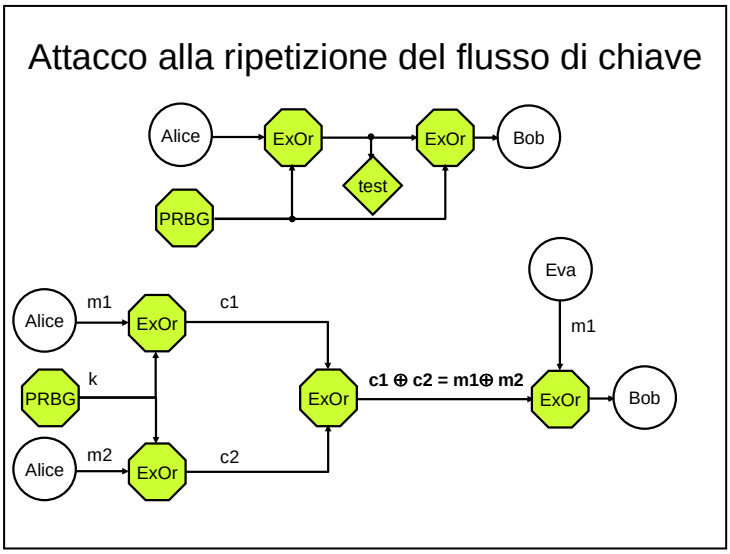
3. Esercitazioni consigliate

3.1 Il Cifrario simmetrico a flusso

Obiettivo formativo – Prendere confidenza con il blocco ExOr. Capire la struttura ed il comportamento di un Cifrario a flusso. Verificare che la ripetizione del flusso di chiave costituisce una pericolosa vulnerabilità.
Riferimenti: Capitoli 3 e 4

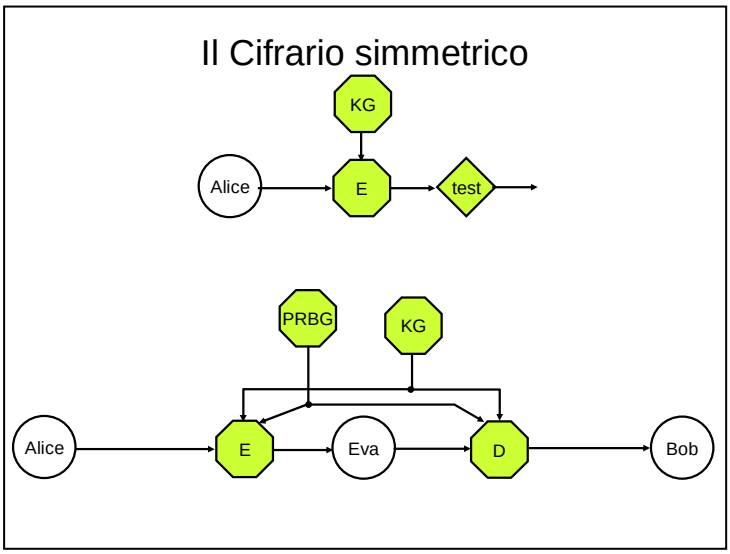


Esperimenti:

1. Verificare la decifrabilità e la casualità di un testo cifrato ottenuto sommando modulo due un testo in chiaro di 20.000 bit generato da Alice ed una chiave casuale di 20.000 bit generata da un PRBG.
2. Si vuole simulare una ripetizione del flusso di chiave. A tal fine si deve creare una seconda istanza del Cifrario a flusso, impiegando però lo stesso flusso di chiave della prima istanza. Digitare nelle due istanze di Alice messaggi corti, di uguale lunghezza e lievemente diversi. Generare un flusso di chiave di uguale lunghezza ed eseguire le due cifrature.
3. Verificare che chi è riuscito ad intercettare i due testi cifrati può sommarli modulo due e venire così a sapere, con poco sforzo, quali sono i bit per cui differiscono i corrispondenti testi in chiaro.
4. Verificare che se Eva ha già messo in chiaro uno dei due messaggi, può digitarlo (impiegando la codifica usata da Alice) e, con un'ulteriore somma modulo due, mettere in chiaro anche l'altro.

3.2 Il Cifrario simmetrico a blocchi

Obiettivo formativo – Prendere confidenza con i blocchi KeyGenerator e Cipher. Provare le diverse modalità d'impiego di un Cifrario a blocchi e verificarne la robustezza agl attacchi attivi.
Riferimenti: Capitolo 4

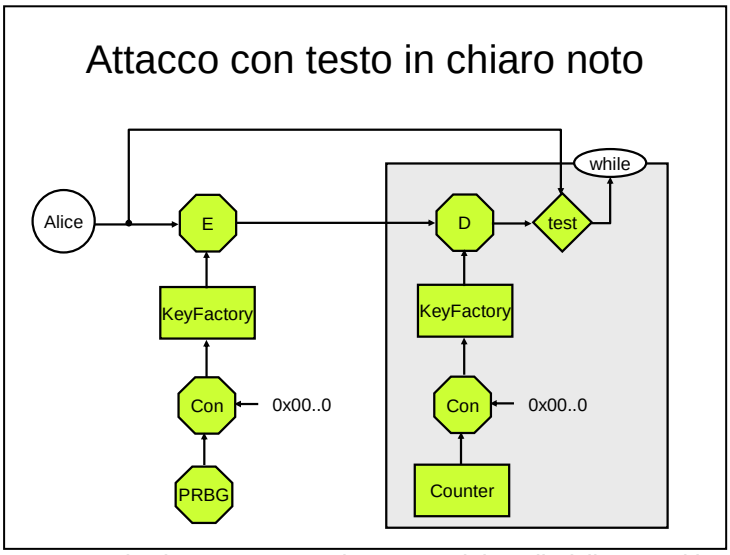


Esperimenti:

1. Istanziare il blocco Cipher e configurarlo, tramite le finestre con i valori dei parametri, per l'algoritmo DES, la direzione ENCRYPT, la modalità di cifratura ECB ed il padding PKCS7.
2. Istanziare un KeyGenerator, configurarlo per il Cifrario prescelto in 1 e digitare la lunghezza della chiave. Eseguire e poi verificare che le due uscite sono identiche. Collegare un'uscita di KG all'ingresso "key" di Cipher.
3. Istanziare Alice a monte del Cipher e farle generare un messaggio lungo tramite lettura di un file.
4. Generare il testo cifrato e verificarne la casualità con i Test del NIST.
5. Provare lo schema con diversi Cifrari ed individuare delle figure di merito che consentano di valutarne l'efficienza.
6. Completare lo schema con un generatore di IV, con la decifrazione e con Bob. Provare le modalità CBC, CFB, OFB, CTR con diversi padding. Individuare il ruolo e l'utilità del vettore di inizializzazione.
7. Inserire Eva sul canale e provare le diverse modalità di cifratura in presenza di modifiche, inserimenti e cancellazione di bit del testo cifrato.

3.3 Attacco con forza bruta alla chiave di un Cifrario

Obiettivo formativo – Prendere confidenza con i blocchi KeyFactory e Counter. Verificare che un attacco con forza bruta ad una chiave poco casuale può ha successo se si conosce una coppia testo in chiaro-testo cifrato.
Riferimenti: Capitolo 4

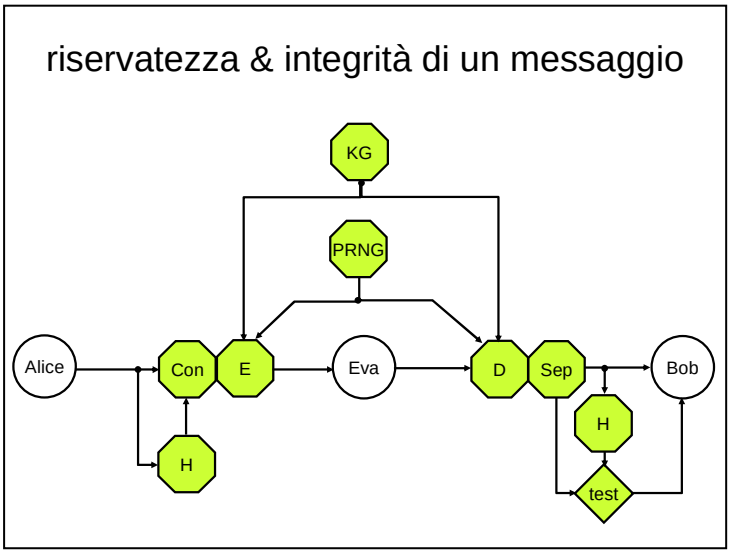


Esperimenti:

1. Istanziare un Cipher e configurarlo per cifrare con l'algoritmo AES un messaggio proveniente da Alice.
2. Consultare in Javadoc la documentazione della classe SecretKeyFactory.
3. Impiegando un PRBG, un blocco di concatenazione e la KeyFactory (per inizializzarla occorre scegliere "algorithm" e "provider"), costruire una chiave AES con un byte casuali e 15 byte a zero.
4. Cifrare il messaggio di Alice e rendere disponibili in ingresso ad una BOX-while il testo cifrato ed il testo in chiaro.
5. Predisporre nella BOX la decifrazione del testo cifrato impiegando come chiave il byte d'uscita di un contatore binario concatenato con 15 byte a zero.
6. Eseguire la BOX e prendere nota del n° di cicli eseguiti. Individuarne il valor medio ripetendo la prova con chiavi diverse (prima di ogni prova occorre mettere 0 nel campo initial state del Counter).
7. Confrontare i dati sperimentali con dati forniti dalla teoria. Classificare la complessità di questo attacco.

3.4 L'accertamento dell'integrità di un messaggio riservato

Obiettivo formativo – Costruire un sistema che consenta di rilevare automaticamente attacchi attivi svolti da un intruso sul testo cifrato.
Riferimenti: Capitolo 4



Esperimenti:

1. Modificare lo schema studiato in 3.2 prevedendo che Alice concateni al testo in chiaro la sua impronta e che Bob, dopo la decifrazione, confronti l'impronta ricevuta con l'impronta del testo in chiaro ricevuto.
2. Configurare i blocchi (Eva in condizioni di non attacco), metterli esecuzione in ordine appropriato ed esaminare cosa riceve Bob.
3. Verificare che il sistema è in grado di rilevare ogni attacco fatto da Eva sul testo cifrato (modifica, inserimento, cancellazione, ripetizione di bit).
4. Impiegare opportunamente i blocchi Counter, Con, Sep, Comp per consentire ad Alice di appendere ai suoi messaggi anche un "serial number" ed a Bob di verificare che non sono messaggi replicati da Eva. Organizzare la prova con Contatori da 1 byte, inizializzati a 0; ricordarsi di incrementare il contatore di Alice ad ogni trasmissione e quello di Bob ad ogni ricezione.
5. Dopo una trasmissione andata a buon fine, salvare il testo cifrato su file, tramite una seconda istanza di Bob, e farlo immettere da Eva sul canale. Verificare che il controllo sul serial number mette Bob in grado di scartare il messaggio.
6. Valutare lo troughput del sistema simulato, mettendo in conto il percorso di elaborazione più lungo.

REPORT N.3

3.1 Il Cifrario simmetrico a flusso

Immagine dello schema studiato	**Osservazioni:**

3.1 Il Cifrario simmetrico a blocchi

Immagine dello schema studiato	**Osservazioni:**

3.3 Attacco con forza bruta alla chiave di un Cifrario

Immagine dello schema studiato	**Osservazioni:**

3.4 L’accertamento dell’integrità di un messaggio riservato

Immagine dello schema studiato	**Osservazioni:**