UC Berkeley — CS194-34 : Undergraduate Cryptography
Prof. Sanjam Garg

Final
December 18, 2019

Final

Name:

SID:

Do not turn this page until your instructor tells you to do so.

- After the exam starts, write your name on every odd-numbered page. We reserve the right to deduct points if you do not, and you will not be allowed to do so after time is called.
- For short question, your answers must be written clearly inside the box region. Any answer outside the box will not be graded. For longer questions, if you run out of space, you must clearly mention in the space provided for the question if part of your answers is elsewhere.
- Try to answer all questions. Not all parts of a problem are weighted equally. Before you answer any question, read the problem carefully. Be precise and concise in your answers.
- You may consult at most *20 sheets of notes.* Apart from that, you may not look at books, notes, etc. Calculators, phones, computers, and other electronic devices are NOT permitted.
- There are **14** pages on the exam (counting this one). Notify a proctor immediately if a page is missing.
- **You have 160 minutes: there are 6 questions on this exam worth a total of 100 points.**

1 Multiple Choice Questions (18 points)

Every question may have one or more choices that are correct. +3 if you mark all the correct choices and only those choices. Every other answer gets 0 points.

1. Let $\mathbb{G}$ be a group of order p and g be a random generator. Let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be an efficiently computable bilinear map. Assume that Computational Diffie-Hellman (CDH) problem is hard on $\mathbb{G}$. Which of the following is/are true?

   (a) The two distributions $\{g, g^a, g^b, c, g^{abc}\}$ for $a, b \leftarrow \mathbb{Z}_p$ and $\{g, g^a, g^b, c, g^d\}$ for $a, b, c, d \leftarrow \mathbb{Z}_p$ are indistinguishable.

   (b) Given g, g^a, g^b for $a, b \leftarrow \mathbb{Z}_p$, it is hard to compute $g^{a/b}$.

   (c) Given g, g^a for $a \leftarrow \mathbb{Z}_p$, it is easy to compute g^{a^2}.

   (d) Given g, g^a for $a \leftarrow \mathbb{Z}_p$, it is hard to compute $g^{1/a}$.

   Your choices: **Solution:** B,D

2. Assume OWFs do not imply public-key encryption. Which of the following is/are implied by OWFs?

   (a) Digital Signatures.

   (b) Decisional Diffie-Hellman Assumption.

   (c) Secret Key Encryption.

   (d) Additively Homomorphic Secret Key Encryption.

   Your choices: **Solution:** A,C

3. Which of the following implies a zero-knowledge proof for NP?

   (a) Pseudorandom Functions.

   (b) $\mathsf{P} = \mathsf{NP}$.

   (c) Decisional Diffie-Hellman Assumption.

   (d) Collision-Resistant Hash functions.

   Your choices: **Solution:** A,B,C,D

4. Let $\mathbb{G}$ be a DDH-hard group of order p and g be a random generator. Which of the following distributions are indistinguishable to (g, g^a, g^b, g^c) for randomly chosen $a, b, c \leftarrow \mathbb{Z}_p$?

   (a) $(g, g^a, g^{1/b}, g^{a/b})$ where $a, b \leftarrow \mathbb{Z}_p$.

   (b) (g, g^a, g^b, g^{a-b}) where $a, b \leftarrow \mathbb{Z}_p$.

(c) $(g, g^a, g^b, g^{a/b})$ where $a, b \leftarrow \mathbb{Z}_p$.

(d) $(g, g^{b_1}, g^{ab_1}, g^{ab_2})$ where $a, b_1, b_2 \leftarrow \mathbb{Z}_p$.

Your choices: **Solution:** A,C,D

5. Which of the following primitives are implied by an Identity-Based Encryption Scheme?

(a) Authenticated Secret key Encryption.

(b) Digital Signatures.

(c) Zero-Knowledge Proofs.

(d) CCA-2 Public-Key Encryption.

Your choices: **Solution:** A,B,C,D

6. Which of the following is/are true?

(a) CBC-MAC using random IV is secure.

(b) Chained CBC mode is insecure.

(c) One round SPN with key mixing step cannot be broken in 2^{64} time.

(d) Define the key space $k = (a, b) \leftarrow \mathbb{Z}_p \times \mathbb{Z}_p$ where p is a prime. Define the encryption to take $x \in \mathbb{Z}_p$ and output $ax + b$. This scheme is not perfectly secure.

Your choices: **Solution:** B

2 Bit Commitment (15 points)

1. Assuming a OWP $f : \{0,1\}^n \to \{0,1\}^n$, construct an efficient algorithm $\mathsf{Com}(b;r)$, where $b \in \{0,1\}$ and $r \in \{0,1\}^\ell$ (for any $\ell \geq n$), such that:

   (a) There does not exist r, r' such that $\mathsf{Com}(0;r) = \mathsf{Com}(1;r')$.

   (b) $\{\mathsf{Com}(0;U_n)\} \approx_c \{\mathsf{Com}(1;U_n)\}$ where U_n is the uniform distribution over $\{0,1\}^n$ and $\approx_c$ denotes computational indistinguishability.

   Prove that your construction satisfies both these properties.

Name:

Solution: $\mathsf{Com}(b; r) = (f(r), h(r) \oplus b)$ where f is a one-way permutation and h is the hardcore bit. Binding follows since f is a one-way permutation and hiding follows from the hardcore bit.

2. Assuming a PRG $f : \{0,1\}^n \to \{0,1\}^{3n}$ (note that f may not be injective), construct an efficient algorithm $\mathsf{Com}(\sigma, b; r)$, where $\sigma \in \{0,1\}^{3n}$, $b \in \{0,1\}$, and $r \in \{0,1\}^{\ell}$ (for any $\ell \geq n$), such that:

   (a) Let $\mathsf{Bad} = \{\sigma \mid \exists r, r' \text{ such that } \mathsf{Com}(\sigma, 0; r) = \mathsf{Com}(\sigma, 1; r)\}$. Then, $\mu(n) = \Pr_{\sigma \leftarrow U_{3n}}[\sigma \in \mathsf{Bad}]$ is a negligible function.

   (b) We have that $\{\sigma, \mathsf{Com}(\sigma, 0; U_n)\} \approx_c \{\sigma, \mathsf{Com}(\sigma, 1; U_n)\}$ for every $\sigma \in \{0,1\}^{3n}$.

   Prove that your construction satisfies both these properties.

Solution: $\mathsf{Com}(\sigma, b; r) = f(r)$ if $b = 0$; else, it is equal to $f(r) \oplus \sigma$ if $b = 1$. Hiding follows from pseudorandomness of f. Binding follows since $|\{f(r) \oplus f(r') : r, r' \in \{0,1\}^n\}| \leq 2^{2n}$ but σ is chosen uniformly from $\{0,1\}^{3n}$.

3 Zero-Knowledge Proofs (15 points)

Let (P_0, V_0) and (P_1, V_1) be *public-coin*[1] honest verifier zero-knowledge proof systems for languages L_0 and L_1 respectively. For simplicity, let us assume that there are three messages in both protocols. The first message is from the prover to the verifier, denoted by α_0 (resp. α_1). The second message is just the verifier's random coins and this will be β_0 (resp. β_1) (we will assume that the length of the verifier's random coins is the same for both proof systems and this length will be n). The final message is from the prover to the verifier, denoted by γ_0 (resp. γ_1). Let S_0 and S_1 be the honest verifier zero-knowledge simulators for the two proof systems. Specifically, for both $c = 0$ and $c = 1$, $S_c(\beta_c)$ (for a randomly chosen β_c) outputs (α_c, γ_c) such that $V_c(\alpha_c, \beta_c, \gamma_c) = 1$ and the distribution of $(\alpha_c, \beta_c, \gamma_c)$ is identical to the real protocol.

Now consider the language $L = \{(x_0, x_1) \mid x_0 \in L_0 \text{ or } x_1 \in L_1\}$. Consider the following honest-verifier zero-knowledge protocol for L.[2] The protocol has a few missing parts and your task is to fill in the details. Also, fill in the details of the simulator S of the constructed honest-verifier, public-coin zero-knoweldge protocol (P, V).

- **Input.** The prover and verifier gets two statements x_0 and x_1 such that $(x_0, x_1) \in L$. The prover additionally obtains a witness (c, w) such that w is a valid witness for $x_c \in L_c$. Note that x_{1-c} may not be in L_{1-c}.
- $P \rightarrow V$: In the first round, prover chooses $\beta_{1-c} \leftarrow \{0,1\}^n$ uniformly at random. It generates the first round message α_c using the prover P_c on input x_c and the witness w. It generates α_{1-c} as **Solution:** $[\mathrm{S}_{1-c}(\beta_{1-c})]_1$. It sends (α_0, α_1) to the verifier.
- $V \rightarrow P$: In the second round, the verifier sends its random coins $\beta \leftarrow \{0,1\}^n$ to the prover.
- $P \rightarrow V$: the last round, the prover generates β_c as **Solution:** $\beta_{1-c} \oplus \beta$.

  It generates the third round message γ_c as **Solution:** $\mathrm{P}_c(x_c, w, \alpha_c, \beta_c)$.

  It generates γ_{1-c} as **Solution:** $[\mathrm{S}_{1-c}(\beta_{1-c})]_2$. It sends $(\beta_0, \beta_1, \gamma_0, \gamma_1)$ to the verifier.
- The verifier accepts if $\beta =$ **Solution:** $\beta_0 \oplus \beta_1$ and $V_0(\alpha_0, \beta_0, \gamma_0) = 1$ and $V_1(\alpha_1, \beta_1, \gamma_1) = 1$.

Give the description of the honest-verifier zero-knowledge simulator for the above protocol.

$S(\beta)$ outputs $(\alpha = (\alpha_0, \alpha_1), \gamma = (\gamma_0, \gamma_1))$ where $\beta_0 \leftarrow \{0,1\}^n$ is chosen uniformly at random,

$\beta_1 =$ **Solution:** $\beta \oplus \beta_1$,

[1] By public-coin we mean that the verifier's messages in the protocol are outcomes of random coin tosses.

[2] Note that this implies that the verifier should not learn where $x_0 \in L_0$ or $x_1 \in L_1$.

$$\alpha_0 = \boxed{\textbf{Solution: } [\mathrm{S}_0(\beta_0)]_1},$$

$$\alpha_1 = \boxed{\textbf{Solution: } [\mathrm{S}_1(\beta_1)]_1},$$

$$\gamma_0 = \boxed{\textbf{Solution: } [\mathrm{S}_0(\beta_0)]_2},$$

$$\gamma_1 = \boxed{\textbf{Solution: } [\mathrm{S}_1(\beta_1)]_2}.$$

4 Pseudorandom Functions (25 points)

1. Assume pseudorandom functions exist. Construct a PRF such that any PPT attacker cannot distinguish between an oracle access to the PRF and an oracle access to a random function but given a single bit of the key, there exists an attacker that can distinguish between these two oracles with non-negligible advantage. Give proofs for all your claims.

Solution: $f_{k,b}(x) = \begin{cases} g_k(x)_{1,\ldots,n-1} \| b & \text{if } x = x^* \\ g_k(x) & otherwise \end{cases}$ where g is a PRF.

2. A weak pseudorandom function $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ is just like any other PRF except that it satisfies a weaker notion of security. Specifically, for any PPT attacker D, there exists a negligible function negl such that:

$$|\Pr[D^{O_k}(1^n) = 1] - \Pr[D^{O'}(1^n)]| \leq \mathsf{negl}(n)$$

where $k \leftarrow \{0,1\}^n$ and O_k is an oracle that on any query, samples a random $x \leftarrow \{0,1\}^n$ and outputs $F_k(x)$. O' is another oracle such that on any query, samples $x \leftarrow \{0,1\}^n$ and outputs $f(x)$ where f is a random function. Notice that the difference between weak PRF and a regular PRF is that in weak PRF, the attacker does not have the ability to choose inputs and learn the output of the function on the chosen inputs. Instead, each time the attacker queries the oracle, he gets the output of the function on randomly chosen inputs.

Construct a CPA-secure secret-key encryption from a weak PRF and prove its security.

Solution:

Name: □

Encryption Scheme:

(a) $\mathsf{Gen}(1^n)$: Sample $k \leftarrow \{0,1\}^n$ and output k.

(b) $\mathsf{Enc}(k,m)$: Sample $r \leftarrow \{0,1\}^n$. Output

$$c := (r, F_k(r) \oplus m)$$

(c) $\mathsf{Dec}(k,c)$: Parse c as $c = (r, c_1)$. Then compute and output

$$m' := F_k(r) \oplus c_1$$

Definition 4.1 (Weak PRF Security)

$\underline{\mathsf{Weak\text{-}PRF\text{-}Game}(1^n, d, \mathcal{A})}$*:*

- *The challenger samples a key $k \leftarrow \{0,1\}^n$ and a function $f : \{0,1\}^n \rightarrow \{0,1\}^n$ uniformly at random.*
- *The adversary $\mathcal{A}$ is given 1^n. $\mathcal{A}$ can make arbitrarily-many oracle queries.*
- *When $\mathcal{A}$ makes an oracle query:*
  - *Pseudorandom Case: If $d = 0$, then the challenger samples $x \leftarrow \{0,1\}^n$, computes $y = F_k(x)$, and sends (x, y) to $\mathcal{A}$.*
  - *Truly Random Case: If $d = 1$, then the challenger samples $x \leftarrow \{0,1\}^n$, computes $y = f(x)$, and sends (x, y) to $\mathcal{A}$.*
- *Finally, $\mathcal{A}$ outputs a bit, which is also the output of the game.*

*F is a **weak PRF** if for any PPT adversary $\mathcal{A}$,*

$$\Big| \Pr\left[\mathsf{Weak\text{-}PRF\text{-}Game}(1^n, 0, \mathcal{A}) \rightarrow 1\right] - \Pr\left[\mathsf{Weak\text{-}PRF\text{-}Game}(1^n, 1, \mathcal{A}) \rightarrow 1\right] \Big| \leq \mathsf{negl}(n)$$

Theorem 4.2 *If F is a weak PRF, then the encryption scheme constructed above satisfies CPA security.*

Proof:

(a) Overview: Let us assume toward contradiction that there is some PPT adversary $\mathcal{A}$ that breaks CPA security for the encryption scheme. Then we will use $\mathcal{A}$ to construct an adversary $\mathcal{B}$ that breaks weak PRF security for F. This is a contradiction because F is known to be a secure weak PRF. Therefore, the initial assumption was false, and in fact, the encryption scheme is CPA-secure.

(b) $\mathcal{B}$ will simulate the CPA security game and run $\mathcal{A}$ as a subroutine.
Construction of $\mathcal{B}$:

   i. The weak PRF challenger samples $k \leftarrow \{0,1\}^n$ and $f : \{0,1\}^n \rightarrow \{0,1\}^n$ uniformly at random. The challenger also takes as input a bit d.

   ii. When $\mathcal{A}$ makes an encryption query on message m, $\mathcal{B}$ handles it as follows:

A. $\mathcal{B}$ makes an oracle query. The weak PRF challenger samples $x \leftarrow \{0,1\}^n$, computes either $y = F_k(x)$ (if $d = 0$) or $y = f(x)$ (if $d = 1$), and sends (x, y) to $\mathcal{B}$.

B. $\mathcal{B}$ computes $c = (x, y \oplus m)$ and sends c to $\mathcal{A}$.

iii. When $\mathcal{A}$ outputs its challenge messages (m_0, m_1), $\mathcal{B}$ samples a bit $b \leftarrow \{0,1\}$ and computes the encryption of m_b using the procedure above. $\mathcal{B}$ sends the resulting ciphertext c^* to $\mathcal{A}$.

iv. $\mathcal{A}$ outputs a guess b' for b. $\mathcal{B}$ checks whether $b = b'$. If so, $\mathcal{B}$ outputs 0, and if not, $\mathcal{B}$ outputs 1.

(c) Pseudorandom Case: If $d = 0$, then $\mathcal{B}$ correctly simulates the CPA security game. In this case, $y = F_k(x)$. So on each encryption query, the ciphertext c is computed from the same distribution as $\mathsf{Enc}(k, m)$. Then

$$\Pr[b = b' | d = 0] = \frac{1}{2} + \mathsf{non\text{-}negl}(n)$$

because $\mathcal{A}$ breaks CPA security for the encryption scheme.

(d) Truly Random Case: We will show that $\Pr[b = b' | d = 1] = \frac{1}{2}$.
With overwhelming probability, no two oracle queries will sample the same x. Then every ciphertext computed during an encryption query takes the form $c = (x, y \oplus m)$, where y is independent of all previous ciphertexts. This is basically one-time pad encryption. Over the randomness of y, c^* is independent of m, so $\mathcal{A}$ has no information about b. Therefore $\Pr[b = b' | d = 1] = \frac{1}{2}$.

(e) In summary, $\mathcal{B}$ breaks the weak PRF security of F because

$$\begin{aligned} &\Big| \Pr\left[\mathsf{Weak\text{-}PRF\text{-}Game}(1^n, 0, \mathcal{B}) \to 1\right] - \Pr\left[\mathsf{Weak\text{-}PRF\text{-}Game}(1^n, 1, \mathcal{B}) \to 1\right] \Big| \\ &= \frac{1}{2} + \mathsf{non\text{-}negl}(n) - \frac{1}{2} \\ &= \mathsf{non\text{-}negl}(n) \end{aligned}$$

■

Name:

3. Consider a PRF $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ where $\mathcal{X}, \mathcal{Y}$ are groups with the group operation denoted by $+$ and $\cdot$ respectively. We say that F is input homomorphic if for any $k \in \mathcal{K}$ and two inputs $x_1, x_2 \in \mathcal{X}$

$$F(k, x_1) \cdot F(k, x_2) = F(k, x_1 + x_2)$$

   (a) Give a construction of collision-resistant hash functions from input homomorphic PRFs. (No need to prove collision-resistance)

Solution: For $n > \log |\mathcal{Y}|$, choose a matrix M randomly from $\mathcal{X}^{2\times n}$ and set the key of the hash function to be $F_k(M)$. On input $x \in \{0,1\}^n$, compute $\prod_i F_k(M_{x_i,i})$ using the homomorphic property.

(b) Give a construction of CPA secure public key encryption from input homomorphic PRFs. Prove its security.

Solution: $(r, F_k(r) \cdot m)$ is a secret key additively homomorphic encryption. This implies an CPA-secure PKE.

5 Private Information Retrieval from DDH (15 points)

In the problem of private information retrieval, there is a server who holds a large database $D \in \{0,1\}^n$. The client wishes to learn the bit in the i-th position of the database, i.e., client wants to learn D_i. The client additionally wishes the server not to learn the position i. One simple way to ensure is to ask the server to send the entire database for every client query. But this has huge communication cost when n is of the order of several thousand GB. We are interested in solutions where the communication cost is substantially lower than the size of the database.

In this problem, we will construct a private information retrieval from DDH where the length of the message from client to server is large (of the order of n) but the communication cost from the server to client is small and independent of the length of the database.

1. Let $(\mathbb{G}, g, p)$ be a DDH-hard group. On input a location $L \in [n]$, the client does the following. It chooses

   $$P = \begin{pmatrix} g_{1,0} & g_{2,0} & \cdots & g_{n,0} \\ g_{1,1} & g_{2,1} & \cdots & g_{n,1} \end{pmatrix}$$

   where each $g_{i,b}$ is chosen uniformly from $\mathbb{G}$. It then chooses a random $r \leftarrow \mathbb{Z}_p$ and computes

   $$Q = \begin{pmatrix} g_{1,0}^r & g_{2,0}^r & \cdots & g_{L-1,0}^r & Q_{L,0} & g_{L+1,0}^r & \cdots & g_{n,0}^r \\ g_{1,1}^r & g_{2,1}^r & \cdots & g_{L-1,1}^r & Q_{L,1} & g_{L+1,1}^r & \cdots & g_{n,1}^r \end{pmatrix}$$

   where $Q_{L,0} =$ **Solution:** $g_{L,0}^r$ and $Q_{L,1} =$ **Solution:** $g \cdot g_{L,1}^r$.

   It sends P, Q to the server.

2. The server on receiving P and Q computes,

   $$h = \prod_{i=1}^{n} P_{i,D_i}$$

   and

   $k =$ **Solution:** $\prod_{i=1}^{n} Q_{i,D_i}$

   and sends (h, k) to the client.

3. The client retrieves D_i as (and argue why it is correct):

Solution: Check if $k = h^r$ and if yes, output 0. Else, output 1.

4. Prove that P, Q hides the location L using the DDH assumption.

Solution: Under the DDH assumption, $(g, g^r, g_{L,1}, g \cdot g_{L,1}^r)$ is indistinguishable to $(g, g^r, g_{L,1}, g^s)$ for a randomly chosen s.

6 Witness Encryption (12 points)

1. A witness encryption scheme for a language $L \in \mathsf{NP}$ is defined as follows:

   - Let x be some statement. The encryption algorithm takes x and a message m and outputs a ciphertext c.
   - The decryption algorithm takes the ciphertext c and a witness w such that (x, w) satisfies the NP relation for L and outputs the message m.

   We require two properties:

   - **Correctness.** For every w such that (x, w) that satisfies the NP relation for L, we require the decryption to output the message m with probability 1.
   - Security. If $x \notin L$, for every two messages m_0, m_1,

   $$\mathsf{Enc}(x, m_0) \approx_c \mathsf{Enc}(x, m_1)$$

   Give a construction of witness encryption scheme for any language $L \in \mathsf{NP}$ using an indistinguishability obfuscator. Prove correctness and security.

   Solution: $\mathsf{Enc}(x, m)$: Output $iO(C_{x,m})$ where $C_{x,m}$ takes a witness w and it outputs m if and only if (x, w) satisfies the NP relation.