MTAT.07.003 Cryptology II
Spring 2010 / Exercise Session VIII

1. Entity authentication protocols are often used to prove liveness of a device or a person. For instance, ATM machines normally ask PIN codes several times during long transactions to assure that the person is still present. Such liveness proofs can be implemented with one-way functions.

   (a) Let $f : \mathcal{X} \to \mathcal{Y}$ be a one-way function and let n be the maximal number of protocol invocations. Then a secret key sk can be chosen as a tuple of random values $x_1, \ldots x_n \xleftarrow{u} \mathcal{X}$ and the corresponding public key pk as a tuple of hash values $f(x_1), \ldots, f(x_n)$.

   Each time when a party wants to prove liveness he or she will release non-published sub-key x_i. The proof is successful if $f(x_i) \stackrel{?}{=} y_i$ where y_i is the ith component of the public key pk.

   Prove that if f is (t, ε_1)-secure one-way function and protocols are executed sequentially, then the probability that a t-time adversary succeeds in the ith authentication without seeing x_i is at most ε.

   (b) Consider a modification of the scheme described above, where $f : \mathcal{X} \to \mathcal{X}$ is a permutation and in the setup phase the party computes an hash chain $x_n \xleftarrow{u} \mathcal{X}$, $x_i \leftarrow f(x_{i+1})$ for $i \in \{1, \ldots, n-1\}$ and publishes $x_0 \leftarrow f(x_1)$ as a public key. In liveness proofs, sub-keys x_i are released one-by-one as before. The proof is valid if $x_{i-1} = f(x_i)$.

   Prove that (t, ε)-secure one-way permutation protocols are executed sequentially, then the probability that a t-time adversary succeeds in the ith authentication without seeing x_i is at most ε.

   How large can be the success probability of a t-time adversary that can attack any of these liveness proofs?

2. The sizes of the public and private of the liveness proofs described in (a) part of the previous exercise grow linearly wrt to the maximal number of invocations. The solution (b) requires a linear amount of work wrt n if we do not store the intermediate states x_i or some liveness proofs are missing.

   (a) Show that we can use a (t, ε_2)-collision resistant hash function family to compact the public key. Describe the corresponding compaction procedure and the resulting proof. Estimate the size of proofs and the amount of time needed to compute them.

   Hint: Binary trees provide an optimal hashing scheme.

   (b) Show that we can use (t, ε_3)-pseudorandom function family $\mathcal{F}$ to compact also the private key sk. Describe the corresponding scheme and recompute the security guarantees.

   Hint: How to stretch randomness in a most optimal way?

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 $f : \{0,1\}^* \times \mathcal{K} \to \mathcal{T}$. More formally, the induced function family $\mathcal{F} \doteq \{f_k\}$ is (t, ε)-pseudorandom.

   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_2 and the authentication tag $f(\mathsf{id}_1 \| r_2, k)$ 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. Analyse 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) Formalise the execution of MAP-1 protocol as a game that ends with 1 iff $\mathcal{A}$ succeeds in deception. Note that $\mathcal{A}$ does not have o respect the temporal order. For example, $\mathcal{A}$ can transfer $\hat{r}_1$ to $\mathcal{P}_1$ before $\mathcal{P}_1$ has released r_1. 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. 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:

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

   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 enc-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.

5. 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 is following:

   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 $\mathcal{V}$.

   However, sometimes one needs to prove that he or she possesses two different secret keys or only one of them. The corresponding proofs are known as conjunctive and disjunctive proofs.

   Conjunctive proof for secret keys sk_0 and sk_1:

   1. The verifier $\mathcal{V}$ chooses $m_0, m_1 \xleftarrow{u} \mathcal{M}$ and sends challenge $\mathsf{Enc}_{\mathsf{pk}_0}(m_0)$ and $\mathsf{Enc}_{\mathsf{pk}_1}(m_1)$ to the prover $\mathcal{P}$.
   2. Given challenge ciphertexts c_0, c_1, the prover $\mathcal{P}$ uses both secret keys sk_0 and sk_1 and replies $\overline{m}_0 \leftarrow \mathsf{Dec}_{\mathsf{sk}_0}(c_0)$ and $\overline{m}_1 \leftarrow \mathsf{Dec}_{\mathsf{sk}_1}(c_1)$ to $\mathcal{V}$.
   3. The verifier $\mathcal{V}$ accepts if $m_0 = \overline{m}_0$ and $m_1 = \overline{m}_1$.

   Disjunctive proof for secret keys sk_0 and sk_1:

   1. The verifier $\mathcal{V}$ chooses $m \xleftarrow{u} \mathcal{M}$ and sends the corresponding challenge $\mathsf{Enc}_{\mathsf{sk}_0}(m; r_0)$ for $r_0 \leftarrow \mathcal{R}$, and $\mathsf{Enc}_{\mathsf{sk}_1}(m; r_1)$ for $r_1 \leftarrow \mathcal{R}$ together with encryptions of random nonces $\mathsf{Enc}_{\mathsf{sk}_0}(r_1)$ and $\mathsf{Enc}_{\mathsf{sk}_1}(r_0)$ to $\mathcal{P}$.
   2. Given challenge ciphertexts c_1, c_2, c_3, c_4, the prover $\mathcal{P}$ uses one of the secret keys sk_i to decrypt a challenge $\overline{m}$ and the nonce $r_{\neg i}$ used to randomise the other encryption $c_{\neg i}$. If $c_{\neg i} = \mathsf{Enc}_{\mathsf{pk}_{\neg i}}(\overline{m}; r_{\neg i})$, the prover $\mathcal{P}$ sends $\overline{m}$ to $\mathcal{V}$, otherwise $\mathcal{P}$ can halt as $\mathcal{V}$ cheats.
   3. The verifier $\mathcal{V}$ accepts if $\overline{m} = m$.

   Consider a simple standalone setting, where a prover and a verifier execute a protocol to be analysed only once in isolation and prove the following facts about conjunctive and disjunctive proofs.

   (a) A prover can succeed in conjunctive proof only if he or she knows both secret keys and a prover fails in disjunctive proof if he or she does not know neither of the secret keys.

(b) Even a malicious verifier cannot reliably detect which secret key is known by the honest prover.

6. Conjunctive and disjunctive proofs of possession can be used as a building-blocks for more complex relations.

   (a) Construct a proof of possession, where the prover has to have at least two secret keys out of three.

   (b) Generalise this construction and show that for any monotone formula $\phi : \{0,1\}^n \to \{0,1\}$. More precisely, let the ith bit of $x \in \{0,1\}^n$ denote whether a prover has a secret key sk_i. Show that there exists a proof of possession, where a prover succeeds only if $\phi(x) = 1$.