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

November 28, 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 $s \leftarrow_{\$} D$.

1. Consider the following language: given a cyclic group $\mathbb{G}$ of prime order p, an instance of the language consists of the statement $\mathsf{inst} = (g, h, x, y)$ and witness $\mathsf{wit} = k$ such that $k = \mathsf{DLOG}_g(x) = \mathsf{DLOG}_h(y)$ (or alternatively $g^k = x$ and $h^k = y$).
   (a) Write an Interactive proof system for the above language.
   (b) Provide arguments for completeness, soundness and honest verifier zero knowledge.
   (c) Can you transform it to one which is non-interactive and fully zero-knowledge? In that case, how do you argue soundness and full zero-knowledge?

   (4+3+3 = 10)

2. Consider three parties A, B and C with private inputs $a, b, c \in \{0, 1\}$ respectively. They want to compute a function $f = a.b + c$ over $\mathbb{Z}_2$ in a secure manner such that at the end of the protocol everyone should have f, but "nothing else".
   (a) Design a protocol which is resilient to at most 1 corrupt party, who can have unbounded computational power.
   (b) Provide security intuitions for the above protocol.

(c) What happens if there are more than 1 corrupt party? Show exactly where the protocol breaks down.

(6+2+2 = 10)

3. Consider the exact same setting as above: three parties A, B and C with private inputs $a, b, c \in \{0, 1\}$ respectively. They want to compute a function $f = a.b + c$ over $\mathbb{Z}_2$ in a secure manner such that at the end of the protocol everyone should have f, but "nothing else".

   (a) Design another protocol which is resilient to at most 2 corrupt parties, such that they are computationally bounded (can run at most $O(\mathsf{poly}(\kappa))$ time.

   (b) Provide security intuition for the above protocol.

   (c) What happens if corrupt parties can have unbounded computational power? Show exactly where the protocol breaks down.

(6+2+2 = 10)