Ruhr-Universität Bochum
Lehrstuhl für Kryptologie und IT-Sicherheit
Prof. Dr. Alexander May
Stefan Hoffmann, Ilya Ozerov

Hausübungen zur Vorlesung

Kryptographie

WS 2013/14

Blatt 10 / 18. Dezember 2013

Abgabe: 14. Januar 2014, 14.00 Uhr (vor der Vorlesung), Kasten NA/02

AUFGABE 1 (5 Punkte)**:**

Betrachten Sie das Spiel $\widetilde{\mathsf{KE}}$ aus der Präsenzübung (Blatt 10, Aufgabe 3), die zugehörige Definition eines *stark sicheren* Schlüsselaustauschprotokolls, sowie das Spiel KE und die Definition eines *sicheren* Schlüsselaustauschprotokolls aus der Vorlesung.

Zeigen Sie, dass jedes *sichere* Schlüsselaustauschprotokoll Π auch *stark sicher* ist.

AUFGABE 2 (5 Punkte)**:**

Sei $\mathcal{G}$ ein ppt-Algorithmus, der zur Eingabe 1^n eine zyklische Gruppe G der Ordnung q und einen Generator g erzeugt, wobei q eine Primzahl der Bitlänge n ist.

Wir definieren das *Inverse Computational Diffie-Hellman Problem* (kurz InvCDH-Problem) wie folgt: Das InvCDH-Problem ist hart bzgl. $\mathcal{G}$, falls für jeden ppt-Algorithmus $\mathcal{A}$ gilt

$$\mathsf{Ws}\left[\mathcal{A}(g, q, g^a) = g^{(a^{-1})}\right] \leq \mathsf{negl}(n).$$

Hierbei wird die Wahrscheinlichkeit über die zufällige Wahl von $(g, q) \leftarrow \mathcal{G}(1^n)$ und $a \in_R \mathbb{Z}_q^*$ sowie $\mathcal{A}$'s Münzwürfe gebildet.

Zeigen Sie: Wenn das InvCDH-Problem hart ist bzgl. $\mathcal{G}$, so ist auch das CDH-Problem hart bzgl. $\mathcal{G}$.

Bitte wenden!

AUFGABE 3 (5 Punkte)**:**

Sei $\mathcal{G}$ ein ppt-Algorithmus, der zur Eingabe 1^n eine zyklische Gruppe G der *primen* Ordnung q und einen Generator g erzeugt, wobei q Bitlänge n hat. Sei zudem das DDH-Problem hart bzgl. $\mathcal{G}$.

Betrachten Sie das folgende Schlüsselaustauschprotokoll:

1) Alice wählt $(g, q) \leftarrow \mathcal{G}(1^n)$, $x \in_R \mathbb{Z}_q^*$, berechnet $\alpha := g^x$ und schickt (g, q, α) an Bob.
2) Bob wählt $y \in_R \mathbb{Z}_q^*$, berechnet $\beta := g^y$, sowie $\gamma_B := \alpha^y$ und schickt β an Alice.
3) Alice berechnet $\gamma_A := \beta^x$, wählt $s, t \in_R \mathbb{Z}_q^*$ mit $s \neq t$ und $\mathrm{ggT}(s, t) = 1$, wählt $z \in_R \mathbb{Z}_q^*$, berechnet $k_A := g^z$, $\sigma := \gamma_A \cdot (k_A)^s$, sowie $\tau := \gamma_A \cdot (k_A)^t$ und schickt (s, t, σ, τ) an Bob.
4) Bob berechnet mit Hilfe des EEA die Bezoutkoeffizienten $u, v \in \mathbb{Z}_q$, so dass $u \cdot s + v \cdot t = 1 \bmod q$ und berechnet $k_B := (\sigma \cdot \gamma_B^{-1})^u \cdot (\tau \cdot \gamma_B^{-1})^v$.

(a) Zeigen Sie, dass Alice und Bob denselben Schlüssel berechnen, d. h. $k_A = k_B$ gilt.

(b) Analysieren Sie die Sicherheit des Protokolls, d. h. beweisen Sie entweder die Sicherheit oder geben Sie einen konkreten Angriff an.

AUFGABE 4 (5 Punkte)**:**

Ein Unternehmen verwendet den öffentlichen RSA-Schlüssel (N, e). Bei einem Sicherheitsupdate wird der Schlüssel auf (N, e') aktualisiert, d. h. der Modulus N bleibt erhalten, nur der Exponent e wird geändert. Dabei wird allerdings darauf geachtet, dass e und e' teilerfremd sind, d. h. dass $\mathrm{ggT}(e, e') = 1$ gilt.

Ein Kunde schickt eine verschlüsselte Nachricht m, noch unter dem alten öffentlichen Schlüssel, an das Unternehmen. Nachdem er auf den Fehler hingewiesen wurde, schickt er die Nachricht erneut, nun verschlüsselt mit dem neuen öffentlichen Schlüssel. Ein Angreifer liest beide Übertragungen mit und erhält $x = m^e \bmod N$ sowie $y = m^{e'} \bmod N$.

(a) Zeigen Sie allgemein, wie der Angreifer die Nachricht m in Zeit polynomiell in $\log(N)$ bestimmen kann. Nehmen Sie dazu an, dass $e, e' < N$ sind.

(b) Bestimmen Sie für $N = 247, e = 11, e' = 17, x = 24$ und $y = 93$ die Nachricht m.