웹 기반(HTML) 간편인증 사용자 인터페이스 예시

간편인증 사용자 인터페이스에서 간편인증 중계서버로 전송하는 파라미터는 [표 1]와 같이 서비스 개시와 간편인증 앱 간의 연결을 위한 필수정보와 선택정보로 구성하며, 정의된 파라미터 이외에 서비스 서버가 추가한 파라미터 정보가 있을 경우 각 중계서버는 확장된 파라미터를 간편인증 앱으로 전달하여야 한다.

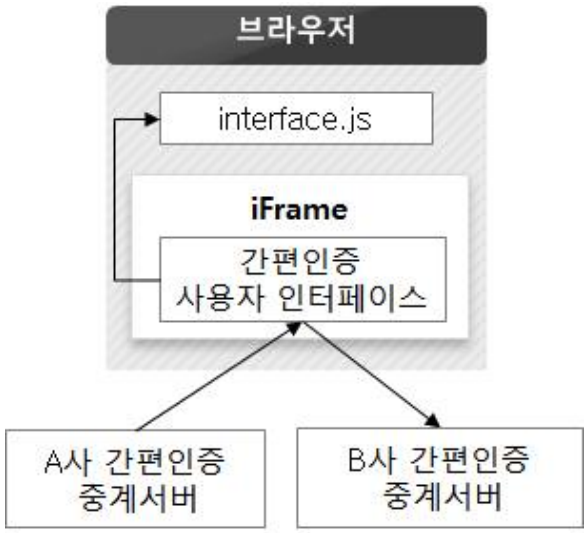
파라미터	구분	설명
cp_info	M	"|"를 구분자로 표현한 서비스서버정보- "domain|cp_code|name"
callbackurl	M	간편인증앱에서 CP서버로 접속하기 위한 정보
server_data	M	서비스서버가 생성한 암호화된 callback 파라미터 정보 base64인코딩(대칭키암호화(txid=xxx&data1=xxx&data2&.....))
telco	O	이용통신사정보 (SKT:0, KT:1, LGU:2)
phone_number	O	휴대폰번호
confirm_code	O	서비스서버에서 생성한 사용자 확인코드

[표 1] 간편인증 중계서버 서비스호출 파라미터

7.2 다중 간편인증서비스간 연계처리

간편인증 중계서버는 동일 매체별로 여러 서비스 제공자가 있을 수 있다. 이 경우 최초에 간편인증 중계서버가 수신한 사용자정보를 타 서비스 제공자로 연계하여 [그림 6]과 같이 사용자가 연계된 간편인증 서비스를 이용할 때 재입력 하지

않도록 입력정보를 타 서비스제공자에게 연계시킬 수 있어야 한다.[부록 1 참조]



[그림 6] 간편인증 중계서버 서비스 연계

7.3 간편인증 서비스 앱과 서비스서버 연계

간편인증 중계서버는 [표 1]와 같이 간편인증 앱과 연계하기 위한 정보를 수신하게 된다. 간편인증 중계서버는 이 정보 중 간편인증 앱이 서비스 서버에 접속하기 위한 정보를 푸쉬 등과 같은 전송방법을 통해 간편인증 앱에 전달할 수 있다.

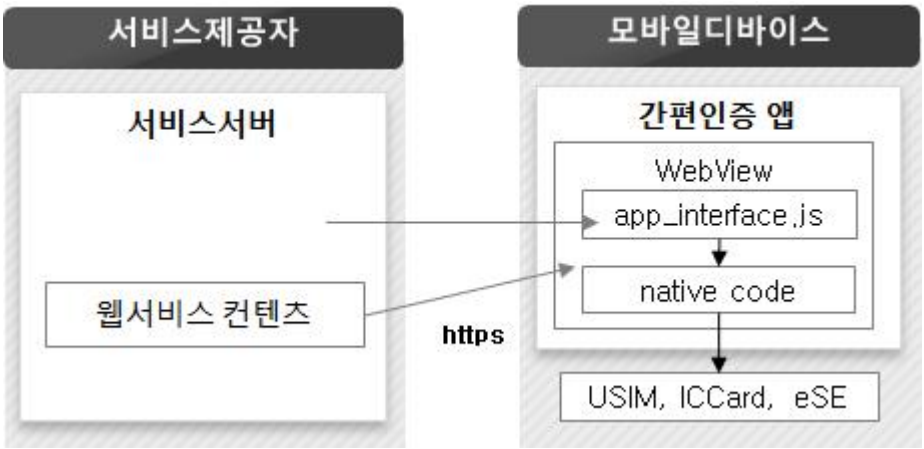
푸쉬와 같은 방법으로 구동된 간편인증 앱은 서비스서버에 접속할 정보 중 접속대상 서비스 서버를 callbackurl로 하고, session_id 및 서비스 서버가 추가한 파라미터들을 포함하여 아래와 같이 서비스 서버에 서비스 요청을 해야 한다.

```
https://callbackurl?server_data=xxxx&서버확장파라미터s=xxxx
```

7.4 간편인증 서비스 앱 구현요구사항

간편인증 앱은 서비스서버와 연계하여 데스크톱 환경에서 필요한 전자서명 및 키 생성 기능을 수행할 수 있으며, 발급되는 인증서는 USIM, ICCard, 트러스트존 등 안전한 저장매체로 저장 및 이용하는 것을 권고한다.

[그림 7]과 같이 서비스 서버는 전자서명을 위해 사용자에게 동의, 입력 등을 요구할 수 있으며, 간편인증 앱은 입력, 화면전환과 같은 다양한 형태의 서비스 구성에 대응할 수 있는 구현과 서비스 서버에 관련 기능 처리를 위한 정보를 제공하는 것을 권고한다.[부록2] 참조



[그림 7] 간편인증 앱 논리구성과 서비스 서버 간 연계 예시

8. 보안 고려사항

본 가이드라인에서 제시하는 간편인증 서비스는 데스크톱 등 웹브라우저에서 별도 설치가 없는 무설치 기반으로 스마트폰 간편인증 앱과 연계 후 서비스 제공자가 요구하는 전자서명을 간편인증 앱을 통해 제출하여 데스크톱에서 필요한 인증처리를 대신해주는 순수 웹 환경으로 공격자로부터 송수신 구간 트래픽 탈취와 사용자 미인증 연결 허용 등의 보안위협들에 노출될 수 있다. 따라서 안전한 간편인증 서비스 제공을 위해서 아래 보안위협을 고려하여 구현할 것을 권고한다.

- 구간암호화

간편인증 중계서버는 https 프로토콜을 이용하여 송수신하는 모든 데이터를 암호화해야 하며, SSL/TLS를 위한 이외의 포트는 서버에서 사용하지 않아야 한다.

간편인증 앱이 서비스 서버에 접속하기 위한 callbackurl의 프로토콜은 "https"로 정의하여야 한다.

- HTTP Request Method

중계서버, 서비스 서버에 http request시 전송하는 method는 "POST" 방식으로 전송할 것을 권고한다.

● 브라우저 Cache 데이터

사용자가 입력한 정보(휴대폰번호 등)를 임시로 보관하는 경우 제3자가 유용하지 못하게 전체정보를 cache하지 않기를 권고한다.

● 사용자 본인확인

웹브라우저와 스마트폰 연결을 위한 이용자 인증절차가 없는 경우, 누군가 의도된 목적으로 간편인증 사용자 인터페이스에 피해자의 연결정보를 입력하고, 피해자가 무의식적으로 연결을 받아들였을 경우 공격자는 피해자의 계정정보 등에 접근할 수 있다. 따라서 간편인증 중계서버 또는 간편인증 앱에서 데스크톱 웹브라우저 이용자와 스마트폰 접속이 동일인임을 확인하는 기능을 구현하여야 한다.

● 서버데이터 보안

서비스 제공자의 데이터는 사용자 웹브라우저 세션과 간편인증 앱 사용자의 구분 및 연결 등의 목적으로 서비스 제공자 서버에서 임의의 데이터를 생성 후 간편인증 중계서버를 통해 간편인증 앱까지 전달하게 되므로, 생성된 임의의 데이터는 경유되는 모든 서버와 간편인증 앱에서 정보를 알 수 없도록 암호화하여 전달하여야 한다.

● 트랜잭션ID 고려사항

간편인증 앱과의 연결을 위한 트랜잭션ID는 웹브라우저와 연결된 세션정보를 그대로 이용할 수 없으며, 안전한 난수생성기를 통해 유일한 트랜잭션ID를 생성하여야 한다.

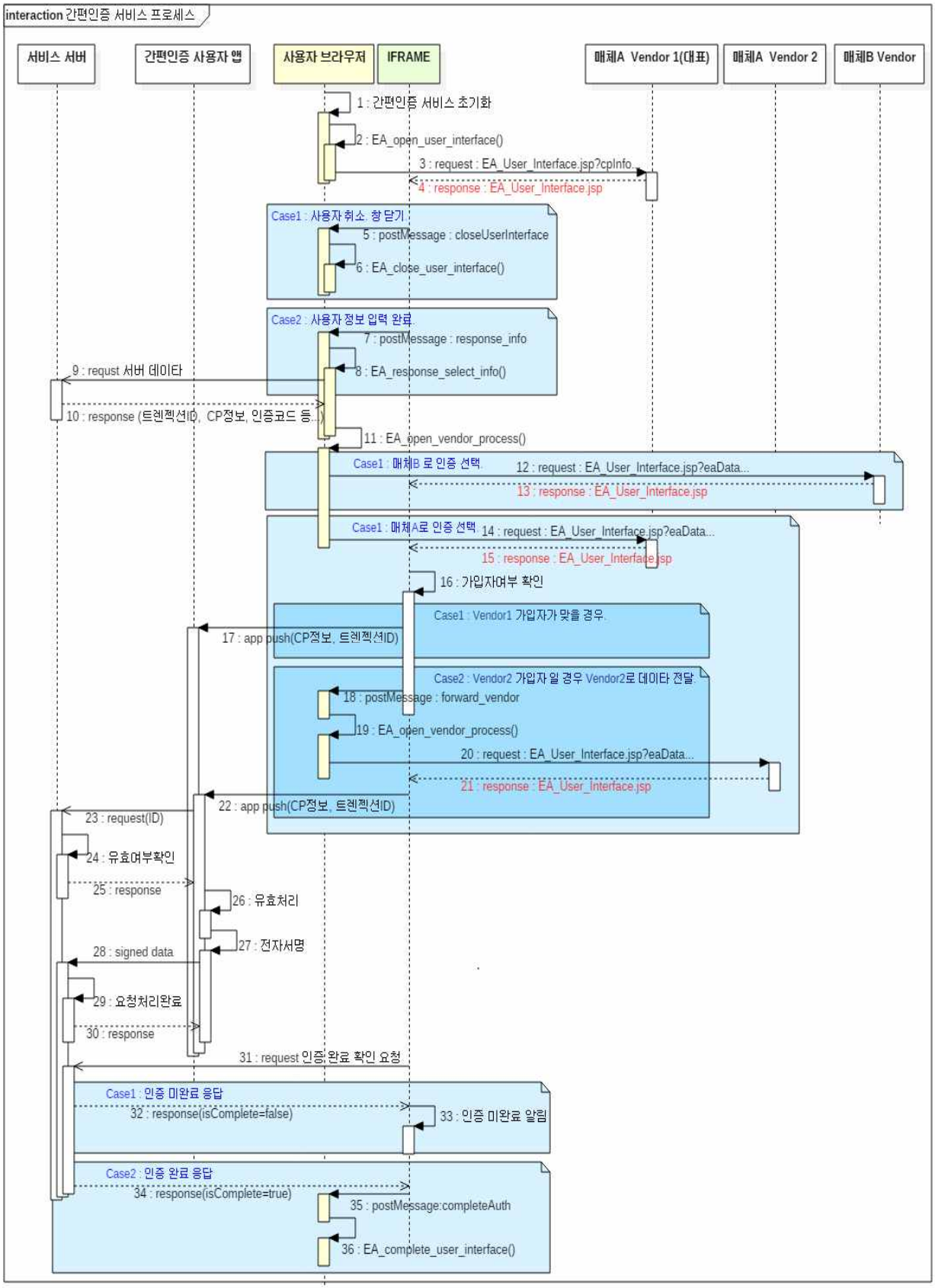
서비스 제공자는 간편인증 앱과의 연결이 이루어진 이후 재사용된 트랜잭션ID의 요청을 거부할 수 있도록 구현하여야 하며, 일정 시간 동안 연결이 없는 트랜잭션은 삭제하는 기능을 갖추어야 한다.

● 간편인증 앱 사용자 입력보안

간편인증 앱은 서비스 제공자로부터 전자서명원문 전체를 수신 받아 전자서명을 할 수도 있지만, 서비스 제공자는 PC에서 입력하는 정보의 보안을 고려해서 데스크톱 웹브라우저에서 주요 민감정보를 입력하지 않고 간편인증 앱을 통해 주

요정보를 입력 후 안전하게 전자서명 하는 과정이 필요할 수 있다. 따라서, 간편인증 앱은 후자의 경우를 고려하여 민감정보 입력보안을 위해 가상키패드와 같은 입력보안기능을 탑재할 것을 권고한다.

* 부록1 간편인증 사용자인터페이스와 서비스서버 연계 Flow 및 javascript API



자바스크립트 함수	함수설명
void EA_init()	초기화. 서비스 호출을 위한 iframe element 생성.
void EA_open_user_interface (*service_callback_function*)	iframe으로 간편인증서비스 대표도메인 사용자 인터페이스 로드.
void EA_close_user_interface()	iframe 영역 닫기.
void EA_response_select_info(dataobj)	인터페이스창 사용자 입력 정보 전달.
void EA_open_vendor_process(forwardurl, dataobj)	서비스 대기창 로드(유사 서비스 간 연계 포함).
void EA_complete_user_interface()	인증프로세스 완료 처리

void EA_init();

간편인증 사용자 인터페이스 초기화

사용자 화면에 iframe 생성

```
( ifram element 생성 예 )
var EA_iframe = '<div id="dscertContainer"><iframe id="EA_frame" name="EA_frame" src="" width="0px"
height="0px" frameborder="0" allowTransparency="true"
style="position:fixed;z-index:100010;top:0px;left:0px;width:100%;height:100%;"></iframe></div>';
$(EA_iframe).appendTo('body');
```

void EA_open_user_interface(service_callback_function)

iframe에 사용자 인터페이스 창 열기.

request get 방식으로 사용자 인터페이스 창으로 파라메타 전달

```
( ifram url 설정 예 )
src="https://www.usimvender1.com?cpInfo=domain|cp_code|name&lgUrl=www.service_server.com:8443"
```

Parameters

service_callback_function 간편인증사용자인터페이스 종료 후 호출될 서비스서버 CallBack 함수

Remark

인증 완료 후 서비스 화면으로 전달 할 결과 데이터 형식을 정의 한다.

void EA_close_user_interface()

간편인증 사용자 인터페이스창(인증 대기화면)을 닫는다.

iframe 내부에서 부모창으로 닫기 요청을 할 경우, key를 송신하여 요청한다.

PostMessage Key : closeUserInterface

void EA_response_select_info(dataobj)

iframe 내부에 사용자 인터페이스 창에서 사용자가 입력한 정보를 부모창에 전달.

서비스 서버로부터 세션 아이디 등 추가 정보를 획득하여 사용자 인터페이스 창에서 입력 받은 데이터와 병합 한 후, 각사의 서비스 대기 화면으로 송신

(서비스 서버와 ajax 통신 데이터 정의)

Parameters

message	원문 메세지

Returns

cpInfo	cp 서버 정보(코드, 도메인, 명칭)
sessionID	간편인증 앱과 동기화를 위한 세션아이디
confirmCode	서비스 서버에서 생성한 인증 코드
callbackUrl	간편인증앱에서 웹으로 접속하기 위한 정보

PostMessage Key : response_info

Parameters

dataObj	- JSON type data object	
	cp_info	서비스 서버 정보
	service_type	인증 타입(usim, ic, trustzone)
	telco	이용통신사 정보(SKT:0, KT:1, LGU:2)
	phone_number	휴대폰 번호

void EA_open_vendor_process(forwardUrl, dataObj)

iframe에 서비스 대기 화면을 열기.

사용자가 선택한 인증 타입에 따라 서비스 대기화면으로 연결한다.

유사서비스가 존재 할 경우 가입여부에 따라, 포워드 요청을 받아 처리 할 수 있다.

PostMessage Key : forward_vendor

Parameters

forwardUrl	URL 문자열. 유사 서비스간 연계를 위한 URL(null 일 경우 설정 되어있는 대표 도메인 URL로 연결)	
dataObj	JSON type data object	
	cp_info	서비스 서버 정보
	service_type	인증 타입(usim, ic, trustzone)
	telco	이용통신사 정보(SKT:0, KT:1, LGU:2)
	phone_number	휴대폰 번호
	session_id	서비스서버에서 생성한 휴대폰과 인증 동기화를 위한 세션아이디
	confirm_code	서비스서버에서 생성한 인증코드
	callbackurl	간편인증앱에서 CP서버로 접속하기 위한 정보

void EA_complete_user_interface()

인증완료 확인 후,

iframe에 서비스 대기 화면을 닫고 (EA_close_user_interface() 호출),

서비스 화면의 다음 단계로 진행한다.

PostMessage Key : completeAuth

부록 2. 간편인증앱과 서비스서버 연동 javascript API

필수 여부	자바스크립트 함수	함수설명
M	String GetAppVersion ()	간편인증앱 버전을 리턴한다.
O	String GetTokenInfo (tokenInfoType)	지정한 매체정보를 리턴한다.
M	void ExitApp()	간편인증앱을 종료한다.
M	void GetVenderInfo(infoType)	앱 제조사, 앱명을 리턴한다.
M	void SetFilter (key, values)	전자서명시 지정한 속성에 해당하는 인증서만 보이도록 설정한다.
O	void AddUnSignedAttribute (oid, oidValueBase64)	전자서명문내에 주어진 oid로 속성을 추가한다.
M	String MakeToBeSignData(form)	form을 QueryString으로 변경한다 (전자서명원문으로 사용할 경우)
M	void StartSignData (signType, signOption, tobeSignData, base64Time, ssn, ase64SignAttribute)	주어진 타입, 데이터를 입력으로 전자서명을 요청한다.
M	void OnSignedDataSubmit(errorCode)	StartSignData()결과로 전자서명이 완료시 호출되는 callback함수
M	String GetSignedDataBase64 ()	base64인코딩된 전자서명 값을 획득한다.
O	String AddUnSignedAttributeWithSignedDataBase64 (signedDataBase64)	SignedData에 특정속성을 추가한다.
M	String GetVIDRandomBase64 ()	전자서명한 인증서의 base64인코딩된 VID값을 리턴한다.
M	String GetSignerSubjectDN ()	전자서명한 인증서의 주체식별정보(DN)를 획득한다.
M	String GetSignerSubjectAltName ()	전자서명한 인증서의 주체대체명을 획득한다.
O	String GetSignerCertificateBase64 ()	전자서명한 인증서의 base64인코딩된 인증서를 획득한다.
O	String GetAuthNumber ()	서비스서버에서 생성한 인증번호를 획득한다.(푸쉬내 포함시)
M	String GetCertCount ()	매체에 저장된 인증서 개수를 획득한다.
O	void SmartMoveCert (moveType)	PC또는 스마트폰SD에 있는 인증서를 저장매체내에 저장한다.
O	void IssueCert (caType, ip, port, refNum, authCode)	지정된 발급정보로 인증서를 발급한다.
O	void UpdateCert (caType, ip, port)	지정된 갱신정보로 인증서를 갱신한다.
O	void RevokeCert (caType, ip, port)	지정된 폐지정보로 인증서를 폐지한다.
O	void HoldCert (caType, ip, port)	지정된 효력정지정보로 인증서를 효력정지 한다.
O	void OnCMPSubmit (errorCode)	인증서 발급/갱신/폐지/효력정지에 대한 수행이 이루어진 후 자동호출 된다. 성공일 경우 주체DN, 발급자DN, 유효기간 시작, 유휴기간 끝 등을 획득할 수 있다.
O	void GetCMPCertInfo (infoCertAttributes)	발급/갱신/폐지/효력정지를 수행한 인증서에 대한 정보를 획득한다.
O	void StartSmartCert (serialNumber)	지정한 일련번호에 해당하는 인증서로 포커스 이동해서 간편인증앱 메인화면을 실행한다.
M	String void GetErrorCode ()	에러코드를 획득한다.
M	String GerErrorMessage (errorCode)	에러코드에 해당하는 메시지를 가져온다.

String GetAppVersion();

간편인증앱 버전을 리턴한다.

Returns 간편인증앱 버전

String GetTokenInfo(tokenInfoType)

지정한 매체정보를 리턴한다.

Parameters

tokenInfoType	INFO_USIM_ICCID – Base64(ICCID Hash value) INFO_USIM_PHONENUMBER – 휴대폰번호

Returns 토큰정보 타입에 해당하는 값

void ExitApp()

간편인증앱을 종료한다.

void GetVenderInfo(infoType)

앱 제조사, 앱명을 리턴한다.

Parameters

infoType	INFO_VENDER_NAME – 앱 제조사명 INFO_APP_NAME – 앱명

Returns 타입에 해당하는 값

void SetFilter(key, values)

전자서명시 지정한 속성에 해당하는 인증서만 보이도록 설정한다.

Parameters

key	"OID" 인증서 정책 OID '|'를 구분자로 사용 예) .2.410.200005.1.1.4|1.2.410.200004.5.3.1.1 "CA" CA인증서의 SubjectDN '|'를 구분자로 사용 예) CN=CrossCert Certificate Authority,OU=AccreditedCA,O=CrossCert,C=KR|CN=yessignCA Class 1,OU=AccreditedCA,O=yessign,C=kr "SPECIFIC" SubjectDN, IssureDN, serialNumber에 해당하는 인증서만 보이도록 설정 '&'를 구분자로 사용 예) 1. subject=테스트-개인-2048&issuer=CrossCertTestCA2&serial=4453, 2. issuer=CrossCertTestCA2 "EXPIRE" 0=만료된 인증서표시, 1=만료된 인증서표시 안함
values	key에 해당하는 값

void AddUnSignedAttribute(oid, oidValueBase64);

전자서명문내에 주어진 oid로 속성을 추가한다. StartSignData(), AddUnSignedAttributeWithSignedDataBase64()를 호출하기 전에 미리 속성을 add해야 한다.

Parameters

oid	Attribute OID
oidValueBase64	base64로 인코딩된 Attribute OID Value

void MakeToBeSignData(form);

form을 QueryString으로 변경한다 (전자서명원문으로 사용할 경우)

Parameters

formHTML formReturns

QueryString

void StartSignData(form, signType, signOption, tobeSignData, base64Time, ssn, base64SignAttribute)

주어진 타입,데이터를 입력으로 전자서명을 요청한다. 서명결과는 OnSignedDataSubmit()을 이용하여 확인할 수 있다.

Parameters

signType	SIGNATURE – 서명값 SIGNEDDATA_PKCS7 – PKCS#7에서 정의한 전자서명 형식 SIGNEDDATA_CMS – CMS에서 정의한 전자서명 형식 SIGNEDDATA_KOSCOM – 코스콤에서 정의한 전자서명 형식
signOption	OPTION_NONE – 옵션을 지정하지 않음 OPTION_NO_CONTENT_INFO – 원본 메시지를 포함하지 않음
tobeSignData	전자서명 하고자 하는 원문
base64Time	서명문내에 입력된 서명시간을 설정한다(Base64 인코딩된 time_t 값). 입력이 null, “””일 경우 현재시간으로 자동 설정된다.
ssn	주민등록번호(사업자등록번호) 입력값이 not null일경우 본인확인 검증 성공일 경우 전자서명 한다.
base64SignAttribute	signAttribute를 base64로 인코딩한 값(null or ""이면 서명원문과 비교하지 않는다.)

void OnSignedDataSubmit(errorCode)

SmartCert.js를 포함하는 웹페이지에 overrding해야 하며, 전자서명이 완료되면 자동 호출된다. 성공일 경우 전자서명 값, VID, 주체DN등을 획득할 수 있다.

```
예)
<HTML>
<script language="javascript" src="SmartCert.js"></script>
.....
<script language=JavaScript>
        OnSignedDataSubmit(errorCode)
        {
                if (errorCode == 0)
                {
                        form.signedData.value = GetSignedDataBase64();
                        form.submit();
                }
                else
                        alert( GetErrorMessage(errorCode) );
        }
</script>
</HTML>
```

String GetSignedDataBase64();

base64인코딩된 전자서명결과를 획득한다. OnSignedDataSubmit() 결과가 성공한 후 사용되어야 한다.

Returns

base64로 인코딩된 전자서명 값

String AddUnSignedAttributeWithSignedDataBase64(signedDataBase64);

이미 생성된 전자서명문 내에 특정속성을 추가할 경우에 사용한다. 예) GetSignedDataBase64() -> AddUnSignedAttribute() -> AddUnSignedAttributeWithSignedDataBase64().

Parameters

signedDataBase64 GetSignedDataBase64() API 호출하여 받아온 서명 데이터

Returns

Base64로 인코딩된OID와 OID Value를 추가한 전자서명 값

String GetVIDRandomBase64();

전자서명한 인증서의 base64인코딩된 VID값을 리턴한다. OnSignedDataSubmit() 결과가 성공한 후 사용되어야 한다.

Returns

Base64로 인코딩된 VID Random

String GetSignerSubjectDN();

전자서명한 인증서의 주체 식별정보를 획득한다. OnSignedDataSubmit() 결과가 성공한 후 사용되어야 한다.

Returns

주체 식별정보

String GetSignerSubjectAltName();

전자서명한 인증서의 주체대체명을 획득한다. OnSignedDataSubmit() 결과가 성공한 후 사용되어야 한다.

Returns

주체대체명

String GetSignerCertificateBase64();

전자서명한 인증서의 base64인코딩된 인증서를 획득한다. OnSignedDataSubmit() 결과가 성공한 후 사용되어야 한다.

Returns

Base64로 인코딩된 전자서명한 인증서

String GetAuthNumber()

서비스서버에서 생성한 인증번호를 획득한다.(푸쉬내 포함시)

Returns

인증번호

int GetCertCount();

매체에 저장된 인증서 개수를 획득한다. SetFilter()와 같이 사용하게 되면 필터링 된 인증서의 개수를 반환된다. 따라서 인증서 발급/갱신 시 SetFilter()와 같이 사용하게 되면 실 저장된 인증서의 개수를 반환하지 않으므로 유의해야 한다.

Returns

인증서 개수

void SmartMoveCert(moveType);

PC또는 스마트폰내 SD에 있는 인증서를 저장한다.

Parameters

moveType	ALL – SD 또는 PC에 인증서를 사용자가 선택하여 저장매체에 저장 SD – SD메모리에 있는 인증서를 저장매체에 저장 PC – PC에 있는 인증서를 저장매체에 저장

void IssueCert(caType, ip, port, refNum, authCode);

인증서를 발급한다. 발급결과는 OnCMPSubmit()을 이용하여 확인할 수 있다.

Parameters

caType	CROSSCERT-한국전자인증CA, KICA 한국정보인증CA, SIGNKOREA 코스콤CA, TRADESIGN-한국무역정보통신CA,YESSIGN-금융결제원CA
ip	CA IP
port	CA Port
refNum	참조번호
authCode	인가코드

void UpdateCert(caType, ip, port);

인증서를 갱신한다. 갱신결과는 OnCMPSubmit()을 이용하여 확인할 수 있다.

Parameters

ip	CA IP
port	CA Port

void RevokeCert(caType, ip, port);

인증서를 폐지한다. 폐지결과는 OnCMPSubmit()을 이용하여 확인할 수 있다.

Parameters

ip	CA IP
port	CA Port

void HoldCert(caType, ip, port);

인증서를 효력정지 한다. 효력정지 결과는 OnCMPSubmit()을 이용하여 확인할 수 있다.

Parameters

ip	CA IP
port	CA Port

void OnCMPSubmit(errorCode)

SmartCert.js를 포함하는 웹페이지에 overriding해야 하며, 인증서 발급/갱신/폐지/효력정지에 대한 수행이 이루어진 후 자동호출 된다. 성공일 경우 주체DN, 발급자DN, 유효기간 시작, 유휴기간 끝 등을 획득할 수 있다.

```
예)
<HTML>
<script language="javascript" src="SmartCert.js"></script>

<script language=JavaScript>
        OnCMPSubmit(errorCode)
        {
                if (errorCode == 0)
                {
                        form.subjectDN.value = GetCMPCertInfo(CERT_ATTRIBUTES.SUBJECT_DN);
                        form.issuerDN.value = GetCMPCertInfo(CERT_ATTRIBUTES.ISSUER_DN);
                        form.validityFrom.value = GetCMPCertInfo(CERT_ATTRIBUTES.VALIDITY_FROM);
                        form.validityTo.value = GetCMPCertInfo(CERT_ATTRIBUTES.VALIDITY_TO);

                        form.submit();
                }
                else
                        alert( GetErrorMessage(errorCode) );
        }
</script>
</HTML>
```

StringGetCMPCertInfo(infoCertAttributes);

발급/갱신/폐지/효력정지를 수행한 인증서에 대한 정보를 획득한다.

Parameters

infoCertAttributes	SUBJECT_DN – 주체DN ISSUER_DN – 발급자DN VALIDITY_FROM – 유효기간 시작 일자 VALIDITY_TO – 유효기간 만료 일자 SERIAL_NUMBER – 일련번호

Returns

인증서 속성 값

void StartSmartCert(serialNumber)

지정한 일련번호에 해당하는 인증서로 포커스를 이동해서 간편인증앱 메인화면을 실행한다.

Parameters

serialNumber	인증서 일련번호(Hexa String)

String GetErrorCode();

에러코드를 획득한다.

Returns
에러코드

String GetErrorMessage(errorCode);

에러코드에 해당하는 메시지를 가져온다.

Returns
에러메시지

에러코드	메시지
0000	**성공**
1001	사용자 취소
2001	검색된 인증서가 없음
3001	본인확인에 실패
3002	개인키에 VID R값 없음
3003	저장공간이 부족
4001	이상징후 감지로 서명중단
4002	기타 인증서 발급 실패
4003	기타 인증서 갱신 실패
4004	효력정지 실패
4005	폐지 실패
5001	알수없는 파라미터(파라미터에 지정한 형태의 속성과 값을 설정 확인)
6001	PIN 잠김
9999	알수없는 오류