

Problem Set 7 Solutions

Problem 1. [40 points] Let $p \geq 3$ be a prime and $g \in \mathbf{Z}_p^*$ a generator of \mathbf{Z}_p^* . (These are public quantities, known to all parties including the adversary.) Consider the key-generation and encryption algorithms below:

Algorithm \mathcal{K}	Algorithm $\mathcal{E}(X, M)$
$x \xleftarrow{\$} \mathbf{Z}_{p-1}^*$	if $M \notin \mathbf{Z}_p^*$ then return \perp
$X \leftarrow g^x \bmod p$	$y \xleftarrow{\$} \mathbf{Z}_{p-1}; Y \leftarrow g^y \bmod p$
return (X, x)	$Z \leftarrow X^y \bmod p; W \leftarrow Y \cdot M \bmod p$
	return (Z, W)

The message space associated to public key X is $\text{Messages}(X) = \mathbf{Z}_p^*$. We let k be the bit-length of p .

1. [10 points] Specify a decryption algorithm \mathcal{D} such that $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is an asymmetric encryption scheme satisfying the correct decryption property. State the running time of your algorithm as a function of k (the lower this is, the more credit you get) and prove that the correct decryption property holds.

The decryption algorithm takes input the secret key x and a ciphertext $C = (Z, W)$ and must return the underlying message M . It works as follows:

algorithm $\mathcal{D}(x, C)$
Parse C as (Z, W)
 $s \leftarrow x^{-1} \bmod (p-1)$
 $Y \leftarrow Z^s \bmod p$
 $M \leftarrow W \cdot Y^{-1} \bmod p$
return M

Note that in the key-generation algorithm x is chosen from \mathbf{Z}_{p-1}^* (and not \mathbf{Z}_{p-1}). This implies that x has an inverse modulo $p-1$. The decryption algorithm begins by computing this inverse and denoting it by s . The fact that s is the inverse of x modulo $p-1$ means that $xs \bmod (p-1) = 1$.

Now, to show that the decryption algorithm is correct we have to show that

$$\mathcal{D}((p, g, x), \mathcal{E}((p, g, X), M)) = M$$

for any $M \in \mathbf{Z}_p^*$. Let $C = (Z, W)$ be an output of $\mathcal{E}(X, M)$. We want to show that $\mathcal{D}(x, C) = M$. Let y be the value chosen by the encryption algorithm such that $Y = g^y \bmod p$. Then

$Z = X^y = g^{xy} \pmod p$. Now, we first claim that Y is correctly re-computed by the decryption algorithm. This is true because modulo p we have:

$$Z^s \equiv (g^{xy})^s \equiv g^{xys \pmod{p-1}} \equiv g^{1 \cdot y \pmod{p-1}} \equiv g^y \equiv Y .$$

Since $W = YM \pmod p$, the decryption algorithm, knowing Y , can recover M via $M \leftarrow WY^{-1} \pmod p$.

The decryption algorithm performs one modular exponentiation, which is $O(k^3)$; a couple of modular inverses, each of which is $O(k^2)$; and a modular multiplication, which is $O(k^2)$. So its running time is $O(k^3)$.

2. [30 points] Show that this scheme is insecure with regard to the ind-cpa property by presenting an adversary A such that $\mathbf{Adv}_{\mathcal{AE}}^{\text{ind-cpa}}(A)$ is high. You should specify the adversary, state its running time in as a function of k (the smaller this is, the more credit you get), state the value of its advantage (the larger this is, the more credit you get) and justify the correctness of the adversary.

As for the El Gamal scheme studied in class, the weakness of this scheme is that given the public key X and a ciphertext $C = (Z, W)$ an adversary can compute the Jacobi symbol of the message M . To illustrate this, let y be the value chosen at random by the encryption algorithm in its computation on input M and output C . Let $Y = g^y \pmod p$ and $Z = X^y \pmod p$. Then we have the following equations, which we justify following their statements:

$$J_p(M) = J_p(WY^{-1} \pmod p) \tag{1}$$

$$= J_p(W) \cdot J_p(Y^{-1} \pmod p) \tag{2}$$

$$= J_p(W) \cdot J_p(Y) \tag{3}$$

$$= J_p(W) \cdot J_p(Z) . \tag{4}$$

Let us explain the reasoning behind the equations above. Equation (1) is true because the 4th line of the encryption algorithm tells us that $M = WY^{-1} \pmod p$. Equation (2) is true because of the Proposition we saw in class stating that $J_p(ab \pmod p) = J_p(a) \cdot J_p(b)$ for all $a, b \in \mathbf{Z}_p^*$. Equation (3) is true because of the Proposition we saw in class stating that $J_p(a) = J_p(a^{-1} \pmod p)$ for all $a \in \mathbf{Z}_p^*$. Finally, we claim that $J_p(Y) = J_p(Z)$, which justifies Equation (4). Why is this claim true? We know that $Z \equiv X^y \equiv g^{xy} \pmod p$. Observe that x is an odd number. (Why? The key-generation algorithm tells us that $x \in \mathbf{Z}_{p-1}^*$, meaning $\gcd(x, p-1) = 1$. But p is odd so $p-1$ is even, and so x must be odd, else $\gcd(x, p-1)$ would be at least two.) Since x is odd, $xy \pmod{p-1}$ is even if and only if y is even. In other words, $g^{xy} \pmod p$ is a square iff $g^y \pmod p$ is a square. That is, $J_p(g^{xy} \pmod p) = J_p(g^y \pmod p)$. But $Z = g^{xy} \pmod p$ and $Y = g^y \pmod p$ so we have justified the claim that $J_p(Y) = J_p(Z)$ and hence justified Equation (4).

The import of Equation (4) is that $J_p(M)$ can be computed if we know $J_p(W)$ and $J_p(Z)$. But W, Z are part of the ciphertext and so an adversary can compute $J_p(W)$ and $J_p(Z)$. Thus an adversary can compute $J_p(M)$ given the (public key and) the ciphertext, which is a security weakness in the scheme.

We capitalize on this in the same way as in the attack on the El Gamal scheme. Our adversary A has access to the oracle $\text{LR}(\cdot, \cdot)$, takes input the public key X , and proceeds as follows:

Adversary $A(X)$

$M_0 \leftarrow g; M_1 \leftarrow g^2 \bmod p$

$C \xleftarrow{\$} \text{LR}(M_0, M_1)$

Parse C as (Z, W)

if $J_p(W) \cdot J_p(Z) = 1$ **then** $d \leftarrow 1$ **else** $d \leftarrow 0$

return d

The adversary picks M_0 to be a non-square, meaning $J_p(M_0) = -1$, and picks M_1 to be a square, meaning $J_p(M_1) = 1$. It then computes $J_p(M_b)$ via Equation (4), where b is the challenge bit chosen in the experiment. If this value is 1 it knows that the chosen message was M_1 , and if not it knows that the chosen message was M_0 .

To see how well this adversary does, we need to compute its advantage

$$\mathbf{Adv}_{\mathcal{AE}}^{\text{ind-cpa}}(A) = \Pr \left[\text{Right}_{\mathcal{AE}}^A \Rightarrow 1 \right] - \Pr \left[\text{Left}_{\mathcal{AE}}^A \Rightarrow 1 \right].$$

Assume $b = 1$. This means that the ciphertext $C = (Z, W)$ obtained by A above is an encryption of M_1 , meaning the experiment generated it via $C \xleftarrow{\$} \mathcal{E}(X, M_1)$. The Equation (4) tells us that $J_p(W) \cdot J_p(Z) = J_p(M_1)$. We know the latter is 1 because $M_1 = g^2 \bmod p$. So A returns 1. In other words, $\Pr \left[\text{Right}_{\mathcal{AE}}^A 1 \Rightarrow 1 \right] = 1$.

On the other hand, assume $b = 0$. This means that the ciphertext $C = (Z, W)$ obtained by A above is an encryption of M_0 , meaning the experiment generated it via $C \xleftarrow{\$} \mathcal{E}(X, M_0)$. The Equation (4) tells us that $J_p(W) \cdot J_p(Z) = J_p(M_0)$. We know the latter is -1 because $M_0 = g$. So A returns 0. In other words, $\Pr \left[\text{Left}_{\mathcal{AE}}^A \Rightarrow 1 \right] = 0$.

Now, plugging this into the advantage formula we get

$$\mathbf{Adv}_{\mathcal{AE}}^{\text{ind-cpa}}(A) = 1 - 0 = 1.$$

The running time of the adversary is $O(k^3)$ since it does some Jacobi symbol computations and these are modular exponentiations via the formula $J_p(a) \equiv a^{(p-1)/2} \pmod{p}$, valid for all $a \in \mathbf{Z}_p^*$, that we proved in class.
