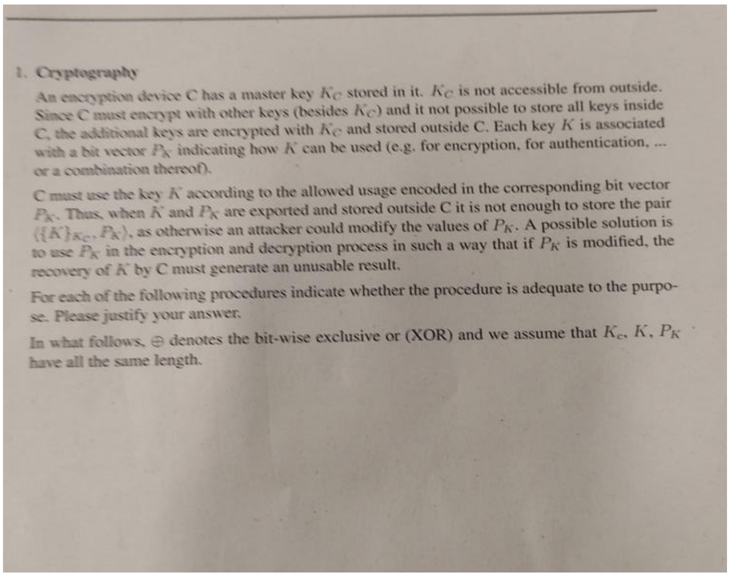


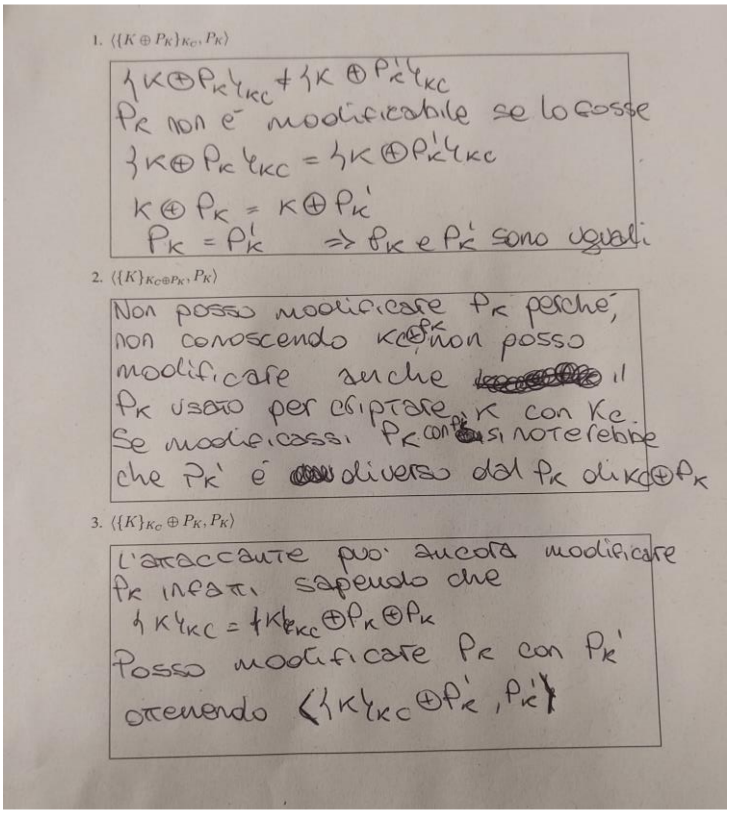
1. **Cryptography**

An encryption device C has a master key K_C stored in it. K_C is not accessible from outside. Since C must encrypt with other keys (besides K_C) and it not possible to store all keys inside C, the additional keys are encrypted with K_C and stored outside C. Each key K is associated with a bit vector P_K indicating how K can be used (e.g. for encryption, for authentication, ... or a combination thereof).

C must use the key K according to the allowed usage encoded in the corresponding bit vector P_K. Thus, when K and P_K are exported and stored outside C it is not enough to store the pair $\langle\{K\}_{K_C}, P_K\rangle$, as otherwise an attacker could modify the values of P_K. A possible solution is to use P_K in the encryption and decryption process in such a way that if P_K is modified, the recovery of K by C must generate an unusable result.

For each of the following procedures indicate whether the procedure is adequate to the purpose. Please justify your answer.

In what follows, $\oplus$ denotes the bit-wise exclusive or (XOR) and we assume that K_c, K, P_K have all the same length.



1. $\langle\{K \oplus P_K\}_{K_C}, P_K\rangle$

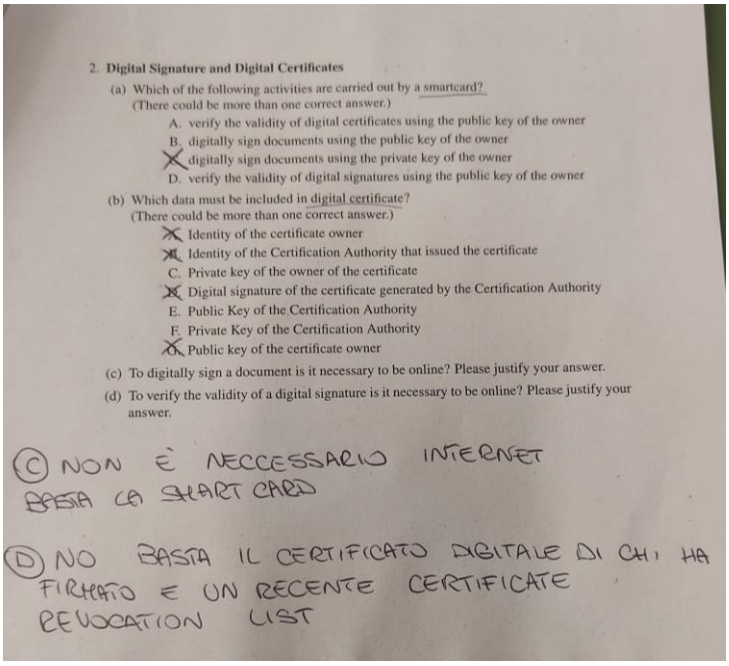
$\{K \oplus P_K\}_{KC} \neq \{K \oplus P_K'\}_{KC}$
P_K non é modificabile se lo fosse
$\{K \oplus P_K\}_{KC} = \{K \oplus P_K'\}_{KC}$
$K \oplus P_K = K \oplus P_K'$
$P_K = P_K'$ $\Rightarrow$ P_K e P_K' sono uguali

2. $\langle\{K\}_{K_C \oplus P_K}, P_K\rangle$

Non posso modificare P_K perché,
non conoscendo $K_C \oplus P_K$ non posso
modificare anche il
P_K usato per criptare K con K_C.
Se modificassi P_K con P_K' si noterebbe
che P_K' é diverso dal P_K di $K_C \oplus P_K$

3. $\langle\{K\}_{K_C} \oplus P_K, P_K\rangle$

L'attaccante può ancora modificare
P_K infatti sapendo che
$\{K\}_{KC} = \{K\}_{KC} \oplus P_K \oplus P_K$
Posso modificare P_K con P_K'
ottenendo $\langle\{K\}_{KC} \oplus P_K', P_K'\rangle$



2. **Digital Signature and Digital Certificates**

(a) Which of the following activities are carried out by a smartcard?
(There could be more than one correct answer.)

A. verify the validity of digital certificates using the public key of the owner
B. digitally sign documents using the public key of the owner
[✓ C.] digitally sign documents using the private key of the owner
D. verify the validity of digital signatures using the public key of the owner

(b) Which data must be included in digital certificate?
(There could be more than one correct answer.)

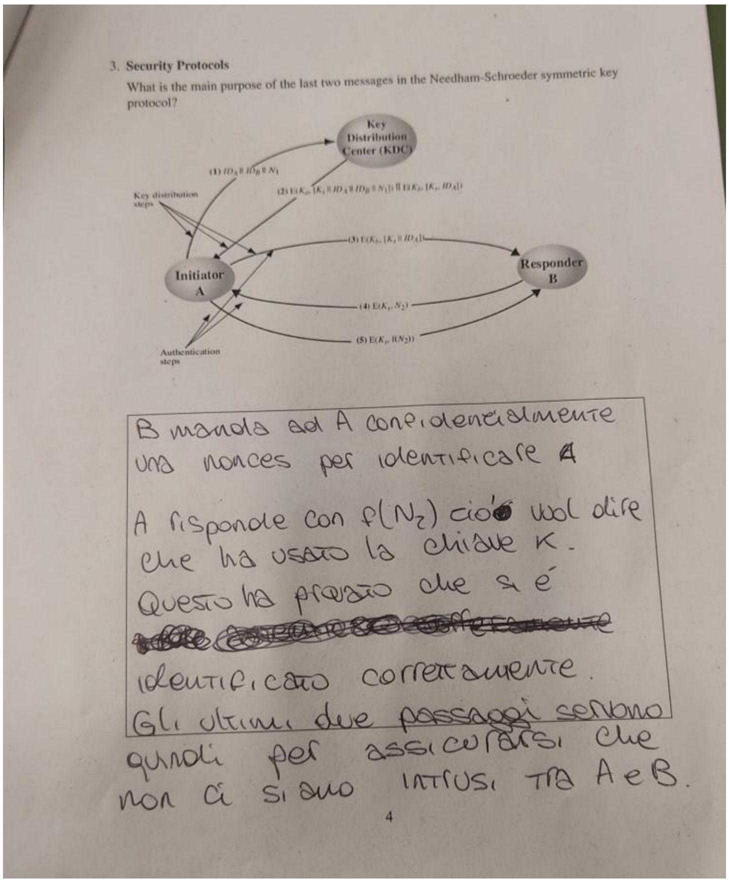
[✓ A.] Identity of the certificate owner
[✓ B.] Identity of the Certification Authority that issued the certificate
C. Private key of the owner of the certificate
[✓ D.] Digital signature of the certificate generated by the Certification Authority
E. Public Key of the Certification Authority
F. Private Key of the Certification Authority
[✓ G.] Public key of the certificate owner

(c) To digitally sign a document is it necessary to be online? Please justify your answer.

(d) To verify the validity of a digital signature is it necessary to be online? Please justify your answer.

Ⓒ NON È NECCESSARIO INTERNET
BASTA LA SMART CARD

Ⓓ NO BASTA IL CERTIFICATO DIGITALE DI CHI HA FIRMATO E UN RECENTE CERTIFICATE REVOCATION LIST



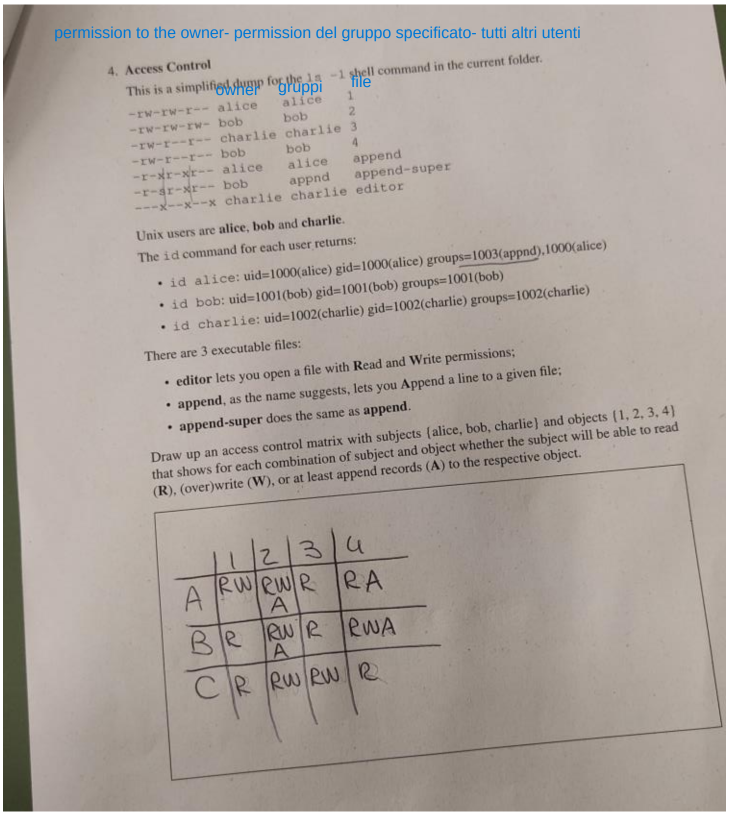
3. **Security Protocols**

What is the main purpose of the last two messages in the Needham-Schroeder symmetric key protocol?

Key Distribution Center (KDC)
(1) $ID_A \| ID_B \| N_1$
(2) $E(K_a, [K_s \| ID_A \| ID_B \| N_1]) \| E(K_b, [K_s, ID_A])$
Key distribution steps
(3) $E(K_b, [K_s \| ID_A])$
Initiator A
Responder B
(4) $E(K_s, N_2)$
(5) $E(K_s, f(N_2))$
Authentication steps

B manda ad A confidenzialmente una nonces per identificare A

A risponde con $f(N_2)$ cioè vuol dire che ha usato la chiave K.
Questo ha provato che si è [illegible] identificato correttamente.
Gli ultimi due passaggi servono quindi per assicurarsi che non ci siano intrusi tra A e B.



permission to the owner- permission del gruppo specificato- tutti altri utenti

4. **Access Control**

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

```
              owner   gruppi    file
-rw-rw-r--  alice     alice     1
-rw-rw-rw-  bob       bob       2
-rw-r--r--  charlie   charlie   3
-rw-r--r--  bob       bob       4
-r-xr-xr--  alice     alice     append
-r-sr-xr--  bob       appnd     append-super
---x--x--x  charlie   charlie   editor
```

Unix users are **alice**, **bob** and **charlie**.

The `id` command for each user returns:

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

There are 3 executable files:

- **editor** lets you open a file with **R**ead and **W**rite permissions;
- **append**, as the name suggests, lets you **A**ppend a line to a given file;
- **append-super** does the same as **append**.

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**), (over)write (**W**), or at least append records (**A**) to the respective object.

	1	2	3	4
A	RW	RW A	R	RA
B	R	RW A	R	RWA
C	R	RW	RW	R