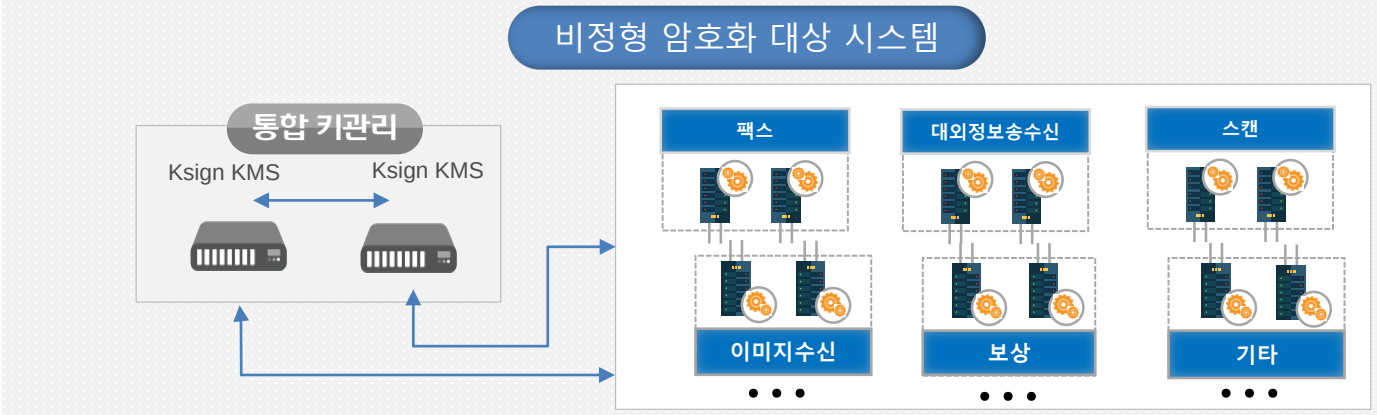
UDE API 사전 가이드

Ksign SecureDB UDE

✓ **Ksign SecureDB** UDE **방식은 Library 형태로 응용 프로그램에서 암/복호화 함수를 호출하여 수행**

구성도



✓ **제공** UDE

제공 UDE

▶JAVA

암 복호화 함수가 내장된 .JAR LIBRARY 제공

▶C#

암 복호화 함수가 내장된 .DLL , .LIB LIBRARY 제공

▶VC++

암 복호화 함수가 내장된 .DLL , .LIB LIBRARY 제공

✓ 암호화 / 복호화 함수 작동 구조

작동구조

▶암호화

● EncryptFileFD 함수 사용

암호화 대상 파일

/Storage/IMG/A.jpeg

원본 파일 이동(백업) -- 해당 부분은 안정화 시 까지만 사용할 예정

/Storage/IMG/enc_bak/A.jpeg

원본 파일 암호화 하여 원본 이름으로 저장

/Storage/IMG/A.jpeg

백업되는 원본파일 경로 /enc_bak 은 config 에서 변경 가능, 하위경로 로 백업 하는 옵션을 사용 하지 않으면 _encbak 접미어 가 붙은 파일로 같은 경로에 복호화 된다
ex)/Storage/IMG/A.jpeg_encbak

▶복호화

● DecryptFileFD 함수 사용

복호화 대상 파일

/Storage/IMG/A.jpeg

암호화 파일 복호화 하여 원본 이름으로 하위 디렉토리에 저장

/Storage/IMG/dec/A.jpeg

복호화 되는 하위 경로 /dec 은 config에서 변경 가능, 하위경로 로 복호화 하는 옵션을 사용 하지 않으면 _dec 접미어 가 붙은 파일로 같은 경로에 복호화 된다
ex)/Storage/IMG/A.jpeg_dec

e-Security Leader

사전환경설정

사전환경설정

✓ 함수 사용 전 사전 환경 설정

사전설정

▶환경변수

● NT계열

암복화 화 함수를 사용하는 WAS나 어플리케이션을 구동하는 계정의 환경변수에
KSIGN UDE Config 파일 경로 설정 필요
환경변수명 : SDBFILEAPICONF_PATH
SDBFILEAPICONF_PATH=C:₩KSIGN₩Config

● UNIX계열

암복화 화 함수를 사용하는 WAS나 어플리케이션을 구동하는 계정의
Profile에 환경변수 KSIGN UDE Config 파일 경로 설정 필요
환경변수명 : SDBFILEAPICONF_PATH
SDBFILEAPICONF_PATH=/home/jeus/ksign/Config

e-Security Leader

API Config

UDE Config

사전설정

헤더명	설명
Use.Domain	키서버에 등록된 도메인 이름으로 GetInstance 함수 호출 여부 [True : 도메인 이름 사용 False : 도메인 IP, Port 사용]
Domain.Name	도메인 이름 입력 (use.Domain=true 시 사용)
Domain.IP	도메인 IP 주소 입력(use.Domain=false 시 사용)
Domain.Port	도메인 Port 번호 입력(use.Domain=false 시 사용)
Use.ServerHA	보안 서버 이중화 사용 여부[True : 보안 서버 이중화 구축, False : 보안 서버 이중화 구축 안 함.]
Server.PrimaryIP	보안 서버 IP 주소 입력
Server.PrimaryPort	보안 서버 Port 번호 입력
Server.SecondIP	보안 서버 IP 주소 입력(use.ServerHA=true 시 사용)
Server.SecondPort	보안 서버 PORT 번호 입력(use.ServerHA=true 시 사용)
Use.ObjectFile	API 정책 파일 사용 여부
ObjectFilePath	API 정책 파일 위치 설정 **(java API 윈도우 일 경우 경로 표시 : C:\\APIPOLICY\\sdbapi.xdb)**
EncDuplicationType	API 이중화 체크 타입 [0: 이중화 체크 안 함, 1:이중화 검출 시 Data 리턴, 2: 이중화 검출 시 exception 리턴
DecDuplicationType	API 이중화 체크 타입 [0: 이중화 체크 안 함, 1:이중화 검출 시 Data 리턴, 2: 이중화 검출 시 exception 리턴
UseFixedEncBakDirectory	**EncryptFileFD 함수 사용시 원본파일 백업 하위 디렉토리 사용 여부 (true : 사용 , false : 사용하지않음 (접미어_encbak))**
FixedEncBakDirectory	**EncryptFileFD 함수 사용시 원본파일 백업 하위 디렉토리 이름**
UseFixedDecDirectory	**DecryptFileFD 함수 사용시 원본파일 백업 하위 디렉토리 사용 여부 (true : 사용 , false : 사용하지않음 (접미어_dec))**
FixedDecDirectory	**DecryptFileFD 함수 사용시 원본파일 백업 하위 디렉토리 이름**

※ Config 는 KSIGN에서 개발/운영 모듈 설치 시 설정하여 제공할 예정
아래 4개의 항목만 각 업무 시스템에 맞게 변경하여 사용

e-Security Leader

API 함수

SDBFileAPI.encryptFileFD

- **함수 원형**

  public static boolean encryptFileFD(String policyName, String sourcePath , boolean isDelSourceFile)
  throws SDBDupException, SDBFileAPIException

- **Parameters**
  - policyName 정책 정보 (Schema.Table.Column)
  - sourcePath 암호화 대상 파일의 Full Path (평문 파일)
  - isDelSourceFile 백업파일 삭제 여부(true: 삭제, false : 삭제 안 함)

- **Returns**
  - 성공 : 암호화 성공 여부(true)
  - 실패 : - SDBFileAPIException (파일 read/write 실패 및 암호화 실패)
    - SDBDupException (sourcePath가 암호화 완료된 파일이고, 이중 암호화 옵션 (Enc.DuplicationType)이 '2' 로 설정되었을 경우 SDBDupException 발생)
    - false (sourcePath가 암호화 완료된 파일이고, 이중 암호화 옵션 (Enc.DuplicationType)이 '1'로 설정되었을 경우 리턴)

- 기능 설명

파일을 암호화 한다. (원본 파일은 사용자 지정 경로로 백업되며, 암호화 파일은 원본 파일명과 동일하게 생성된다)
sdbfileapi.conf (profile에는 SDBFILEAPICONF_PATH 환경변수 추가) 파일을 확인하고 옵션을 자동적으로 체크한다.
이미 암호화가 완료된 파일이 입력되면 Enc.DuplicationType 옵션에 따라 SDBDupException 또는 false를 리턴한다.

SDBFileAPI.decryptFileFD

- **함수 원형**

  public static boolean decryptFileFD(String policyName , String sourcePath)

  throws SDBDupException, SDBFileAPIException

- **Parameters**
  - policyName 정책 정보 (Schema.Table.Column)
  - sourcePath 복호화 대상 파일의 Full Path

- **Returns**
  - 성공 : 복호화 성공 여부(true)
  - 실패 : - SDBFileAPIException (파일 read/write 실패 및 복호화 실패)
    - SDBDupException (sourcePath가 평문 파일이고, 이중 암호화 옵션 (Dec.DuplicationType)이 '2' 로 설정되었을 경우 SDBDupException 발생)
    - false (sourcePath가 평문 파일이고, 이중 암호화 옵션 (Dec.DuplicationType)이 '1'로 설정되었을 경우 리턴)

- 기능 설명

파일을 복호화 한다. (원본 파일은 변경하지 않으며, 복호화가 완료된 파일은 사용자 지정 경로에 생성된다)
sdbfileapi.conf (profile에는 SDBFILEAPICONF_PATH 환경변수 추가) 파일을 확인하고 옵션을 자동적으로 체크한다.
이중 복호화 체크시 평문 파일이 들어올 경우, 평문 파일을 Copy하여 복호화 파일을 생성하고, 그 후에, Dec.DuplicationType 옵션에 따라 SDBDupException 또는 false를 리턴한다.

SDBFileAPI.encryptFileT

- **함수 원형**

  public static boolean encryptFileT(String policyName, String sourcePath , String targetPath, boolean isDelSourceFile)
  throws SDBDupException, SDBFileAPIException

- **Parameters**
  - policyName 정책 정보 (Schema.Table.Column)
  - sourcePath 암호화 대상 파일의 Full Path (평문 파일)
  - targetPath 암호화 파일이 생성될 Full Path (암호화 생성 파일)
  - isDelSourceFile 백업파일 삭제 여부(true: 삭제, false : 삭제 안함)

- **Returns**
  - 성공 : 암호화 성공 여부(true)
  - 실패 : - SDBFileAPIException (파일 read/write 실패 및 암호화 실패)
    - SDBDupException (sourcePath가 암호화 완료된 파일이고, 이중 암호화 옵션 (Enc.DuplicationType)이 '2' 로 설정되었을 경우 SDBDupException 발생)
    - false (sourcePath가 암호화 완료된 파일이고, 이중 암호화 옵션 (Enc.DuplicationType)이 '1'로 설정되었을 경우 리턴)

- 기능 설명

파일을 암호화 한다. (sourcePath의 파일을 읽어서 targetPath에 암호화 파일을 생성한다)
sdbfileapi.conf (profile에는 SDBFILEAPICONF_PATH 환경변수 추가) 파일을 확인하고 옵션을 자동적으로 체크한다.
이미 암호화가 완료된 파일이 입력되면 Enc.DuplicationType 옵션에 따라 SDBDupException 또는 false를 리턴한다.

SDBFileAPI.decryptFileT

- **함수 원형**

  public static boolean decryptFileT(String policyName , String sourcePath, String targetPath)

  throws SDBDupException, SDBFileAPIException

- **Parameters**
  - policyName 정책 정보 (Schema.Table.Column)
  - sourcePath 복호화 대상 파일의 Full Path
  - targetPath 복호화 파일 저장 위치 Full Path

- **Returns**
  - 성공 : 복호화 성공 여부(true)
  - 실패 : - SDBFileAPIException (파일 read/write 실패 및 복호화 실패)
    - SDBDupException (sourcePath가 평문 파일이고, 이중 암호화 옵션 (Dec.DuplicationType)이 '2' 로 설정되었을 경우 SDBDupException 발생)
    - false (sourcePath가 평문 파일이고, 이중 암호화 옵션 (Dec.DuplicationType)이 '1'로 설정되었을 경우 리턴)

- 기능 설명

파일을 복호화 한다. (sourcePath의 파일을 읽어서 targetPath에 복호화 파일을 생성한다)
sdbfileapi.conf (profile에는 SDBFILEAPICONF_PATH 환경변수 추가) 파일을 확인하고 옵션을 자동적으로 체크한다.
이중 복호화 체크시 평문 파일이 들어올 경우, 평문 파일을 Copy하여 복호화 파일을 생성하고, 그 후에, Dec.DuplicationType 옵션에 따라 SDBDupException 또는 false를 리턴한다.

e-Security Leader

API Sample

JAVA API Sample

```
import com.ksign.securedb.fileapi.SDBDupException;
import com.ksign.securedb.fileapi.SDBFileAPI;
import com.ksign.securedb.fileapi.SDBFileAPIException;

public class Test {
	public static void main(String[] args) {
		String policyName = "DBSEC.ALG.SEED";
		String sourcePath = "/storage/img/A.jpg";
		String targetPath = "/storage/img/enc/A.jpg";
		boolean isDelSourceFile = false;

		try {
			boolean result = SDBFileAPI.encryptFileFD(policyName, sourcePath, isDelSourceFile);
			result = SDBFileAPI.decryptFileFD(policyName, sourcePath);
			result = SDBFileAPI.encryptFileT(policyName, sourcePath, targetPath, isDelSourceFile);
			result = SDBFileAPI.decryptFileT(policyName, sourcePath, targetPath);
		} catch (SDBDupException dupEx) {
			// ...
		} catch (SDBFileAPIException faEx) {
			// ...
		} catch (Exception e) {
			// ...
		}
	}
}
```