CAS CS 538. Solutions to Problem Set 4

Due electronically via gradescope, Tuesday February 17, 2026 11:59pm

Om Khadka, U51801771

Problem 1. (34 points) Problem 3.2 from Boneh-Shoup. Let $\mathcal{E} = (E, D)$ be a cipher defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ where $\mathcal{K} \subseteq \mathcal{M}$. Let $\mathcal{E}' = (E', D')$ be a cipher where encryption is defined as $E'((k_1, k_2), m) := (E(k_1, k_2), E(k_2, m))$. Show that if $\mathcal{E}$ is semantically secure then so is $\mathcal{E}'$. Note that problem is analogous to problem 3 on problem set 1, replacing perfect secrecy with semantic security.
Hint: use a hybrid argument with two intermediate games between Semantic Security Experiment 0 and Semantic Security Experiment 1. You will need to define two other games; let's call them Game h0 and Game h1 ("h" is for "hybrid") . Then you will have three separate proofs:

- Semantic Security Experiment 0 is indistinguishable from Game h0.
- Game h0 is indistinguishable from Game h1.
- Game h1 is indistinguishable from Semantic Security Experiment 1.

We suggest denoting by p_0, p_{h0}, p_{h1}, p_1 the probability that the adversary outputs 1 in each of the games.

Each of these three proofs will be, itself, a reduction. Recall that a reduction itself has two parts: an algorithm and an analysis of advantage (which consists of computing two probabilities and subtracting them). The probability computations will be very simple in this problem.

Solution. The solution to this proof will be (like the hint stated) done via chaining the various changes we do in the encryption scheme, proving and reducing each part until we end up with the final product. By proving each part of the chain is semantically secure, I can prove via transitivity that the end-product will then also be secure.

Let $\mathcal{E}$ be the defined cipher stated above. Then, $\mathcal{E}'$ will defined as the following:

- Key space is $\mathcal{K}' = \mathcal{K} \times \mathcal{K}$
- Encrpytion is $\mathcal{E}'((k_1, k_2), m) = (\mathcal{E}(k_1, k_2), E(k_2, m))$

Let $\mathcal{A}$ be some efficent adversay, that'll attack the semantic security of $\mathcal{E}'$. I will now split this game up into 4 games, bounded to $\mathcal{A}$'s advantage. Also, fix some element $k_0 \in \mathcal{K}$ (since $\mathcal{K} \neq \theta$).

- $(\mathcal{E}(k_1, k_2), \mathcal{E}(k_2, m_0))$
  - Game 0; The actual $\mathcal{E}'$ experiment.
- $(\mathcal{E}(k_1, k_0), \mathcal{E}(k_2, m_0))$
  - Game 1; Encrypting with k_0 instead of k_2
- $(\mathcal{E}(k_1, k_0), \mathcal{E}(k_2, m_1))$
  - Game 2; Encrypting m_1 instead of m_0

- $(\mathcal{E}(k_1, k_2), \mathcal{E}(k_2, m_1))$
  - Game 3; Actual $\mathcal{E}'$ experiment, but now with m_0.

Let W_j denote the chance that $\mathcal{A}$ outputs 1 in game j. Then, by the triangle inequality:

$$\text{SSadv}[\mathcal{A}, \mathcal{E}'] = |\Pr[W_0] - \Pr[W_3]| \leqslant |\Pr[W_0] - \Pr[W_1]| + |\Pr[W_1] - \Pr[W_2]| + |\Pr[W_2] - \Pr[W_3]|$$

I'll go into the reduction of each term to prove $\mathcal{E}$'s semantic security.

Game 0 $\approx$ Game 1
Let $\mathcal{B}_1$ be an adversary attaking $\mathcal{E}$:

- Gets challenge ciphertext $c*$ from $\mathcal{E}'$ (either k_2, k_0).
- Samples $k_2 \leftarrow \mathcal{K}$ independently.
- Computes $c_2 \leftarrow \mathcal{E}(k_2, m_0)$.
- Gives $(c*, c_2)$ to $\mathcal{A}$.
- Outputs the result of $\mathcal{A}$.

Looking at this now, note two key things. Firstly, if $c* = \mathcal{E}(k_1, k_2)$, then $\mathcal{B}_1$ simulates Game 0 perfectly. Also, if $c* = \mathcal{E}(k_1, k_0)$, $\mathcal{B}_1$ will then also simulate Game 1 perfectly. Thus,

$$|\Pr[W_0] - \Pr[W_1]| = \text{SSadv}[\mathcal{B}_1, \mathcal{E}]$$

Game 1 $\approx$ Game 2
Now let $\mathcal{B}_2$ attack $\mathcal{E}$:

- Gets challenge ciphertext c^* from $\mathcal{E}'$ (either m_0, m_1 under key of k_2).
- Samples $k_1 \leftarrow \mathcal{K}$ independently.
- Computes $c_1 \leftarrow \mathcal{E}(k_1, k_0)$.
- Gives (c_1, c^*) to $\mathcal{A}$.
- Outputs result of $\mathcal{A}$.

Now look at this game. If $c^* = \mathcal{E}(k_2, m_0)$, then $\mathcal{B}_2$ will perfectly simulate Game 1. In the same manner, if $c^* = \mathcal{E}(k_2, m_1)$, then $\mathcal{B}_2$ will perfectly simulate Game 2. Thus,

$$|\Pr[W_1] - \Pr[W_2]| = \text{SSadv}[\mathcal{B}_2, \mathcal{E}]$$

Game 2 $\approx$ Game 3
Let $\mathcal{B}_3$ attack $\mathcal{E}$ now (symmetric to $\mathcal{B}_1$ btw).

- Gets challenge ciphertext c^* from $\mathcal{E}'$ (either k_0, k_2).
- Samples $k_2 \leftarrow \mathcal{K}$ independently, then compute $c_2 \leftarrow \mathcal{E}(k_2, m_1)$.
- Give (c^*, c_2) to $\mathcal{A}$.
- Output result of $\mathcal{A}$.

This game also ends up with a similar result as the previous games; If $c^* = \mathcal{E}(k_1, k_0)$, then $\mathcal{B}_3$ simulates Game 2 perfectly, and if $c^* = \mathcal{E}(k_1, k_2)$, then $\mathcal{B}_3$ simulates Game 3 perfectly. Thus,

$$|\Pr[W_2] - \Pr[W_3]| = \mathsf{SSadv}[\mathcal{B}_3, \mathcal{E}]$$

By combining these 3 bounds:

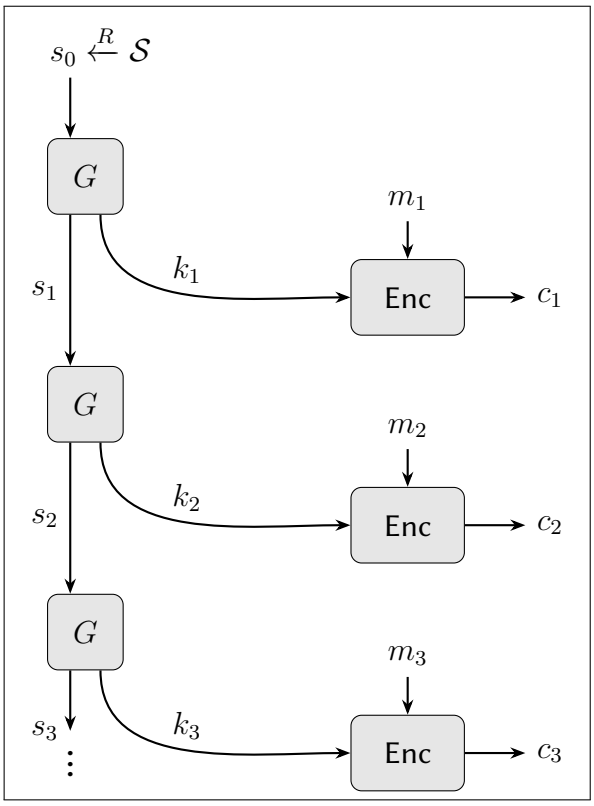
$$\mathsf{SSadv}[\mathcal{A}, \mathcal{E}'] \leqslant \mathsf{SSadv}[\mathcal{B}_1, \mathcal{E}] + \mathsf{SSadv}[\mathcal{B}_2, \mathcal{E}] + \mathsf{SSadv}[\mathcal{B}_3, \mathcal{E}]$$

And since $\mathcal{E}$ is semantically secure, then each $\mathsf{SSadv}[\mathcal{B}_i, \mathcal{E}]$ will then be negligible. Thus, the sum wil lalso be neglibile, and $\therefore \mathsf{SSadv}[\mathcal{A}, \mathcal{E}']$ is negligible from all efficent $\mathcal{A}$, proving the initial statement.

Problem 2. NOTE: We are postponing this problem. This problem will appear on Discussion 5 on Feb 18; a different problem related to the symmetric ratchet will appear on the next homework assignment. So if you have done this problem, you have not wasted time.

The Signal protocol is the most widely used end-to-end encrypted messaging protocol in the world, used by over a billion people daily. It's the core cryptography underneath WhatsApp, Google Messages, Facebook Messenger, and the eponymous Signal app.

An important property Signal provides is *forward secrecy*: each time a message is sent, the encryption key is changed so that message remains hidden from this moment forward, even if an attacker obtains future encryption keys. The mechanism Signal uses to update the keys is called a *symmetric ratchet*[1]. The core ingredients to a symmetric ratchet are a PRG G and a semantically secure cipher $(\mathsf{Enc}, \mathsf{Dec})$.



Symmetric Ratchet

$s_0 \xleftarrow{R} \mathcal{S}$
for $i \twoheadleftarrow 1$ to ∞:
  $(s_i, k_i) \twoheadleftarrow G(s_{i-1})$
  Delete s_{i-1} from memory
  Use k_i to encrypt/decrypt the i-th message
  Delete k_i from memory

Figure 1: The Signal symmetric ratchet

When Alice and Bob want to communicate with a symmetric ratchet, they begin by agreeing on a uniformly random PRG seed s_0. If Alice is sending the first message, she runs the seed s_0 through G to obtain a new seed s_1 as well as a key k_1. She encrypts her message under k_1 with Enc and send the ciphertext to Bob. Now Bob runs his copy of s_0 through G to learn the same key k_1 and decrypt the message. This process can repeat essentially forever: with each new message, Alice and Bob use the PRG to update their state and produce a fresh encryption key. Every time Alice or Bob updates their state with G, they delete the previous state and key from memory. This ensures that if they are compromised, the attacker only learns the current state.

[1]The full Signal protocol is sometimes called the double ratchet protocol, because it interleaves a symmetric ratchet with another mechanism called an asymmetric ratchet. The asymmetric ratchet provides backwards security, meaning that messages are secure from key compromises that occurred in the past.

This mechanism is called a *ratchet* because it's easy to advance the state forward, but it's hard to recover a previous state once you've deleted it from memory.

In this problem you'll show that the symmetric ratchet has a property necessary for forward secrecy: specifically, that there is no efficient way to learn a past key given a later state and key. (This property is necessary, but not sufficient, because what you really want is that past encryptions are still semantically secure; but we are giving you a simpler problem here.) Let $G : \{0,1\}^\ell \to \{0,1\}^{2\ell}$ be a secure PRG. For all $n \in \mathbb{N}$, let $G^n : \{0,1\}^\ell \to \{0,1\}^{(n+1)\ell}$ be the n-wise sequential composition of G, i.e.

$G^n(s)$

$s_0 \twoheadleftarrow s$
for $i \twoheadleftarrow 1$ to n:
  $(k_i, s_i) \twoheadleftarrow G(s_{i-1})$
output $(k_1, \ldots, k_n, s_n)$

Prove that there does not exist a PPT algorithm $\mathcal{A}$ such that $\Pr[\mathcal{A}(k_n, s_n) == k_{n-1}]$ is non-negligible, when k_n, s_n, and k_{n-1} are values produced by G^n with a random seed.

You can use the fact that G^n is a secure PRG, as proven in section 3.4.2 of the textbook. Suppose there exists such an algorithm $\mathcal{A}$, and use it to construct an adversary that breaks the PRG game for G^n.

Solution. *Skipped for now.*

Problem 3. (66 points, at 8 each except (c) which is 2)

Your proofs for this problem may rely only on the theorems we provide below (note the optional references to their proofs in Shoup's A Computational Introduction to Number Theory and Algebra https://shoup.net/ntb) and problems from Discussion 3, which is posted also under Piazza resources. In particular, you may not (and do not need to) invoke other theorems from number theory or abstract algebra. When proving each part, it may be very helpful to use previous parts. The parts will be graded independently, so you can use the previous parts even if you have not solved them.

Theorem 1 (Division theorem, Shoup Theorem 1.4)**.** *Let $a, b \in \mathbb{Z}$ with $b > 0$. Then there exists unique $q, r \in \mathbb{Z}$ such that $a = bq + r$ and $0 \leqslant r < b$. (q is called "quotient" and r is called "remainder".)*

A *divisor* d of a is any integer d such that there exists an integer q with $qd = a$. A *common* divisor of a and b is an integer that is a divisor of a and a divisor of b.

Theorem 2 (Greatest common divisor theorem, Shoup Theorem 1.7)**.** *Let $a, b \in \mathbb{Z}$. Unless $a = b = 0$, the set of common divisors of a and b has a maximum, called the greatest common divisor of a and b and denoted by $\gcd(a, b)$. We will say that a and b are* relatively prime *whenever $\gcd(a, b) = 1$.*

Theorem 3 (Shoup Theorem 4.4)**.** *Let $a, b \in \mathbb{Z}$. The* extended Euclidean algorithm *computes $d, s, t \in \mathbb{Z}$ in time polynomial in $\log a \cdot \log b$, such that $d = \gcd(a, b)$ and*

$$as + bt = d. \tag{1}$$

Equation (**??**) is also known as the Bézout's identity, and the integers s and t are called Bézout's coefficients for (a, b).

Thus, if a and n are relatively prime, then Extended Euclidean Algorithm gives us s and t such that $as + nt = 1$, so $as = 1 - nt \equiv_n 1$, and therefore $s \bmod n \in \mathbb{Z}_n$ is the number that, when multiplied by a, is congruent to 1 modulo n. We will call s a *multiplicative inverse of a modulo n*, write $s = a^{-1} \bmod n$, and, when working modulo n, we will use the term "divide by a" to mean "multiply by s" (just like in arithmetic over real numbers).

Note that division does not always exist — it exists if and only if a and n are relatively prime.

Let p be a prime. Since every element of $\mathbb{Z}_p^*$ is relatively prime with p whenever p is prime, inverses for elements of $\mathbb{Z}_p^*$ always exist.

(a) Prove that for any integer $a \in \mathbb{Z}_p^*$, the values $a, 2a, 3a, ..., (p-1)a$ (all reduced modulo p) hit every element in the set $\mathbb{Z}_p^* = \{1, 2, 3, \ldots, p-1\}$ exactly once.

Hint: think of multiplication by a modulo p as a function. The statement is simply asking you to prove that it is a bijection. Since the domain and the target are finite and of the same size, it suffices to prove that it is a surjection (onto) or injection (1-to-1). You can choose whether to prove surjectivity or injectivity, and them simply write "a surjective (respectively, injective) function between domain and range of the same finite size is bijective." The existence of multiplicative inverses proven could be useful.

Solution. Your solution goes here

(b) Fermat's little theorem: prove that for any $a \in \mathbb{Z}_p^*$, $a^{p-1} \equiv_p 1$. (Or equivalently, $a^{p-1} = 1$ if you consider multiplications modulo p. Here, we write $\equiv_p$ to emphasize the underlying modulus.)

Hint: multiply together the $(p-1)$ values $a, 2a, 3a, \ldots, (p-1)a$ and use part (a).

Solution. Your solution goes here

(c) Watch this video https://www.youtube.com/watch?v=sB8vIbsssUo. Pronounce Fermat correctly as the video explains. We will give you two points on an honor system if you tell us you did it.

Solution. Your solution goes here

(d) Let $a \in \mathbb{Z}_p^*$. The *order* of a, denoted by $\mathrm{ord}(a)$ is the smallest positive integer k such that $a^k \equiv_p 1$. Prove that k divides $p-1$.

Hint: do a proof by contradiction. Let $p-1 = kq + r$, where $q, r \in \mathbb{Z}$ and $0 \leqslant r < k$ by the division theorem. Consider a^r.

Solution. Your solution goes here

(e) Again, let $\mathrm{ord}(a) = k$. Prove that the k powers of a modulo p (namely $\{1 = a^0, a = a^1, a^2, \ldots, a^{k-1}\}$) are all distinct.

Solution. Your solution goes here

(f) Prove that if the order of a is k and $x \equiv_k y$ for some integers x and y, then $a^x \equiv_p a^y$ (note the different subscripts of $\equiv$).

Solution. Your solution goes here

(g) Prove the converse of part (f): if the order of a is k and $a^x \equiv_p a^y$ for some integers x and y, then $x \equiv_k y$ (note the different subscripts of $\equiv$). Thus combining this part and the previous part, you know that for any a of order k, exponents work modulo k.

Solution. Your solution goes here

(h) Let $b = a^x \bmod p$. Let c be the multiplicative inverse of b. Prove that c is a nonnegative power of a: that is, prove that there exists a nonnegative integer y such that $a^y \equiv_p c$.

Solution. Your solution goes here

(i) Suppose g has order $p-1$ (we know such a g exists for any prime p, though we have not proven this fact). Suppose $k \mid p-1$. We want to show that there exists an element of order k. Specifically, show that $g_k := g^{(p-1)/k}$ has order k.

Solution. Your solution goes here

For any k that divides $p-1$, we can consider the set G_k of powers of g_k. This set has size k (because powers of g_k repeat after k multiplications by g_k). Multiplying elements of G_k gives you other elements of G_k (because $g_k^x \cdot g_k^y = g_k^{x+y}$), as does dividing (by part (h)). Thus, G_k is a cyclic group of order k: "cyclic" because it consists of all the powers of one element g_k, "group" because it is closed under multiplication and division, "of order k" because it contains k elements. We will use such groups throughout the semester. The value g_k is called a *generator* of G_k, and G_k is sometimes denoted by $\langle g_k \rangle$.