MTAT.07.003 Cryptology II
Spring 2008 / Homework 7

1. Consider the standard simulators $\mathcal{S}_1$ and $\mathcal{S}_2$ for the Blum coin flipping protocol that were described in the lecture. Assume that initial states of parties ϕ_1 and ϕ_2 are empty, i.e., the both parties start from a scratch. Estimate the real and ideal world output distributions $\boldsymbol{\psi}$ and $\boldsymbol{\psi}^\circ$ under the following conditions.

    (a) The commitment scheme is totally non-hiding, e.g. $(m, m) \leftarrow \mathsf{Com}(m)$, and the party $\mathcal{P}_2^*$ always sets $b_2 = b_1$ and outputs $\psi_2 = b_1 \oplus b_2$.

    (b) The commitment scheme is totally non-binding, e.g. $(0, m) \leftarrow \mathsf{Com}(m)$, and the party $\mathcal{P}_1^*$ sets $b_1 = 1 \oplus b_2$ and chooses $\psi_1 \xleftarrow[u]{} \{0, 1\}$.

    (c) The commitment scheme is perfectly hiding and binding but the party $\mathcal{P}_1^*$ always halts if $b_1 \oplus b_2 = 0$.

    (d) The commitment scheme is $(k \cdot t, \varepsilon_1)$-hiding and (t, ε_2)-binding and both parties always output $\psi_i \in \{0, 1\}$.

2. Consider the parallel composition of the Blum coin flipping protocol discussed in the lecture. Let ℓ denote the number of protocols.

    (a) Construct the simulators for malicious participants $\mathcal{P}_1^*$ and $\mathcal{P}_2^*$. You may mimic the construction of the simulator for the Blum protocol. That is, $\mathcal{S}_1$ should provide all possible replies $b_2^1, \ldots, b_2^\ell$ to extract $b_1^1, \ldots, b_1^\ell$ that correspond to the commitments $c_1, \ldots, c_\ell$. Similarly, $\mathcal{S}_2$ can repeat the original protocol until $\boldsymbol{b}_1 \oplus \boldsymbol{b}_2 = \boldsymbol{y}$.

    (b) Assume that the commitment scheme is (t, ε_1)-extractable and (t, ε_2)-equivocable, i.e. there exists a modified setup procedure $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}^*$ such that we can use $\mathsf{Extr}_{\mathsf{sk}}(\cdot)$ to break the hiding property and $\mathsf{Com}^*_{\mathsf{sk}}()$ and $\mathsf{Equiv}_{\mathsf{sk}}(\cdot)$ functions to break the binding property. Construct the corresponding simulators $\mathcal{S}_1$ and $\mathcal{S}_2$ for the original protocol and the corresponding parallel compositions.

    Estimate the statistical distance between the real and ideal world outputs.

3. Consider the following entity authentication protocol proposed by Bellare and Rogaway. In the MAP-1 protocol, parties $\mathcal{P}_1$ and $\mathcal{P}_2$ share the secret key $k \xleftarrow[u]{} \mathcal{K}$ of a (t, ε)-pseudorandom function $f : \{0, 1\}^* \times \mathcal{K} \to \mathcal{T}$.

    1. $\mathcal{P}_1$ sends a random nonce $r_1 \xleftarrow[u]{} \mathcal{R}$ to $\mathcal{P}_2$.
    2. $\mathcal{P}_2$ generates a random nonce $r_2 \xleftarrow[u]{} \mathcal{R}$ and sends the identities $\mathsf{id}_1, \mathsf{id}_2$, nonces r_1, r_2 and the authentication tag $f(\mathsf{id}_1 \| \mathsf{id}_2 \| r_1 \| r_2, k)$ to $\mathcal{P}_1$.
    3. $\mathcal{P}_1$ replies id_1, r_b and the authentication tag $f(\mathsf{id}_1 \| r_b, k)$ to $\mathcal{P}_2$.

    Parties $\mathcal{P}_1$ and $\mathcal{P}_2$ halt if the received messages are not in correct form. Otherwise, both parties are convinced that they are indeed talking with each other. Consider the security of Map-1 protocol in the standalone setting, where $\mathcal{P}_1$ and $\mathcal{P}_2$ run a single instance of the protocol by sending messages through the adversary $\mathcal{A}$ who can alter, drop or insert messages into the conversation. The adversary $\mathcal{A}$ succeeds in deception if both parties reach accepting state but the adversary has altered some messages.

(a) Estimate the probability that the adversary $\mathcal{A}$ sends $\hat{r}_1 \neq r_i$ to $\mathcal{P}_2$ and still succeeds in deception.

(b) Estimate the probability that the adversary $\mathcal{A}$ sends $(\hat{\mathsf{id}}_1, \hat{\mathsf{id}}_2, \hat{r}_1, \hat{r}_2) \neq (\mathsf{id}_1, \mathsf{id}_2, r_1, r_2)$ to $\mathcal{P}_1$ and still succeeds in deception.

(c) Estimate the probability that $\mathcal{A}$ sends $(\hat{\mathsf{id}}_1, \hat{r}_2) \neq (\mathsf{id}_1, r_2)$ to $\mathcal{P}_1$ and still succeeds in deception.

(d) Summarise the results and give the final bound on deception.

4. Consider the security of the Map-1 protocol in the the strongest model, where the adversary $\mathcal{A}$ can force parties $\mathcal{P}_1$ and $\mathcal{P}_2$ to start as many instances of authentication protocols as he or she likes.

   (a) Formalise the notion of deception so that it provides the strongest possible security guarantees. As an hint, note that in the ideal implementation the adversary can either transfer all messages without changes or stop the protocol instance. The same original message should not be used in several protocol instances.

   (b) Show that each protocol instance is determines unique nonces r_1 and r_2 and estimate the corresponding collision probability.

   (c) Give the final bound on the deception under the assumption that there are no nonce collisions. Compute the final deception bound.

5. The Kerberos protocol is uses a trusted key generation server $\mathcal{T}$ to set up shared keys between participants $\mathcal{P}_1, \ldots, \mathcal{P}_n$. Let $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a IND-CCA2 secure symmetric cryptosystem. Then in a setup phase, each party $\mathcal{P}_i$ shares a secret key $\mathsf{sk}_i \leftarrow \mathsf{Gen}$ with the trusted server $\mathcal{T}$. To set up a new session key $\mathsf{sk}_{ij} \leftarrow \mathsf{Gen}$ between $\mathcal{P}_i$ and $\mathcal{P}_j$, the parties $\mathcal{P}_1$, $\mathcal{P}_2$ and $\mathcal{T}$ execute the following protocol.

   1. $\mathcal{P}_i$ sends $\mathsf{id}_i, \mathsf{id}_j$ and a random nonce $r_1 \xleftarrow[u]{} \mathcal{R}$ to the server $\mathcal{T}$.
   2. $\mathcal{T}$ generates a new session key $\mathsf{sk}_{ij} \leftarrow \mathsf{Gen}$ and sends back:

$$\begin{aligned} \mathsf{ticket} &\leftarrow \mathsf{Enc}_{\mathsf{sk}_j}(\mathsf{sk}_{ij}, \mathsf{id}_i, \text{expiration time}) \enspace , \\ \mathsf{enc\text{-}info} &\leftarrow \mathsf{Enc}_{\mathsf{sk}_i}(\mathsf{sk}_{ij}, r_1, \text{expiration time}, \mathsf{id}_j) \enspace . \end{aligned}$$

   3. $\mathcal{P}_i$ decrypts $\mathsf{enc\text{-}info}$ creates another nonce $r_2 \xleftarrow[u]{} \mathcal{R}$ and sends ticket and $\mathsf{Enc}_{\mathsf{sk}_{ij}}(\mathsf{id}_i, r_2)$ to $\mathcal{P}_j$, who replies $\mathsf{Enc}_{\mathsf{sk}_{ij}}(r_2)$.

   Participants halt if some messages are not in expected form. An adversary $\mathcal{A}$ succeeds in deception if either $\mathcal{P}_1$ or $\mathcal{P}_2$ reach the accepting state but one of them has a fraudulent output.

(a) Estimate the probability that $\mathcal{P}_i$ accepts altered $\mathsf{enc\text{-}info}$.

(b) Estimate the probability that $\mathcal{P}_j$ accepts altered ticket.

(c) Estimate the probability that $\mathcal{P}_j$ halts but $\mathcal{P}_i$ accepts.

(d) Give the final bound on the deception probability.

6. Let $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be (t, ε)-IND-CCA2 secure cryptosystem such that the message space $\mathcal{M}$ is an additive group. Then the classical challenge-response protocol for proving the possession of sk:

   1. The verifier $\mathcal{V}$ chooses $m \xleftarrow[u]{} \mathcal{M}$ and sends $\mathsf{Enc}_{\mathsf{pk}}(m)$ to the prover $\mathcal{P}$.
   2. Given a challenge c, the prover $\mathcal{P}$ replies $\overline{m} \leftarrow \mathsf{Dec}_{\mathsf{sk}}(c)$.
   3. The verifier $\mathcal{V}$ accepts if $m = \overline{m}$ to $\mathcal{V}$.

   is also $(t, \frac{1}{|M|} + \varepsilon)$-secure in the most powerful attack scenario. Prove that the following AND and OR compositions are also secure.

   AND composition for secret keys sk_1 and sk_2:

   1. The verifier $\mathcal{V}$ chooses $m_1, m_2 \xleftarrow[u]{} \mathcal{M}$ and sends two challenges $\mathsf{Enc}_{\mathsf{pk}_1}(m_1)$ and $\mathsf{Enc}_{\mathsf{pk}_2}(m_2)$ to the prover $\mathcal{P}$.
   2. Given challenge ciphertexts c_1, c_2, the prover $\mathcal{P}$ uses both secret keys sk_1 and sk_2 and replies $\overline{m}_1 \leftarrow \mathsf{Dec}_{\mathsf{sk}_1}(c)$ and $\overline{m}_2 \leftarrow \mathsf{Dec}_{\mathsf{sk}_2}(c)$ to $\mathcal{V}$.
   3. The verifier $\mathcal{V}$ accepts if $m_1 = \overline{m}_1$ and $m_2 = \overline{m}_2$.

   OR composition for secret keys sk_1 and sk_2:

   1. The verifier $\mathcal{V}$ chooses $m \xleftarrow[u]{} \mathcal{M}$ and sends the corresponding challenge pair $\mathsf{Enc}_{\mathsf{sk}_1}(m; r_1)$ for $r_1 \leftarrow \mathcal{R}$, and $\mathsf{Enc}_{\mathsf{sk}_2}(m; r_2)$ for $r_2 \leftarrow \mathcal{R}$ to $\mathcal{P}$.
   2. Given challenge ciphertexts c_1, c_2, the prover $\mathcal{P}$ uses one of the secret keys sk_1 and sk_2 to decrypt a challenge and then uses a commitment scheme to create a commitment to the answer $\overline{m}$.
   3. The verifier $\mathcal{V}$ sends m, r_1 and r_2 to the prover $\mathcal{P}$ who recomputes ciphertext c_1 and c_2 to make sure that they are indeed encryptions of the same message. If not the prover $\mathcal{P}$ halts, otherwise the prover $\mathcal{P}$ reveals the decommitment value.
   3. The verifier $\mathcal{V}$ opens the commitment and verifies that $\overline{m} = m$.

   More precisely, prove the following facts and provide the corresponding security guarantees.

   (a) The verifier $\mathcal{V}$ cannot cheat in the OR composition.

   (b) The AND and OR compositions are functional and secure in the most powerful attack scenario.

   (c) The verifier cannot distinguish whether the prover knows the secret key sk_1 or sk_2.

   (d) How to generalise this approach for any monotone formula? What could be the potential applications?