MTAT.07.003 Cryptology II
Spring 2009 / Exercise session XII

1. Sometimes it is inherently possible to implement the desired functionality without breaching desired security. For instance, consider a two-party protocol for addition, where at the end both parties obtain $x_1 + x_2$.

   (a) Show that any protocol that implements addition also reveals the input of the opponent. Generalise the result and show that it is always possible to deduce something about the opponents input unless the desired output is not a constant (protocol is actually redundant).

   (b) Show that if a malicious party knows the parity of the opponents input before the protocol, then he or she can control the output parity in the addition protocol. Generalise the result and show that given enough information the adversary can always partially control the opponents output unless the output is constant.

2. To illustrate some aspects of secure computations, consider a simultaneous broadcast protocol used for the rock-paper-scissors game:

   - $\mathcal{P}_1$ computes $\mathsf{pk} \leftarrow \mathsf{Gen}$. $(c, d) \leftarrow \mathsf{Com}_{\mathsf{pk}}(x_1)$ and sends pk, c to $\mathcal{P}_2$.
   - $\mathcal{P}_2$ replies x_2 to $\mathcal{P}_1$ who after that releases d to $\mathcal{P}_1$,
   - $\mathcal{P}_2$ computes $x_1 \leftarrow \mathsf{Open}_{\mathsf{pk}}(c, d)$ and both parties output x_1, x_2.

   Let $\mathcal{P}_2^*$ be a semi-malicious adversary that chooses $\hat{x}_2$ as an efficiently computable randomised function $f(c, x_2)$ but outputs $(x_1, \hat{x}_2)$ as an output. For simplicity, assume that the input of party $\mathcal{P}_1$ consists only form the protocol input x_1 and thus $\mathcal{P}_1$ outputs $(x_1, \hat{x}_2)$.

   (a) Compute the corresponding output distribution if the initial input distribution $\mathfrak{D}$ is a uniform distribution over $\mathbb{Z}_3 \times \mathbb{Z}_3$ and

   $$\alpha(x_1, x_2, \hat{x}_2) = \Pr\left[\mathsf{pk} \leftarrow \mathsf{Gen}, (c, d) \leftarrow \mathsf{Com}_{\mathsf{pk}}(x_1) : f(c, x_2) = \hat{x}_2\right]$$

   is independent form x_2. To get concrete results, fix some concrete values for the α table.

   (b) Compute the output distribution in the ideal world under the assumption that the commitment scheme is perfectly hiding and we use a naive simulation construction

   $\mathcal{S}^{\mathcal{P}_2^*}(x_2)$
   $\lceil$ $\mathsf{pk} \leftarrow \mathsf{Gen}$
   $(c, d) \leftarrow \mathsf{Com}_{\mathsf{pk}}(0)$
   Send $\hat{x}_2 \leftarrow \mathcal{P}_2^*(\sigma_2, x_2, c)$ to $\mathcal{T}$.
   Given $x_1, \hat{x}_2$ from $\mathcal{T}$ set $(c, d) \leftarrow \mathsf{Com}_{\mathsf{pk}}(x_1)$.
   Rewind $\mathcal{P}_2^*$ and output whatever $\mathcal{P}_2^*$ on pk, c, d does. $\rfloor$

What is the corresponding statistical difference between real and ideal world implementations?

3. Consider the implementation of simultaneous broadcast primitive detailed in the previous exercise. For simplicity and concreteness, let the input distribution and $\mathcal{P}_2^*$ be the same as in the previous exercise.

   (a) Compute the distance between ideal and real world distributions if we use an inefficient but perfect simulator

   $\mathcal{S}_\circ^{\mathcal{P}_2^*}(x_2)$
   - $\mathsf{pk} \leftarrow \mathsf{Gen}$
   - $(c, d) \leftarrow \mathsf{Com}_{\mathsf{pk}}(0)$
   - Send $\hat{x}_2 \leftarrow \mathcal{P}_2^*(\sigma_2, x_2, c)$ to $\mathcal{T}$.
   - Given $x_1, \hat{x}_2$ from $\mathcal{T}$, generate all runs between $\mathcal{P}_1(x_2)$ and $\mathcal{P}_2(x_2)$:
     - Choose randomness $\omega_1 \leftarrow \Omega_1$ and $\omega_2 \leftarrow \Omega_2$ for $\mathcal{P}_1$ and $\mathcal{P}_2^*$.
     - Run the protocol and store the output $(x_1, \overline{x}_2)$ of $\mathcal{P}_2^*$ into a list $\mathcal{L}$.
   - Let $\mathcal{L}_{\hat{x}_2}$ be the list of pairs $(x_1, \overline{x}_2) \in \mathcal{L}$ such that $\hat{x}_2 = \overline{x}_2$.
   - Choose uniformly a pair for the list $\mathcal{L}_{\hat{x}_2}$ and output it.

   Show that if the commitment scheme is perfectly hiding then the output distributions in the real and ideal world coincide.

   (b) Show that the simulator

   $\mathcal{S}^{\mathcal{P}_2^*}(x_2)$
   - $\mathsf{pk} \leftarrow \mathsf{Gen}$
   - $(c, d) \leftarrow \mathsf{Com}_{\mathsf{pk}}(0)$
   - Send $\hat{x}_2 \leftarrow \mathcal{P}_2^*(x_2, c)$ to $\mathcal{T}$.
   - Given $x_1, \hat{x}_2$ from $\mathcal{T}$ rewind until success.
     - $(c, d) \leftarrow \mathsf{Com}_{\mathsf{pk}}(x_1)$
     - If $\mathcal{P}_2^*(x_2, c) \neq \hat{x}_2$ repeat the cycle.
   - Output whatever $\mathcal{P}_2^*$ does.

   provides same output distribution as the simulator $\mathcal{S}_\circ$ provided that $\mathcal{S}$ stops before the time-bound t_{id}. Compute the failure probability for a fixed value of t_{id} and estimate the final statistical distance between real and ideal world distributions.

4. The Blum coin-fipping protocol is very similar to the simultaneous broadcast protocol described and analysed in previous exercises.

   (a) This resemblance is not a coincidence. Prove that given an ideal implementation of addition protocol, it is trivial to implement coin-flipping protocol. Also, prove that given an ideal addition protocol it

is trivial to implement simultaneous broadcast protocol, where both parties learn their inputs and vice versa.

(b) Construct a coin-flipping protocol from the ideal simultaneous broadcast protocol and substitute the ideal implementation with the protocol analysed in previous exercises. Compare the end result with the description of the Blum protocol:

- $\mathcal{P}_1$ generates $b_1 \xleftarrow{u} \{0,1\}$, computes $\mathsf{pk} \leftarrow \mathsf{Gen}$, $(c,d) \leftarrow \mathsf{Com}_{\mathsf{pk}}(b_1)$ and sends pk, c to $\mathcal{P}_2$.
- $\mathcal{P}_2$ generates $b_2 \xleftarrow{u} \{0,1\}$ to $\mathcal{P}_1$ who after that releases d to $\mathcal{P}_1$.
- $\mathcal{P}_2$ computes $b_1 \leftarrow \mathsf{Open}_{\mathsf{pk}}(c,d)$ and both parties output $b_1 \oplus b_2$.

5. Let $\pi_{\circledast}$ denote the the coin-flipping protocol that uses an ideal simultaneous broadcast primitive and π the Blum protocol.

(a) Construct a simulator for $\mathcal{P}_1^*$ for the protocol $\pi_{\circledast}$. Next, modify the simulator so that it works with the Blum coin-flipping protocol. For that recall that the simulator for $\mathcal{P}_1^*$ for the simultaneous broadcast protocol consists of an input extractor block

$\mathcal{K}^{\mathcal{P}_1^*}(\sigma_1, x_1)$

- Generate randomness $\omega_1 \leftarrow \Omega_1$ for $\mathcal{P}_1^*$.
- $(\mathsf{pk}, c) \leftarrow \mathcal{P}_1^*(\sigma_1, x_1; \omega_1)$
- Use rewinding to get
  - $d_0 \leftarrow \mathcal{P}_1^*(0), d_1 \leftarrow \mathcal{P}_1^*(1)$,
- Reveal the actual inputs:
  - $\hat{x}_1^0 \leftarrow \mathsf{Open}_{\mathsf{pk}}(c, d_0)$
  - $\hat{x}_1^1 \leftarrow \mathsf{Open}_{\mathsf{pk}}(c, d_1)$
- If $\bot \neq \hat{x}_1^0 \neq \hat{x}_1^1 \neq \bot$ then output $\mathsf{double\text{-}opening}$.
- If $\hat{x}_1^0 \neq \bot$ output $\hat{x}_1^0$ else output $\hat{x}_1^0$.

followed by the protocol simulation block

$\mathcal{S}_{\circ}^{\mathcal{P}_1^*}(x_1, \omega_1, x_2)$

- $(\mathsf{pk}, c) \leftarrow \mathcal{P}_1^*(\sigma_1, x_1; \omega_1)$
- $d \leftarrow \mathcal{P}_1^*(x_2)$
- If $\mathsf{Open}_{\mathsf{pk}}(c,d) = \bot$ then order $\mathcal{T}$ to halt the computations.
- Output whatever $\mathcal{P}_1^*$ outputs.

(b) Analyse the quality and running time of these simulators. Show that both achieve perfect simulation of the output distributions, i.e., the real and ideal world distributions coincide.

6. Let $\pi_{\circledast}$ denote the the coin-flipping protocol that uses an ideal simultaneous broadcast primitive and π the Blum protocol.

   (a) Construct a simulator for $\mathcal{P}_2^*$ for the protocol $\pi_{\circledast}$. Next, modify the simulator so that it works with the Blum coin-flipping protocol. For that recall that the simulator for $\mathcal{P}_1^*$ for the simultaneous broadcast protocol consists of an input extractor block

$$
\begin{array}{l}
\mathcal{K}^{\mathcal{P}_2^*}(\sigma_2, x_2) \\
\left[\begin{array}{l}
\text{Generate randomness } \omega_2 \leftarrow \Omega_2 \text{ for } \mathcal{P}_2^*. \\
\mathsf{pk} \leftarrow \mathsf{Gen} \\
(c, d) \leftarrow \mathsf{Com}_{\mathsf{pk}}(0) \\
\text{Return } \hat{x}_2 \leftarrow \mathcal{P}_2^*(x_2, c).
\end{array}\right.
\end{array}
$$

   followed by the protocol simulation block

$$
\begin{array}{l}
\mathcal{S}_{\circ}^{\mathcal{P}_2^*}(x_2, \omega_2, x_1, \hat{x}_2) \\
\left[\begin{array}{l}
\text{Rewind until success.} \\
\quad \left[\begin{array}{l}
(c, d) \leftarrow \mathsf{Com}_{\mathsf{pk}}(x_1) \\
\text{If } \mathcal{P}_2^*(x_2, c; \omega_2) \neq \hat{x}_2 \text{ repeat the cycle.}
\end{array}\right. \\
\text{Output whatever } \mathcal{P}_2^* \text{ does.}
\end{array}\right.
\end{array}
$$

   (b) Analyse the quality and running time of these simulators. Show that the real and ideal world distributions are statistically close if the number of rewinds is high enough.