

**Kiwinest 요구사항 명세서**

2024.05.18 3차 수정 – 최창호 Choi Chang Ho



1. **Introduction 개요**
   1. Purpose 목적
   2. Document Convention 문서 규칙
   3. Reference 참조
2. **Overall description 전체 설명**

2.1 Project perspective 프로젝트 조망

2.2 Project functions 프로젝트 기능

2.3 Target user classes 목표 사용자 계층

2.4 Constraints 제약사항

2.5 Assumptions and dependencies 가정과 종속관계

2.6 User Documentation 사용자 문서

1. **Specific requirements 세부적 요구사항**
   1. Interface requirements 인터페이스 요구사항
   2. Functional requirements 기능적 요구사항
   3. Maintainability & Portability 유지보수와 호환성

**1. Introduction 개요**

Kiwinest는 실시간으로 문서 또는 코드를 암호화하여 보관하는 소프트웨어다. 실시간 암호화란 사용자 개입 없이 데이터가 저장되기 직전에 자동으로 암호화되고 로드된 직후에 해독된다는 의미이다. 올바른 비밀번호/키 파일 또는 올바른 암호화 키를 사용하지 않으면 암호화된 볼륨에 저장된 데이터를 읽을(복호화할) 수 없다. Kiwinest는 Flutter 프레임워크를 통해 개발되며 Windows, Mac 환경에서 구동할 수 있다.

* 1. **Purpose 목적**

RSA는 암호화와 디지털 서명을 위한 알고리즘으로 널리 사용된다. 하지만 상업용/개인용으로 나온 유료 소프트웨어 또는 오픈소스 소프트웨어는 존재하지 않는다. Kiwinest는 범람하는 정보의 시대에서 기업용/개인용 RSA 암호화 소프트웨어의 입지를 확대하고자 개발되었다.

* 1. **Document Convention 문서규칙**
  2. **Reference 참조**

**2. Overall description 전체 설명**

* 1. **Project Perspective 프로젝트 조망**

Kiwinest는 기본적으로 RSA 암호화 프로그램이다. 2021년 제작된 파이썬 코드 암호화 소스코드 ‘RSA2C.py(깃헙링크)’를 시작으로 이에 문서 암호화 기능과 액세스 관리, GUI를 추가한 것이 Kiwinest이다. Kiwinest는 2024년 2분기 안으로 Github를 통해 공개되며, Github 커뮤니티나 Kiwinest 자체 사이트에서 프로젝트에 기여를 할 수 있다. 추후에는 파티션 분리 암호화 등이 구현될 수 있다.

* 1. **Project functions 프로젝트 기능**
     1. **RSA text encryption RSA 텍스트 암호화**

소인수 분해, 모듈러 연산, 오일러 Phi 함수, 유클리드 호제법 등의 수학적 개념들로 구성된 RSA는 높은 안전성을 보장하기에 다음과 같은 소(小)기능에 사용된다.

문서 암호화: 문서를 불러온 후 문서의 모든 텍스트를 10진수 정수로 암호화하여 기밀성을 유지하고 보안을 강화하기 위해 문서를 암호화하는 과정이다. 일반적으로 사용자는 RSA 알고리즘과 키를 사용하여 문서를 암호화하고, 이를 해독하기 위해서는 암호화된 문서를 복호화하기 위한 암호화 키가 필요하다.

코드 암호화: 코드를 불러온 후 코드의 모든 텍스트를 10진수 정수로 암호화하여 기밀성을 유지하고 보안을 강화하기 위해 문서를 암호화하는 과정이다. 일반적으로 사용자는 RSA 알고리즘과 키를 사용하여 문서를 암호화하고, 이를 해독하기 위해서는 암호화된 문서를 복호화하기 위한 암호화 키가 필요하다.

* + 1. **Secure folder 보안 폴더**

2.2.1과 동일하게 RSA를 사용하며 각 OS의 문서 폴더에 생성된 ‘Nest 파일’ 에 대한 접근을 관리한다. Nest 파일은 파일 탐색기나 Finder를 통해서 내용을 볼 수 없으며 Nest에 대한 접근은 Kiwinest를 통해서만 가능하다. Kiwinest에서의 접근은 암호 입력으로만 할 수 있다.

* + 1. **Cycle clean-up 사이클 클린-업**

RSA 암호는 안전성을 늘리기 위해 키 값의 크기를 늘려야 하는데, 이는 자연히 많은 계산 리소스를 필요로 한다. 때문에 Kiwinest는 안전성과 계산 리소스 또한 확보하기 위해 주기적으로 낮은 상태의 키 값을 새롭게 갱신하고 암호화/복호화하는 사이클 클린-업을 사용한다. 이는 설정화면에서 끄거나 킬 수 있다.

* 1. **Target user classes 목표 사용자 계층**

Kiwinest는 가상머신 등의 샌드박스 기법을 사용하지 않고, 소규모로 운영되는 회사 또는 SOHO에서 비용을 들이지 않고 문서와 소스코드를 암호화하는 데에 도움을 준다. 또한, 사내 클라우드나 가상머신을 사용하는 비중이 적은 위 두 형태의 기업은 높은 근무 유연성을 가지고 있는데, 이는 플랫폼 간 기능 차이가 없고 단순히 설치만으로 사용할 수 있는 Kiwinest가 적합하다. 이는 개인적 사유로 문서/코드를 암호화하고자 하는 개인이나 1인 사업가에도 동일하게 적용된다.

* 1. **Constraints 제약사항**

해당 프로그램은 GPL(General Public License)로 배포되기 때문에 해당 프로그램을 포크(fork)할 시 GPL을 명시해야 한다. 또한, GPL과 호환되지 않은 라이선스는 포크된 프로그램에서 사용할 수 없다.

* 1. **Assumptions and dependencies 가정과 종속관계**

Kiwinest는 본 문서 1차 개정판 수정일인 2024/03/21 기준 Windows11 23H2, macOS Sonoma 14.4(23E214)로 개발되었고, 사용자가 해당 버전들과 호환되는 OS를 사용한다는 가정에 종속된다. 암호화/복호화에 걸리는 시간은 사용자의 하드웨어 사양에 종속되며, 사용자의 부주의에 의한 기기 결함에는 관련이 없다.

* 1. **User Documentation 사용자 문서**



**3. Specific requirements 세부적 요구사항**

* 1. **Interface requirements 인터페이스 요구사항**
     1. **사용자 인터페이스**

사용자가 프로그램을 실행하고 암호화/복호화 작업을 수행할 수 있는 사용자 인터페이스가 있어야 한다. 또한 사용자 인터페이스는 직관적이고 사용하기 쉬워야 하며 암호화 및 복호화에 대한 명확한 지침을 제공해야 한다.

* + 1. **외부 시스템 인터페이스**

프로그램이 다른 시스템과 데이터를 주고받을 수 있어야 한다(예: 파일 시스템). 그리고 외부 시스템과의 통신에 사용되는 프로토콜과 데이터 형식에 대한 명세가 필요하다.

* 1. **Functional requirements 기능적 요구사항**
     1. **키 생성/관리**

프로그램은 RSA 알고리즘을 사용하여 공개 키와 개인 키를 생성한다. 난수 생성은 C 언어의 표준 라이브러리를 사용한다. 또한 프로그램은 생성된 키를 안전하게 관리하고 사이클링하며, 사용자에게 키를 백업하거나 복구할 수 있는 기능을 제공받아야 한다.

* + 1. **문서/코드의 암호화와 복호화**

사용자는 선택한 파일을 암호화 및 복호화 할 수 있고, 암호화된 파일은 복호화 시 원본과 동일한 형태를 유지해야 한다.

* + 1. **오류 처리**

프로그램은 예상치 못한 상황이 발생할 경우 적절한 오류 메시지를 표시해야 한다. 오류 처리 메커니즘이 구현되어야 하고, 사용자가 오류를 이해하고 문제를 해결할 수 있도록 명시적으로 인터페이스에 표시해야 한다.

* 1. **Maintainability & Portability 유지보수와 호환성**
     1. **유지보수 요구사항**

본 프로그램은 버그 수정에 대한 프로세스와 책임을 명확하게 정의해야 한다. 또한 커뮤니티에서는 새로운 기능이나 기능 개선 사항에 대한 우선순위를 설정하는 방법을 명시해야 하며, 보안 업데이트의 우선순위 및 프로세스를 문서화하고 업데이트를 신속하게 제공해야 한다.

* + 1. **호환성 요구사항**

본 프로그램은 프로그램이 명시한 운영 체제에서 동작할 수 있도록 운영 체제 호환성을 보장해야 한다. 또한 커뮤니티에서는 호환되는 데이터 형식 및 파일 시스템 버전에 대한 요구 사항을 문서화하고, 최대한 다양한 데이터 형식 또는 파일 시스템과의 호환을 보장해야 한다.