Alice Bob

6. Generate private key: a

7. Compute public key: $A = g^a \mod p$

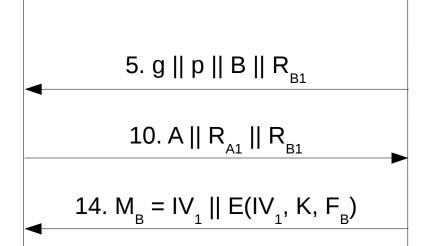
8. Generate random number: R_{A1}

9. Compute shared key: K = B^a mod p

- 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 $\rm F_{\rm B}$ and assign to $\rm IV_{\rm 2}$



23.
$$M_A = E(IV_2, K, F_A)$$

31. Secure communication can proceed using K

- 1. Select Diffie-Hellman parameters: g, p
- 2. Generate private key b
- 3. Compute public key: $B = g^b \mod p$
- 4. Generate random number: $R_{\rm B1}$
- 11. Verify $R_{_{\rm B1}} \parallel R_{_{\rm A1}}$
- 12. Compute shared key: $K = A^b \mod p$
- 13. Retrieve facial recognition parameters: $F_{_{\rm B}}$

- 24. Record channel state information CSI_B from transmission of M_A
- 25. Extract final bytes from $F_{\rm B}$ and assign to $IV_{\rm 2}$
- 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: $FL_{R} = H(F_{\Delta})$
- 29. Verify that $L_{_{\rm B}}$ and $FL_{_{\rm B}}$ are overlapping locations
- 30. Visually verify the intended face was selected