Description of the Woo Lam mutual authentication protocol and an attack

The Woo Lam mutual authentication protocol P_{WL} allows to establish a fresh symmetric key using a trusted server S and can be described as follows.

1. $A \rightarrow B: \langle A, N_a \rangle$
2. $B \rightarrow A: \langle B, N_b \rangle$
3. $A \rightarrow B: \{\langle A, \langle B, \langle N_a, N_b \rangle\rangle\rangle\}_{K_{as}}$
4. $B \rightarrow S: \langle A, \langle B, \langle \{\langle A, \langle B, \langle N_a, N_b \rangle\rangle\rangle\}_{K_{as}}, \{\langle A, \langle B, \langle N_a, N_b \rangle\rangle\rangle\}_{K_{bs}} \rangle\rangle\rangle$
5. $S \rightarrow B: \langle \{\langle B, \langle N_a, \langle N_b, K_{ab} \rangle\rangle\rangle\}_{K_{as}}, \{\langle A, \langle N_a, N_b, K_{ab} \rangle\rangle\rangle\}_{K_{bs}} \rangle$
6. $B \rightarrow A: \langle \{\langle B, \langle N_a, \langle N_b, K_{ab} \rangle\rangle\rangle\}_{K_{as}}, \{\langle N_a, N_b \rangle\}_{K_{ab}} \rangle$
7. $A \rightarrow B: \{N_b\}_{K_{ab}}$

Initial knowledge: For any agent C we suppose that K_{cs} is a symmetric keys whose value is initially known only by agent C and the server S.

Data generated during the protocol: N_a is a nonce generated by A. N_b is a nonce generated by B. K_{ab} is a fresh symmetric key created at message 5 by the server S.

Protocol description: Alice starts the protocol by sending her identity A together with a freshly generated random number N_a. Bob replies by sending his identity B together with a freshly generated random number N_b.
Next, Alice sends both identities and both nonces encrypted with the symmetric key she shares with the server S to Bob ($\{A, B, N_a, N_b\}_{K_{as}}$). Bob constructs the same message as Alice but encrypts it with the key K_{bs}. He sends both ciphertexts to the server S (message 4).
The server decrypts both messages and checks that they contain the same plaintexts and that the symmetric keys used correspond to the identities in the messages. The server then generates a key to be shared between A and B. Next the server constructs two ciphertexts $\{B, N_a, N_b, K_{ab}\}_{K_{as}}$ and $\{A, N_a, N_b, K_{ab}\}_{K_{bs}}$ which he sends to B (message 5).
B decrypts the second ciphertext and checks that it contains the expected identity and nonces. He then forwards the first ciphertext to A together with an encryption of the nonces N_a, N_b with the key K_{ab} he just obtained.
Finally Alice decrypts the first ciphertext, checks the values of the identity and nonces and obtains the shared key K_{ab}. She uses this key to decrypt the second ciphertext and verify that the nonces have the expected values. She finally sends nonce N_b encrypted with K_{ab} to Bob, who verifies that the ciphertext contains the correct nonce.

Security properties:

- *Confidentiality:* the protocol must guaranty the secrecy of K_{ab}, i.e. in every session, the value of K_{ab} must be known only by the participants playing the roles of A, B and S.
- *Mutual authentification:* When A successfully finishes a protocol run it must have been with B and vice versa.

Cost of the protocol:
Suppose m_i ($1 \leq i \leq 7$) denotes the ith message of P_{WL}.

$$\begin{aligned} f(P_{WL}) &= \textstyle\sum_{1 \leq i \leq 7} m_i \\ &= 2 + 2 + 165 + 332 + 330 + 228 + 12 \\ &= 1071 \end{aligned}$$

(Note that the WL protocol was not designed with the cost function in mind.)

Attack on the Woo-Lam mutual authentication protocol

We now describe an attack where the intruder I makes B believe that he is executing the protocol with A. The attack requires two sessions. In the description below we denote the n^{th} message of the first session by $i.n$ and the n^{th} message of the second session by $ii.n$. The attack relies on the fact that B could confuse in the message $i.1$ a nonce (N_a in the normal execution) with the identity B. Moreover the intruder replaces some ciphertexts with random values r, r', r''. In the attack these replaced ciphertexts are however never decrypted. Finally B is tricked in accepting the nonce N'_b, i.e. the nonce generated by B in session ii, as a symmetric key shared with A. Both authentication and confidentiality of the key are obviously broken.

$$\begin{array}{llcll}
i.1. & I(A) & \rightarrow & B: & A, B \\
i.2. & B & \rightarrow & I(A): & B, N_b \\
i.3. & I(A) & \rightarrow & B: & r \\
i.4. & B & \rightarrow & I(S): & A, B, r, \{A, B, B, N_b\}_{K_{bs}} \\
ii.1. & I(A) & \rightarrow & B: & A, N_b \\
ii.2. & B & \rightarrow & I(A): & B, N'_b \\
ii.3. & I(A) & \rightarrow & B: & r' \\
ii.4. & B & \rightarrow & I(S): & A, B, r', \{A, B, N_b, N'_b\}_{K_{bs}} \\
i.5. & I(S) & \rightarrow & B: & r'', \{A, B, N_b, N'_b\}_{K_{bs}} \\
i.6. & B & \rightarrow & I(A): & r'', \{B, N_b\}_{N'_b} \\
i.7. & I(A) & \rightarrow & B: & \{N_b\}_{N'_b}
\end{array}$$