COMPUTER SECURITY

Corso di Laurea Magistrale in Ingegneria Informatica
Prof. Alessandro Armando

14 gennaio 2019

Attenzione: Si risponda alle domande utilizzando lo spazio apposito.
Tempo per lo svolgimento: 2 ore.

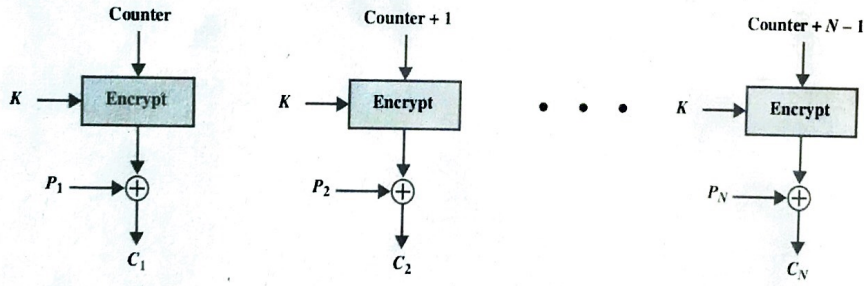
Nome e Cognome: _____

Matricola: _____

---

1. **Crittografia**

   Il seguente schema crittografico per *block encription*. **Counter** è una sequenza di bit arbitraria della stessa lunghezza b dei blocchi di dati e l'operazione di somma è da intendersi modulo 2^b.



   (a) Si disegni il corrispondente schema crittografico da usarsi in fase di decifratura.

Soluzione.

(b) Tale schema crittografico si presta a implementazioni più o meno efficienti rispetto allo schema CBC visto a lezione? Si giustifichi la risposta data.

Soluzione.

2. **Memorizzazione di Password** Un modo per supportare la verifica delle credenziali degli utenti nei sistemi operativi e' quello di memorizzare per ciascun utente le seguenti informazioni:

   - Nome dell'account, ad esempio `armando`
   - l'hash della password dell'utente.

   A tale schema tuttavia se ne preferisce uno che richiede la memorizzazione per ciascun utente le seguenti informazioni:

   - Nome dell'account, ad esempio `armando`
   - un numero n, detto **salt**
   - l'hash della concatenazione della password dell'utente con il salt n.

   Perche'? Si discuta l'importanza della lunghezza del *salt*.

   Soluzione.

3. **Crittografia a Chiave Pubblica** Alice deve inviare un file M di grosse dimensioni a Bob (ad esempio un filmato di qualche Gbyte) in modo tale che sia garantita la l'integrità ma non la confidenzialità della trasmissione. Quale tra le seguenti procedure è quella più adatta allo scopo?

Soluzione.

- ☐ Alice calcola ed invia a Bob il ciphertext ottentuto cifrando M con la chiave pubblica di Bob.
- ☐ Alice genera un modo (pseudo)casuale una chiave simmetrica K ed invia a Bob il ciphertext ottenuto cifrando M con K.
- ☐ Alice genera un modo (pseudo)casuale una chiave simmetrica K ed invia a Bob (1) il ciphertext ottenuto cifrando M con K e (2) il ciphertext ottentuto cifrando K con la chiave pubblica di Bob.
- ☐ Alice genera un modo (pseudo)casuale una chiave simmetrica K ed invia a Bob (1) il ciphertext ottenuto cifrando M con K, (2) K e il ciphertext ottentuto cifrando K con la propria chiave privata.

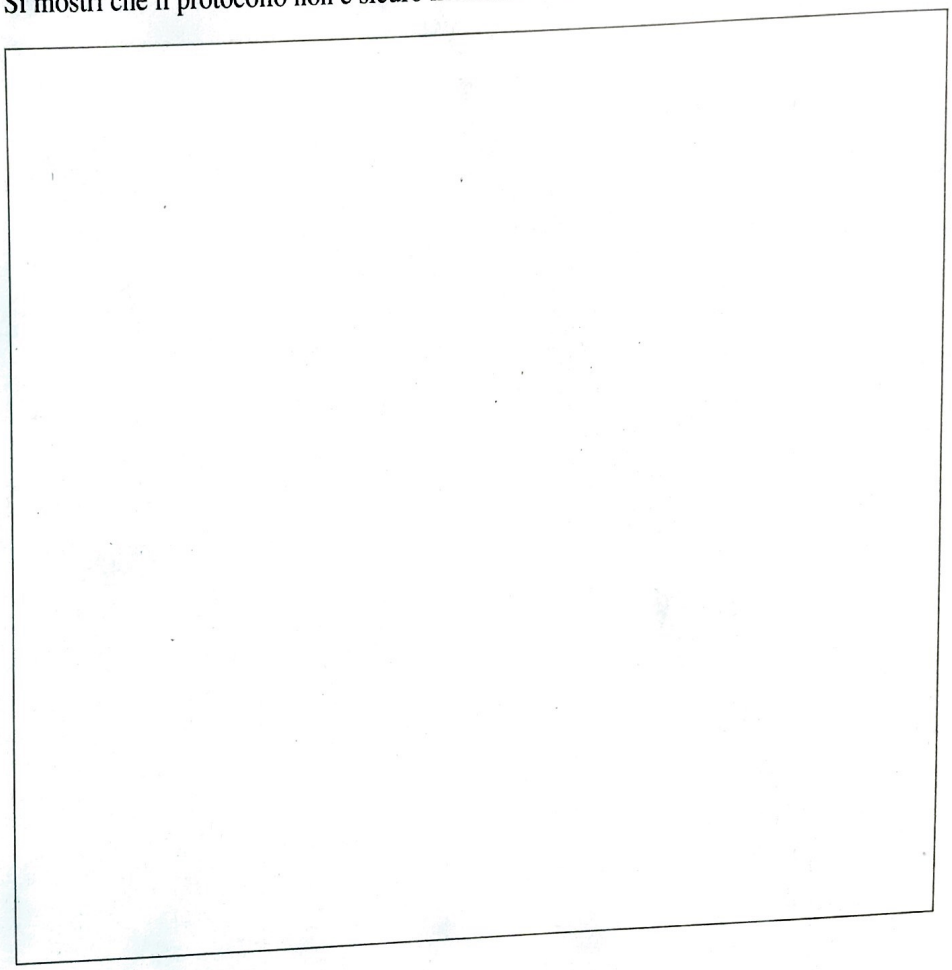
4. **Domanda #2**

Si consideri il seguente protocollo:

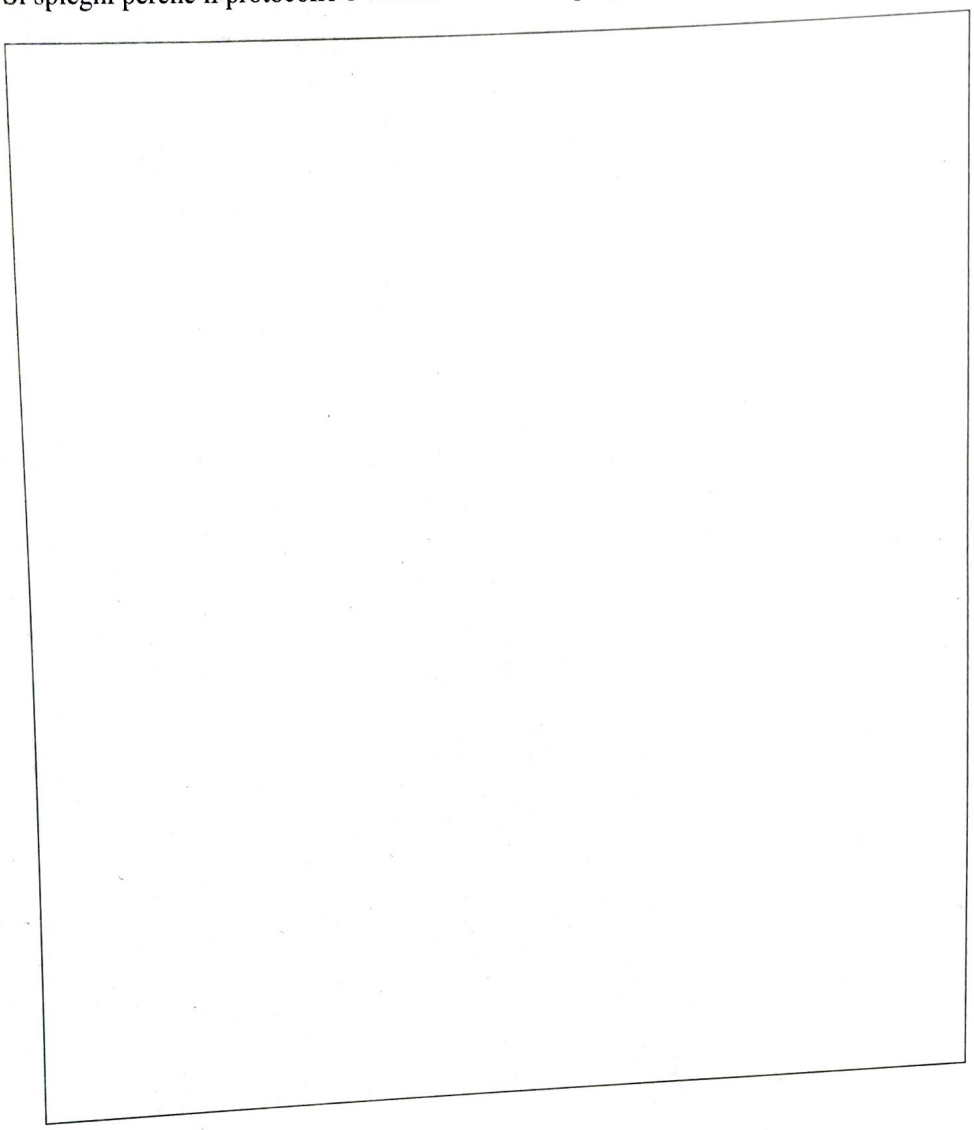
$$1.\ C \rightarrow AS : C, S, N_C$$
$$2.\ AS \rightarrow C : AS, \{AS, C, N_C, K_S\}K_{AS}^{-1}$$

in cui un client C vuole communicare con un server S avendo la garanzia di farlo effettivamente con S. Siccome C non possiede la chiave pubblica di S, C interpella un Autentication Server AS per ottenerla. Ovviamente si assume che C possegga la chiave pubblica di AS, ovvero K_{AS}.

(a) Si mostri che il protocollo non è sicuro mostrando un attacco allo stesso.



(b) Si spieghi perchè il protocollo è vulnerabile e se ne proponga una versione corretta.



5. **Secure Programming**

(a) Write a program (preferably in C) suffering from a buffer overflow.

Soluzione.

(b) Modify the program so to prevent the buffer overflow.

Soluzione.

6. **Access Control**

This is a simplified dump for the `ls -l` shell command in the current folder.

```
-r--r----- alice   admin    1
-r--r--r-- bob     bob      2
-rw-rw---- charlie charlie  3
-rw-r----- charlie admin    4
---x--x--x alice   alice    editor
---x--s--- bob     admin    editor-super
```

Unix users are **alice**, **bob** and **charlie**. **root** is the system administrator.

The `id` command for each user returns:

- `id alice`: uid=1000(alice) gid=1000(alice) groups=1000(alice),1003(admin)
- `id bob`: uid=1001(bob) gid=1001(bob) groups=1001(bob)
- `id charlie`: uid=1002(charlie) gid=1002(charlie) groups=1002(charlie),1003(admin)

There are 2 executable files:

- **editor** lets you open a file with **R**ead and **W**rite capabilities;
- **editor-super** does the same as **editor**.

Draw up an access control matrix with subjects {alice, bob, charlie} and objects {1, 2, 3, 4} that shows, for each combination of subject and object, whether the subject will be able to read (**R**), and/or write (**W**) the respective object.

NOTE: **root** should not appear in the matrix.