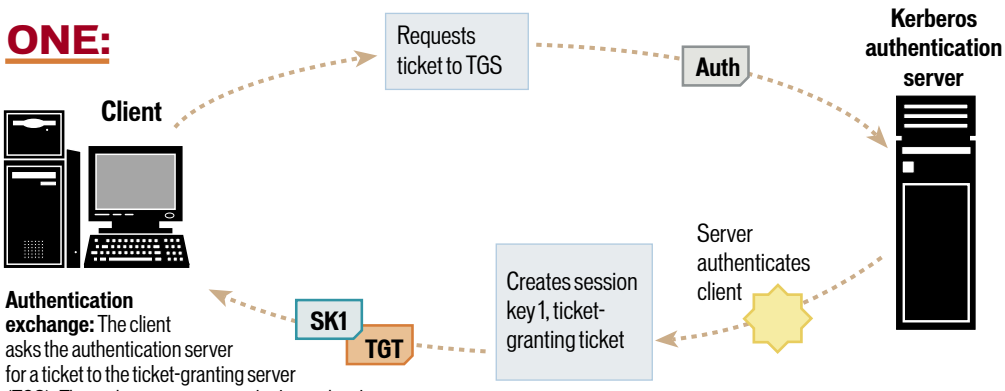
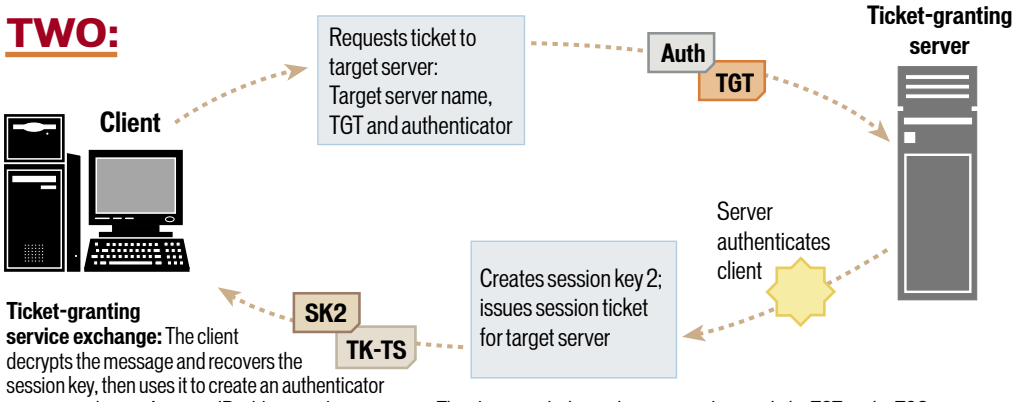
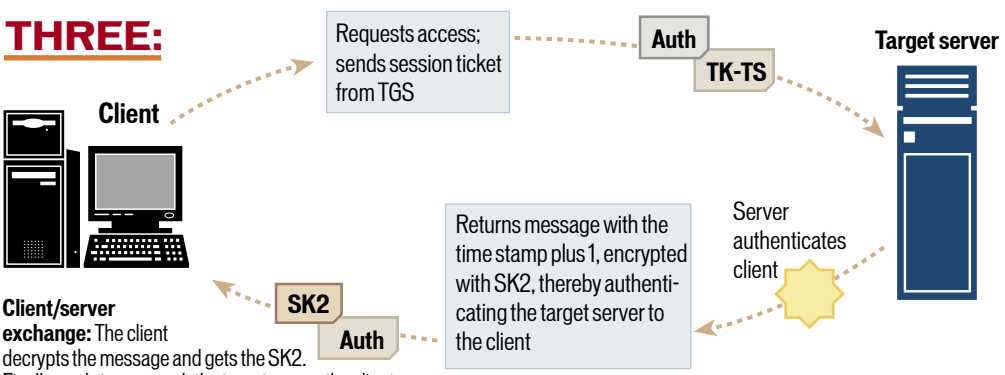
Sharing a Secret: How Kerberos Works



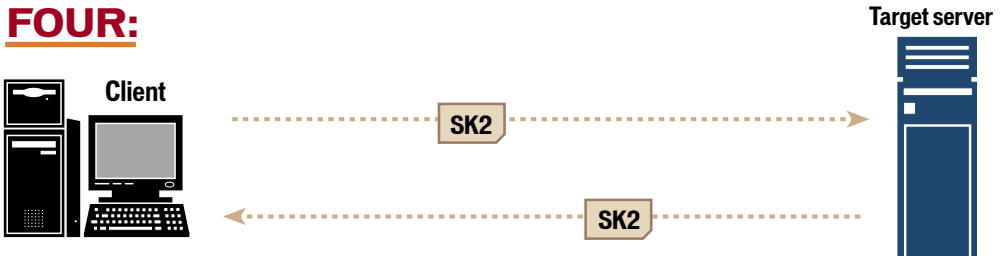
Authentication exchange: The client asks the authentication server for a ticket to the ticket-granting server (TGS). The authentication server looks up the client in its database, then generates a session key (SK1) for use between the client and the TGS. Kerberos encrypts the SK1 using the client's secret key. The authentication server also uses the TGS's secret key (known only to the authentication server and the TGS) to create and send the user a ticket-granting ticket (TGT).



Ticket-granting service exchange: The client decrypts the message and recovers the session key, then uses it to create an authenticator containing the user's name, IP address and a time stamp. The client sends this authenticator, along with the TGT, to the TGS, requesting access to the target server. The TGS decrypts the TGT, then uses the SK1 inside the TGT to decrypt the authenticator. It verifies information in the authenticator, the ticket, the client's network address and the time stamp. If everything matches, it lets the request proceed. Then the TGS creates a new session key (SK2) for the client and target server to use, encrypts it using SK1 and sends it to the client. The TGS also sends a new ticket containing the client's name, network address, a time stamp and an expiration time for the ticket – all encrypted with the target server's secret key – and the name of the server.



Client/server exchange: The client decrypts the message and gets the SK2. Finally ready to approach the target server, the client creates a new authenticator encrypted with SK2. The client sends the session ticket (already encrypted with the target server's secret key) and the encrypted authenticator. Because the authenticator contains plaintext encrypted with SK2, it proves that the client knows the key. The encrypted time stamp prevents an eavesdropper from recording both the ticket and authenticator and replaying them later. The target server decrypts and checks the ticket, authenticator, client address and time stamp. For applications that require two-way authentication, the target server returns a message consisting of the time stamp plus 1, encrypted with SK2. This proves to the client that the server actually knew its own secret key and thus could decrypt the ticket and the authenticator.



Secure communications: The target server knows that the client is who he claims to be, and the two now share an encryption key for secure communications. Because only the client and target server share this key, they can assume that a recent message encrypted in that key originated with the other party.

KEY: Auth Authenticator | SK1 Session key | TGT Ticket

Editor's note: This description was adapted and considerably simplified from *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, 2nd Edition, by Bruce Schneier (Wiley, 1995).