 **Domanda d'esame:** quante firme digitali bisogna verificare affinché il browser abbia certezza che il contenuto sia dell'URL?

Risposta: bisogna farne tante quante è la catena dei certificati.

Protocolli di sicurezza storici

Needham-Schroeder (1978)

Supponiamo che A e B debbano autenticarsi a vicenda.

1. $A \rightarrow B : \{N_a\}_{k_B}$
2. $B \rightarrow A : N_a$

Questo protocollo basilare autentica A con B e viceversa, con freshness.

A crea una nonce e la cifra con la chiave pubblica di B , questo gliela rimanda indietro in chiaro (si è quindi autenticato perché la decifrata con la propria chiave pubblica). Il messaggio che trasporta la nonce garantisce freshness.

Affinché possa esserci freshness il protocollo deve essere iniziato da chi la garanzia di freshness la vuole, in questo caso A .

Siccome il protocollo basilare prevedeva che B mandasse un messaggio ad A codificato con la chiave privata, non si poteva garantire freshness.

Equivalentemente per autenticare B con A possiamo fare:

1. $A \rightarrow B : N_a$
2. $B \rightarrow A : \{N_a\}_{k_B^{-1}}$

I due protocolli sono equivalenti perché entrambi garantiscono autenticazione con freshness.

Un estensione del protocollo può essere:

1. $A \rightarrow B : \{N_a\}_{k_B}$
2. $B \rightarrow B : N_a$
3. $B \rightarrow A : \{N_b\}_{k_a}$
4. $A \rightarrow B : N_b$

I primi due step autenticano B con A , gli altri due il viceversa.

Possiamo "comprimere" gli step 2 e 3 in:

$$B \rightarrow A : \{N_a, N_b\}_{k_a}$$

Otteniamo quindi:

$$\begin{aligned} 1. & A \rightarrow B : \{N_a\}_{k_B} \\ 2. & B \rightarrow A : \{N_a, N_b\}_{k_a} \\ 3. & A \rightarrow B : N_b \end{aligned}$$

C'è il problema di capire inizialmente per B chi sia il mittente al primo messaggio cifrato. Si può risalire al mittente grazie a protocolli di trasporto sottostanti, senza però avere pretese di sicurezza.

Una soluzione equipollente può essere esplicitare il nome del mittente allo step 1.

$$\begin{aligned} 1. & A \rightarrow B : \{A, N_a\}_{k_B} \\ 2. & B \rightarrow A : \{N_a, N_b\}_{k_a} \\ 3. & A \rightarrow B : N_b \end{aligned}$$

Anche in presenza di un attaccante che inoltri la nonce al posto di B questo non può sovvertire la proprietà di autenticazione.

A partire da questa prima autenticazione è poi possibile scambiarsi una chiave di sessione per comunicare. Non è una buona scelta utilizzare la nonce N_b come chiave di sessione perchè sebbene un attaccante non possa violare la proprietà di sicurezza può esfiltrare la nonce (violazione della sicurezza della nonce).

Un ulteriore step per garantire la sicurezza della nonce N_b in modo da utilizzarla come chiave di sessione può essere cifrarla.

$$\begin{aligned} 1. & A \rightarrow B : \{A, N_a\}_{k_B} \\ 2. & B \rightarrow A : \{N_a, N_b\}_{k_a} \\ 3. & A \rightarrow B : \{N_b\}_{k_b} \end{aligned}$$

In questo modo otteniamo:

- autenticazione reciproca
- segretezza della chiave di sessione
- freshness

Attraverso le trasformazioni applicate ai protocolli basilari siamo arrivati al protocollo **Needham-Schroeder (1978)**.

La variante di questo protocollo che non cifra l'ultima nonce, è detta Helsinki.

Attacco di Lowe al protocollo Needham-Schroeder

Gavin Lowe nel **1995** dimostra come il protocollo **Needham-Schroeder** sia attaccabile dal seguente scenario, senza toccare le assunzioni crittografiche. Questo porterà poi all'utilizzo di metodi formali per la dimostrazione della correttezza dei protocolli.

Supponiamo di avere tre agenti, A , B , C , dove C è l'utente malevolo.

A inizia comunicando con l'utente malevolo C .

1. $A \rightarrow C : \{A, N_a\}_{k_c}$
2. $C \rightarrow B : \{A, N_a\}_{k_b}$
3. $B \rightarrow C : \{N_a, N_b\}_{k_a}$
4. $C \rightarrow A : \{N_a, N_b\}_{k_a}$

In questo scenario, A che voglia parlare con C in realtà si ritrova a parlare indirettamente con B , mentre C funge da ponte, inoltrando i messaggi di A a B e reinoltrando la risposta di B ad A (la nonce creata da B , il messaggio cifrato con la chiave di B).

La ricezione del messaggio al passaggio 3 però agli occhi di A non mostra alcuna irregolarità. Intuiamo però il problema.

Effettuati i 4 punti di autenticazione, A estrae la nonce / chiave di sessione e comunica con C .

5. $A \rightarrow C : \{N_b\}_{k_c}$
6. $C \rightarrow B : \{N_a\}_{k_b}$

B crede quindi che il suo interlocutore sia A . Ritiene di aver autenticato A ma il suo interlocutore è C . Questo è un man-in-the-middle al protocollo di Needham-Schroeder.

L'apprendimento della nonce N_b permette all'attaccante di sovvertire la proprietà l'autenticazione, rendendo il protocollo insicuro.

Needham-Schroeder sostennero che l'attacco di Lowe non fosse tale, dicendo che **il protocollo non era pensato per il modello di attaccante che Lowe aveva introdotto**. C'è da dire che l'attacco ha comunque valenza.

Protocollo Needham-Schroeder nel modello General Attacker

Lo stesso scenario, riproposto per il modello di General Attacker (e non più Dolev-Yao), presenta altre problematiche.

Supponiamo quindi l'esistenza di una doppia attività offensiva, sia da C che da B .

B può exploitare la vulnerabilità trovata da Lowe fingendosi C con A , visto che anche B conosce le nonces (attacco di retaliation, vendetta).

Fix al protocollo Needham-Schroeder per evitare l'attacco di Lowe

Sono di seguito proposte più modifiche.

$$1. A \rightarrow B : \{\{A, N_a\}_{k_a^{-1}}\}_{k_b}$$

$$1. A \rightarrow B : \{\{A, N_a\}_{k_a^{-1}}\}_{k_b}$$

$$2. B \rightarrow A : \{\{N_a, N_b\}_{k_b^{-1}}\}_{k_a}$$

$$2. B \rightarrow A : \{\{N_a, N_b\}_{k_b^{-1}}\}_{k_a}$$

$$2. B \rightarrow A : \{N_a, N_b, B\}_{k_a}$$

$$1. A \rightarrow B : \{A, B, N_a\}_{k_b}$$

Verificare quale di queste modifiche sia un fix.

Per fare questa verifica, introduciamo il fix, riproduciamo lo scenario e vediamo se risolviamo il problema.

La prima proposta non può rappresentare un fix perchè non agisce informando A della presenza di B , in quanto non si agisce sulla nonce N_b .

La seconda proposta è un fix invece. A per estrarre N_b dovrebbe usare la chiave pubblica di B , ma il suo intento era dialogare con C .


Anche la terza proposta è un fix anzi, è proprio questa a rendere la soluzione precedente un fix (di fatto il punto 1 della seconda soluzione è irrilevante).

La quarta soluzione è un fix perchè C non può decifrare né alterare il contenuto del messaggio. Non conoscerà N_b e non può spacciarsi per B . Questo è un fix a livello "informativo". Viene **esplicitato** il destinatario B che sta rispondendo.

Se l'attaccante facesse forgering di questo messaggio A lo capirebbe, facendo abort della comunicazione.

L'ultima soluzione non è un fix. Per quanto non sia modificabile, è ricreabile dall'attaccante e inoltrabile al destinatario.

Notiamo come il positioning del messaggio sia importante, è difficile realizzare un fix al primo messaggio.

 **Domanda d'esame:** perchè la quarta proposto è una soluzione?

Elenco di lettura

- [Protocollo Needham-Schroeder](#)
- Autori Boyd-Mathuria, per un libro sulla sicurezza
- [Gavin Lowe](#)
- [Protocollo Woo-Lam](#)