Grundlagen Informationsicherheit und Datensicherheit, Blatt 6

Lukas Baur, 3131138
Felix Bühler, 2973410
Marco Hildenbrand, 3137242

30. Januar 2018

Problem 1: Protocol for Mutual Authentication

Reflection attack:
Eine Person, die N_B korrekt entschlüsselt hat, ist jemand, der den KEY kennt (Alice). Allerdings kennt Bob selbst den KEY auch! Der Angreifer kann also die gesendeten Nachrichten auzeichnen und später nochmal senden und sich damit als Alice ausgeben.

E = Evil (= Attacker)

1. Verbindung 1: A $\rightarrow$ B : $enc_s^k(N_A)$
2. Verbindung 1: B $\rightarrow$ A : $enc_s^k(N_B), N_A$
3. Verbindung 1: A $\rightarrow$ B : N_B

4. Verbindung 2: E $\rightarrow$ A : $enc_s^k(N_B)$
5. Verbindung 2: A $\rightarrow$ E : $enc_s^k(N'_A), N_B$
6. Verbindung 3: E $\rightarrow$ A : $enc_s^k(N'_A)$
7. Verbindung 3: A $\rightarrow$ E : $enc_s^k(N''_A), N'_A$
8. Verbindung 2: A $\rightarrow$ E : N'_A

Problem 2: Kerberos

Im folgenden gehen wir von folgender Anfangssituation aus:

$$\left|\ \text{C}\ \right| \xrightarrow{cert_C, sig_{\hat{k}_C}(t_C, n_2), C, TGS, n_1} \left|\ \text{E}\ \right| \xrightarrow{cert_E, sig_{\hat{k}_E}(t_C, n_2), E, TGS, n_1} \left|\ \text{KAS}\ \right|$$

(a)

Die Vorgestellte Idee stellt **keine** sichere Lösung dar:

Bei einem MitM sendet der Server unter anderem $enc^s_k(AK, n_1, t_{KAS}, TGS, E)$ zurück. Diesen Teil kann E enschlüsseln (mit dem Key k, den er auch bekommt) und sendet dann die substituierte Nachricht zurück:

$$\left|\ \text{E}\ \right| \xleftarrow{enc^a_{k_E}(cert_{KAS}, sig_{\hat{k}_{KAS}}(k, n_2)), E, TGT, enc^s_k(AK, n_1, t_{KAS}, TGS, E)} \left|\ \text{KAS}\ \right|$$

$$\left|\ \text{C}\ \right| \xleftarrow{enc^a_{k_C}(cert_{KAS}, sig_{\hat{k}_{KAS}}(k, n_2)), C, TGT, enc^s_k(AK, n_1, t_{KAS}, TGS, C)} \left|\ \text{E}\ \right|$$

(b)

Die Vorgestellte Idee stellt eine **sichere Lösung** dar:

Der Name, für den der TGT bereitgestellt wird, wird von KAS-Server mit dessen privaten Schlüssels verschlüsselt. Diese Signatur kann E nicht fälschen.

(c)

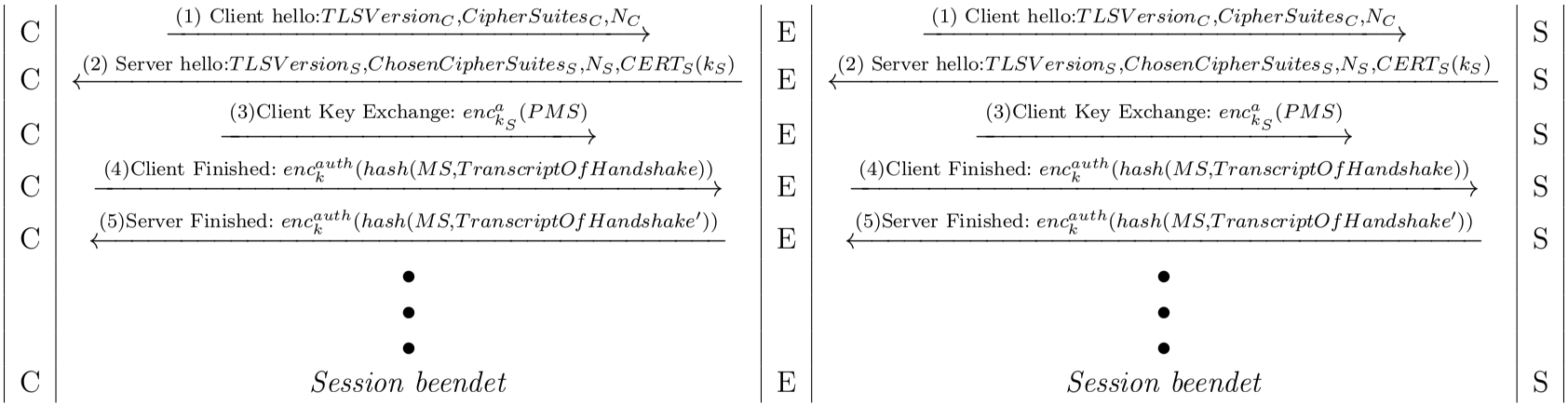
Die Vorgestellte Idee stellt **keine** sichere Lösung dar:

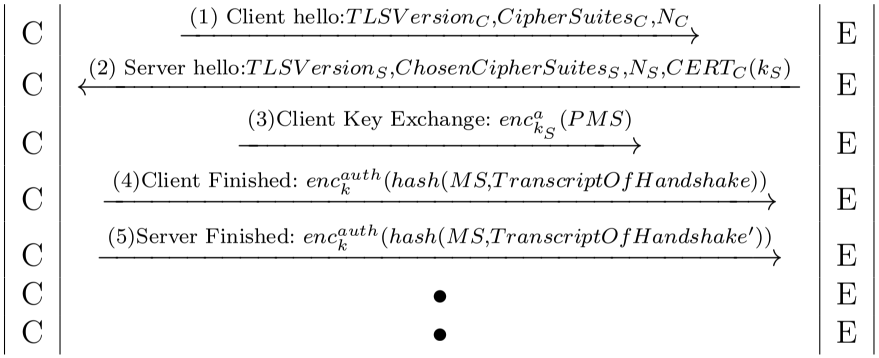
Die Nachricht, in der der Name eingetragen und verschlüsselt wurde, lässt sich von E entschlüsseln, da er den privaten Schlüssel hierfür besitzt. Ein Substituieren ist dann problemlos möglich.

$$\left|\ \text{E}\ \right| \xleftarrow{enc^a_{k_E}(E, cert_{KAS}, sig_{\hat{k}_{KAS}}(k, n_2)), E, TGT, enc^s_k(AK, n_1, t_{KAS}, TGS)} \left|\ \text{KAS}\ \right|$$

$$\left|\ \text{C}\ \right| \xleftarrow{enc^a_{k_C}(C, cert_{KAS}, sig_{\hat{k}_{KAS}}(k, n_2)), C, TGT, enc^s_k(AK, n_1, t_{KAS}, TGS)} \left|\ \text{E}\ \right|$$

Problem 3: TLS and Randomness



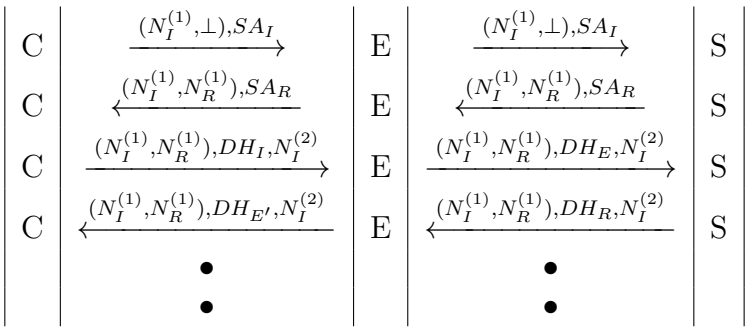


Resultat:

Der Angreifer E lässt C hier im Glauben eine Verbindung mit Server S aufgebaut zu haben. Dem ist aber nicht so, der Angreifer kann nun ggf. alte Nachrichten zwischen Server und Client erneut senden. Ein Auslesen der Nachrichten ist ihm aufgrund des geheimen Keys k allerdings nicht möglich.

Problem 4: Attack on IKEv1 Main Mode

Die dritte Nachricht in diesem Protokoll enthält keine Information über den wahren Empfänger. Der Empfänger weiß demnach nicht sicher, ob die Nachricht für ihn bestimmt war. Mittels einer Authentifizierung zu Beginn könnte das Problem behoben werden.
Die Attacke läuft ähnlich zu der vorgestellten Methode aus der Vorlesung (Brute-Forcing)



$k1$ ist hierdurch bekannt. $k2$ kann durch Brute-Force herausgefunden werden.