실적설명서

※ 컴퓨터로 작성한 뒤 PDF로 저장하여 원서접수시스템에 업로드 하십시오.(양식 변경 불가, 서명하지 않음.)

인적 사항

지원자 성명	박재현	수험번호	22[redacted]
지원학과	웹프로그래밍과	연락처	010-[redacted]

실적 내역

실적물 제목	KoalaSign
보조영상 주소	

KoalaSign(코알라사인)은, 자바스크립트 상에서, 공동인증서(구 공인인증서)를 사용한 전자서명법상 전자서명인증을 구현하고 전자서명인증업무를 제공하도록 돕는 소프트웨어의 모임입니다.
이 모임은 현재 다음 세 가지로 구성되어 있습니다:

- ✓ accredited-certificate
- ✓ @koalasign/storagelist
- ✓ Koala

먼저, accredited-certificate는 주어진 공동인증서를 공동인증서 규격[KCAC.TS.CERTPROF]에 따라 해석하는 자바스크립트 모듈입니다. 따라서 공동인증서를 직접 처리하는 기능을 구현하는 데에 핵심적으로 사용됩니다. 이 모듈은 드라이브에 저장된 공동인증서를 해석하는 것을 목표로 하므로, 규격[KCAC.TS.UI]에서 정한 디스크 내 공동인증서의 저장방법에 따라, 주어진 공동인증서는 DER 인코딩된 것이어야 합니다.
이 모듈은 여러 가지 사용 환경을 효율적으로 지원하기 위해, 모듈이 사용되는 자바스크립트 환경에서 지원되는 경우,

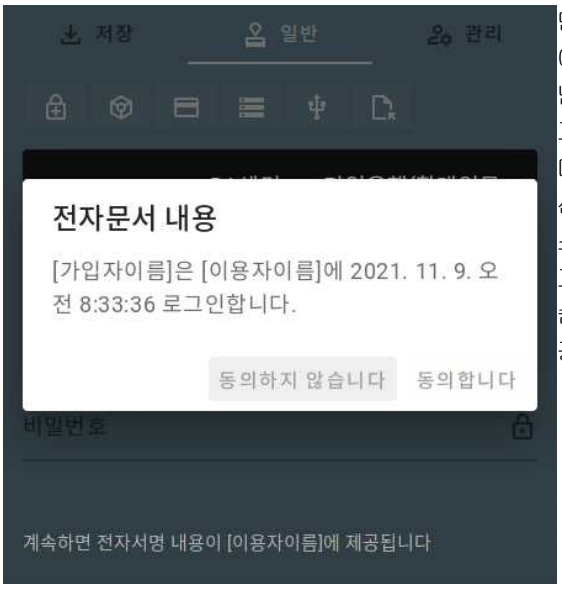
- ✓ BER(DER도 BER에 해당합니다) 규격의 키를 자바스크립트 객체로 변환하는 라이브러리 ASN1js,
- ✓ 객체로 변환된 키가 X.509 인증서일 경우 그 인증서의 내용을 정리하는 라이브러리 PKIjs,
- ✓ 그리고 버전 15.6.0부터 OpenSSL 기능이 활성화된 빌드인 경우 PKI 기능을 내장하는 자바스크립트 런타임 Node.js

등의 보조 소프트웨어를 사용합니다.
해석된 공동인증서는 공동인증서를 구분하는 명칭인 인증서(공개키) 또는 전자서명생성키(비밀키)이며, 이로부터 원하는 정보를 얻거나 전자서명인증에 필요한 비밀키를 복호화할 수 있습니다. 현재 accredited-certificate 모듈은 전자서명인증에 사용되는 '가입자 전자서명 인증서'만 정상적인 공동인증서로 판단하여 해석하고 있으며 나머지 특수한 공동인증서를 정상적으로 지원하기 위한 작업이 진행중입니다.
이 모듈은 모든 공동인증서 기술규격을 만족하는 것을 지향합니다. 이미, 가입자 소프트웨어가 반드시 지원해야 하는 해시 알고리즘[KCAC.TS.HASH]을 모두 지원하는 등 규격의 대부분을 만족합니다.
타입스크립트로 이전(migration)하는 작업 중에는 git stash 사용을 잘못하여 파일을 전부 지워버려 아직 타입스크립트를 잘 지원하지 않습니다.

다음으로, @koalasign/storagelist는 디스크 드라이브 상에 저장된 공동인증서의 목록을 만드는 타입스크립트 모듈입니다. 이 모듈은 파일시스템을 사용하므로 Node.js 환경에서만 작동됩니다.
[KCAC.TS.UI]에 따르면 공동인증서는 디스크에 저장되어도 무관하며, 따라서 가입자 대부분은 별도의 보안 하드웨어 없이 디스크 저장 방법을 사용합니다. 이 모듈은 이러한 가입자들이 사용할 공동인증 소프트웨어에서 더 나은 사용자 경험(User eXperience)을 위해 사용될 수 있도록 구현하였습니다.
시스템에 마운트된 저장장치 파티션의 목록을 가져오려면 일반적으로 시스템 플랫폼에 따라 win32, libusb 등의 저수준(low-level) 소프트웨어를 사용합니다. 이미 이런 형태로 구현된 라이브러리 drivelist를 사용하여 신뢰성을 높였습니다. 해당 라이브러리는, Node.js 상에서 컴파일된 바이너리의 프로세스를 실행하는 대신, C 어댑터(Node.js 네이티브 애드온)와 GYP(Generate Your Projects) 기술을 사용하여 직접 동적 라이브러리 호출을 하도록 구현되어있습니다.
@koalasign/storagelist는 시스템 드라이브와 USB 저장장치를 탐색하며, 시스템 드라이브 외의 다른 하드디스크 파티션이나 광학 디스크(CD 등)는 기본적으로 인증서 저장소 목록에 포함하지 않습니다.

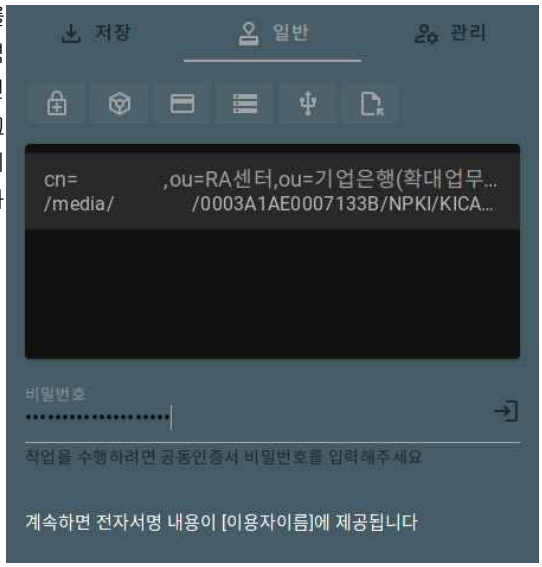
마지막 Koala는 KoalaSign 프로젝트의 최-고수준(highest-level) GUI 소프트웨어입니다. 공동인증 가입자 소프트웨어에 해당하는 도구이며 전자서명 가입자 S/W 표준 인터페이스 제작 가이드라인(Guideline on a user interface for digital signature software developers)을 기준으로 구현하였습니다.
Koala는, Electron과 유사한 특징을 보이는 시스템 소프트웨어 프레임워크인 Tauri로 만들어졌습니다. Tauri는 Rust로 개발되었으며 웹사이트를 별도의 소프트웨어로 만들어주는 도구입니다. 플랫폼마다 내장된 시스템 브라우저; 웹뷰를 이용하므로, Electron과 다르게 완성된 소프트웨어에 브라우저와 Node.js 바이너리를 내장하지 않아 용량이 작고 첫 가동 시간(startup time)이 매우 짧습니다. 이렇게 하면 Node.js 내장 모듈들을 Rust 코드와 자바스크립트 프로세스 간 통신으로 대체하여 사용해야 하지만 정말 안전합니다.
Koala의 가입자 기능은 현재 저장(공동인증서 다운로드), 일반(전자서명), 관리(공동인증서 관리) 세 가지이며 전자서명 관련 기능만 완전히 작동합니다. 공동인증서 검증을 위한 OCSP 기능, 보안토큰 및 스마트카드 전자서명 기능, 그리고 다운로드 및 저장 기능은 개발 중에 있습니다.
안타깝지만 Koala는 KoalaSign의 나머지 소프트웨어인 accredited-certificate와 @koalasign/storagelist를 모두 사용하지 않습니다. 두 가지 모듈 모두 Node.js에 대한 의존성을 일부 가지고 있으며, 파일시스템 기능을 사용하지 않는 경우 Node.js에 의존하지 않게 할 수 있으나 완벽하지 못하여 해당 모듈들은 개선 후에 사용할 계획입니다. 일시적으로는 ECMAScript 표준인 SubtleCrypto를 위주로 사용하고 있습니다.
Rust 코드를 이용해, (현재는 Windows 플랫폼을 한하여) 비밀번호 입력 화면 상태에서 메모리 보호와 스크린 캡처 보호 기능이 동작하도록 구현하였습니다. 이 기능은 가입자 소프트웨어로서 상당히 차별화된 기능입니다.
디자인은 어떨까요? 아름답습니다. Google이 Material UI 표준을 제시하고 직접 선보인 Material Components Web(MCW)을 사용하므로 최신 스마트폰의 UI를 닮았습니다. 사용자 인터페이스(UI) 구현에 사용된 라이브러리는 정확하게는 SMUI(Svelte Material UI)인데, 이 라이브러리는 MCW의 기능 이상으로 표준을 지키도록 강제하고 있으므로 흐트러짐 하나 없이 대칭적입니다. SMUI의 이름을 들으니 하나 궁금해지셨을 것으로 생각합니다. 바로 Svelte입니다. Koala 웹의 프레임워크는 Svelte로, 가상 DOM(Document Object Model)과 실시간 업데이트를 사용하지 않고 Svelte 코드에 작성된 스크립트를 완전히 트랜스파일(transpile)하여 웹의 아름다움을 지키고 불필요한 성능 낭비를 하지 않습니다.
이제 Koala는 어떻게 만들어졌나 알아봅시다. Koala는 자바스크립트 소프트웨어로서 구상되었으나 Tauri를 사용하므로 기본적으로 Rust 소프트웨어입니다. 대신에 이 프로젝트 안에는 package.json을 기준으로 자바스크립트 프로젝트가 하나 존재합니다. 이 자바스크립트 프로젝트는 빌드 스크립트를 통해 폴더 하나 분량의 정적 웹사이트로 만들어질 수 있는 형태이며 빌드된 이 정적 사이트와 Rust 코드에서 빌드된 바이너리 코드가 모두 합쳐져 ELF, EXE와 같은 실행 가능한 앱으로 완성됩니다. 자바스크립트 프로젝트를 한번 봅시다. 이 프로젝트는 Webpack과 같으면서도 다른 목표를 지닌 코드 번들러(bundler) Rollup을 사용합니다. 웹 프레임워크로는 플레인(plain) Svelte를 직접 사용하며 이러한 플레인 환경에서도 효율적인 개발을 위해

페이지네이션(pagination, 또는 라우팅) 기능을 구현하는 Routify를 사용합니다. 모든 실행 스크립트와 페이지별 스크립트는 타입스크립트로 작성했습니다. 또, SMUI의 편리함을 유지하면서 맞춤 디자인을 할 수 있는 Advanced Styling Method로, 여러 테마(theme)를 구현하는 동시에 SCSS/Sass와 PostCSS를 이용해 시스템 웹뷰의 렌더링 엔진과 상관없이 같은 디자인을 보여줄 수 있도록 하였습니다.
다음은 지금까지 진행된 개발의 상황을 대략 알 수 있는 두 장의 사진입니다. Koala를 이용한 전자서명의 과정을 설명과 함께 봅시다. 모든 디자인은 프로토타입이며 특히 색상은 모두 변경할 예정입니다.



먼저 서명할 텍스트 또는 파일(즉, 전자문서)을 사용자에게 확인하도록 합니다. 단, 로그인 화면에서 Koala로 넘긴 경우 등 가입자가 이미 전자문서의 내용을 파악하고 있는 경우에는 생략할 수 있습니다.
다음으로 공동인증서 목록에서 서명에 사용할 인증서를 선택합니다. 위의 토글 버튼으로 왼쪽부터 Zbox 저장소, 보안토큰, 스마트카드, 하드 디스크, USB 디스크, 그리고 직접 설정한 저장소의 공동인증서를 표시하거나 숨길 수 있습니다만, 현재 하드 디스크와 USB 디스크의 공동인증서만 이용할 수 있습니다.

인증서를 선택하였다면 비밀번호라고 적힌 텍스트 상자를 클릭하고 인증서 비밀번호를 입력합니다. 비밀번호를 입력하기 시작하면 텍스트 상자 오른쪽의 자물쇠 아이콘이 전자서명 목적에 맞는 아이콘으로 바뀝니다. 사진에는 로그인 목적의 전자서명이므로 로그인 아이콘이 보입니다. 비밀번호를 다 입력하였다면 키보드의 엔터 키를 누르거나 로그인 아이콘을 클릭하여 로그인을 완료합니다.



지금까지 KoalaSign 프로젝트의 구성을 함께 보았습니다. 우리 프로젝트가 매력적이라고 생각하시면 좋겠습니다. 고맙습니다. 끝.

본 실적설명서에 작성한 내용은 본인이 만든(또는 팀과 함께 작업한) 실적물에 대한 설명이며, 실적물의 대리 제작, 타인의 저작물 도용 등 결격사유가 발견 될 경우 불이익을 받을 수 있음을 확인합니다. 또한, 필요시 실적물을 제출하여 평가받을 수 있음을 알고 있습니다.

2021년 11월 9일 / 지원자 박 재 현