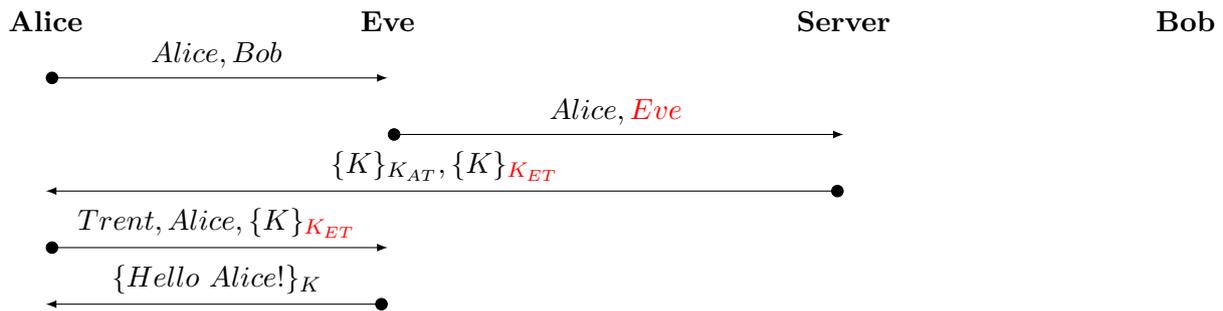


2.1. Warm Up



Assumptions

- (1) Eve ist Nutzer des Protokolls, damit ein *Key Encryption Key* (KEK) K_{ET} zwischen ihr und dem KDC existiert.
- (2) Eve unterliegt dem Angreifermodell *aktiver MitM*.

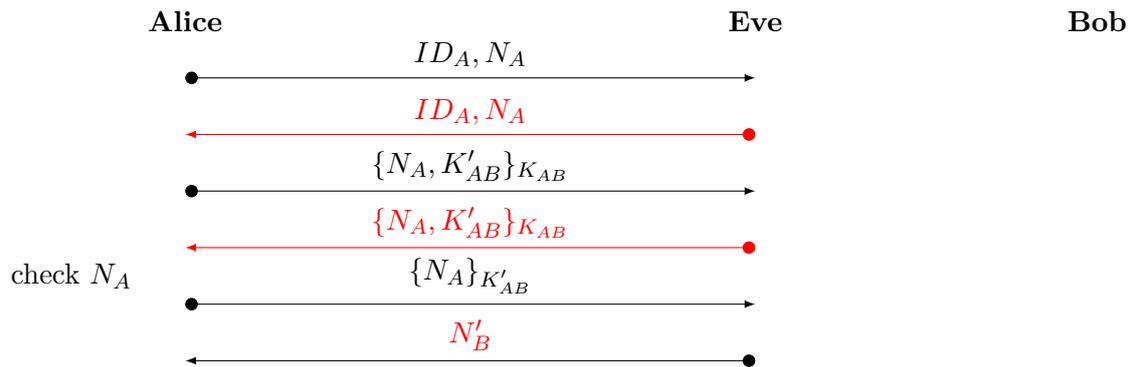
Attack

1. Eve fängt die erste Nachricht $[Alice, Bob]$ von Alice ab, ersetzt *Bob* durch ihren eigenen Namen und leitet die Nachricht anschließend an den Server weiter.
2. Der Server antwortet mit $[\{K\}_{K_{AT}}, \{K\}_{K_{ET}}]$. Eve lässt diese Nachricht durch.
3. Nachdem Alice den Session Key erhalten hat, schickt sie den mit K_{ET} verschlüsselten Session Key K an Bob. Damit Bob weiß, mit wem der Session Key ist und wie dieser verschlüsselt ist, schickt Alice nicht nur das Chiffre, sondern *Trent, Alice* ebenfalls mit.
4. Eve fängt die Nachricht $[Trent, Alice, \{K\}_{K_{ET}}]$ ab. Aus (1) folgt, dass Eve in Besitz von K_{ET} ist und kann daher den Session Key K entpacken. Um das Protokoll zu beenden, und Alice endgültig davon zu überzeugen, dass sie Bob ist, schickt Eve $\{Hello Alice!\}_K$ an Alice zurück.

Evaluation

-

2.2. Key Establishment



Assumptions

- (1) Eve unterliegt dem Angreifermodell *aktiver MitM*.

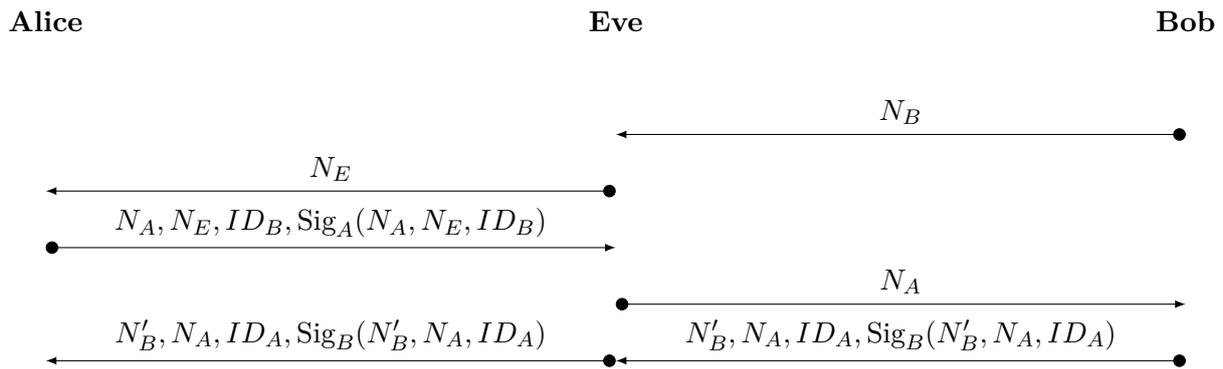
Attack

1. Alice startet das Protokoll mit der Nachricht $[ID_A, N_A]$ an Bob, die als Challenge dient. Bevor die Nachricht Bob erreicht, reflektiert Eve diese zurück an Alice.
2. Alice hat somit ihre eigene Challenge selbst erhalten und beantwortet diese, dem Protokoll konform, mit der Response $[\{N_A, K'_{AB}\}_{K_{AB}}]$. Erneut, bevor die Nachricht Bob erreicht, reflektiert Eve diese zurück an Alice und beantwortet damit die ihr gestellte Challenge.
3. Alice erhält die Response von Eve, im Glauben sie wäre von Bob und verifiziert nach Entschlüsselung die Nonce N_A . Die Verifikation ist bei diesem Angriff immer erfolgreich, weshalb Alice mit $[\{N_A\}_{K'_{AB}}]$ an Bob antwortet.
4. Eve fängt diese Nachricht wieder ab und sendet ohne Verifikation der Nonce N_A eine Nonce N'_B an Alice zurück, um das Protokoll zu beenden. Eve kann die Nonce nicht verifizieren, da sie nicht in Besitz des Session Keys K'_{AB} ist.
5. Alice ist nun in dem Glauben, dass sie mit Bob einen neuen Session Key ausgehandelt hat. Stattdessen hat Bob nichts von der Protokollausführung mitbekommen.

Evaluation

-

2.3. Authentication without Trusted Party



Assumptions

- (1) Eve unterliegt dem Angreifermodell *aktiver MitM*.

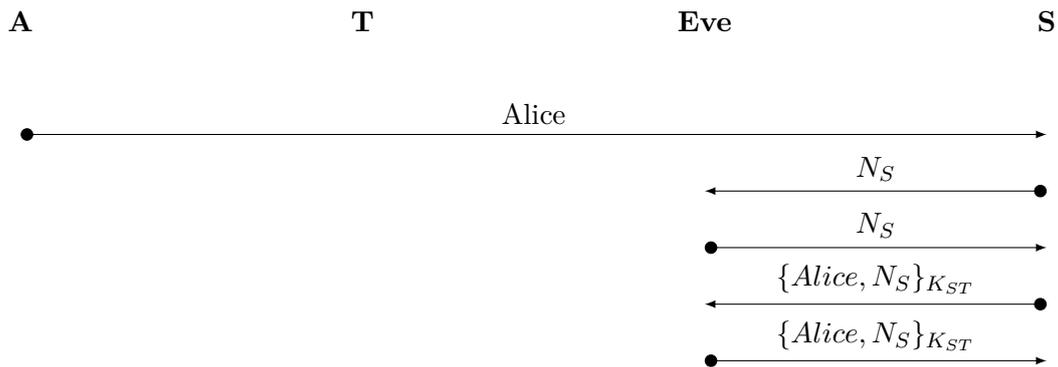
Attack

1. Bob startet das Protokoll, indem er eine Nonce N_B an Alice schickt. Bevor Alice diese Nachricht erhält, ersetzt Eve diese Nonce mit ihrer eigenen.
2. Alice sendet in ihrer Antwort an Bob eine eigene Nonce N_A , die manipulierte Nonce N_E , die ID von Bob ID_B und eine Signatur über diese Daten mit ihrem eigenen Schlüssel. Sie erwartet nun in ihrer Antwort eine neue Nonce von Bob, ihre eigene Nonce und ihre ID mit einer Signatur über diese Daten von Bob. Schickt Eve, getarnt als Alice, Alices Nonce an Bob.
3. Bob antwortet dann gemäss des Protokolls mit einer Nonce von ihm N'_B , Alices Nonce (die Eve ihm geschickt hat), Alices ID, da er meint, dass die Nachricht von Alice kam, und die Signatur über die Nachricht.
4. Da diese Daten von Alice erwartet werden, kann Eve diese Nachricht einfach an Alice weiterleiten.
5. Alice prüft die Signatur, die valide ist, sofern Bob keine falsche Signatur erstellt hat, und meint, dass Eve nun Bob ist.

Evaluation

-

2.4. Authentication with Trusted Party



Assumptions

- (1) Eve unterliegt dem Angreifermodell *aktiver MitM*.

Attack

1. A baut eine Kommunikation mit S auf, indem sie die Nachricht Alice an S schickt.
2. S antwortet mit einer Nonce. Eve fängt diese ab und sendet sie einfach an S zurück.
3. S erwartet hier die Nonce verschlüsselt von Alice. Daher sendet S einfach die beiden Nachrichten, mit dem gemeinsamen Schlüssel mit T verschlüsselt, an T.
4. Eve fängt diese Nachricht ab. Da der Trent sich nicht authentifizieren muss, kann Eve einfach die Nachricht an S zurück schicken.

Evaluation

-