ECE498AM Final

Instructor: Andrew Miller

December 10, 2019

Name _____ NetID _____

Instructions

Please either 1) typeset your answers in LaTeX and submit a gradescope PDF file or else 2) write answers by hand and turn in a upload a scanned PDF file on gradescope. We prefer to read succinct and precise answers. If you can be precise while being succinct with your answers, please try. Total points = 125

Due date: 11:59pm, Dec 10, 2019.

Academic integrity reminder: We treat academic integrity very seriously. You are supposed to do this exam individually. You may refer to lecture notes, optional textbooks, or internet searches. However, you should not talk about the problems with peers or ask for help online.

How multiple choice is graded. Multiple choice questions may have multiple correct answers. If there are N multiple correct answers, then each correct answer is worth $1/N$ of the total points for the question. You lose $1/N$ points for every wrong answer you circle. The total cannot go negative. In code, `score = points * max(num_correct - num_wrong, 0) / total_correct`

Clarity, succinctness, writing your name and Netid: [5 pts].

1 Fault Tolerant Broadcast Protocols [14pts]

Consider the following definition of a protocol for Fault Tolerant Broadcast (Byzantine Generals) in a synchronous network. In this setting there are n nodes, and up to t of them may be Byzantine, where $n = 3t+1$.[1] One of the nodes, the Leader, which may or may not be one of the Byzantine nodes, starts out with an input $v \in \{0, 1, ..., m\}$. The goal of the protocol is to reach agreement on the Leader's value.

The desired security properties are:

- Safety: all honest parties decide on the same value v (if they decide)
- Validity: if the leader is honest, all parties decide on its input (if they decide)
- Liveness: if the leader is honest, all honest parties decide

The protocol definition is:

- Leader(v) does the following (v is the leader's input):
  - send PRE-PREPARE(v) to every other party
- Party p_i does the following:
  - on receiving PRE-PREPARE(v) from Leader for some value v:
    send PREPARE(v) to every party (only if PREPARE hasn't been sent yet)
  - on receiving PREPARE(v) messages from $2t + 1$ parties for some value v:
    send COMMIT(v) to every party (only if COMMIT hasn't been sent yet)
  - after receiving COMMIT(v) messages from $t + 1$ parties,
    decide(v)

1.1 Reasoning about protocols [7pts]

Which of the following statements are true? (circle the number of the statements that are true):

1. Whether or not the Leader is honest, every honest party receives at least one PRE-PREPARE message
2. Whether or not the Leader is honest, every honest party eventually receives PREPARE(v) messages from at least $t + 1$ nodes
3. Whether or not the Leader is honest, if p_i receives $2t + 1$ PREPARE messages, then $t + 1$ of these are for the same value v
4. If the Leader is honest, then p_i receives PREPARE messages from at least $2t + 1$ parties for the same value v
5. At least $t + 1$ nodes are honest
6. If p_i received COMMIT messages from $t + 1$ parties, then at least t of those parties are honest
7. An honest party could send $PREPARE(0)$ and also later send $PREPARE(1)$

1.2 Complete the safety proof[7pts]

A proof of the safety property begins:

Proof. Suppose for contradiction that in a run of this protocol, an honest party p_i receives 1) PREPARE(0) messages from a subset of parties P, with $|P| = 2t + 1$, and also 2) PREPARE(1) messages from a subset of parties Q, with $|Q| = 2t + 1$.

Complete this proof by explaining how to reach a contradiction.

[1] This $n = 3t + 1$ is not the optimal fault tolerance in the synchronous setting, but it makes this question easier to assume.

2 Lattice Based Attacks: Biased Nonce Sense [14pts]

We studied lattice based against poorly implemented RSA ecryption schemes. This paper by Nadia Heninger and Joachim Breitner uses similar techniques to find bitcoin private keys from signatures. Biased Nonce Sense: Lattice Attacks against Weak ECDSA Signatures in Cryptocurrencies [2]. Consider the following signature scheme: The public domain parameters for an elliptic curve digital signature include an elliptic curve E over a finite field and a base point g of order p on E.

2.1 [4pts]

Fill in the verify method for the given signature scheme based on section 3 in the paper.

- KeyGen$(1^\lambda, r) \rightarrow$ pk, sk: takes in a security parameter λ and randomness r. Samples sk acoording to randomness r and returns a tuple consisting of public key and secret key as (g^{sk}, sk)
- Sign(pk, sk, h, r) : $\rightarrow \sigma$ takes in a public key pk, secret key sk, message hash h and m outputs a signature σ as follows:

$$k_1 \xleftarrow{\$} \{0,1\}^{32}$$

$$k = \lfloor \frac{SHA256(\text{"}HelloWorld!\text{"})}{2^{32}} \rfloor + k_1$$

$$Q = g^k$$

$$(x_r, y_r) = Q$$

$$(r, s) \leftarrow (x_r, k^{-1}(h + sk * x_r))$$

- Verify() $\rightarrow$ {0,1}

2.2 [4pts]

The introduction of the paper starts with "if an ECDSA private key is ever used to sign two messages with the same signature nonce, the private key is trivial to compute". Clearly describe how to compute the private key from the two signed messages with the same nonce.

[2] https://eprint.iacr.org/2019/023.pdf

2.3 [4pts]

In our class, we looked at LLL and coppersmith algorithms for figuring out RSA private keys. This paper uses a similar method called the "hidden number problem". Describe what is a hidden number problem and create a instance of hidden number problem based on signatures (r_i, s_i) on messages h_i.

2.4 [2pts]

What is the suggested counter measure for this attack?

3 Reading Cryptography Library Documentation [12pts]

The crypto library NaCL(pronounced "salt") is a modern cryptographic library, that applies many practical engineering choices. These are discussed in detail in the technical paper, "The security impact of a new cryptographic library." by Daniel J. Bernstein, Tanja Lange, and Peter Schwabe.[3]. NaCL is not maintained anymore, but a fork a it, libsodium [4] is actively in use. Refer to the technical paper and the website documentation to answer the following questions.

3.1 Libsodium includes an implementation of the following crypto primitives (circle all that apply):

1. **Public key signatures** a. RSA b. ECDSA c. EdDSA
2. **Symmetric encryption** a. AES b. Salsa20 c. ChaCha20
3. **Hash functions** a. SHA-1 b. SHA-256 c. SHA-512 d. MD5 e. BLAKE2b

[3] `https://cr.yp.to/highspeed/coolnacl-20120725.pdf`
[4] `https://libsodium.gitbook.io/doc/`

3.2 The `crypto_box_easy` interface provides which of the following security properties:

a. Authentication b. Encryption c. Replay prevention d. Forward secrecy

3.3 Identify four specific design decisions made in NaCL to improve security.

Note: By *specific*, I mean that just saying "sidechannels" does not count!

4 Threshold Cryptography and Secret Sharing [14pts]

Alice wishes to split her Crypto Egg private key into three backup copies. She uses the Shamir's Secret sharing program (SSSS) to generate three files. She writes down one of them on a piece of paper and stores it in her closet. She keeps one on a USB drive she carries in her pocket. She sends one of them to her trusted friend Bob. Bob decides to use SSSS again to split up his share for Alice's key into four shares such that he only needs two of them to get back the Alice's share. Alice receives an anonymous message that appears to be encrypted using her Crypto Egg public key.

4.1 [2pts]

Describe the steps Alice must take to decrypt the message. Do not include any more steps than necessary!

4.2 [2pts]

Describe a scenario where enough of Alice's key shares are stolen so an attacker can decrypt the message.

4.3 [2pts]

Describe a scenario where Alice loses enough keys that she cannot decrypt the message.

4.4 [2pts]

Recall that Shamir's Secret Sharing represents a secret by a polynomial over a finite field. Fill in the blanks. The degree of the Alice's polynomial in this case is ____. Alice's configuration is referred to as ____-of-____ secret sharing. The degree of the Bob's polynomial in this case is ____. Bob's configuration is referred to as ____-of-____ secret sharing.

4.5 [3pts]

What is the unique degree-bound 5 polynomial f over the finite field $\mathbb{F}_{59}$ that satisfies the following constraints:

$$\begin{aligned} f(0) &= 0 \\ f(1) &= 28 \\ f(2) &= 54 \\ f(3) &= 29 \\ f(4) &= 39 \\ f(5) &= 10 \end{aligned}$$

You can do this by hand or using any tools you like.

4.6 [3pts]

How many degree-3 (or smaller degree) polynomials of $\mathbb{F}_{59}$ satisfy the constraints $f(0) = 0$, $f(1) = 37$, and $f(2)$ is even?

5 Interactive Proofs [20 pts]

This question involves a variant of the Sigma protocols for Zero Knowledge Proofs discussed in class. You'll have to work through the protocol construction, definitions, and proofs.

Let $\mathcal{G}_\lambda$ be a family of groups in which the discrete log problem is hard, and the order of each group in the family is $|\mathcal{G}_\lambda| = p_\lambda = 2^{\mathsf{poly}(\lambda)}$. g and h are generators of the cyclic group amongst whom the discrete log relation is unknown.

Alice knows three secret keys $a \xleftarrow{\$} \mathbb{Z}_p, b \xleftarrow{\$} \mathbb{Z}_p, c \xleftarrow{\$} \mathbb{Z}_p$, such that her public keys are $A = g^a, B = g^b, C = g^c$. (Assume that Alice posted A, B, C publicly on Piazza).

Alice posts a commitment D on Piazza and claims that $D = g^{ab}h^c$. Help Alice create a zero knowledge proof to prove this, without revealing any other information a, b and c?

5.1 [4 pts]

Illustrate an ideal functionality below that could carry out this protocol:

5.2 [4 pts]

Write the goal for the necessary ZK proof scheme using Camenisch-Stadler notation.

$$ZK\{(\_) : \_\}$$

Simplify or rearrange the C-S specification at this point to simplify the next steps.

Hint: It may help to create one or more ephemeral commitments.

5.3 [4 pts]

Finish constructing the protocol (P, V) below, where P is the prover and V is the verifier, such that $\left[P(1^\lambda, a) \leftrightarrow V(1^\lambda, A)\right]$ emulates the ideal functionality .

1. Round 1 (commit):
2. Round 2 (challenge): V sends the challenge ch to P where

$$ch \xleftarrow{\$} \mathbb{Z}_p \backslash \{0\}$$

3. Round 3 (response): What does P send to V in response?
4. Verification: What does V do with the response?

5.4 [4 pts]

Define the "Honest Verifier Zero Knowledge" (aka "Simulatable") property for this scheme, in terms of View_P and View_V. Be sure to explain what the views consist of.

Give a construction for the simulator and prove it satisfies the definition.

5.5 [4 pts]

Define the "Extractability" property for this scheme.

Give a construction for the extractor $\mathcal{E}$ and prove it extracts correctly with non-negligible probability.

6 Symmetric Encryption Security Definitions [16pts]

This question uses the following security definitions for symmetric encryption:

- **Indistinguishability under Chosen Plaintext (IND-CPA).** In this setting, the adversary has access to just an encryption oracle.

$$\forall \mathcal{A}. \Pr\left[\begin{array}{rl} k \leftarrow \mathsf{Gen}(1^\lambda); & \\ (m_0, m_1) \leftarrow \mathcal{A}^{\mathsf{Enc}_k(\cdot)}(1^\lambda); & \\ b \xleftarrow{\$} \{0,1\}; & \\ c \leftarrow \mathsf{Enc}_k(m_b); & \\ b' \leftarrow \mathcal{A}^{\mathsf{Enc}_k(\cdot)}(1^\lambda, c) & : b = b' \end{array}\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda)$$

- **Indistinguishability under Chosen Ciphertexts (IND-CCA1).** In this setting, the adversary has access to both an encryption oracle and a decryption oracle before the challenge.

$$\forall \mathcal{A}.\, \Pr\left[\begin{array}{r} k \leftarrow \mathsf{Gen}(1^\lambda); \\ (m_0, m_1) \leftarrow \mathcal{A}^{\mathsf{Enc}_k(\cdot),\mathsf{Dec}_(\cdot)}(1^\lambda); \\ b \xleftarrow{\$} \{0,1\}; \\ c \leftarrow \mathsf{Enc}_k(m_b); \\ b' \leftarrow \mathcal{A}^{\mathsf{Enc}_k(\cdot)}(1^\lambda, c) \quad : b = b' \end{array}\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda)$$

- **Indistinguishability under Chosen Ciphertexts (IND-CCA2).** In this setting, the adversary has access to both an encryption oracle and a decryption oracle even after the challenge.

$$\forall \mathcal{A}.\, \Pr\left[\begin{array}{r} k \leftarrow \mathsf{Gen}(1^\lambda); \\ (m_0, m_1) \leftarrow \mathcal{A}^{\mathsf{Enc}_k(\cdot),\mathsf{Dec}_(\cdot)}(1^\lambda); \\ b \xleftarrow{\$} \{0,1\}; \\ c \leftarrow \mathsf{Enc}_k(m_b); \\ b' \leftarrow \mathcal{A}^{\mathsf{Enc}_k(\cdot),O(k,c,\cdot)}(1^\lambda, c) \quad : b = b' \end{array}\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda)$$

  where the oracle O allows the adversary to see the decryption of an arbitrary ciphertext except for the challenge ciphertext c.

$$O(k, c, c') := \begin{cases} \bot & c = c' \\ \mathsf{Dec}_k(c') & c \neq c' \end{cases}$$

6.1 Application Analysis [8pts]

Alice, a student at Indiana University, is designing a Piazza-based service for her class which allows them to bet on college football games.[5] The way Alice has implemented her service, she creates a post the day before each game. Students submit their bets by posting an encrypted message containing a guess of the scores, (e.g., "7-2") in a reply to the post. When the game starts, the post is locked, and after the game has concluded, Alice automatically decrypts all the bets and posts them back in a reply so that everyone can keep track of who's doing well.

Alice already has a well-known public key, which she uses throughout the semester. In the initial version of her service, she chooses a public-key authenticated encryption scheme that is known to be IND-CPA secure.

Note: Alice already anticipated one potential problem involving replay attacks. To prevent Bob from copying someone else's bid, if the same encryption shows up multiple times in any of the posts on Piazza, she discards everything except the one with the earliest timestamp. Assume that this works OK - there is no way for an attacker to "front-run" an honest user's post and sneak a reply in with an earlier timestamp.

Explain to Alice why IND-CPA is not enough to guarantee that Bob's encrypted bets stay confidential until the game is over. Give a construction of an IND-CPA encryption scheme and describe a scenario in which an attacker learns one of Bob's bets early.

[5] According to this website, college sports betting is legal in Indiana but not in Illinois, though we aren't sure about the details. It's just an example, use your imagination but don't get in trouble. https://www.legalsportsreport.com/sports-betting/ncaaf/

Explain to Alice why IND-CCA1 also does not guarantee that Bob's encrypted bets stay confidential. Give a construction for an IND-CCA1 encryption scheme and describe a scenario in which an attacker learns one of Bob's bets early.

Hint: you may assume that there are same days in the season in which the football team plays a double-header (two games on the same day). In this case, there would be two separate Piazza posts on the same day, bets can be collected during both.

In her second version, Alice uses an encryption scheme that is IND-CCA2 secure. Explain why this is appropriate to ensure confidentiality.

6.2 Security proof of encryption scheme [8pts]

Let $f : \{0,1\}^\lambda \times \{0,1\}^* \to \{0,1\}^\lambda$ be a pseudorandom function family. Consider the following encryption scheme:

- $\mathsf{Gen}(1^\lambda)$: $k_1 \xleftarrow{\$} \{0,1\}^\lambda, k_2 \xleftarrow{\$} \{0,1\}^\lambda$, return (k_1, k_2)
- $\mathsf{Enc}_{k_1,k_2}(m)$:
  sample $r \leftarrow \{0,1\}^\lambda$
  $c_1 \leftarrow f_{k_1}(r \| m)$
  $c_2 \leftarrow m \oplus f_{k_2}(r)$
  output (r, c_1, c_2)

- $\mathsf{Dec}_{k_1,k_2}((r, c_1, c_2))$:
  Fill in an appropriate decryption scheme yourself

  TODO: Your answer goes here

Prove that this scheme is secure under IND-CCA2.

Given an adversary A that breaks _____ we need to construct an $\mathcal{A}'$ that breaks _____.

Define how requests to the encryption and decryption oracles are handled.

Use a sequence of hybrid games and reduction proofs to show that $\mathcal{A}'$ is as successful as $\mathcal{A}$.

Hint: The proof in Pass and Shelat, **7.1.3 A CCA2-Secure Encryption Scheme** can be used as a starting point. However, the scheme in this question is slightly different, so the proof must be adapted. Similar to this proof, you can assume that scheme from **99.1: Many-message Encryption Scheme** is IND-CPA secure.

7 Hash-based data structures [20pts]

Consider the following one-time signature scheme, inspired by Winternitz. Assume that $\mathcal{H} : \{0,1\}^* \to \{0,1\}^\lambda$ is a hash function modeled as a random oracle. This one-time signature scheme is for signing a single message in the message space $\mathcal{M} = \{0, 1, 2, ..., 7\}$, in other words a three-bit message.

- Define two hash functions using the tags `L` and `R`, so $\mathcal{H}_{\mathtt{L}}(x) = \mathcal{H}(\text{“}\mathtt{L}\text{”}\|x)$, and $\mathcal{H}_{\mathtt{R}}(x) = \mathcal{H}(\text{“}\mathtt{R}\text{”}\|x)$.
- $\mathsf{KeyGen}(1^\lambda)$ returns sk, pk:

  $\mathsf{sk} \leftarrow \{0,1\}^\lambda$

  $\mathsf{pk} := (\mathcal{H}^8_{\mathtt{L}}(\mathsf{sk}), \mathcal{H}^8_{\mathtt{R}}(\mathsf{sk}))$

  return sk, pk
- $\mathsf{Sign}(\mathsf{sk}, m \in \mathcal{M})$ returns σ

  $m_{\mathtt{L}} := 8 - m$

  $m_{\mathtt{R}} := m$

  return $(\mathcal{H}^{m_{\mathtt{L}}}_{\mathtt{L}}(\mathsf{sk}), \mathcal{H}^{m_{\mathtt{R}}}_{\mathtt{R}}(\mathsf{sk}))$
- $\mathsf{Verify}(\mathsf{pk}, m, \sigma)$ returns $\{0,1\}$

  $v_{\mathtt{L}} := m$

  $v_{\mathtt{R}} := 8 - m$

parse pk as $(\mathsf{pk}_{\mathtt{L}}, \mathsf{pk}_{\mathtt{R}})$

parse σ as $(\sigma_{\mathtt{L}}, \sigma_{\mathtt{R}})$

return 0 if $\mathcal{H}_{\mathtt{L}}^{v_{\mathtt{L}}}(\sigma_{\mathtt{L}}) \neq \mathsf{pk}_{\mathtt{L}}$

return 0 if $\mathcal{H}_{\mathtt{R}}^{v_{\mathtt{R}}}(\sigma_{\mathtt{R}}) \neq \mathsf{pk}_{\mathtt{R}}$

return 1 otherwise

7.1

This scheme is defective. Explain a scenario in which a forgery attack can occur.

7.2

Give a formal definition of unforgeability. Fix this scheme without changing the size of the keys or signatures.

8 Design question [10 points]

In order to attract customers, most Cafes have started a loyalty program. A user can register to a loyalty program at a cafe with their email address. At the UIUC campus, we have Cafe Kopi and Espresso Royale who have started offering such a loyalty program. Owners of Cafe Kopi and Espresso Royale decide to create a new promotion for users who are a member of both loyalty programs.

Design a protocol that both Cafes can follow in order to implement the joint loyalty problem, satisfying the following goals:

- Starting out, assume that Cafe Kopi and Espresso Royale each have a list of customers that have already registered for their individual programs. To simplify the problem, you can assume that a correct commitment to each list has already been posted in a public place.
- Cafe Kopi and Espresso Royale should both learn which email addresses are a member of both programs.
- If a user Alice is *not* a member of Cafe Kopi's loyalty program, then Cafe Kopi should not learn any information about whether or not Alice is a member of Espresso Royale's loyalty program, and vice versa.

You can use any cryptography primitives you have learned throughout the course to solve this problem.

8.1 Semi-honest version:

You may assume that Cafe Kopi and Espresso Royale both follow whatever protocol you provide, but you must still ensure the privacy guarantees.

8.2 Malicious case version:

Now consider that either Cafe may deviate from the protocol in order to interfere with the results or compromise the privacy. First, describe an attack that would allow Espresso Royale to violate the privacy property. Next, describe a way to improve the scheme to withstand such attacks.

8.3 Realistic case:

Suppose that we could not assume that the correct lists of registered customers were already committed to in public. What other ideas do you have about how to prevent manipulation here? Feel free to make additional assumptions, but justify why they are realistic and why they would not compromise the intended privacy and functionality goals.