Solution Sheet #3

Advanced Cryptography 2022

Solution 1 The Goldwasser-Micali Cryptosystem

1. By construction, we have $n = pq$, $\left(\frac{z}{p}\right) = -1$, and $c \equiv r^2 z^b \pmod{n}$. We have $\left(\frac{c}{p}\right) = \left(\frac{r^2 z^b}{p}\right)$ since p divides n. Thus,

$$\left(\frac{c}{p}\right) = \left(\frac{r^2 z^b}{p}\right) = \left(\frac{z}{p}\right)^b = (-1)^b$$

   So, the decryption of c produces b.

2. Key generation: to generate the primes p and q of bit size s requires $\mathcal{O}(s^4)$ by using Miller-Rabin primality testing, square-and-multiply exponentiation, and schoolbook multiplication. The Legendre symbol requires $\mathcal{O}(s^2)$ which is negligible, as well as computing $n = pq$. So, key generation works in $\mathcal{O}(s^4)$.

   Encryption: this requires a constant number of multiplications which are $\mathcal{O}(s^2)$.

   Decryption: this requires a Legendre symbol, so $\mathcal{O}(s^2)$ as well.

3. (a) In the KR problem, an instance is a pair (n, z) such that $n \in \mathcal{N}$ and $\left(\frac{z}{p}\right) = \left(\frac{z}{q}\right) = -1$ where $n = pq$ is the factoring of n. The solution to the problem is p. Or, equivalently, q which plays a symmetric role.

   (b) Clearly, factoring n solves the problem: by submitting n to an oracle solving Fact, we get p and q so we can yield p.

   Conversely, with an oracle solving the KR problem, we can define an algorithm to factor n. For this, we just need to find one z satisfying $\left(\frac{z}{p}\right) = \left(\frac{z}{q}\right) = -1$ and feed (n, z) to the oracle solving KR. By construction, we have

$$\left(\frac{z}{n}\right) = \left(\frac{z}{p}\right)\left(\frac{z}{q}\right) = 1$$

   If we pick a random z satisfying $\left(\frac{z}{n}\right) = 1$, we have $\left(\frac{z}{p}\right) = \left(\frac{z}{q}\right)$ but this can be 1 or -1. If this is -1 (which happens with probability $\frac{1}{2}$), feeding (n, z) to the KR oracle yield p. We can check that p solve the Fact problem and stop. If it is $+1$, it is bad luck as we have a bad z and we don't know. Thus, feeding (n, z) to the KR oracle may give anything. However, if it gives something which solves the Fact oracle, we are happy anyway and we can stop. Otherwise, we can start again with a new z. Eventually, we find a good z and the solution to Fact.

   So, KR and Fact are equivalent.

4. (a) In the DP problem, an instance is defined by a triplet (n, z, c) where $n \in \mathcal{N}$ (let write $n = pq$), $z \in \mathbf{Z}_n^*$ is a non-quadratic residue with $\left(\frac{z}{n}\right) = 1$, and $c = r^2 z^b \bmod n$ for some $r \in \mathbf{Z}_n^*$ and a bit b. The problem is to find b.

   (b) Clearly, with an oracle solving QR, we can solve DP: we just submit (n, c) to the QR oracle and obtain b. Indeed, $r^2 z^b \bmod n$ is a quadratic residue if and only if $b = 0$.

   To show the converse, we assume an oracle $\mathcal{O}$ solving the DP problem and construct an algorithm to solve the QR one. Given a QR instance (n, c), we pick $z \in \mathbf{Z}_n^*$ such that $\left(\frac{z}{n}\right) = 1$ and consider the function $f_z : y \mapsto \mathcal{O}(n, z, y)$.

   If z is a quadratic residue, we observe that for any b, $r^2 z^b \bmod n$ is uniformly distributed in the set of quadratic residues modulo n. So, this is independent from b. Thus, $f_z(r^2 z^b \bmod n)$ is a random bit independent from b. If now z is a non-quadratic residue, $f_z(r^2 z^b \bmod n) = b$. By taking b uniformly distributed, we can easily identify in which case we are. We can thus iterate until we have a good z which is a non-quadratic residue. Then, we can compute $f_z(c)$ and get the solution to the QR problem.

   So, DP and QR are equivalent.

Solution 2 The CPA-secure PKC from the deterministic PKC (HW 1, 2019)

1. Consider the following adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$.

   Adversary: $\mathcal{A}_1(pk)$
   $m_0 \xleftarrow{\$} \mathcal{M}$
   $m_1 \xleftarrow{\$} \mathcal{M} \setminus \{m_0\}$
   $s_1 \leftarrow \mathcal{C}.\mathsf{Enc}(pk, m_0)$
   return m_0, m_1, s_1

   Adversary: $\mathcal{A}_2(c, s_1)$
   if $c = s_1$ **then**
   | **return** 0
   else
   | **return** 1
   end

   If $\mathcal{C}$ is deterministic, $\mathcal{C}.\mathsf{Enc}(pk, m) = \mathcal{C}.\mathsf{Enc}(pk, m') \iff m = m'$. Then, we have

   $$\Pr\left[\mathsf{IND\text{-}CPA}_{\mathcal{C}}^{\mathcal{A}}(0, \lambda) = 1\right] = 0 \qquad \text{and} \qquad \Pr\left[\mathsf{IND\text{-}CPA}_{\mathcal{C}}^{\mathcal{A}}(1, \lambda) = 1\right] = 1.$$

   The advantage $\mathsf{Adv}_{\mathcal{A},\mathcal{C}}^{\mathsf{IND\text{-}CPA}}(\lambda) = 1$ for any $\mathcal{C}$. Hence, there is no IND-CPA-secure deterministic PKC.

2. Consider the following adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$.

   Adversary: $\mathcal{A}_1(pk)$
   $m_0 \leftarrow 0$
   $m_1 \xleftarrow{\$} \mathcal{M}_2 \setminus \{0\}$
   $s_1 \leftarrow \bot$
   return m_0, m_1, s_1

   Adversary: $\mathcal{A}_2(c, s_1)$
   $c_1, c_2 \leftarrow c$
   if $c_1 = c_2$ **then**
   | **return** 0
   else
   | **return** 1
   end

   If m is zero, $\mathsf{Enc}_1(pk, m \oplus r) = \mathsf{Enc}_1(pk, r)$ because Enc_1 is deterministic. Therefore, $c_1 = c_2$ if c_1 is the encryption of 0, which is m_0. So, we have

   $$\Pr\left[\mathsf{IND\text{-}CPA}_{\mathcal{C}_2}^{\mathcal{A}}(0, \lambda) = 1\right] = 0 \qquad \text{and} \qquad \Pr\left[\mathsf{IND\text{-}CPA}_{\mathcal{C}_2}^{\mathcal{A}}(1, \lambda) = 1\right] = 1.$$

   Hence, we have $\mathsf{Adv}_{\mathcal{A},\mathcal{C}_2}^{\mathsf{IND\text{-}CPA}}(\lambda) = 1$, and $\mathcal{C}_2$ is not IND-CPA-secure.

3. If $\mathcal{C}_1$ is the plain RSA and $\mathcal{M}_2$ is a multiplicative group, the ciphertext $c = (c_1, c_2)$ can be written as follows:

$$(c_1, c_2) = ((mr)^e \bmod n, r^e \bmod n)$$

where (e, n) is a public key pair in the plain RSA. Then, we can deduce that

$$c_1 \equiv m^e c_2 \pmod{n}$$

Now, consider the following adversary $\mathcal{A}$:

Adversary: $\mathcal{A}_1(pk, m_0, m_1, c)$
$e, n \leftarrow pk$
$c_1, c_2 \leftarrow c$
if $c_1 \equiv m_0^e c_2 \pmod{n}$ **then**
  return 0
else
  return 1
end

Since $c_1 \equiv m_0^e c_2 \pmod{n}$ always holds if c is an encryption of m_0, the guess of $\mathcal{A}$ is always correct. Hence, the advantage of $\mathcal{A}$ is 1 and $\mathcal{C}_2$ is not IND-KPA-secure.