Mid Semester - Fall, 2022-2023
M. Tech. Cryptology & Security
Topics in Cryptology (Part-A Only)

September 22, 2022

Maximum Marks: 50
Time: 3 hr (flexible)
Open note/book Exam

Instructions

- Maximum marks is 50. Total marks provided in the paper: 60, among which Part-A contains 30 and Part-B contains 30.
- Use separate sheets for Part-A and Part-B and clearly mark on top of your answer sheets.
- You can use any book/notes during the exam, but not internet.

Part-A

Notations. The set $\{1, 2, 3, \ldots\}$ of all positive integers are denoted by $\mathbb{N}$. The set of all integers are denoted by $\mathbb{Z}$. The ring of all integers modulo n is denoted by $\mathbb{Z}_n$. Below $\kappa \in \mathbb{N}$ denotes the security parameter throughout (for example, if 128 bit security is desired from the system, then κ is set to 128, this is often equal to the key-length). For any randomized algorithm A we denote $y \leftarrow \mathsf{A}(x)$. Sometimes such algorithms are determinized by making the randomness explicit as $y := \mathsf{A}(x; r)$. An inherently deterministic algorithm D is denoted as $y := \mathsf{D}(x)$. A uniform random sample from a domain D is denoted as $s \leftarrow_{\$} D$. We assume $\mathbb{G}$ to be a cyclic group of prime order p. Let g be a generator of the group.

1. Recall the standard El-Gammal Encryption for messages in $\mathbb{G}$:

   - $\mathsf{KGEN}(1^\kappa) \to (\mathsf{pk}, \mathsf{sk})$:
     (a) randomly sample $x \leftarrow_{\$} \mathbb{Z}_p$, and set $\mathsf{sk} := x$;
     (b) compute $\mathsf{pk} := g^x$.
   - $\mathsf{ENC}(\mathsf{pk}, m) \to c$: It is given that $m \in \mathbb{G}$. Then sample $r \leftarrow_{\$} \mathbb{Z}_p$ and compute $c_1 := g^r$ and $c_2 := \mathsf{pk}^r \cdot m$, and set $c := (c_1, c_2)$.
   - $\mathsf{DEC}(\mathsf{sk}, c) =: m$: Parse $\mathsf{sk} := x$, then decrypt $m := c_2 \cdot (c_1^x)^{-1}$.

   Now answer the following questions:

   (a) Assume $m \in \{0,1\}^C$ when $C = O(1)$ – that means C is a small constant. For example, C can be 16.
      i. How would you modify El-Gammal encryption to encrypt such plaintexts?
      ii. What would be the computational complexity of decryption in concrete terms (as a function of C) for the modified encryption scheme?
      iii. Would the modified scheme remain additively homomorphic?

(3+1+2 = 6)

(b) Now assume $m \in \{0,1\}^\kappa$

i. How would you change the scheme to encrypt such message? (Hint: Use hash functions!)

ii. What happens to the additive homomorphic properties in this case?

(3+2 = 5)

2. Consider the following encryption scheme, obtained by slightly changing El-Gammal:

- $\mathsf{KGEN}(1^\kappa) \to (\mathsf{pk}, \mathsf{sk})$:
  (a) let $\mathsf{H} : \mathbb{G} \to \mathbb{Z}_p$ be a hash function, which behaves as a random function (random oracle);
  (b) randomly sample $x \leftarrow_\$ \mathbb{Z}_p$, and set $\mathsf{sk} := x$ and $\mathsf{pk} := (g^x, \mathsf{H})$.
- $\mathsf{ENC}(\mathsf{pk}, m) =: c$: It is given that $m \in \mathbb{G}$. Then compute $r := \mathsf{H}(m)$ and compute $c_1 := g^r$ and $c_2 := \mathsf{pk}^r \cdot m$, and set $c := (c_1, c_2)$.
- $\mathsf{DEC}(\mathsf{sk}, c) =: m$: Parse $\mathsf{sk} := x$, then decrypt $m := c_2 \cdot (c_1^x)^{-1}$.

Assume that DDH is *hard* in $\mathbb{G}$, that is given g, g^x, g^y, where $(x, y) \leftarrow_\$ \mathbb{Z}_p^2$, it is computationally *hard* to distinguish between g^{xy} and a randomly sampled $h \leftarrow_\$ \mathbb{G}$.

- Is the above scheme secure (more formally, CPA-secure)? Argue in favor or against.

(4)

3. Assume that the CDH is *hard* in group $\mathbb{G}$, that is: given g, g^x, g^y computing g^{xy} is *hard*, where $(x, y) \leftarrow_\$ \mathbb{Z}_p^2$. Now consider the following assumption, called computational Square-DH (CSDH): given g, g^x computing g^{x^2} is computationally *hard* (again $x \leftarrow_\$ \mathbb{Z}_p$). Show that, these assumptions are *equivalent*, that is, specifically:

(a) If CDH is *easy*, then so is CSDH. Or in other words, given a solver for CDH, one can use that to solve CSDH *efficiently*. Note that, a solver for CDH, on input (g, g^x, g^y) (for uniform random $(x, y) \in \mathbb{Z}_p^2$), returns g^{xy}. This is also known as black-box or oracle access.

(b) In the other direction, show that if CSDH is easy, then so is CDH. Specifically, show that given a solver for CSDH (that, on input (g, g^x), returns g^{x^2}), one can *easily* solve CDH.

(c) How many times the reduction has to access the given oracle in each case?
(Hint: Remember that, when you are using any algorithm as oracle, that only gives you correct answer if the input is distributed identically as actual inputs. For example, you can not give (g, g^x, g^x) as input to the CDH solver, because the exponents (x, x) is not uniform random in $\mathbb{Z}_p^2$. The CDH solver will be able to detect that the second and third inputs are identical and may not return the answer.)

(3+3+2 = 8)

4. Let π be a generic two party computation (2PC) protocol which computes any boolean function $\{0,1\} \times \{0,1\} \to \{0,1\}$ between any two parties P_1 and P_2 when P_1 holds the first input x_1 and P_2 holds the second input x_2 and at the end of the protocol both of them get the output y. Now consider a *specific* function $y = f(x_1, x_2)$ which is computed by P_1 and P_2 using π. At the end of the protocol party P_1 is able to compute x_2 and party P_2 is able to compute x_1. Then answer each of the following questions with arguments.

(a) Is it possible to conclude whether π is secure or insecure? Argue with a concrete example.

(b) Now consider that the same protocol is used to compute the AND function $g(x_1, x_2) = x_1 \wedge x_2$. If P_1's input is 1 then it is able to recover P_2's input x_2. Does the conclusion about the security of π change or not?

(c) Now consider the same protocol is used to compute the OR function $h(x_1, x_2) = x_1 \vee x_2$. If P_1's input is 1 then in the end it recovers P_2's input x_2. How does this fact change your conclusion about π, if at all?

(3+2+2 = 7)