Explicitness

La forma del messaggio è cruciale ed il messaggio in sé trasporta contenuto informativo. Se le identità del mittente e del destinatario sono significative per la corretta interpretazione del messaggio è prudente **menzionarle esplicitamente**. Chiamiamo questo "**principio di esplicitazione**." Si dice che il messaggio è **pienamente informativo** se rispetta questo principio.

Notare bene che: un messaggio cifrato con la chiave pubblica di B esprime che B è stato capace di estrarla ma non implica che sia stato poi l'unico a riceverla. C'è differenza tra estrarre e conoscere.

Protocollo Woo-Lam

È un protocollo risalente agli anni 80, basato su crittografia simmetrica.
Dati due agenti legittimi che vogliano comunicare, con le rispettive chiavi a lungo termine valide ma non abbiano ancora le chiavi di sessione, introduciamo il concetto di **TTP (Trusted Thrid Party)**, un database di tutte le chiave a lungo termine.

Il TTP può essere visto come il file delle password sotto UNIX.

Il protocollo Woo-Lam è così fatto:

$$
\begin{aligned}
&1.\ A \to B : A \\
&2.\ B \to A : N_b \\
&3.\ A \to B : \{N_b\}_{K_a} \\
&4.\ B \to TTP : \{A, \{N_b\}_{Ka}\}_{K_b} \\
&5.\ TTP \to B : \{N_b\}_{K_b}
\end{aligned}
$$

In questo protocollo viene a mancare la privacy di chi inizia il protocollo. Il mittente esplicita a tutti la sua identità. Notiamo inoltre che la prima nonce è in chiaro, rinunciando alla segretezza di quest'ultima.

Gli step **2** e **3** sono challenging per B, per capire se A è effettivamente chi dice di essere. Il messaggio inoltrato al punto **3** è cifrato con la chiave di A, B non può decifrarlo, per questo ricorre al TTP.

Grazie al layer esterno di cifratura del punto **4** l'attaccante non può alterare il crypto-testo.
A può inoltre legge il mittente dal livello di trasporto, può quindi fare abort se il mittente non è quello che si aspetta.

Al punto **5** TTP estrae il criptotesto, decifrando prima con la chiave di B e poi di A. Se le componenti sono comprensibili allora sono affidabili (se la decription va a buon fine). Ottiene una nonce e la rimanda a B.

La nonce in questo caso è cifrata perché mandarla in chiaro rappresenterebbe un problema. Un attaccante potrebbe estrarla dal punto 2 ed inoltrarla a B, se non fosse cifrata. Inoltre messaggi uguali rendono gli step del protocollo ambigui. L'ambiguità genera vulnerabilità perché non si ha certezza dello step del protocollo in cui ci troviamo.

Per differenziare il messaggio del punto **5** da quello del punto **3** potremmo utilizzare anche altre misure, ad esempio concatenare al messaggio in chiaro il messaggio "TTP" ma non sarebbe altrettanto robusto (si può fare forgering di un messaggio simile).

In questo protocollo l'autenticazione dei due agenti avviene **alla fine**. Un man-in-the-middle in questo caso è inoffensivo, può ascoltare una comunicazione che non comprende e se provasse ad alterare i messaggi porterebbe all'abort della comunicazione. Il man-in-the-middle non è però l'unica offensiva.

Attacco parallel-session a Woo-Lam

L'attaccante C costruisce due sessioni parallele per poter impersonare l'agente legittimo all'ultimo step. C ricostruisce quindi la sessione. In presenza di **session ID** questo non sarebbe possibile.

$$1.\ C \to B : A$$
$$1'.\ C \to B : C$$

$$2.\ B \to A : N_b$$
$$2'.\ B \to C : {N_b}'$$

$$3. C \to B : \{N_b\}_{K_c}$$
$$3'. C \to B : \{N_b\}_{K_c}$$

$$4. B \to TTP : \{A, \{N_b\}_{K_c}\}_{K_b}$$
$$4'. B \to TTP : \{C, \{N_b\}_{K_c}\}_{K_b}$$

$$5. TTP \to B : \{{N_b}''\}_{K_b}$$
$$5'. TTP \to B : \{N_b\}_{K_b}$$

C inizia spacciandosi per A, inizia poi la sessione parallela come sé stesso, B inoltra la nonce N_b a C convinto di star lanciando un challenge ad A. Manda poi la nonce ${N_b}'$ per la sessione con C. La nonce ${N_b}'$ va invece a perdersi, C la butta via.

I messaggi dei punti **3** e **3'** sono uguali. Ricevuti questi due criptotesti, B li inoltra a TTP. B non tiene uno stato dei messaggi perché sarebbe dispendioso, potrebbe inoltre ricevere più messaggi legittimi a causa del livello di trasporto.

Al punto **4** TTP non riesce ad ottenere nulla di sensato dal tentativo di decryption visto che identità e chiave divergono ($N_b{''}$). Nello step **4'** la decryption va a buon fine, il che risolve la challenge di risolvere N_b, challenge che B aveva associato ad A.
Quindi B ha motivo di credere che la sua controparte A sia disponibile, dialogherà invece con C.

Il problema si potrebbe risolvere aggiungendo al quinto step l'esplicitazione dell'agente coinvolto al messaggio. Visto che il riconoscimento avviene solo nello step 5 è possibile applicare una soluzione solo a livello 5.

Vedendo un messaggio del tipo $\{N_b, C\}_{K_b}\ B$ farebbe detection della presenza dell'attaccante, visto che avrebbe evidenza che è C a rispondere.

IPSec

IPSec aggiunge un layer di sicurezza a livello IP. **È una suite di protocolli (AH, ESP, IKE) che distribuisce la chiave di sessione e poi la utilizza per cifrare i pacchetti di livello IP.**

- **AH**: per l'autenticazione, basandosi su chaffing & winnowing
- **ESP**: per cifrare il paylod del pacchetto
- **IKE**: scambio di chiavi, basato su Diffie-Hellmann

È opzionale per IPv4 e mandatorio per IPv6. Ogni pacchetto è arricchito con componenti crittografiche per la sicurezza.

Chaffing & winnowing

È una buona alternativa alla cifratura. Può essere intesa come una forma di steganografia.

1. Il mittente S e il ricevente R concordano una chiave k mediante protocollo Diffie-Hellmann
2. S spedisce a R coppia $m, MAC(m, k)$
3. S spedisce a R molte copie false (chaff) x, y
4. R seleziona coppia giusta verificandone MAC

Obiettivo: ottenere la proprietà di segretezza senza l'utilizzo di una misura crittografica. La versione di segretezza ottenuta dalla crittografia è detta **non-deducibilità**. Anche quella ottenuta in questo caso è una forma di non-deducibilità, anche detta **indistinguibilità**.

Una tecnica di questo tipo è differente dalla cifratura e sicuramente meno scalabile.