

1) CRITTOGRAFIA SIMMETRICA

$$X \oplus X = \emptyset$$

$$X \oplus \emptyset = X$$

$M = \text{Messaggio}$ $M = M_1 M_2 \dots M_N$

$$E = \{\dots\}_E$$

$$H(M_1) = E(K, M_1)$$

$$H(M_1 \dots M_i M_{i+1}) = E(K, H(M_1 \dots M_i) \oplus M_{i+1}) \quad \text{for } i = 1, \dots, n-1$$

↑ blocchi del messaggio tutti della stessa lunghezza

- Lo schema non è sicuro, dati messaggi $A_1 A_2$ ed un arbitrario blocco $B_1 \rightarrow$ determinare B_2 tale che $H(B_1 B_2) = H(A_1 A_2)$ sapendo che H non è weak resistant.

$$H(B_1 B_2) = E(K, H(B_1) \oplus B_2) = E(K, E(K, B_1) \oplus B_2)$$

$$H(A_1 A_2) = E(K, H(A_1) \oplus A_2) = E(K, E(K, A_1) \oplus A_2)$$

$$\underline{E(K, E(K, B_1) \oplus B_2)} = \underline{E(K, E(K, A_1) \oplus A_2)}$$

Sappiamo che $E(K, \dots)$ è uguale quindi lo leviamo e confrontiamo solo gli argomenti.

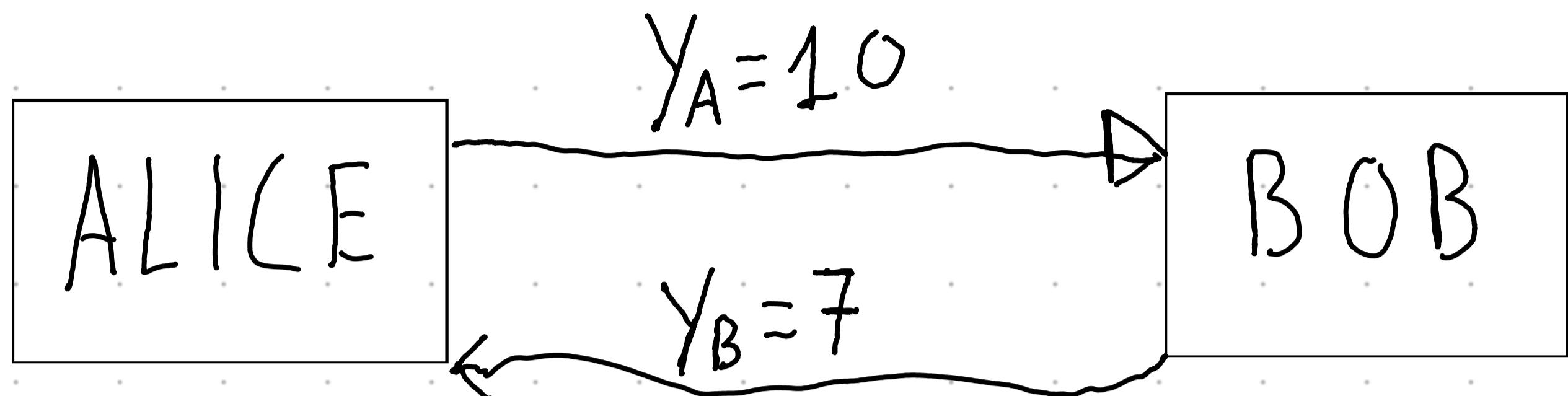
$$E(K, B_1) \oplus B_2 = E(K, A_1) \oplus A_2$$

Mettendo \oplus in entrambi i lati con $E(K, B_1)$ e semplificando

$$B_2 = E(K, B_1) \oplus E(K, A_1) \oplus A_2$$

2) CRITTOGRAFIA A CHIAVE PUBBLICA (DIFFIE HELLMAN)

$$q = 11 \quad \alpha = 2$$



$$K = y_B^{X_A} \bmod q = y_A^{X_B} \bmod q$$

$$7^{X_A} \bmod 11 = 10^{X_B} \bmod 11$$

quindi dallo schema dato ci sappiamo che
 $10^7 \bmod 11 = 10$ e $7^5 \bmod 11 = 10$, da questo
ricaviamo che $X_A = 5$, $X_B = 7$ ed infine $K_{A,B} = 10$

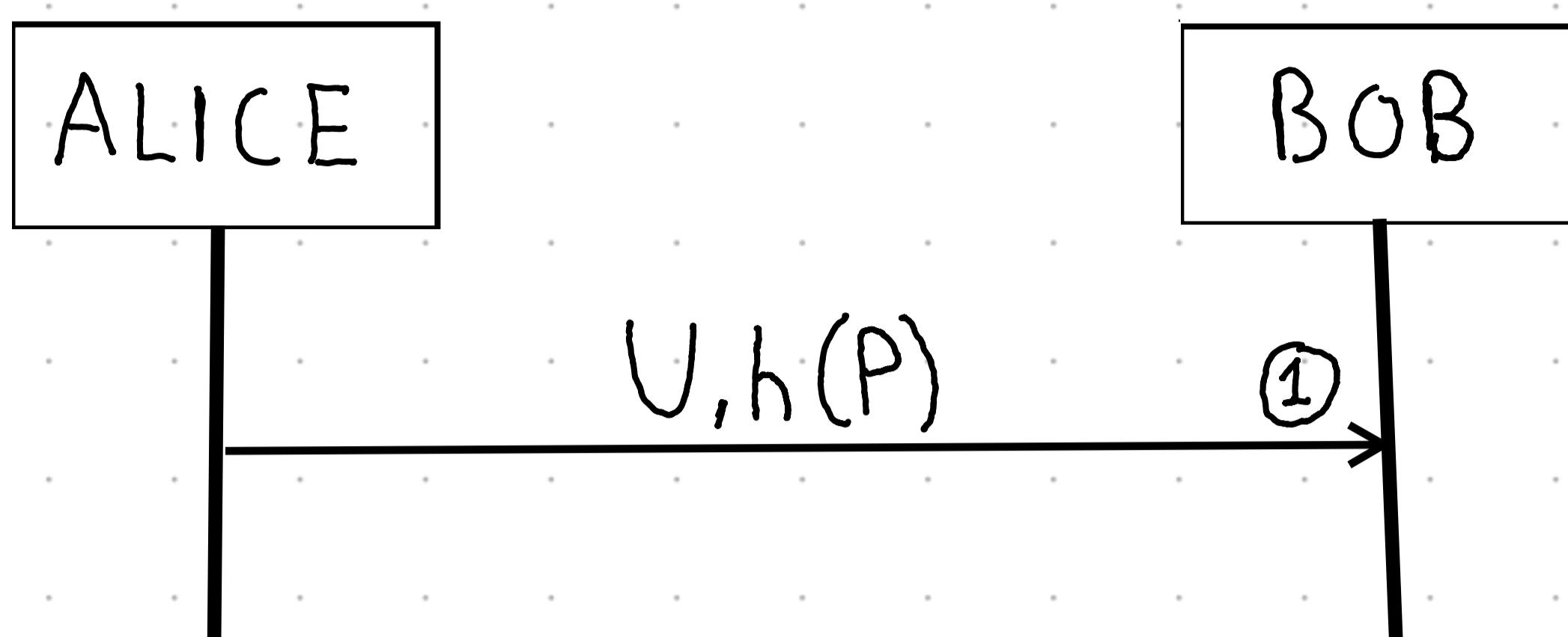
$$7^5 \bmod 11 = 10^7 \bmod 11$$

$$\downarrow \\ 10$$

$$\downarrow \\ 10$$

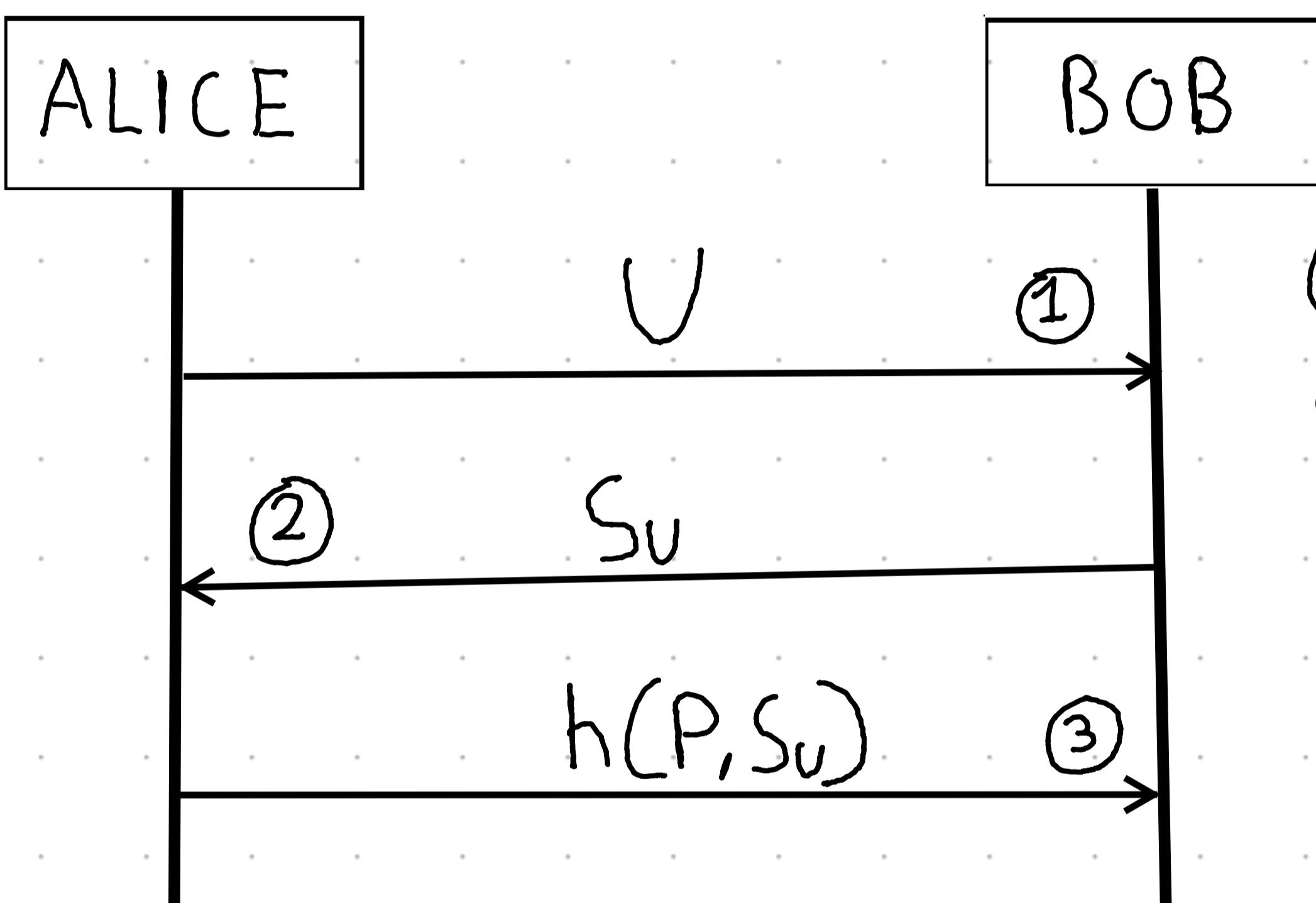
3) SECURITY PROTOCOL

(a)



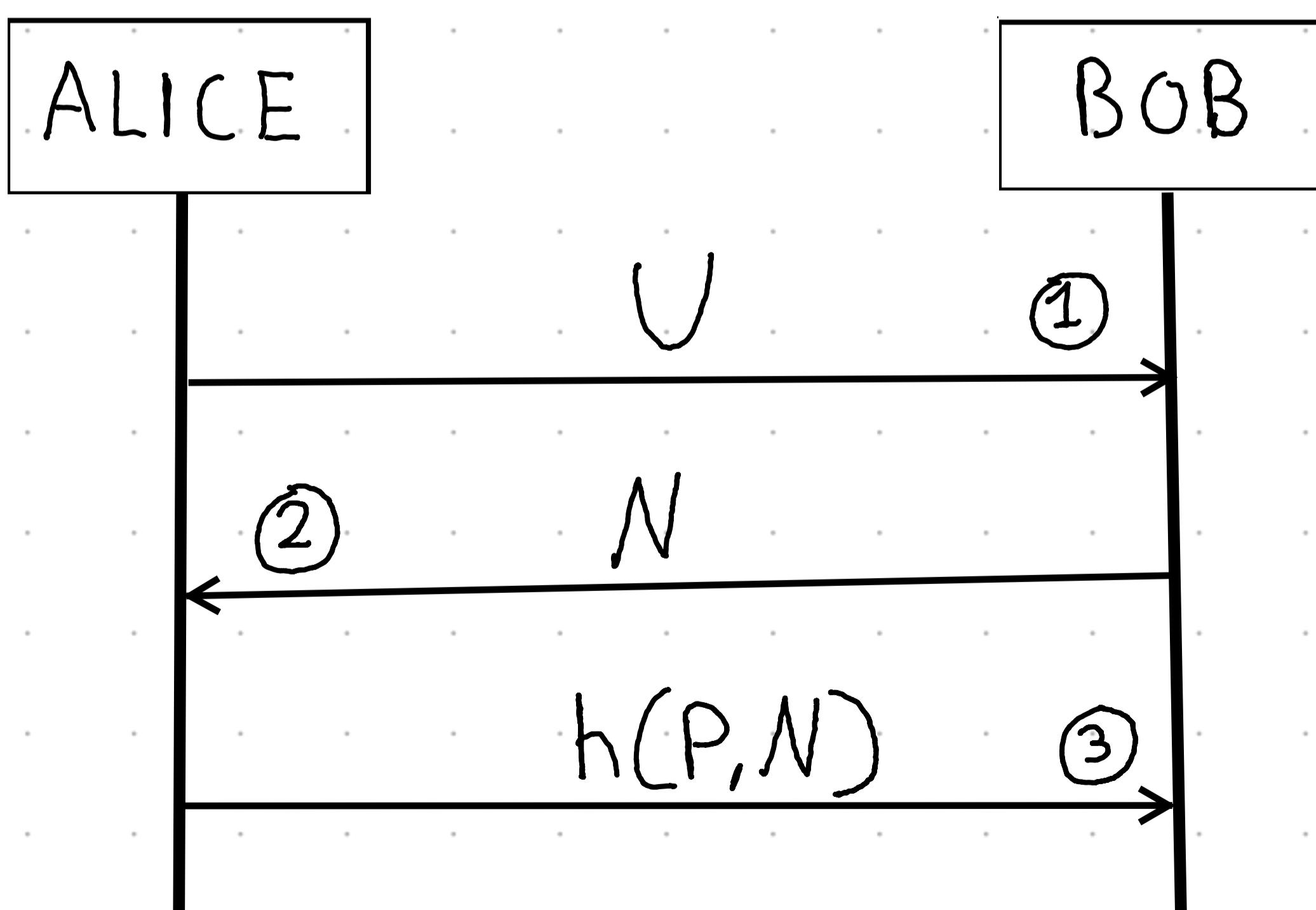
Questo protocollo è vulnerabile ad attacchi di tipo replay, un attaccante può intercettare il messaggio per poi riutilizzarlo in futuro per autenticarsi come Alice.

(b)



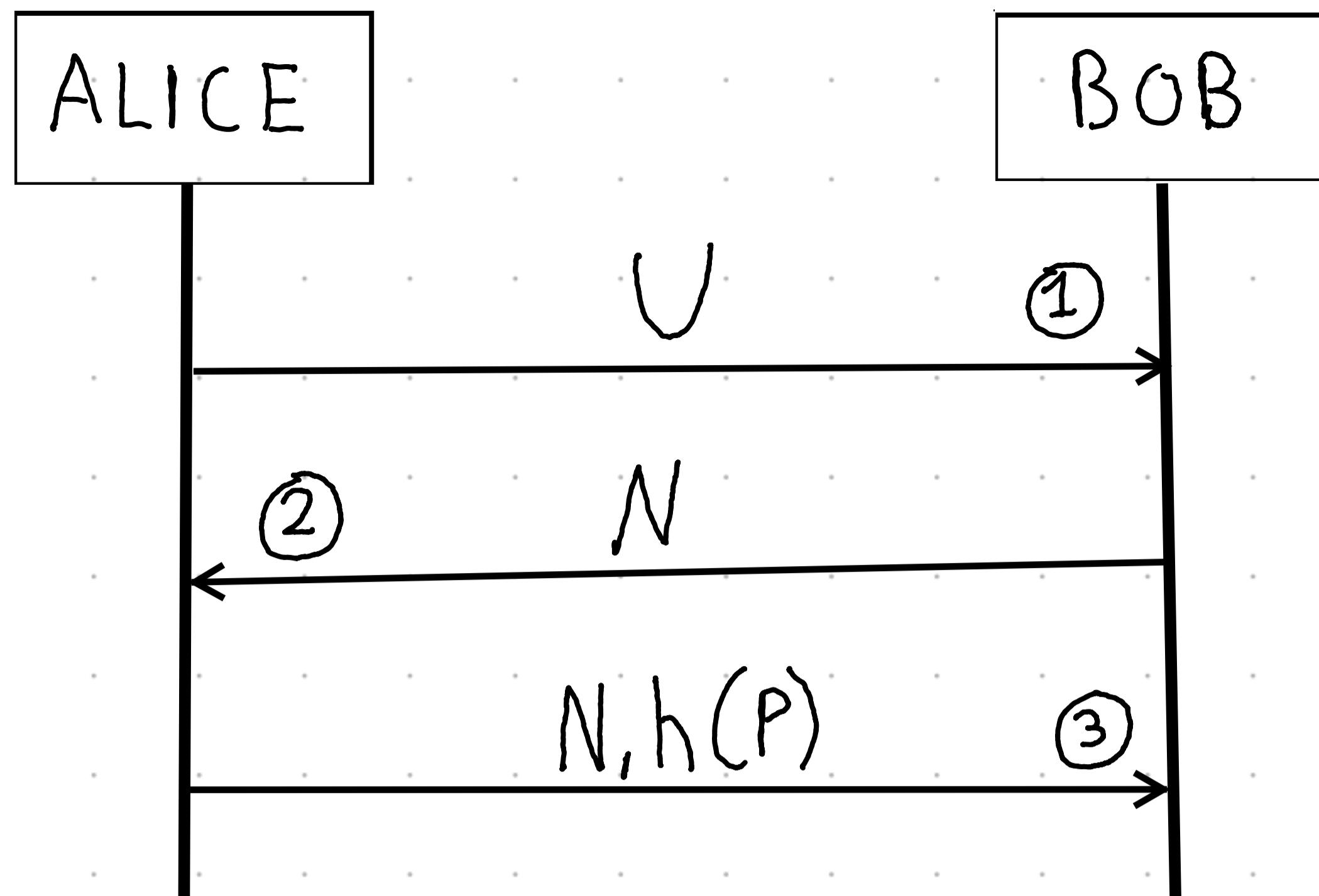
Questo protocollo è vulnerabile ad attacchi di tipo replay, un attaccante può intercettare il messaggio per poi riutilizzarlo in futuro per autenticarsi come Alice, aggiunta di un salt statico non aggiunge freschezza al messaggio.

(c)



Questo protocollo appare come sicuro, ovviamente il nonce deve cambiare ogni login.

(d)



Questo protocollo appare come sicuro, ovviamente il nonce deve cambiare ogni login.

Ovviamente un man in the middle può intercettare prima username e poi la password, così da attaccare la password via bruteforce.

4) WEB SECURITY

È vittima di un XSS attack (cross-site-scripting). Un test che si può provare ad eseguire è quello di provare a scrivere al posto test uno script in JS e vedere se viene eseguito.

Può anche essere vittima di CSRF, cioè si obbliga utente a fare una richiesta tipo "change password"