Advanced Cryptography
University of Michigan, Winter 2016

Homework 6

Instructor: Chris Peikert
Student: SOLUTIONS

This homework is due by **7pm on April 8** via the course Canvas page. Start early!

Instructions. Solutions must be *typed*, preferably in LaTeX (a template for this homework is available on the course web page). Your work will be graded on *correctness*, *clarity*, and *concision*. You should only submit work that you believe to be correct; if you cannot solve a problem completely, you will get significantly more partial credit if you clearly identify the gap(s) in your solution. It is good practice to start any long solution with an informal (but accurate) "proof summary" that describes the main idea.

You may collaborate with others on this problem set and consult external sources. However, you must ***write your own solutions*** and ***list your collaborators/sources*** for each problem.

1. Do Exercise 11.17 in Katz-Lindell.

Solution: The scheme is not necessarily CCA-secure. The problem is that the underlying CPA-secure encryption scheme may be *malleable*, which makes the derived scheme malleable as well. Concretely, suppose that Enc of the underlying scheme always produces a ciphertext that ends with a zero bit, and the decryption algorithm Dec ignores this last bit of the ciphertext. It is trivial to construct a CPA-secure encryption scheme having this property from any CPA-secure scheme.

For such an underlying encryption scheme, we can mount a successful chosen-ciphertext attack on the derived scheme. First query the left/right oracle with distinct messages m_0, m_1, and given the challenge ciphertext $c = (c_1 = \mathsf{Enc}_{pk}(r), c_2 = \mathsf{Enc}'_{H(r)}(m_b))$, let c'_1 be c_1 with its last bit flipped (to one). Query the decryption oracle on $c' = (c'_1, c_2) \neq c$, which will return m_b. From this we learn b, meaning our attack has advantage 1.

2. Do Exercise 11.21 in Katz-Lindell.

Solution: First observe that $\mathsf{lsb}(x) = \mathsf{half}(2^{-1} \cdot x)$, where $2^{-1} \in \mathbb{Z}_N^*$ denotes the multiplicative inverse of two modulo N (which exists because N is odd). Indeed, if $\mathsf{lsb}(x) = 0$ then $2^{-1} \cdot x = \frac{x}{2} < \frac{N}{2}$, and if $\mathsf{lsb}(x) = 1$ then $2^{-1} \cdot x = \frac{x+N}{2} > \frac{N}{2}$.
Now let $\mathcal{A}$ be an efficient algorithm which given $N, e, y = x^e$ computes $\mathsf{half}(x)$ with some advantage. Using $\mathcal{A}$ we build an algorithm $\mathcal{A}'$ for computing $\mathsf{lsb}(x)$ with the same advantage: on input $N, e, y = x^e$ it simply outputs $\mathcal{A}(y \cdot (2^{-1})^e)$. We have:

$$\begin{aligned}\Pr_{N,e,x \leftarrow \mathbb{Z}_N^*}[\mathcal{A}'(x^e) = \mathsf{lsb}(x)] &= \Pr[\mathcal{A}((2^{-1} \cdot x)^e) = \mathsf{lsb}(x) = \mathsf{half}(2^{-1} \cdot x)] \\ &= \Pr_{N,e,x' \leftarrow \mathbb{Z}_N^*}[\mathcal{A}((x')^e) = \mathsf{half}(x')].\end{aligned}$$

Because lsb is a hard-core predicate for RSA, the left-hand side must be at most $1/2 + \operatorname{negl}(n)$, hence so is the right-hand side, which means that half is also a hardcore predicate for RSA.

3. Recall that for a (possibly randomized) algorithm G that takes input from X, we define $G^{\$}$ to be the *randomized* oracle that, when queried (on no input), chooses a fresh $x \leftarrow X$ uniformly at random and outputs $(x, G(x))$.

Let $\Pi = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ be a signature scheme. We say that Π is *randomly unforgeable under random-message attack* if the adversary, given oracle access to $\mathsf{Sign}^{\$}_{sk}$, has only negligible probability of outputting a forgery on a *randomly chosen* message that we provide it (along with the verification key). I.e., for any efficient $\mathcal{A}$,

$$\Pr_{(vk,sk)\leftarrow \mathsf{Gen}(1^n);\; m^*\leftarrow \mathcal{M}}[\mathcal{A}^{\mathsf{Sign}^{\$}_{sk}}(vk, m^*) \text{ outputs } \sigma^* \text{ s.t. } \mathsf{Ver}(vk, m^*, \sigma^*) \text{ accepts}] = \operatorname{negl}(n).$$

Prove that textbook RSA (construction 12.5 in Katz-Lindell) is randomly unforgeable under random message attack, under the RSA assumption.

Solution: Let $\mathcal{A}$ be an efficient adversary against random unforgeablity of textbook RSA under random message attack. Using $\mathcal{A}$ we build an adversary $\mathcal{A}'$ that solves the RSA problem with the same advantage as $\mathcal{A}$. Given an instance of the RSA problem (N, e, y), we define $\mathcal{A}'$ to work as follows:

- It provides (N, e) as the verification key and $m^* = y$ as the random challenge message to $\mathcal{A}$.
- Whenever $\mathcal{A}$ queries its oracle, $\mathcal{A}'$ chooses a random $\sigma \in \mathbb{Z}_n^*$ and returns $(m = \sigma^e, \sigma)$ as the message-signature pair.
- $\mathcal{A}'$ outputs whatever $\mathcal{A}$ outputs as the forged signature σ^* for message $m^* = y$.

To see that $\mathcal{A}$ perfectly simulates the random unforgeability under random message game, first notice that the challenge message $m^* = y$ is uniformly random in $\mathbb{Z}_N^*$, by construction. Secondly, because $\gcd(e, \phi(N)) = 1$, each response $(m = \sigma^e, \sigma)$ from the $\mathsf{Sign}^{\$}$ oracle is made up of a uniformly random $m = \sigma^e \in \mathbb{Z}_N^*$ and its correct signature $\sigma = m^{1/e} \in \mathbb{Z}_N^*$. Therefore,

$$\Pr[\mathcal{A}'(N, e, x^e) = x] = \Pr[\sigma^* \leftarrow \mathcal{A}^{\mathsf{Sign}^{\$}_{sk}}(vk, m^*) : (\sigma^*)^e = m^*],$$

where the first probability is over $(N, e, d) \leftarrow \mathsf{GenRSA}(1^n), x \leftarrow \mathbb{Z}_N^*$ and the second one is over $(vk = (N, e), sk = (N, d)) \leftarrow \mathsf{Gen}(1^n), m^* \leftarrow \mathbb{Z}_N^*$. By assumption on the RSA, the left-hand side is negligible, hence so is the right-hand side, and the proof is complete.

4. In class we saw that the DDH assumption can be used for public-key encryption; here you will show that it is very useful for *symmetric* primitives too.

For a cyclic group $G = \langle g \rangle$ of prime order q, recall that the DDH assumption says that

$$(g, g^a, g^b, g^{ab}) \overset{c}{\approx} (g, g^a, g^b, g^c),$$

where $a, b, c \leftarrow \mathbb{Z}_q$ are uniformly random and independent. By grouping the elements appropriately, we can view this assumption in matrix form:

$$g^{\begin{pmatrix} 1 & a \\ 1 \cdot b & a \cdot b \end{pmatrix}} \overset{c}{\approx} g^{\begin{pmatrix} 1 & a \\ b & c \end{pmatrix}},$$

where g^M (for a matrix M over $\mathbb{Z}_q$) is the matrix over G obtained by raising g to each entry of M. Observe that in the left-hand matrix, the two rows are linearly dependent (over $\mathbb{Z}_q$), while in the right-hand matrix they are very likely not to be.

(a) Prove that the DDH assumption implies that, for any positive integer w (bounded by a polynomial in the security parameter),

$$g^{\begin{pmatrix} a_1 & a_2 & \cdots & a_w \\ a_1 \cdot b & a_2 \cdot b & \cdots & a_w \cdot b \end{pmatrix}} \overset{c}{\approx} g^{\begin{pmatrix} a_1 & a_2 & \cdots & a_w \\ c_1 & c_2 & \cdots & c_w \end{pmatrix}},$$

where $a_i, b, c_i \leftarrow \mathbb{Z}_q$ are all uniformly random and independent. (*Hint*: try a hybrid argument over the columns.)

Solution: We prove the claim by a hybrid argument. Informally, the hybrid distributions successively replace each column with one made up of two uniformly random and independent group elements, instead of one where the second entry is always the bth power of the first.

Define hybrids $H_0, H_1, \ldots, H_w$ as follows: H_0 chooses $a_1, \ldots, a_w \leftarrow \mathbb{Z}_q$ and $b \leftarrow \mathbb{Z}_q$ uniformly and independently at random and outputs

$$g^{\begin{pmatrix} a_1 & a_2 & \cdots & a_w \\ a_1 \cdot b & a_2 \cdot b & \cdots & a_w \cdot b \end{pmatrix}}.$$

For $i \in [w]$, H_i chooses $a_1, \ldots, a_w \leftarrow \mathbb{Z}_q$, $c_1, \ldots, c_i \leftarrow \mathbb{Z}_q$ and $b \leftarrow \mathbb{Z}_q$ uniformly and independently at random and outputs

$$g^{\begin{pmatrix} a_1 & a_2 & \cdots & a_i & a_{i+1} & \cdots & a_w \\ c_1 & c_2 & \cdots & c_i & a_{i+1} \cdot b & \cdots & a_w \cdot b \end{pmatrix}}.$$

Clearly, H_0 and H_w are respectively the left and right distributions from the problem statement. We prove that $H_0 \overset{c}{\approx} H_w$ by showing that $H_{i-1} \overset{c}{\approx} H_i$ for all $i \in [w]$, under the DDH assumption.

Let $\mathcal{A}$ be an efficient algorithm that attempts to distinguish between H_{i-1} and H_i. Define $\mathcal{A}'$ as follows: given a DDH instance $(g, A = g^a, B = g^b, C = g^c)$ where either $c = ab$ or $c \in \mathbb{Z}_q$ is uniformly random, $\mathcal{A}'$ chooses $a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_w \leftarrow \mathbb{Z}_q$, $c_1, \ldots, c_{i-1} \leftarrow \mathbb{Z}_q$ and runs $\mathcal{A}$ on input

$$\begin{pmatrix} g^{a_1} & g^{a_2} & \cdots & g^{a_{i-1}} & A & g^{a_{i+1}} & \cdots & g^{a_w} \\ g^{c_1} & g^{c_2} & \cdots & g^{c_{i-1}} & C & B^{a_{i+1}} & \cdots & B^{a_w} \end{pmatrix},$$

and outputs the same accept/reject decision as $\mathcal{A}$.

Notice that $B^{a_j} = g^{a_j b}$ for each $j > i$. So the above matrix is distributed like H_{i-1} or H_i, depending on whether $c = ab$ or c is uniformly random, respectively. Therefore, $\mathbf{Adv}^{\text{DDH}}(\mathcal{A}') = \mathbf{Adv}_{H_{i-1}, H_i}(\mathcal{A})$. Because the left-hand side is negligible by the DDH assumption, so is the right-hand side, and the proof is complete.

(b) Using the previous part, prove that the DDH assumption implies that, for any positive integers w, h (both bounded by a polynomial in the security parameter),

$$g^{\left(a_i \cdot b_j\right)_{i \in [h], j \in [w]}} \overset{c}{\approx} g^{\left(c_{i,j}\right)_{i \in [h], j \in [w]}},$$

where $a_i, b_j, c_{i,j} \leftarrow \mathbb{Z}_q$ are all uniformly random and independent. (*Hint*: modify the previous hint.)

Solution: We prove the claim by a hybrid argument. Similarly the previous part, the hybrid distributions successively replace each row with a row of uniformly random and independent group elements.

Define hybrids $H_0, H_1, \ldots, H_h$ as follows: H_0 chooses $a_1, \ldots, a_h \leftarrow \mathbb{Z}_q$ and $b_1, \ldots, b_w \leftarrow \mathbb{Z}_q$ uniformly and independently, and outputs

$$g^{\left(a_i \cdot b_j\right)_{i\in[h],j\in[w]}}.$$

For $k \in [h]$, H_k chooses uniform and independent $a_{k+1}, \ldots, a_h \leftarrow \mathbb{Z}_q$, $b_1, \ldots, b_w \leftarrow \mathbb{Z}_q$, and $c_{i,j} \leftarrow \mathbb{Z}_q$ for $i \in [k]$ and $j \in [w]$ and outputs

$$\left(g^{\begin{cases} c_{i,j} & \text{if } i \le k \\ a_i \cdot b_j & \text{otherwise} \end{cases}} \right)_{i\in[h],j\in[w]}.$$

Clearly, H_0 and H_h are respectively the left and right distributions from the problem statement. We show that $H_0 \overset{c}{\approx} H_h$ by proving that $H_{k-1} \overset{c}{\approx} H_k$ for each $k \in [h]$, assuming the conclusion from the previous part. Let $\mathcal{A}$ be an arbitrary efficient algorithm that attempts to distinguish between H_{k-1} and H_k. Define $\mathcal{A}'$ as follows: it is given as input

$$\begin{pmatrix} B_1 & B_2 & \cdots & B_w \\ D_1 & D_2 & \cdots & D_w \end{pmatrix},$$

where each $B_j = g^{b_j}$ for uniform independent $b_j \in \mathbb{Z}_q$, and $D_j = g^{d_j}$ where either $d_j = a \cdot b_j$ for some uniform $a \in \mathbb{Z}_q$, or every $d_j \leftarrow \mathbb{Z}_q$ is uniform and independent. Then $\mathcal{A}'$ chooses uniform $c_{i,j} \leftarrow \mathbb{Z}_q$ for $i \in [k-1]$ and $j \in [w]$, $a_{k+1}, \ldots, a_h \leftarrow \mathbb{Z}_q$ and runs $\mathcal{A}$ on input

$$\left(G_{i,j} = \begin{cases} g^{c_{i,j}} & \text{if } i < k \\ D_j & \text{if } i = k \\ B_j^{a_i} = g^{a_i \cdot b_j} & \text{if } i > k \end{cases} \right)_{i\in[h],j\in[w]}.$$

It is easy to verify that the above matrix is distributed like H_{k-1} or H_k, depending on whether $d_j = a \cdot b_j$ or the d_j are uniformly random, respectively. Therefore, $\mathcal{A}'$ and $\mathcal{A}$ have the same advantage, and $\mathcal{A}'$ has negligible advantage by the previous part, which completes the proof.

(c) Construct a PRG that stretches about $2n \lg q$ bits to about $n^2 \lg q$ bits, and is secure under the DDH assumption. (For simplicity, the output need not literally be made up of bits.)

Solution: Consider the function $f \colon \mathbb{Z}_q^{2n} \to G$ which on input $(a_1, \ldots, a_n, b_1, \ldots b_n)$ outputs $g^{\left(a_i \cdot b_j\right)}$. From the previous part we see that if the input is chosen uniformly, then the output is indistinguishable from n^2 uniformly random and independent elements of G (under the DDH assumption). The input has length about $2n \log q$, and the output has length about $n^2 \log q$.

5. Suppose that there are k parties who may act as recipients of encrypted messages. In a *multi-recipient* encryption scheme, a sender wants to transmit a (possibly different) message m_i to each recipient i by broadcasting a single ciphertext c to all the parties. An obvious way of doing this is for the recipients to each generate a public/secret key (pk_i, sk_i) for an encryption scheme, and for the sender to broadcast $c = (\mathsf{Enc}(pk_i, m_i; r_i))_{i\in[k]}$, where each r_i denotes fresh randomness used for encrypting m_i.

   Now suppose that the sender uses the *same* randomness for encrypting each of the messages, i.e., the sender broadcasts $c = (\mathsf{Enc}(pk_i, m_i; r))_{i\in[k]}$, where the same r is used for every ciphertext component. If the underlying encryption scheme is ElGamal where everyone uses the same group description (G, g, q), is this scheme still IND-CPA secure under DDH? Prove your answer.

Solution: The modified scheme remains IND-CPA secure under DDH.

First notice that we can slightly modify the proof from the first part of the previous question to show the following extended version:

$$g^{\begin{pmatrix} 1 & a_1 & a_2 & \cdots & a_w \\ r & a_1 \cdot r & a_2 \cdot r & \cdots & a_w \cdot r \end{pmatrix}} \overset{c}{\approx} g^{\begin{pmatrix} 1 & a_1 & a_2 & \cdots & a_w \\ r & c_1 & c_2 & \cdots & c_w \end{pmatrix}},$$

where $a_i, r, c_i \leftarrow \mathbb{Z}_q$ are all uniformly random and independent. (Here we have replaced b with r.)

In the scheme described in the question, receiver i's public key is $A_i = g^{a_i}$, and the ciphertext encrypting message m_i is $(R = g^r, m_i \cdot A_i^r = m_i \cdot g^{a_i \cdot r})$. Therefore, the entire view of the adversary in an IND-CPA attack is just the left-hand side above, with m_i multiplied into the $(i+1)$st entry of the second row.

By a trivial reduction, it follows that the adversary's entire view is indistinguishable (under DDH) from the right-hand side above, with m_i multiplied into the $(i+1)$st entry of the second row. But here the view is just random independent group elements (excepting g) that are independent of the messages, because $m_i \cdot g^{c_i}$ is uniformly random and independent of everything else in the view. This implies IND-CPA security, as we have seen in many other examples.