MTAT.07.003 Cryptology II
Spring 2008 / Homework 6

1. To minimise memory footprint in servers, operational information is often stored by clients and provided on demand. Web cookies are the most famous example. Such a storage strategy opens up new attack vectors, since malicious clients can provide inconsistent data that might lead to crashes or a malicious code injection.

   (a) Design simple integrity tests based on collision resistant hash function if the stored data is always used as a single unit.

   (b) Provide a solution if stored data is structured and only few substructures are used in each operation. For example, the entire file system is stored at client cite who can potentially alter it.

   (c) Design a data protection model for BitTorrent like application, where the data is hosted by many potentially malicious sub-servers and a client assembles the entire file by combining the data streams.

   (d) The MD5 hash function was recently shown to be weak, i.e., it is possible to find collisions. However, the attacker cannot control the values of colliding messages. Are now all integrity protection mechanisms based on MD5 insecure or not?
   Clarification: The MD5 hash function is iterative

   $$f^*(m_1, \dots m_n) = f(f(\cdots f(f(\mathsf{iv}, m_1), m_2), \dots, m_{n-1}), m_n)$$

   where $f : \mathcal{T} \times \mathcal{M} \to \mathcal{T}$ is a dedicated compression function.

   Compute all corresponding security guarantees provided that the hash function is sampled form the (t, ε)-collision-resistant function family $\mathcal{H}$.

2. There are several other properties that hash function families can have besides collision resistance.

   - A hash function family $\mathcal{H}$ is (t, ε)-secure one-way function family if for any t-time adversary $\mathcal{A}$

     $$\Pr\left[h \xleftarrow{u} \mathcal{H}, m \xleftarrow{u} \mathcal{M} : \mathcal{A}(h, h(m)) = m\right] \leq \varepsilon \ .$$

   - A hash function family $\mathcal{H}$ is (t, ε)-secure against second preimage if for any t-time adversary $\mathcal{A}$

     $$\Pr\left[h \xleftarrow{u} \mathcal{H}, m \xleftarrow{u} \mathcal{M}, m_1 \leftarrow \mathcal{A}(h, m_0) : m_0 \neq m_1 \wedge h(m_0) = h(m_1)\right] \leq \varepsilon \ .$$

   Establish the corresponding homological classification of these three properties under the assumption that $\mathcal{H}$ is a compressing function family. Provide the corresponding reductions.

   (a) Show that collision resistance implies security against second preimage attacks.

(b) Show that security against second preimage attacks implies one-wayness.

(c) Give interpretation to all three properties. Is the MD5 function still secure against second preimage attacks?

($\star$) Give the corresponding separations that show that the corresponding inclusions are strict under the assumption that $\mathcal{H}$ is compressing function family.

3. The main drawback of the modified Naor commitment scheme is message expansion—to commit one bit we must send n bits. One possibility is to increase the size of the message space. Let the message space $\mathcal{M}$ be a subset of a finite field $(\mathbb{F}; +, \times)$ such that we can treat all elements of $\mathbb{F}$ as n-bit strings. Then we can define modified commitment scheme:

$$\begin{array}{lll}
\mathsf{Gen} & \mathsf{Com}_{\mathsf{pk}}(x) & \mathsf{Open}_{\mathsf{pk}}(c, d) \\
\left[\begin{array}{l} \mathsf{pk} \xleftarrow{u} \mathbb{F}^* \\ \mathsf{return}\ \mathsf{pk} \end{array}\right. & \left[\begin{array}{l} d \leftarrow \{0,1\}^k \\ c \leftarrow f(d) \oplus x \cdot \mathsf{pk} \\ \mathsf{return}\ (c, d) \end{array}\right. & \left[\begin{array}{l} y \leftarrow c \oplus f(d) \\ \mathsf{if}\ y \notin \mathsf{pk} \times \mathcal{M}\ \mathsf{then\ return}\ \perp \\ \mathsf{else\ return}\ y \times \mathsf{pk}^{-1} \end{array}\right.
\end{array}$$

Establish the corresponding security guarantees under the assumption that $f : \{0,1\}^k \to \{0,1\}^n$ is a (t_1, ε_1)-pseudorandom generator.

How big must be the message space $\mathcal{M} \subseteq \mathbb{F}$ to achieve reasonable security guarantees against double openings?

Hint: How many decommitment pairs can lead to a double opening? How is this number related to the size of $\mathbb{F}$ and $\mathcal{M}$?

4. Another way to improve the modified Naor commitment scheme is to use a collision resistant hashing to build a list commitment scheme on top of the ordinary commitment scheme:

$$\begin{array}{ll}
\mathsf{Gen} & \mathsf{Com}_{\mathsf{pk},h}(x_1, \ldots, x_\ell) \\
\left[\begin{array}{l} \mathsf{pk} \xleftarrow{u} \{0,1\}^n \\ h \xleftarrow{u} \mathcal{H} \\ \mathsf{return}\ (\mathsf{pk}, h) \end{array}\right. & \left[\begin{array}{l} (c_i, d_i) \leftarrow \mathsf{Naor\text{-}Com}_{\mathsf{pk}}(x_i),\ i = 1, \ldots, \ell \\ c_* \leftarrow h(c_1, \ldots, c_\ell) \\ \mathsf{return}\ (c_*, (c_1, \ldots, c_\ell, d_1, \ldots, d_\ell)) \end{array}\right.
\end{array}$$

where the decommitment procedure just verifies $c_* = h(c_1, \ldots, c_\ell)$ and restores $x_i \leftarrow \mathsf{Open}_{\mathsf{pk}}(c_i, d_i)$ for $i = 1, \ldots, \ell$.

(a) Establish security guarantees under the assumption that the basic commitment scheme is (t_1, ε_1)-hiding and (t_2, ε_2)-binding and $\mathcal{H}$ is a (t_3, ε_3)-collision resistant hash function family.

(b) Can we use a pseudorandom generator f for compacting the decommitment? What happens if we generate $d_0, \ldots, d_{\ell-1}$ by stretching a single master seed d_*? Provide corresponding security guarantees.

(c) Modify the compaction strategy so that it is possible to open individual bits without leaking information about the others.

5. One of the most elegant properties of additively homomorphic commitments is the ability to do verifiable shuffling. As an example consider the following card shuffling protocol:

   $\mathcal{P}_1$ generates a random permutation $\pi : \{1, \ldots, 36\} \to \{1, \ldots, 36\}$. Let P be the corresponding 36×36 zero-one matrix such that $\pi(\boldsymbol{y}) = P\boldsymbol{y}$ for any n-element vector $\boldsymbol{y}$ and let $(c_{ij}, d_{ij}) \leftarrow \mathsf{Com}_{\mathsf{pk}}(p_{ij})$. Next, $\mathcal{P}_1$ sends the matrix of commitments c_{ij} to $\mathcal{P}_2$.

   $\mathcal{P}_2$ computes randomly shuffled card pack. First $\mathcal{P}_1$ chooses a random permutation $x_1, \ldots, x_{36}$ of the set $\{1, \ldots, 36\}$. Next, $\mathcal{P}_2$ computes

   $$e_i \leftarrow c_{i1}^{x_1} \cdot c_{i2}^{x_2} \cdots c_{in}^{x_n} c_i^* \ ,$$

   where $(c_i^*, d_i^*) \leftarrow \mathsf{Com}_{\mathsf{pk}}(0)$, and sends $e_1, \ldots, e_n$ to $\mathcal{P}_2$.

   (a) Prove that the values $e_1, \ldots, e_n$ are indeed randomly shuffled commitments of $x_1, \ldots, x_n$.

   (b) Prove that neither $\mathcal{P}_1$ not $\mathcal{P}_2$ cannot guess where is the commitment to 36 among $e_1, \ldots, e_n$ if commitment is (t, ε)-hiding.

   (c) Prove that $\mathcal{P}_1$ and $\mathcal{P}_2$ can release cards one by one and one can detect cheating in the release phase if commitment scheme is (t, ε_1)-binding.

   (d) How $\mathcal{P}_1$ can prove that c_{ij} are indeed commitments to the permutation matrix under the assumption that c_{ij} are guaranteed to be commitments of zeros or ones?

   Hint: Can one characterise permutation matrices in terms of row and column sums.

   ($\star$) Use cut-and-choose techniques to make the protocol secure against malicious corruption in the dealing phase.

6. Consider the following simple user-aided key agreement protocol. The public key pk of a server $\mathcal{P}_1$ is known to all participants. If a participant $\mathcal{P}_2$ wants to connect to $\mathcal{P}_1$ it generates a random session key $k \xleftarrow{u} \mathcal{K}$ and a short authentication nonce $r \xleftarrow{u} \{0, \ldots, 9999\}$ and sends $\mathsf{Enc}_{\mathsf{pk}}(k \| r)$ to $\mathcal{P}_1$. The server $\mathcal{P}_1$ recovers k and r and sends r as an SMS to the client $\mathcal{P}_2$. The client $\mathcal{P}_2$ halts if the SMS does not correspond to his or her authentication nonce. Prove that the protocol is secure under the assumptions that the cryptosystem is IND-CCA2 secure and no adversary can alter the SMS. The adversary can alter the ciphertext.

7. To prove that the Blum coin flipping protocol between parties $\mathcal{P}_1$ and $\mathcal{P}_2$ is secure in a strong sense, we need to construct code wrappers for the ideal world that simulate the missing real world messages. For clarity, let $\mathcal{P}_1^*$

and $\mathcal{P}_2^*$ denote potentially malicious participants and let ω_1 and ω_2 denote the random coins used by them. Let the simulators be the following

$\mathcal{S}_1^{\mathcal{P}_1^*}(y, \mathsf{pk})$

- $\omega_1 \leftarrow_u \Omega_1,\ c \leftarrow \mathcal{P}_1^*(\mathsf{pk}; \omega_1)$
- $d_0 \leftarrow \mathcal{P}_1^*(0; \omega_1),\ d_1 \leftarrow \mathcal{P}_1^*(1; \omega_1)$
- $b_1^0 \leftarrow \mathsf{Open}_{\mathsf{pk}}(c, d_0),\ b_1^1 \leftarrow \mathsf{Open}_{\mathsf{pk}}(c, d_0)$
- if $\perp \neq b_1^0 \neq b_1^1 \neq \perp$ then Failure
- if $b_1^0 = \perp = b_1^1$ then
  - Send the Halt command to $\mathcal{T}$.
  - Choose $b_2 \leftarrow_u \{0,1\}$ and rerun the protocol with ω_1 and b_2.
  - Return whatever $\mathcal{P}_1^*$ returns.
- if $b_1^0 = \perp$ then $b_1 \leftarrow b_0^1$ else $b_1 \leftarrow b_1^0$
- $b_2 \leftarrow b_1 \oplus y$
- Rerun the protocol with ω_1 and b_2
- if $b_1^{b_2} = \perp$ then Send the Halt command to $\mathcal{T}$.
- Return whatever $\mathcal{P}_1^*$ returns.

$\mathcal{S}_2^{\mathcal{P}_2^*}(y, \mathsf{pk})$

- $\omega_2 \leftarrow_u \Omega_2$,
- For $i = 1, \ldots k$ do
  - $b_1 \leftarrow_u \{0,1\}$
  - $(c, d) \leftarrow \mathsf{Com}_{\mathsf{pk}}(b_i)$
  - $b_2 \leftarrow \mathcal{P}_2^*(\mathsf{pk}, c; \omega_2)$
  - if $b_1 \oplus b_2 = y$ then
    - Send d to $\mathcal{P}_2^*$ and output whatever $\mathcal{P}_2$ outputs.
- return Failure

Prove following the following statements under the assumption that the commitment scheme is $(k \cdot t, \varepsilon_1)$-hiding and (t, ε_2)-binding.

(a) The failure probability of $\mathcal{S}_1$ is bounded by ε_2 and the failure probability of $\mathcal{S}_2$ is bounded by $2^{-k} + \varepsilon_1$.

(b) Show that if y is chosen uniformly from $\{0,1\}$, then b_2 computed by $\mathcal{S}_1$ has a uniform distribution over $\{0,1\}$ and is independent of ω_1. Assume that b_2 has uniform distribution if $\mathcal{S}_1$ fails.

(c) Conclude that the output distributions of $\mathcal{S}_1^{\mathcal{P}_1}$ and $\mathcal{P}_2$ in the ideal world coincide with the real world outputs when $\mathcal{S}_1$ does not fail.

(d) Show that if y is chosen uniformly from $\{0,1\}$, then b_1 computed by $\mathcal{S}_2$ has a uniform distribution over $\{0,1\}$ and is independent of ω_2. Assume that b_1 has uniform distribution if $\mathcal{S}_1$ fails.

(e) Conclude that the output distributions of $\mathcal{P}_1$ and $\mathcal{S}_2^{\mathcal{P}_2}$ in the ideal world coincide with the real world outputs when $\mathcal{S}_2$ does not fail.

8. Usually, we need many random bits in the computations and thus one must repeat the Blum coin flipping protocol.

   (a) Construct the corresponding simulator $\mathcal{S}_2$ if the protocols are executed sequentially. The simulator must assure that

   $$b_1^1 \oplus b_2^1 = y_1, \ldots, b_1^\ell \oplus b_2^\ell = y_\ell$$

   for random bits $y_1, \ldots, y_\ell$ generated by the trusted third party $\mathcal{T}$.

   (b) Construct the corresponding simulator $\mathcal{S}_2$ for $\mathcal{P}_2^*$ if the protocols are executed in parallel, i.e. $\mathcal{P}_1$ sends commitments $c_1, \ldots, c_\ell$ as a single message. Why do we get an exponential blowup in the running time of $\mathcal{S}_2$. Does a similar inefficiency appear also for $\mathcal{S}_1$?

   (c) Construct a non-rewinding simulator $\mathcal{S}_2$ for the Blum protocol under the assumption that the commitment scheme is equivocable.

   Hint: Use the modified setup procedure $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}^*$ to avoid the need for rewinding.

   (d) Show that there exists an efficient simulator $\mathcal{S}_2$ for the parallel composition of Blum protocols if the commitment scheme is equivocable. Why could a commitment scheme that is extractable and equivocable simultaneously improve the situation even further?