Utilizzo Librerie JCA/JCE

Sviluppare una classe Java **Incapsula** che, utilizzando in maniera opportuna le librerie JCA/JCE, permette a due utenti di scambiare messaggi in modo tale che i seguenti requisiti siano soddisfatti (non necessariamente tutti per ogni messaggio).

- Confidenzialità
- Integrità
- Autenticazione
- Non ripudio

Non è necessario realizzare un'applicazione client/server. Il messaggio da inviare è contenuto in un file (e.g., documento.pdf) e il messaggio che dovrà essere inviato sarà contenuto in un altro file (e.g., documento.pdf.ts).

La confidenzialità del messaggio dovrà essere sempre garantita tramite l'uso di un cifrario ibrido. Il messaggio sarà cifrato con un cifrario simmetrico (la chiave simmetrica è selezionata dal mittente) e la chiave simmetrica è cifrata, a sua volta, con un cifrario asimmetrico (con la chiave pubblica del destinatario).

Il messaggio può anche essere firmato.

È necessario definire un formato del file (insieme di campi per rappresentare la modalità di codifica utilizzata).

Si tenga presente che il destinatario e il mittente non sono fissati a priori (il formato del file deve prevedere un campo mittente ed un campo destinatario).

Supporre che le chiavi pubbliche/private di cifratura e di firma di un utente siano autentiche (non usiamo i certificati).

La classe **Incapsula** deve supportare le seguenti primitive crittografiche, modi operativi e padding. Fare attenzione alla lunghezza delle chiavi e del digest. Esse dipendono dal tipo di primitiva selezionato.

Cifrari simmetrici

- AES (128 bit), DES, DESede

Modi operativi

- ECB, CBC, CFB

Utilizzati con i cifrari a blocchi, gli ultimi due richiedono l'IV (*Initialization Vector*) la cui dimensione dipende dal tipo di cifrario.

Padding

- PKCS5Padding

Cifrario asimmetrico

- RSA con dimensione chiavi: 1024 oppure 2048
- Padding: PKCS1Padding oppure OAEPPadding

Schema di firma

- DSA con dimensione chiavi: 1024 oppure 2048
- Tipo di firma: SHA1withDSA, SHA224withDSA, SHA256withDSA

Le chiavi private di cifratura/firma devono essere memorizzare in un file cifrato. 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

Attenzione: la firma DSA è descritta con la notazione ASN.1 ed è DER encoded (codifica con una sequenza di byte). La sequenza di byte che rappresenta la firma DSA a 1024 bit (2048) può essere lunga da 46 a 48 (da 60 a 62) byte. Il formato DER della firma DSA è il seguente:

- Il primo byte contiene 0x30 (significa che si sta codificando una sequenza).
- Il secondo byte contiene la lunghezza rimanente della sequenza in byte 44/45/46 oppure 60/61/62.
- I rimanenti 44/45/46 (60/61/62) byte contengono la codifica della firma vera e propria.

È necessario leggere tutti i 46/47/48 (60/61/62) byte della sequenza come array di byte e poi convertirli con JCA/JCE in una firma DSA.