3. In Kerberos, the Ticket Granting Server includes the identity of the server (S) in its response to client: $E_{K_{CG}}(S, K_{CS})$. What would happen if the identity of the server were not included? Does it lead to an attack?

4. Alice and Bob wants to establish a secure channel to communicate securely. Suppose Alice and Bob have long term RSA public and private key pairs: (pk_A, sk_A) and (pk_B, sk_B), respectively. They can use both the RSA signature and encryption algorithms and they know each other's long term public keys. Show how Alice and Bob can achieve forward secrecy.

Alice: Long term RSA public and secret key pairs: (pk_A, sk_A)
Bob: Long term RSA public and secret key pairs: (pk_B, sk_B)

Initially they know each others long term public keys.

Bob:

Sends random N to Alice

Alice:

- Generates temporary RSA key pair for encryption ($T\ pk_A, T\ sk_A$)
- Creates new message s.t. $N \| T\ pk_A$ and signs it with sk_A, resulting in signature S
- Sends $T\ pk_A \| S$ to Bob

Bob:

- Verifies signature S using pk_A
- Now has $T\ pk_A$
- Creates random session key K (symmetric key)
- Encrypts K with her $T\ pk_A$, resulting in K' (using any symmetric key algorithm)
- Using K, he encrypts his message M, resulting in C
- Sends $K' \| C$ to Alice

Alice:

- Decrypts K' using her new private key $T\ sk_A$, resulting back in K
- Decrypts ciphertext C using session key K, resulting in Bob's message M (using symmetric key algorithm choosen by bob)
- Discard session key K and her temporary RSA key pair ($T\ pk_A, T\ sk_A$)

This process can be repeated, If message receiving party generates new temporary RSA key pair for encryption, in order for sending party to generate new random session key. Thus by generating new ession key K and new temporary RSA key pairs ($T\ pk, T\ sk$), and discarding the old ones, we can achieve forward secrecy.