Hochschule RheinMain
Fachbereich DCSM - Informatik
Marc Stöttinger

Security
SoSe 23
LV 4120, 7240

Übungsblatt 9

Aufgabe 9.1 (Challenge Response Protocols):

a) Erstellen Sie ein einseitiges Authentifizierungsprotokoll mit dem sich die App beim B2D-Server authentifiziert. Um die Sicherheit zu erhöhen, soll für das Challenge-Response-Verfahren RSA genutzt werden. Für den Ablauf des Protokolls soll auf der APP Seite nur der öffentliche Schlüssel des Schlüsselpaars des Servers benötigt werden, die App besitz kein eigenes privates, öffentliches Schlüsselpaar. Was für Sicherheitsziele muss der Public Key erfüllen?

b) Ändern Sie das Protokoll dahin gehend ab, dass nun der B2D-Server den öffentlichen Schlüssel der App besitzt und die App sich ebenfalls gegenüber dem Server authentifizieren muss.Was ist der Vorteil gegenüber dem ersten Protokoll?
In diesem Protokoll muss der öffentliche Schlüssel nicht das Schutzziel vertrauenswürdig erfüllen, damit sinkt die Angriffsoberfläche.

c) Erweitern Sie nun das Challenge-Response-Verfahren zu einem Mutual-Challenge-Response-Verfahren bei dem sich der B2D-Server nun gegenüber der APP authentifiziert.

d) Was für Vorteile bieten Challenge-Response-Verfahren mit asymmetrischer Kryptographie im Gegensatz zu Verfahren mit symmetrischer Kryptographie?

Aufgabe 9.2 (Needham-Schroeder-Protokoll):

Herr Seky hat sich überlegt, um die Anmeldung der Blutspende APP abzusichern, den registrierten Benutzern einen QR-Code zukommen zu lassen, um die Anmeldung vollständig abzuschliessen. Der QR-Code für die finale Registrierung wird von einem unabhängigen Registrierungsserver (nicht dem B2D-Server) erstellt, bei dem sich die Benutzer zuerst mit der B2D-App registrieren müssen. Dieser Registrier-Server erstellt per Registrierung ein individuellen Sessionschlüssel $K_{Session}$. Um die Sicherheit zu erhöhen, möchte er das Needham-Schroeder-Protokoll benutzen, auf welchem auch das Kerberos-Protokoll beruht. Der QR-Code ist ein Ciphertext, der $K_{Session}$ in verschlüsselter From enthält. Der Schlüssel ist sowohl individuell für den B2D-Server verschlüsselt $ENC_{K_{B2DS_{pb}}}(K_{Session})$ als auch für die APP $ENC_{K_{USER1_{pb}}}(K_{Session})$. Daneben enthält der Ciphertext im QR-Code noch die Adresse vom B2D-Server an den der verschlüsselte Sessionschlüssel für den B2D-Server gesendet werden soll. Um die

Sicherheit zu erhöhen, will Herrr Seky anstelle von symmetrischer Verschlüsselung für das symmetrische Needham-Schroeder-Protokoll RSA (Schulbuchvariante von RSA, kein Padding und Hashing wird verwendet) als Verschlüsselung nutzen. Nachdem der K_{Session} sowohl in der APP als auch auf den B2D-Server entschlüsselt wurde, soll ein Mutual Challenge-Response-Verfahren zwischen App und B2D-Server ausgeführt werden, um sich gegenseitig zu autorisieren und um die Registrierung abzuschliessen.

a) Skizzieren Sie das Protokoll, welches sich Herr Seky überlegt hat. Überlegen Sie, welche Schlüssel im Vorfeld zwischen B2D-Server, App und dem Registrierungsserver ausgetauscht werden müssen, bevor das Protokoll ausgeführt wird. Gehen Sie davon aus, dass jeder Benutzer (z.B. $USER1$) in der APP ein individuelles asymmetrisches Schlüsselpäärchen erzeugt hat.

b) Herrn Seky fällt sehr spät auf (das Protokoll ist schon in den Betrieb gegangen), dass es keine Authentizitätsprüfung zwischen der APP und dem B2D-Server gibt. Ihm fällt auf, dass der Registrierungsserver schon den öffentlichen Schlüssel von der APP hat ($K_{USER1_{pb}}$). Deswegen kommt er auf die Idee, dass sich die APP beim Registrierungssgserver mit einem Signatur basierten Challenge-Response-Verfahren authentifizieren kann. Bevor das angepasste, auf asymmetrischer Verschlüsselung basierte Needham-Schroeder-Protokoll durchgeführt wird, soll folgendes nicht-zustandsbehaftetes Authentisierungsprotokoll ausgeführt werden:

   1. APP sendet Anfrage Req_{USER1} an Registrierungssgserver
   2. Registrierungssgserver generiert einen Nonce N_{auth} und sendet diesen an die APP
   3. Die APP signiert N_{auth} mit $sign_{K_{USER1_{pr}}}(N_{auth,N_{USER1}})$
   4. Der Registrierungssgserver kann somit die Echtheit von N_{auth} und N_{USER1} überprüfen und weiß im Nachgang, dass die erste Nachricht in dem angepassten Needham-Schroeder-Protokoll von der APP stammt.

   Herr Seky will noch zusätzlich ermöglichen, bei einer starken Auslastung redundant Anfragen abarbeiten zu können, deswegen stellt die APP nach einer Wartezeit (5 Sekunden) wiederholende Anfragen mit der gleichen N_{auth}. Nach der dritten Anfrage werden die nachfolgenden Anfragen mit neuer N_{auth} gestellt.

   Warum ist dieses Authentifizierungsprotkoll eine Schwachstelle von RSA (Schulbuchvariante von RSA, kein Padding und Hashing)? Zeigen Sie, wie Sie diese ausnutzen können.

c) Überlegen Sie sich, wie Sie das Needham-Schroeder-Protokoll anstatt nur mit asymmetrischer Verschlüsselung mit Signaturen und asymmetrischer Verschlüsselung umsetzen können.

Aufgabe 9.3 ((Vertiefende Aufgabe:) X3DH):

a) Sie kennen aus der vorherigen Übung und der Vorlesung das Diffie-Hellmann Protokoll. Welches Sicherheitsziel wird nicht erfüllt, wenn DH für einen Schüsselautausch allein genutzt wird und ermöglicht deswegen einen MITM-Angriff? Wie können Sie dieses Problem umgehen?

b) Lesen Sie sich das Dreifach-Deffi-Hellman-Protokoll (X3DH), welches in dem Chat-Pprogramm *signal* genutzt wird, unter `https://www.signal.org/docs/specifications/x3dh/` durch und zeichnen Sie den Protokollablauf auf, wenn Alice den Sessionschlüssel ohne die One-time-key Option direkt von Bob's Parameter ableiten möchte.

c) Die Schlüsselableitefunktion $KDF(\cdot)$ vermischt die drei Deffie-Hellmann geteilten Geheimnisse $DH1, DH2, DH3$, um einen Sessionschlüssel SK abzuleiten. Erläutern Sie warum diese gemacht wird und welche geteilte Geheimnis welches Sicherheitsziel zum Protokoll beiträgt.

d) Was muss Bob berechnen, um auf den gleichen SK zu kommen?