Mid Semester - Fall, 2023
M. Tech. Cryptology & Security
Advanced Cryptology
(with Answers)

November 25, 2025

Maximum Marks: 30
Maximum Time: 3 hr
Open note/book Exam

Instructions

- Maximum marks is 30. Total marks provided in the paper: 35.
- Be short and precise. Partial marking will be provided for a partially correct answer/attempt.
- You can use any book/notes during the exam, but *not internet*.

Notations. 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). 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. Consider the following secure messaging scheme between Alice and Bob.
   - Alice wants to send a message $m \in \{0,1\}^n$ to Bob. Alice and Bob do not have a shared key, neither they are connected by a secure channel. The adversarial model is eavesdropping. That is, Alice and Bob are honest, and want to protect privacy of their communications against a potential eavesdropper Eve, who is a passive attacker (and is not doing a man-in-the-middle).
   - Alice and Bob perform a Diffie-Hellman key-exchange. Alice's secret is $a \in \mathbb{Z}_p$, public value is $A = g^a \in \mathbb{G}$; Bob's secret is $b \in \mathbb{Z}_p$ and public value is $B = g^b \in \mathbb{G}$. Let $C = g^{ab} \in \mathbb{G}$ be the derived "key".
   - They have a hash function $\mathsf{H} : \mathbb{G} \to \{0,1\}^n$ modeled as a random oracle. They both apply it on C to get $k := \mathsf{H}(C)$.
   - Alice sends $c := k \oplus m$; Bob decrypts $m := c \oplus k$.
   - Now answer the following questions:
     (a) Write the eavesdropping security game for the secure messaging setting against a passive eavesdropper.

     Answer. We define the eavsedropping security game *abstractly* as follows:

     – The challenger plays the roles of Alice and Bob, and the adversary plays Eve. The adversary may issue random oracle queries for H to the challenger.

- The challenger sends A and B to the adversary as Alice and Bob's public values.
- When the adversary returns two challenge messages m_0, m_1, then the challenger randomly chooses m_b and sends over "the symmetric-key encryption" (details are not part of this game) c of m_b to adversary, where the key is derived from A and B.
- The adversary sends a guess b' and the game outputs 1 if and only if $b = b'$.

We say that the secure messaging scheme is secure if and only if the probability that the game outputs 1 is bounded by $|1/2 - \mathsf{negl}(\lambda)|$. □

(b) Briefly describe the hybrids.

Answer. Let GAME_0 be the game where $b = 0$. Then we define three main hybrids (the hybrids for GAME_1 are analogous) as follows:

- HYB_1: In this hybrid, the only change from HYB_0 is that: the derived value C is chosen uniformly at random from $\mathbb{G}$.
- HYB_2: In this hybrid, the only change from HYB_1 is that: $\mathsf{H}(C)$ is replaced with a uniform random k in $\{0,1\}^n$, which has no connection with H.
- HYB_3: In this hybrid, the only change from HYB_2 is that: the ciphertext c is chosen uniformly at random from $\{0,1\}^n$.

□

(c) Argue the security arguments of each hybrid (hint: there are three hybrids).

Answer. The security arguments are as follows:

- GAME_0 and HYB_1 are indistinguishable due to DDH. The reduction, on receiving a triple (g^a, g^b, h) (where h is either g^{ab} or uniform over $\mathbb{G}$), emulates as follows:
  - Sends $A = g^a$ and $B = g^b$ to the challenger.
  - Simulates the RO queries on H by lazy sampling of a random function (consistently replying for prior queries and choosing uniform random values for new queries).
  - Use h as C.

  Clearly, the games GAME_0 and HYB_1 are perfectly emulated when $h = g^{ab}$ and is uniform random respectively.
- HYB_1 and HYB_2 are identical, as long as the adversary does not make a RO query $\mathsf{H}(C)$. Because, in game HYB_2, k is uniformly random and is not programmed to $\mathsf{H}(C)$. Since C is uniformly chosen at random, the probability of adversary making such query is negligible.
- HYB_2 and HYB_3 are perfectly indistinguishable due to one-time pad security.

□

(2+3+3 = 8)

2. Given a random oracle $\mathsf{H} : \{0,1\}^n \to \{0,1\}^{n+1}$:

- Design a length-doubling PRG $\{0,1\}^n \to \{0,1\}^{2n}$ and briefly provide arguments why that is secure.

Answer (partial). There are two steps. First, we prove that $\mathsf{H} : \{0,1\}^n \to \{0,1\}^{n+1}$ is a PRG. Then in the second step one just uses Blum-Micali hybrid arguments.

In the random oracle model, the PRG-game changes slightly, in that the adversary also queries the RO on H. However, since by definition of PRG, the adversary obtains $\mathsf{H}(s)$ for a uniform random seed s in the real PRG game, the probability that H is queried on s prior to the response is negligible (precisely, $\frac{1}{2^n}$). Furthermore, we note that the only way to distinguish between $\mathsf{H}(s)$ and a uniform random value in $\{0,1\}^n$ is to explicitly make an RO query on s. This implies H is a PRG. *The second step can be completed following a standard textbook.* □

(3)

3. Recall the DDH assumption: for any PPT adversary A, it is hard to distinguish between the following triples in $\mathbb{G}$:

   - (g^a, g^b, g^{ab}) where a, b are uniform random in $\mathbb{Z}_p$;
   - (g^a, g^b, g^c) where a, b, c are uniform random in $\mathbb{Z}_p$.

   Now consider the ℓ-extended DDH assumption: for any PPT adversary A it is hard to distinguish between the following $2\ell + 1$ tuples in $\mathbb{G}$:

   - $(g^a, g^{b_1}, g^{ab_1}, g^{b_2}, g^{ab_2}, \ldots, g^{b_\ell}, g^{ab_\ell})$ where $a, b_1, \ldots, b_\ell$ are uniform random in $\mathbb{Z}_p$;
   - $(g^a, g^{b_1}, g^{c_1}, g^{b_2}, g^{c_2}, \ldots, g^{b_\ell}, g^{c_\ell})$ where $a, b_1, \ldots, b_\ell, c_1, \ldots, c_\ell$ are uniform random in $\mathbb{Z}_p$.

   Now answer the following questions:

   (a) Argue that as long as DDH holds in $\mathbb{G}$, ℓ-extended DDH also holds in $\mathbb{G}$ for any ℓ is polynomial in κ. Ideally, this should use a single reduction (*not* hybrids).

   Answer. If there is a PPT adversary for ℓ-extended DDH, then we can construct a PPT reduction to break DDH as follows:

   - The reduction gets DDH-tuple (g^x, g^y, h) from the challenger, where h is either g^{xy} or uniformly random in $\mathbb{G}$.
   - The reduction constructs a challenge $(g^a, g^{b_1}, h_1, g^{b_2}, h_2, \ldots, g^{b_\ell}, h_\ell)$ for the ℓ-extended DDH adversary where each h_i is either g^{ab_i} or uniformly at random. The reduction works as follows:
     - Set $g^a := g^x; g^{b_1} = g^y$ and $h_1 := h$ (a, b_1 unknown) .
     - Sample $\ell - 1$ random $r_2, \ldots, r_\ell$ in $\mathbb{Z}_p$.
     - For all $i \in \{2, \ldots, \ell\}$: set $g^{b_i} := (g^y)^{r_i}$ and $h_i := h^{r_i}$ (b_i unknown)

     Now let us argue that the challenge has the correct distribution. Note that, the reduction implicitly sets $b_i := r_i b$ without knowing b_i. Clearly, all b_i are uniformly distributed.[1] Furthermore, if the received tuple is a DDH tuple, that is $h = g^{ab}$, then each $h_i = g^{ab_i}$, as desired for the ℓ-extended DDH tuple. In the other case, when the received tuple is a random tuple, that is $h = h^c$ for a uniform random c, then each $h_i = h^{c_i}$ for uniform random $c_i = r_i c$.

   □

   (b) Can you prove the same when ℓ is exponential in κ? Argue in favor of your answer.

   Answer. The reduction's running time is proportional to ℓ. So if ℓ is exponential, then the reduction would not be a PPT algorithm. Hence the security argument does *not* go through. □

(4+2)

4. Let $F : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$ be a PRF (the key is κ-bit and input/output are n-bits). Now, consider the following extended function $F' : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^{n+1}$ which is constructed as follows:

   - $F'_k(x) = F_k(x) \| k_0$ (where k_0 is the first bit of the key k).

   Now answer the following about function F':

   - Is F' a PRF or not? Argue in favor of your answer.

[1] It is worth noting that, though it appears that $r_i b$'s are correlated, that is not the case, because no other information of r_i are given (for example, giving away g^{r_i} would have made the distribution correlated). So, we can just use simple re-randomization technique to "simulate" ℓ-extended DDH game.

Answer. If the adversary makes Q queries to the challenger, then for the real PRF game it ends up receiving values where the last bit would always be the same, that is k_0. For the random game, this happens with probability $1/2^Q$. So, the adversary is able to distinguish except with probability $1/2^Q$. □

(3)

5. Consider a the following three (Vanilla) El-Gammal ciphertexts with respect to public key $h = g^x$ for messages $m_1, m_2 \in \mathbb{G}$.

   - $c_1 = (g^{r_1}, h^{r_1} m_1)$
   - $c_2 = (g^{r_2}, h^{r_2} m_2)$
   - $c_{12} = (g^{r_{12}}, h^{r_{12}} m_1 m_2)$

   A prover wants to prove to a verifier that c_{12} encrypts products of the plaintexts encrypted in c_1 and c_2. The public inputs (instances) are g, h, c_1, c_2, c_{12}.

   (a) Give a brief outline for a sigma-protocol for proving the above relation. (no need to prove security of standard/well-known protocols)

   Answer. Both prover and verifier knows $c_1 = (R_1, E_1), c_2 = (R_2, E_2), c_{12} = (R_{12}, E_{12})$ plus the public key h (in addition to generator g). Both parties compute $R^* = R_1 R_2 / R_{12}$ and $E^* = E_1 E_2 / E_{12}$. Then they execute a Chaum-Pederson proof of equal exponent with respect to bases g and h. Clearly the prover is able to prove equal exponent, if and only if the plaintexts cancel out as $R^* = g^\delta$ $E^* = h^\delta$ where $\delta = r_1 + r_2 - r_{12} \bmod p$. Hence the verifier can conclude that the ciphertexts indeed has the desired form. □

   (b) Does the prover have to know the plaintexts m_1, m_2 for the above proof? Argue in favor of your answer.

   Answer. The prover *does not* have to know the plaintexts. This is despite the fact that the Chaum-Pederson proof offers knowledge soundness – that is, the prover needs to know the exponent δ in order to prove equality of the exponent (soundness argument follows from a construction of a PPT knowledge extractor). However, this does not imply that the prover needs to know the plaintexts. So, for the given statement, existential soundness suffices, which follows from the knowledge soundness of the Chaum-Pedersom. □

   (3+2)

6. Recall Pedersen's commitment for public key $h = g^x$.

   - To commit to a message $m \in \mathbb{Z}_p$, the committer generates the commitment $c := g^m h^r$. The opening consists of the pair (m, r).

   Now answer the following:

   (a) If the committer somehow knows x, what would be the issue? Argue with a concrete attack.

   Answer. If x is known, binding can be broken by producing a pair of distinct commitment-opening (m, r) and (m', r') such that $m + rx = m' + r'x$, where m, r, r' can be arbitrary and $m' := x(r - r') + m \bmod p$. □

   (b) Design a non-interactive proof-system, in that a prover wants to prove to a verifier, the *knowledge of opening* (m, r), given (c, g, h) as public input (instance). (hint: you may use random oracles) without actually opening.

Answer. The public input/instance are (g, h, c) and the prover has private input/witness (m, r). They can execute a simple sigma-protocol with Fiat-Shamir transform (for non-interactivity) as follows:

- The prover samples uniform random μ, ρ in $\mathbb{Z}_p$ to construct a random instance $\alpha := g^{\mu} h^{\rho}$. Then it uses an appropriate hash function H to compute $\beta := \mathsf{H}(c, g, h, \alpha)$. Then it computes: $\gamma := m\beta + \mu$ and $\delta := r\beta + \rho$. The proof consists of (α, γ, δ)
- The verifies, on receving (α, γ, δ), first computes $\beta := \mathsf{H}(c, g, h, \alpha)$ and then checks $(c^{\beta}.\alpha = g^{\gamma} h^{\delta})$

□

(c) Argue completeness, soundness (proof of knowledge) and zero-knowledge.

Answer (partial). We argue completeness, knowledge soundness and zero-knowledge as follows:

- *Completeness is left as a straightforward exercise.*
- Soundness can be proven by constructing a knowledge extractor as follows:
  - The extractor forks/rewinds the random oracle to get two responses (β, β') for the same $\mathsf{H}(c, g, h, \alpha)$, and subsequently $\gamma := m\beta + \mu$ and $\delta := r\beta + \rho$ plus $\gamma' := m\beta' + \mu$ and $\delta' := r\beta' + \rho$.
  - Both the proofs are accepting. So, $c^{\beta}\alpha = g^{\gamma} h^{\delta}$ and $c^{\beta'}\alpha = g^{\gamma'} h^{\delta'}$. The extractor then divides them to compute $c^{\beta - \beta'} = g^{\gamma - \gamma'} h^{\delta - \delta'}$. This can also be written as $g^{m(\beta - \beta') + xr(\beta - \beta')} = g^{(\gamma - \gamma') + x(\delta - \delta')}$.
  - Now, it can extract $m = (\gamma - \gamma')/(\beta - \beta')$ and $r = (\delta - \delta')/(\beta - \beta')$ over $\mathbb{Z}_p$.
  - To argue why the extraction works, we can observe that first we have a 1-degree polynomial in x in the exponent in each side of the above equation (after division). Now, since x is chosen uniformly at random, by Schwartz-Zippel Lemma, we conclude that the polynomial must be identical except with probability $1/p$. Secondly, since both β and β' are uniformly chosen, $\beta - \beta'$ is non-zero with overwhelming probability.
- The protocol achieves full zero-knowledge due to Fiat Shamir. The simulator works as follows:
  - It has only the public input (g, c, h).
  - It samples uniform random γ, δ, β and then compute $\alpha := g^{\gamma} h^{\delta} / c^{\beta}$. Finally it programs random oracle at $\mathsf{H}(c, g, h, \alpha) := \beta$.
  - Note that, the distribution is perfectly identical to a real simulation as μ, ρ acts as one-time pads and hence yield uniformly distributed γ and δ in the real simulation. The only issue is the random oracle is programmed later after β is chosen – so the simulation would fail to answer any RO query of the form $\mathsf{H}(\cdots, \alpha)$ before choosing β, γ, δ. However, since α is uniquely determined by uniform random β, γ, δ, the probability that the adversary makes a RO query on a correct α, before the simulator chooses β, γ, δ is statistically negligible, as long as the adversary can make a bounded number of RO queries. For example, the adversary can be allowed to make $2^{\epsilon \log p}$ many queries for any $\epsilon < 1$.

□

(d) Among these properties which one would still hold even if the adversary has unbounded power.

Answer. Completeness always holds, as there is no adversary involved. However, here the ZK argument is statistical as long as the adversary is allowed to make a bounded number of queries. Of course, if the adversary can exhaust the domain of H, then the programmability would fail. So ZK also holds for unbounded adversaries with bounded RO-queries as explained above. □

(3+3+3+1)