Institut für Theoretische Informatik
Jun.-Prof. Dr. D. Hofheinz

Stammvorlesung Sicherheit im Sommersemester 2014

Übungsblatt 6

Hinweis: Übungsblätter können freiwillig bei Jessica Koch, Raum 256, Geb. 50.34 („Info-Bau“) bis zur Übung am 14.7.14 zur Korrektur abgegeben werden. Die Korrektur dient nur der Selbstkontrolle; es gibt keine Punkte und keinen Klausur-Bonus.

Aufgabe 1. Wir betrachten ein Public-Key-Identifikationsprotokoll $(\mathsf{Gen}, \mathsf{P}, \mathsf{V})$, bei dem ein Benutzer P einen Benutzer V davon überzeugen will, dass P einen diskreten Logarithmus $\log_g h$ modulo p, mit ungeradem primem $p \in \mathbb{N}$ und $g, h \in \mathbb{Z}_p^*$, kennt. Wir nehmen dazu an, dass beide Benutzer dem Protokoll folgen. Dabei sei der Beweisalgorithmus P, der Verifikationsalgoritmus V sowie der Protokollablauf folgendermaßen definiert:

(1) Die Parametergenerierung $\mathsf{Gen}(1^k)$ wählt zuerst ein ungerades primes $q \in \mathbb{N}$ und setzt $p := 2q + 1$, sodass p prim ist. Weiterhin sei $g \in \mathbb{Z}_p^*$ ein Element der Ordnung $q := \mathsf{ord}(g)$. Wir ziehen zufällig gleichverteilt ein $s \in \mathbb{Z}_{\mathsf{ord}(g)}$ und setzen $h := g^s \bmod p$. ($\mathsf{ord}(g)$ gebe dabei die Ordnung von g in $\mathbb{Z}_p^*$ an.) Anschließend wird ein Public-Key $pk := (\mathbb{Z}_p^*, g, h)$ und ein Secret-Key $sk := (\mathbb{Z}_p^*, g, s)$ ausgegeben.

(2) $\mathsf{P}(sk)$ wird mit Eingabe des Secret-Keys sk gestartet, wählt ein zufällig gleichverteiltes $r \in \mathbb{Z}_{\mathsf{ord}(g)}$ und gibt $\mathsf{out}_\mathsf{P} := X := g^r \bmod p$ aus.

(3) Der Verifikationsalgorithmus $\mathsf{V}(pk, \mathsf{out}_\mathsf{P})$ erhält als Eingabe den Public-Key pk und die P-Ausgabe $X := \mathsf{out}_\mathsf{P}$ aus (2); anschließend wird ein $b \in \{0, 1\}$ zufällig gleichverteilt gezogen und $\mathsf{out}_\mathsf{V} := b$ ausgegeben.

(4) $\mathsf{P}(sk, \mathsf{out}_\mathsf{V})$ erhält zu sk die V-Ausgabe $b := \mathsf{out}_\mathsf{V}$ aus (3), berechnet $y := r + b \cdot s \bmod \mathsf{ord}(g)$ und gibt $\mathsf{out}_\mathsf{P} := y$ aus.

(5) $\mathsf{V}(pk, \mathsf{out}_\mathsf{P})$ erhält zu pk die P-Ausgabe $y := \mathsf{out}_\mathsf{V}$ aus (4), überprüft, ob $g^y \bmod p = h^b X \bmod p$ gilt. Falls ja, wird 1 ausgegeben; anderenfalls 0.

Wir sagen, dass V den Beweis von P akzeptiert, wenn nach (mehrfacher) Anwendung des Protokolls immer $\mathsf{V}(pk, \mathsf{out}_\mathsf{P}) = 1$ in (5) gilt.

(a) Zeigen Sie die Korrektheit von $(\mathsf{Gen}, \mathsf{P}, \mathsf{V})$.

(b) Falls V in (3) $b = 0$ zieht, ist y in (4) offensichtlich unabhängig von s. (Das bedeutet, dass jemand ohne Kenntnis von sk in (2) $g^r \bmod p$ und in (4) r senden würde, sodass V akzeptiert.) Falls V in (3) $b = 1$ wählt, ist dies nicht der Fall. Wie könnte jemand, der sk nicht kennt, in diesem Fall dennoch Elemente X in (2) und y in (4) erzeugen, sodass V in (5) 1 ausgibt? Wie groß ist die Erfolgswahrscheinlichkeit?

(c) Geben Sie einen Simulator $\mathcal{S}$ in der Rolle von P (analog zum Graph-Dreifärbbarkeitsbeispiel aus der Vorlesung) an.

Aufgabe 2. Wir betrachten im Folgenden das Jacobi-Symobol $\left(\frac{a}{n}\right)$ für zwei natürliche Zahlen $a, n \in \mathbb{N}$. Dabei gilt, falls $n = p \in \mathbb{P}$ eine Primzahl ist, folgendes:

$$\left(\frac{a}{p}\right) := a^{\frac{p-1}{2}} \bmod p = \begin{cases} 1 & \text{wenn } a \text{ quadratischer Rest modulo } p \text{ ist} \\ -1 & \text{wenn } a \text{ quadratischer Nichtrest modulo } p \text{ ist} \\ 0 & \text{wenn } a \text{ ein Vielfaches von } p \text{ ist} \end{cases}$$

Desweiteren gelten folgende Rechenregeln für ungerades $n \in \mathbb{N}$:

* $\left(\frac{a}{n}\right) = \left(\frac{a \bmod n}{n}\right), \left(\frac{-1}{n}\right) = (-1)^{\frac{n-1}{2}}$
* $\left(\frac{a_1 a_2}{n}\right) = \left(\frac{a_1}{n}\right)\left(\frac{a_2}{n}\right), \left(\frac{a}{mn}\right) = \left(\frac{a}{m}\right)\left(\frac{a}{n}\right)$

(a) Berechnen Sie die Jacobi-Symbole von $\left(\frac{15}{35}\right)$, $\left(\frac{32}{33}\right)$, $\left(\frac{17}{39}\right)$

(b) Zeigen Sie, dass -1 für $n = pq$ mit Primzahlen p und q für die $p \equiv q \equiv 3 \bmod 4$ gilt, kein quadratischer Rest mod n ist, jedoch das Jacobi-Symbol 1 ergibt. (Solche Zahlen n nennt man Blum-Integer.)

(c) Wir betrachten nun folgendes Identifikationsprotokoll, das auf der Schwierigkeit der Berechnung von Quadratwurzeln in der Gruppe $\mathbb{Z}_n^*$ für $n = pq$ Blum-Integer besteht, sofern n nicht leicht zu faktorisieren ist:

Eine vertrauenswürdige Instanz veröffentlicht einen RSA-Modulus $n = pq$, wobei n ein Blum-Integer ist. Ein Prover P möchte einem Verifier V gegenüber beweisen, dass er in Besitz eines Geheimnisses ist. Dazu wählt P geheime $s \xleftarrow{\$} \mathbb{Z}_n$, $t_1 \xleftarrow{\$} \{-1, 1\}$ und veröffentlicht $v = t_1 \cdot s^2 \bmod n$.

Protokoll:

1. P wählt $r \xleftarrow{\$} \mathbb{Z}_n$, $t_2 \xleftarrow{\$} \{-1, 1\}$, berechnet $x = t_2 \cdot r^2 \bmod n$, sendet x an V.
2. V wählt $b \xleftarrow{\$} \{0, 1\}$, sendet b an P.
3. P berechnet $y = r \cdot s^b \bmod n$, sendet y an V.
4. V überprüft, ob $y^2 = \pm\, x \cdot v^b \bmod n$ gilt.

(i) Zeigen Sie die Korrektheit des Protokolls für ehrlichen P und V.

(ii) Geben Sie die Erfolgswahrscheinlichkeit für einen unehrlichen P an, falls das Protokoll k mal durchgeführt wird.

(iii) Zeigen Sie durch Angabe eines Simulators, dass die Zero-Knowledge-Eigenschaft gilt.

(iv) Welche Information würde V über v lernen, falls wir das Protokoll ohne die zufälligen Vorzeichen t_1, t_2 durchführen würden? (Überlegen Sie dazu, was P gegenüber V beweist, falls es keine Vorzeichen gibt)

Aufgabe 3. **Achtung**: Aufgabe wurde geändert! Es gibt keine Set-Befehle und Write-Rechte implizieren **keine** Read-Rechte!

Gegeben sei das folgende System im Bell-LaPadula-Modell:

- Subjektmenge $\mathcal{S} = \{\text{Alice}, \text{Bob}, \text{Carol}\}$
- Objektmenge $\mathcal{O} = \{D_1, D_2, D_3, D_4\}$
- Menge der Zugriffsoperationen $\mathcal{A} = \{\texttt{read}, \texttt{write}, \texttt{append}, \texttt{execute}\}$
- Menge der Sicherheitsstufen $L = \{\textit{Verwaltung}, \textit{Lehre}, \textit{Forschung}, \textit{Präsidium}\}$ mit der partiellen Ordnung definiert durch $\textit{Lehre} \leq \textit{Verwaltung} \leq \textit{Präsidium}$ und $\textit{Lehre} \leq \textit{Forschung} \leq \textit{Präsidium}$
- Zugriffskontrollmatrix M gegeben durch

	D_1	D_2	D_3	D_4
Alice	`r,w,a`	`r`	`r,w,a`	`r,x`
Bob	`r,w,a`	`r,w,a`	`r,w,a`	`r,x`
Carol	`r`	`r`	`r,w,a`	`r,w,a,x`

- Zuordnung der Sicherheitsstufen $F = (f_S, f_C, f_O)$ gegeben durch

	f_S	f_C
Alice	*Verwaltung*	*Verwaltung*
Bob	*Forschung*	*Lehre*
Carol	*Präsidium*	*Forschung*

	f_O
D_1	*Verwaltung*
D_2	*Lehre*
D_3	*Forschung*
D_4	*Präsidium*

Geben Sie für die folgende Liste von Zugriffsanforderungen und Anfragen zur Änderung der Sicherheitsstufe an, ob das System Zugriff gewährt beziehungsweise die Sicherheitsstufe ändert oder nicht. Berücksichtigen Sie erfolgreiche Zugriffe und Änderungen der Sicherheitsstufe für die nachfolgenden Zugriffsanforderungen. Geben Sie für abgelehnte Anfragen als Begründung an, welche Eigenschaft(en) verletzt würde(n), wenn der Zugriff beziehungsweise die Änderung der Sicherheitsstufe erlaubt würde. Dabei bezeichne (s, o, l) eine Zugriffsanforderung von Subjekt s auf Objekt o mit der Zugriffsart l. Gehen Sie von einem leeren Initialzustand aus.

1. $(\mathit{Alice}, D_1, \mathtt{a})$
2. $(\mathit{Bob}, D_2, \mathtt{w})$
3. $(\mathit{Bob}, D_3, \mathtt{r})$
4. $(\mathit{Carol}, D_3, \mathtt{r})$
5. $(\mathit{Carol}, D_2, \mathtt{w})$
6. $(\mathit{Alice}, D_2, \mathtt{r})$
7. $(\mathit{Bob}, D_4, \mathtt{r})$
8. $(\mathit{Bob}, D_2, \mathtt{a})$