Alice

Bob

1. Select Diffie-Hellman parameters: g, p
2. Generate private key b
3. Compute public key: $B = g^b \bmod p$
4. Generate random number: R_{B1}

5. $g \,||\, p \,||\, B \,||\, R_{B1}$ →

6. Generate private key: a
7. Compute public key: $A = g^a \bmod p$
8. Generate random number: R_{A1}
9. Compute shared key: $K = B^a \bmod p$

10. $A \,||\, R_{A1} \,||\, R_{B1}$ ←

11. Verify $R_{B1} \,||\, R_{A1}$
12. Compute shared key: $K = A^b \bmod p$
13. Retrieve facial recognition parameters: F_B

14. $M_B = IV_1 \,||\, E(IV_1, K, F_B)$ →

15. Record channel state information CSI_A from transmission of M_B
16. Compute decryption: $F_B = D(IV_1, K, M_B)$
17. Compute signal origin location: $L_A = G(CSI_A)$
18. Run facial recognition and compute the location of the recognized face: $FL_A = H(F_B)$
19. Verify that L_A and FL_A are overlapping locations
20. Visually verify the intended face was selected
21. Retrieve facial recognition parameters: F_A
22. Extract the final bytes from F_B and assign to IV_2

23. $M_A = E(IV_2, K, F_A)$ →

24. Record channel state information CSI_B from transmission of M_A
25. Extract final bytes from F_B and assign to IV_2
26. Compute decryption: $F_A = D(IV_2, K, M_A)$
27. Compute signal origin location: $L_B = G(CSI_B)$
28. Run facial recognition and compute the location of the recognized face: $FL_B = H(F_A)$
29. Verify that L_B and FL_B are overlapping locations
30. Visually verify the intended face was selected

31. Secure communication can proceed using K