**Entities (Note there will be more entities):**

1. Server
2. ClientA
3. ClientB

**Items**

1. SRSA – Servers RSA public and private key pair
2. NS – nonce from Server, server and ClientA replay protection
3. KSA – Session key between Server and ClientA
4. KSB – Session key between Server and ClientB
5. KAB – Session key between ClientA and ClientB
6. EA – Encryption key for ClientA communication to ClientB
7. DA – Decryption key for ClientA communication from ClientB
8. EB – Encryption key for ClientB communication to ClientA
9. DB – Decryption key for ClientB communication from ClientA
10. AuthA – integrity verification for messages ClientA sends
11. AuthB – integrity verification for messages ClientB sends

**Scenario:**

ClientA wants to talk to ClientB

**Step 1 - Initial Exchange (Login):**

1. ClientA 🡪 SRSA{uname} 🡪 Server
2. ClientA 🡨 SRSA[NS , salt] 🡨 Server
3. ClientA 🡪SRSA{uname, hash(pw, salt), NS} 🡪 Server
4. Server decrypts and checks database for ClientA information.
5. ClientA 🡨 SRSA[gs mod p] 🡨 Server
6. ClientA 🡪 SRSA{ga mod p} 🡪 Server
7. Both agree on KSA = gsa mod p

**Step 2 - Server Gives ClientA Buddy List**

1. KSA{ClientA buddy list}
2. ClientA receives his buddy list

**Step 3 - ClientA Requests to talk to ClientB**

1. ClientA 🡪 KSA{ClientA, “Want to talk to ClientB”} 🡪 Server
2. ClientA 🡨 KSA{KAB , “ClientB”, ticket to ClientB = KSB{ClientA, KAB}} 🡨 Server

**Step 4 – ClientA sends ticket to ClientB (Mutual Authentication)**

1. ClientA 🡪 KAB{ClientA, ClientB}, ticket to ClientB 🡪 ClientB
2. ClientA 🡨 KAB{ClientB, ClientA} 🡨 ClientB

**Step 5 – Derive Encryption and Decryption keys and well as message authentication keys**

1. ClientA computes EA = KAB + 1 for encryption and DA = KAB + 2 for decryption
2. ClientB computers EB = KAB + 2 for encryptions and DB = KAB + 1 for decryption
3. ClientA computes AuthA = KAB + 3
4. ClientB computes AuthB = KAB + 4

**Step 6 – Communication**

ClientA and ClientB communicate with their derived keys.