Il progetto consiste nello sviluppo di un KeyRing per conservare chiavi crittografiche (simulazione della classe Java KeyStore che consente di memorizzare chiavi e certificati). Il KeyRing deve essere disponibile in due versioni: una privata, l'altra pubblica.

- Nel KeyRing privato devono essere conservate le chiavi pubbliche/private e quelle segrete del gruppo. Il KeyRing è memorizzato in un file cifrato con AES/CFB/ PKCS5Padding e la chiave di cifratura è generata tramite la tecnica Password-Based Key-Derivation.

- Nel KeyRing pubblico, memorizzato in un file, devono essere conservate le chiavi pubbliche (di cifratura e di firma) degli altri gruppi del corso di Sicurezza Informatica.

Il *record* generico del KeyRing prevede almeno due campi: alias e chiave. Se necessario aggiungere altri campi (ad esempio un campo associato a una chiave segreta potrebbe essere la modalità di cifratura).

Ogni gruppo deve avere
- un nome
- una coppia di chiavi pubblica/privata per cifrare (chiave RSA di 1024 bit)
- una coppia di chiavi pubblica/privata per firmare (chiave DSA di 1024 bit)
- una chiave AES di 128 bit
- una chiave DESede di 168 bit

L'alias delle chiavi pubbliche/private di NomeGruppo sono:
- per cifrare: NomeGruppo_EPK
- per decifrare: NomeGruppo_ESK
- per firmare NomeGruppo_SSK
- per verificare: NomeGruppo_SPK

Le chiavi possono essere memorizzare nel KeyRing come array di byte. Una chiave può essere convertita in array di byte tramite il metodo getEncoded. Ogni chiave è codificata in un array di byte secondo un formato standard.

Formato chiave	Algoritmo RSA	Algoritmo DSA
Privata	PKCS#8	PKCS#8
Pubblica	X.509	X.509

Le chiavi, dopo essere state recuperate del KeyRing sotto forma di array di byte, devono essere convertite nella Key appropriata (e.g., PrivateKey, PublicKey) tramite KeyFactory.

Il KeyRing deve prevedere i metodi load/store per leggere/scrivere il KeyRing da/in un file e i metodi getKey/setKey per leggere/aggiungere dal/al KeyRing una chiave associata ad un alias.

Testare il KeyRing sviluppato.
Ogni gruppo deve memorizzare le chiavi pubbliche di tutti gli altri gruppi. Ogni gruppo potrebbe esportare le proprie chiavi pubbliche in due file NomeGruppo_EPK.txt e NomeGruppo_SPK.txt e distribuirli agli altri gruppi.

Ogni gruppo deve inviare ad almeno cinque altri gruppi un file contenete il seguente testo

```
**********************************************
* Laurea Magistrale in Ingegneria Informatica
* Corso di Sicurezza Informatica
* Messaggio del xx/12/2016
* Dal gruppo:
* Al gruppo:
* Nonce:
**********************************************
```

Il Nonce è un valore casuale di 20 byte diverso per ogni file. Il Nonce è generato con SecureRandom, e codificato in Base64.

Il file deve essere e firmato con DSA (modalità SHA1withDSA), cifrato con una chiave AES di 128 bit (modalità AES/CBC/PKCS5Padding) la chiave AES deve essere cifrata con RSA (modalità RSA/ECB/OAEPWithSHA-256AndMGF1Padding).

Ogni gruppo deve essere in grado di decifrare tutti i file ricevuti dagli altri gruppi e verificare le rispettive firme.

Ogni gruppo deve essere in grado di cifrare e decifrare un file con le proprie chiavi AES (AES/CBC/PKCS5Padding) e DESede (DESede/CBC/PKCS5Padding).

Un Tutorial sulla generazione e verifica di firme è disponibile su
http://docs.oracle.com/javase/tutorial/security/apisign/index.html