MTAT.07.003 Cryptology II
Spring 2012 / Exercise session ?? / Example Solution

Exercise (Witness indistinguishability of disjunctive composition). *Let $\mathbb{G}$ be a discrete logarithm group with a prime number q elements. Use the Schnorr protocol*

$$
\begin{array}{ccc}
\mathcal{P} & & \mathcal{V} \\
r \xleftarrow{u} \mathbb{Z}_q & & \\
 & \xrightarrow{\qquad\quad \alpha = g^r \qquad\quad} & \\
 & & \beta \xleftarrow{u} \mathbb{Z}_q \\
 & \xleftarrow{\qquad\qquad \beta \qquad\qquad} & \\
 & \xrightarrow{\qquad \gamma = x\beta + r \qquad} & \\
 & & [g^{\gamma} \stackrel{?}{=} \alpha y^{\beta}]
\end{array}
$$

to construct a proof of knowledge $\textsc{pok}[\exists x_1 \exists x_2 : y_1 = g^{x_1} \vee y_2 = g^{x_2}]$. *Give a complete description of the provers behaviour if it knows only x_1, only x_2 or both secrets. Use game rewriting to show that the output distribution of a verifier does not depend which of those provers it interacts.*

Solution. Recall that the disjunctive composition is a three move protocol, where the prover sends two commitment messages α_1 and α_2. After that the verifier sends a challenge β, which is then freely decomposed into sub-challenges β_1 and β_2 and augmented with corresponding responses γ_1 and γ_2. Next, the verifier checks that $\beta_1 + \beta_2 = \beta$ and that individual protocol transcripts $(\alpha_1, \beta_1, \gamma_1)$ and $(\alpha_2, \beta_2, \gamma_2)$ are valid. The overall structure of disjunctive composition is depicted below.

$$
\begin{array}{ccc}
\mathcal{P} & & \mathcal{V} \\
 & \xrightarrow{\qquad\qquad \alpha_1, \alpha_2 \qquad\qquad} & \\
 & & \beta \xleftarrow{u} \mathbb{Z}_q \\
 & \xleftarrow{\qquad\qquad \beta \qquad\qquad} & \\
 & \xrightarrow{\qquad \beta_1, \beta_2, \gamma_1, \gamma_2 \qquad} & \\
 & & [\beta_1 + \beta_2 \stackrel{?}{=} \beta] \\
 & & [g^{\gamma_1} \stackrel{?}{=} \alpha_1 y^{\beta_1}] \wedge [g^{\gamma_2} \stackrel{?}{=} \alpha_2 y^{\beta_2}]
\end{array}
$$

If the prover knows both secret exponents x_1 and x_2, then it can carry out both protocols as usual and we get the following detailed description of a protocol execution:

$$
\begin{array}{ccc}
\mathcal{P} & & \mathcal{V} \\
r_1 \xleftarrow{u} \mathbb{Z}_q & & \\
r_2 \xleftarrow{u} \mathbb{Z}_q & & \\
\alpha_1 \leftarrow g^{r_1} & & \\
\alpha_2 \leftarrow g^{r_2} & & \\
 & \xrightarrow{\qquad\qquad \alpha_1, \alpha_2 \qquad\qquad} & \\
 & & \beta \xleftarrow{u} \mathbb{Z}_q \\
 & \xleftarrow{\qquad\qquad \beta \qquad\qquad} & \\
\beta_2 \xleftarrow{u} \mathbb{Z}_q & & \\
\beta_1 \leftarrow \beta - \beta_2 & & \\
\gamma_1 \leftarrow x_1 \beta_1 + r_1 & & \\
\gamma_2 \leftarrow x_2 \beta_2 + r_2 & & \\
 & \xrightarrow{\qquad \beta_1, \beta_2, \gamma_1, \gamma_2 \qquad} & \\
 & & [\beta_1 + \beta_2 \stackrel{?}{=} \beta] \\
 & & [g^{\gamma_1} \stackrel{?}{=} \alpha_1 y^{\beta_1}] \wedge [g^{\gamma_2} \stackrel{?}{=} \alpha_2 y^{\beta_2}]
\end{array}
$$

Prover knows only x_1. When the prover does not know x_2, it cannot create the response for the second sub-challenge β_2 without creating a simulated protocol transcript first. As a result, the prover must fix the value β_2 before the challenge β is generated. The corresponding protocol execution is depicted below.

$$
\begin{array}{ccc}
\mathcal{P} & & \mathcal{V} \\
\begin{array}{c} r_1 \xleftarrow{u} \mathbb{Z}_q \\ \beta_2 \xleftarrow{u} \mathbb{Z}_q \\ \gamma_2 \xleftarrow{u} \mathbb{Z}_q \\ \alpha_1 \leftarrow g^{r_1} \\ \alpha_2 \leftarrow \frac{g^{\gamma_2}}{y_2^{\beta_2}} \end{array} & & \\
 & \xrightarrow{\alpha_1, \alpha_2} & \\
 & & \beta \xleftarrow{u} \mathbb{Z}_q \\
 & \xleftarrow{\beta} & \\
\begin{array}{c} \beta_1 \leftarrow \beta - \beta_2 \\ \gamma_1 \leftarrow x_1\beta_1 + r_1 \end{array} & & \\
 & \xrightarrow{\beta_1, \beta_2, \gamma_1,\ \gamma_2} & \\
 & & \begin{array}{c} [\beta_1 + \beta_2 \stackrel{?}{=} \beta] \\ [g^{\gamma_1} \stackrel{?}{=} \alpha_1 y^{\beta_1}] \wedge [g^{\gamma_2} \stackrel{?}{=} \alpha_2 y^{\beta_2}] \end{array}
\end{array}
$$

Prover knows only x_2. When the prover does not know x_1, it cannot create the response for the second sub-challenge β_1 without creating a simulated protocol transcript first. As a result, the prover must fix the value β_1 before the challenge β is generated. The corresponding protocol execution is depicted below.

$$
\begin{array}{ccc}
\mathcal{P} & & \mathcal{V} \\
\begin{array}{c} r_2 \xleftarrow{u} \mathbb{Z}_q \\ \beta_1 \xleftarrow{u} \mathbb{Z}_q \\ \gamma_1 \xleftarrow{u} \mathbb{Z}_q \\ \alpha_1 \leftarrow \frac{g^{\gamma_1}}{y_1^{\beta_1}} \\ \alpha_2 \leftarrow g^{r_2} \end{array} & & \\
 & \xrightarrow{\alpha_1, \alpha_2} & \\
 & & \beta \xleftarrow{u} \mathbb{Z}_q \\
 & \xleftarrow{\beta} & \\
\begin{array}{c} \beta_2 \leftarrow \beta - \beta_1 \\ \gamma_2 \leftarrow x_2\beta_2 + r_2 \end{array} & & \\
 & \xrightarrow{\beta_1, \beta_2, \gamma_1,\ \gamma_2} & \\
 & & \begin{array}{c} [\beta_1 + \beta_2 \stackrel{?}{=} \beta] \\ [g^{\gamma_1} \stackrel{?}{=} \alpha_1 y^{\beta_1}] \wedge [g^{\gamma_2} \stackrel{?}{=} \alpha_2 y^{\beta_2}] \end{array}
\end{array}
$$

Witness indistinguishability. In the following we show that the verifier cannot distinguish which of those three modes the honest prover runs. As the first step, we show that the verifier cannot distinguish whether the prover knows x_1 and x_2 or only x_1. For that, we construct games for both execution modes where the output is determined by the verifier $\mathcal{V}_*$ and then show that both games produce identical output distributions. The game $\mathcal{G}_0$ corresponds to the case where the prover knows x_1 and x_2 and game $\mathcal{G}_3$ to the case where prover knows only x_1:

$$\mathcal{G}_0 \qquad \left[\begin{array}{l} r_1 \xleftarrow[u]{} \mathbb{Z}_q \\ r_2 \xleftarrow[u]{} \mathbb{Z}_q \\ \alpha_1 \leftarrow g^{r_1} \\ \alpha_2 \leftarrow g^{r_2} \\ \beta \leftarrow \mathcal{V}_*(\alpha_1, \alpha_2) \\ \beta_2 \xleftarrow[u]{} \mathbb{Z}_q \\ \beta_1 \leftarrow \beta - \beta_2 \\ \gamma_1 \leftarrow x_1\beta_1 + r_1 \\ \gamma_2 \leftarrow x_2\beta_2 + r_2 \\ \textbf{return } \mathcal{V}_*(\beta_1, \beta_2, \gamma_1, \gamma_2) \end{array}\right.$$

$$\mathcal{G}_3 \qquad \left[\begin{array}{l} r_1 \xleftarrow[u]{} \mathbb{Z}_q \\ \beta_2 \xleftarrow[u]{} \mathbb{Z}_q \\ \gamma_2 \xleftarrow[u]{} \mathbb{Z}_q \\ \alpha_1 \leftarrow g^{r_1} \\ \alpha_2 \leftarrow \dfrac{g^{\gamma_2}}{y_2^{\beta_2}} \\ \beta \leftarrow \mathcal{V}_*(\alpha_1, \alpha_2) \\ \beta_1 \leftarrow \beta - \beta_2 \\ \gamma_1 \leftarrow x_1\beta_1 + r_1 \\ \textbf{return } \mathcal{V}_*(\beta_1, \beta_2, \gamma_1, \gamma_2) \end{array}\right. .$$

Since the sampling of β_2 does not depend on β in $\mathcal{G}_0$ we can move it towards the beginning of the game and get a new game with the same output distribution:

$$\mathcal{G}_1 \qquad \left[\begin{array}{l} r_1 \xleftarrow[u]{} \mathbb{Z}_q \\ \beta_2 \xleftarrow[u]{} \mathbb{Z}_q \\ r_2 \xleftarrow[u]{} \mathbb{Z}_q \\ \alpha_1 \leftarrow g^{r_1} \\ \alpha_2 \leftarrow g^{r_2} \\ \beta \leftarrow \mathcal{V}_*(\alpha_1, \alpha_2) \\ \beta_1 \leftarrow \beta - \beta_2 \\ \gamma_1 \leftarrow x_1\beta_1 + r_1 \\ \gamma_2 \leftarrow x_2\beta_2 + r_2 \\ \textbf{return } \mathcal{V}_*(\beta_1, \beta_2, \gamma_1, \gamma_2) \end{array}\right. .$$

Let us now concentrate on the variables $r_2, \alpha_2, \beta_2, \gamma_2$. Note that even for fixed β_2 the value γ_2 must be uniformly distributed as r_2 is uniformly distributed. Hence, we can pick γ_2 and then calculate r_2 from it:

$$\mathcal{G}_2 \qquad \left[\begin{array}{l} r_1 \xleftarrow[u]{} \mathbb{Z}_q \\ \beta_2 \xleftarrow[u]{} \mathbb{Z}_q \\ \gamma_2 \xleftarrow[u]{} \mathbb{Z}_q \\ r_2 \leftarrow \gamma_2 - x_2\beta_2 \\ \alpha_1 \leftarrow g^{r_1} \\ \alpha_2 \leftarrow g^{r_2} \\ \beta \leftarrow \mathcal{V}_*(\alpha_1, \alpha_2) \\ \beta_1 \leftarrow \beta - \beta_2 \\ \gamma_1 \leftarrow x_1\beta_1 + r_1 \\ \textbf{return } \mathcal{V}_*(\gamma_1, \gamma_2, \beta_1, \beta_2) \end{array}\right. .$$

Now note that $\alpha_2 = g^{r_2} = g^{\gamma_2 - x_2\beta_2} = \frac{g^{\gamma_2}}{y^{\beta_2}}$ and thus $\mathcal{G}_2$ is equivalent to $\mathcal{G}_3$.

The sequence of game transformations that show that knowledge of x_2 is indistinguishable form the knowledge of x_1 and x_2 is analogous, except for a minor detail how β_1 and β_2 are generated:

$$\begin{array}{ll}
\mathcal{G}_0 & \overline{\mathcal{G}}_3 \\
\left[\begin{array}{l}
r_1 \xleftarrow{u} \mathbb{Z}_q \\
r_2 \xleftarrow{u} \mathbb{Z}_q \\
\alpha_1 \leftarrow g^{r_1} \\
\alpha_2 \leftarrow g^{r_2} \\
\beta \leftarrow \mathcal{V}_*(\alpha_1, \alpha_2) \\
\beta_2 \xleftarrow{u} \mathbb{Z}_q \\
\beta_1 \leftarrow \beta - \beta_2 \\
\gamma_1 \leftarrow x_1\beta_1 + r_1 \\
\gamma_2 \leftarrow x_2\beta_2 + r_2 \\
\textbf{return } \mathcal{V}_*(\beta_1, \beta_2, \gamma_1, \gamma_2)
\end{array}\right.
&
\left[\begin{array}{l}
r_2 \xleftarrow{u} \mathbb{Z}_q \\
\beta_1 \xleftarrow{u} \mathbb{Z}_q \\
\gamma_1 \xleftarrow{u} \mathbb{Z}_q \\
\alpha_1 \leftarrow \dfrac{g^{\gamma_1}}{y_1^{\beta_1}} \\
\alpha_2 \leftarrow g^{r_2} \\
\beta \leftarrow \mathcal{V}_*(\alpha_1, \alpha_2) \\
\beta_2 \leftarrow \beta - \beta_1 \\
\gamma_1 \leftarrow x_1\beta_1 + r_1 \\
\textbf{return } \mathcal{V}_*(\beta_1, \beta_2, \gamma_1, \gamma_2)
\end{array}\right. .
\end{array}$$

It is easy to see that it does not matter if we first pick β_1 and then calculate $\beta_2 = \beta - \beta_1$ in $\mathcal{G}_0$ or if we first pick β_2 and then calculate $\beta_1 = \beta - \beta_2$. After doing this switch, we will arrive at the setting where the games are analogously aligned as in the first equivalence proof and we can do analogous proof that games $\mathcal{G}_0$ and $\mathcal{G}_3$ are identical. As a result, we have shown

$$\mathcal{G}_3^{\mathcal{V}_*} \equiv \mathcal{G}_0^{\mathcal{V}_*} \equiv \overline{\mathcal{G}}_3^{\mathcal{V}_*}$$

and thus by transitivity the verifier will not be able to distinguish between any of these games. Note that the proof holds also for malicious adversary who could create β as it wishes, since during the game rewriting we made no assumption about β.