

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF WISCONSIN-MADISON
Math/CS/E C E 435 — Final Exam — Spring 2026

NAME: (as it appears on Canvas)

EMAIL: @wisc.edu

PROFESSOR: MIKHAIL IVANOV

INSTRUCTIONS

- Time: **120 minutes**.
- This exam paper contains 12 pages.
- Please write your name on every page.
- No Calculators, No Notecards, No Notes
- With the exception of the True/False questions, Multiple Choice questions, and Short Answer questions you must justify your claims and use complete sentences in proofs.
- You must use correct notation to receive full credit.
- For multiple choice questions with answers listed by $\bigcirc$, choose one answer and completely fill the circle.

Modular arithmetic

Number a has a multiplicative inverse modulo n (i.e. $a \in \mathbb{Z}_N^*$) if and only if $\gcd(a, N) = 1$.

Shift cipher

$P = C = K = \mathbb{Z}_N$, encryption: $e_k(x) = x + k \bmod N$, decryption: $d_k(y) = y - k \bmod N$.

Affine cipher

$P = C = \mathbb{Z}_N$, $K = \mathbb{Z}_N^* \times \mathbb{Z}_N$, $k = (a, b)$, encryption: $e_k(x) = ax + b \bmod N$, decryption: $d_k(y) = a^{-1}(y - b) \bmod N$.

Monoalphabetic substitution cipher

$P = C = \mathbb{Z}_N$, $K = \Sigma_N$, $k = \sigma \in K$ is a permutation of the alphabet, encryption: $e_\sigma(x) = \sigma(x)$, decryption: $d_\sigma(y) = \sigma^{-1}(y)$.

Vigenère cipher

$P = C = K = \mathbb{Z}_N^n$, encryption: $e_k(x)_i = x_i + k_i \bmod N$, decryption: $d_k(y)_i = y_i - k_i \bmod N$.

Hill cipher

$P = C = \mathbb{Z}_N^n$, key space consists of invertible $n \times n$ matrices with coefficients from $\mathbb{Z}_N$, i.e. matrices K such that $\gcd(\det(K), N) = 1$, encryption: $e_K(X) = KX$, decryption: $e_K(Y) = K^{-1}X$.

Cryptanalytic attack types:

- Ciphertext-only attack
- Known-plaintext attack
- Chosen-plaintext attack

Electronic Codebook Mode (ECB)

$y_i = e_k(x_i)$, $n \geq 1$

Cipher Block Chaining mode (CBC)

$y_0 = \text{IV}$, $y_n = e_k(x_n \oplus y_{n-1})$ for $n \geq 1$.

Counter Mode (CTR)

$y_n = e_k(\text{IV} \parallel n) \oplus x_n$, for $n \geq 0$

Addition on Elliptic curve $y^2 = x^3 + ax + b$

$$k = \frac{y_2 - y_1}{x_2 - x_1} = \frac{3x^2 + a}{2y} \bmod p, \quad x_3 = k^2 - x_1 - x_2 \bmod p, \quad y_3 = k(x_1 - x_3) - y_1 \bmod p$$

ECDSA

Public/private keys: An elliptic curve $E_{a,b}(\mathbb{Z}_p)$, a point A that generates a cyclic group of prime order q , a random integer d with $0 < d < q$, $B = dA$. Public key: p, a, b, A, q, B . Private key: d .

Signature of message m : a random ephemeral key k_E with $0 < k_E < q$, $R = k_E A = (x_R, y_R)$, $r = x_R$, $s \equiv (h(m) + d \cdot r)k_E^{-1} \bmod q$. Signature (r, s) .

Verification: $P = (s^{-1}h(m))A + (s^{-1}r)B$, then $x_P \equiv r \bmod q \implies$ valid signature.

Problem 1. (8 points) Answer Only

- 1) In hybrid encryption, the public key is used to encrypt the message, and the symmetric key is used only to sign.

☐ TRUE
☐ FALSE

Solution: FALSE

- 2) In AES-CBC mode, encryption cannot be parallelized because the encryption of block c_i depends on the ciphertext of the previous block c_{i-1} , but decryption can be parallelized.

☐ TRUE
☐ FALSE

Solution: TRUE

- 3) If an attacker intercepts a Diffie–Hellman key exchange where Alice sends $A = g^a \bmod p$ and Bob sends $B = g^b \bmod p$, the attacker can easily compute the shared secret S if they can solve the Discrete Logarithm Problem.

☐ TRUE
☐ FALSE

Solution: TRUE

- 4) El Gamal encryption is deterministic; if you encrypt the same message m twice with the same public key, you will always generate the exact same ciphertext (c_1, c_2) .

☐ TRUE
☐ FALSE

Solution: FALSE

Problem 2 (6 points) Answer Only

1) Suppose Alice and Bob share a secret key k used with a secure MAC algorithm. Alice sends (m, t) , where $t = \text{MAC}_k(m)$, and a attacker intercepts it. The attacker wants to replace Alice's message with $(m\|1, t')$, with t' chosen so that Bob will accept this modified message as valid. Is it feasible for an attacker to do this, without knowing the key k ?

- ☐ Yes, because the attacker knows m and anyone can compute the hash of $m\|1$
- ☐ Yes, because MACs provide confidentiality, not integrity
- ☐ Yes, because MACs are deterministic
- ☐ No, because MACs are unforgeable
- ☐ No, because MACs are collision resistant
- ☐ No, because MACs are deterministic

Solution: No, because MACs are unforgeable

2) Is it good to replace Diffie–Hellman Key Exchange with Elliptic Curves Diffie–Hellman Key Exchange?

- ☐ Yes, because scheme with same level of security would be faster
- ☐ Yes, because scheme with same level of security would use less memory
- ☐ Yes, because is can provide higher level of security with same amount of resources
- ☐ Yes, all above
- ☐ No, computations are more complicated, so it is slower
- ☐ No, it was broken in July 2022 by Castryck and Decru

Solution: Yes, all above

Problem 3. (6 points) Suppose you encrypt a message using an affine cipher, then encrypt the ciphertext using another affine cipher. Is there any advantage to doing this, rather than using a single affine cipher? Why or why not?

Solution: Suppose the first affine cipher is $e_1(x) = ax + b \bmod N$, and second one is $e_2(x) = cx + d \bmod N$, so $e_2(e_1(x)) = c(ax + b) + d = acx + bc + d \bmod N$ still is a linear transformation, still just an affine cipher.

Problem 4. (10 points)

Alice encrypted her long message

$$x_1, x_2, \dots, x_n$$

using block cipher using block cipher mode of operation. One block was lost during transmission, so Bob received

$$y_1, y_2, \dots, y_{n-1}.$$

Bob knows about it, but doesn't know which block is lost. How many blocks of plaintext Bob able to recover from received ciphertext? Standard assumptions about plaintext recognition holds. You can assume IV was transmitted correctly.

1) Alice used ECB mode

Solution: Bob can decrypt $n - 1$ blocks of the plaintext by $d_k(y_i)$. Looking on decryption Bob can recognize (maybe?) the place of the lost plaintext block.

2) Alice used CBC mode

Solution: Bob can decrypt $n - 1$ blocks of the plaintext by $d_k(y_i) \oplus y_{i-1}$. Exactly one block will be corrupted, this is the place of lost block, so we have $n - 2$ correct blocks and place where 2 blocks are lost.

3) Alice used CTR mode

Solution: Bob can decrypt $n - 1$ blocks of the plaintext. Suppose block j was lost. Bob can decrypt blocks by $e_k(\text{IV} \parallel i) \oplus y_i$. When he will see corrupted block j , he knows which block is missing and continues to decrypt blocks by $e_k(\text{IV} \parallel i + 1) \oplus y_i$.

Problem 5. (6 points) Bob has noticed that somebody can sign messages in his name. Upon inspection of his signature software, he notices that the random number generator always chooses the same ephemeral key for the ECDSA scheme. Since his implementation lacks a good RNG, Bob implements the following countermeasure. He modifies the random number generator such that the new key is computed by multiplying the value of the previous key with 3, i.e., $k_{E,i+1} = 3 \cdot k_{E,i}$. Describe how an attacker can compute the secret key d if he gets two consecutive messages and signature tuples, (m_i, r_i, s_i) and $(m_{i+1}, r_{i+1}, s_{i+1})$.

Solution: An attacker can use following system of equations

$$\begin{cases} s_i & \equiv (h(x_i) + dr_i)k_{E,i}^{-1} \pmod{q} \\ s_{i+1} & \equiv (h(x_{i+1}) + dr_{i+1})(3k_{E,i})^{-1} \pmod{q} \end{cases} \iff \begin{cases} s_i \cdot k_{E,i} - r_i \cdot d & \equiv h(x_i) \\ 3s_{i+1} \cdot k_{E,i} - r_{i+1} \cdot d & \equiv h(x_{i+1}) \end{cases}$$

this is a 2 times 2 linear system of equations, variables are $k_{E,i}$ and d . This system has unique solution (which gives us secret d) iff determinant is invertible module q .

Determinant is $s_i r_{i+1} - 3r_i s_{i+1}$, i.e. for a given non zero r_i, r_{i+1}, s_i there is only one s_{i+1} which given 0 i.e. attacker can recover d with probability $1 - \frac{1}{q}$.

Problem 6. (15 points)

University wants to hold a charity auction and needs your help! They have designed a secure bidding protocol as follows:

- Each party chooses a bid x .
- Each party applies a function **ENVELOPE** to x and broadcasts **ENVELOPE**(x) to every other party.
- Once everyone has posted their commitment, each party reveals their bid and any other information relevant to the scheme.

There are two key properties we want to guarantee:

- **Binding:** If a party chooses x , sends **ENVELOPE**(x), and then reveals $x' \neq x$, other parties can detect that x' is invalid with overwhelming probability.
- **Hiding:** **ENVELOPE**(x) should not leak more than a negligible amount of information about x .

For each of the following schemes, select whether the scheme provides binding, hiding, both, or neither.

- You should assume that the range of all possible bids is **small** enough to brute-force,
- The auction is run exactly once.
- Assume that all parties have access to the trusted public keys of every other party and Diffie–Hellmann parameters: generator g and large prime p .

1) **ENVELOPE**(x) = $\text{AES}_k(x)$ where k is a publicly-known symmetric key shared by everyone. To reveal, each party sends x .

- ☐ Binding
- ☐ Hiding
- ☐ Both
- ☐ Neither

Solution: Binding

2) **ENVELOPE**(x) = $g^x \bmod p$. To reveal, each party sends x .

- ☐ Binding
- ☐ Hiding
- ☐ Both
- ☐ Neither

Solution: Binding

3) (x, y) denotes a tuple of two values, x and y .

$$\text{ENVELOPE}(x) = \begin{cases} (x-1, x) & \text{with probability } 1/2 \\ (x, x+1) & \text{with probability } 1/2 \end{cases}$$

To reveal, each party sends x .

- ☐ Binding
- ☐ Hiding
- ☐ Both
- ☐ Neither

Solution: Neither

4) Design a scheme that was not mentioned in a previous subpart. Your scheme should be both binding and hiding. Assume you have access to an unlimited source of random bits. Given x , how do you compute $\text{ENVELOPE}(x)$?

Do not forget to provide the list of values sent in the reveal step (including x) and how to verify $\text{ENVELOPE}(x)$ given the values from the reveal step.

Solution: The core idea is that we need some sort of randomness to avoid brute-force (hiding), and some sort of one-way, collision resistant function to achieving binding. $\text{HMAC}_k(x)$ or $H(x\|k)$ with fix length k is hiding if k is unknown and randomly generated. Fixed length of k protect us from changing x as well. To reveal, each party sends x and k .

Problem 7. (9 points)

1) List the points on the elliptic curve $E : y^2 \equiv x^3 - 2 \pmod{7}$.

Solution:

$$\begin{array}{c|c|c|c|c|c|c|c|c} x \pmod{7} & 0 & 1 & 2 & 3 & 4 \equiv -3 & 5 \equiv -2 & 6 \equiv -1 \\ \hline x^3 \pmod{7} & 0 & 1 & 1 & -1 & 1 & -1 & -1 \\ \hline x^3 - 2 \pmod{7} & -2 \equiv 5 & -1 \equiv 6 & -1 \equiv 6 & -3 \equiv 4 & -1 \equiv 6 & -3 \equiv 4 & -3 \equiv 4 \end{array}$$

$$\begin{array}{c|c|c|c|c|c|c|c|c} y \pmod{7} & 0 & 1 & 2 & 3 & 4 \equiv -3 & 5 \equiv -2 & 6 \equiv -1 \\ \hline y^2 \pmod{7} & 0 & 1 & 4 & 2 & 2 & 4 & 1 \end{array}$$

LHS has values $\{0, 1, 4\}$, RHS has values $\{4, 5, 6\}$, they shared only value 4, now we can find all possible values.

$$E(\mathbb{Z}_7) = \{\mathcal{O}, (3, 2), (3, 5), (5, 2), (5, 5), (6, 2), (6, 5)\}.$$

2) Find the sum $(3, 2) + (3, 2)$ on E .

Solution:

$$k = \frac{3x^2 + a}{2y} = \frac{27}{4} \equiv -1 \cdot 2 \equiv -2 \pmod{7} \quad (2 \cdot 4 \equiv 1 \pmod{7})$$

$$x_3 = k^2 - x_1 - x_2 = 4 - 3 - 3 \equiv -2 \equiv 5 \pmod{7}$$

$$y_3 = k(x_1 - x_3) - y_1 = -2(3 - 5) - 2 \equiv 2 \pmod{7}$$

$$(3, 2) + (3, 2) = (5, 2)$$

3) Find the sum $(3, 2) + (5, 5)$ on E .

Solution: After previous part:

$$(3, 2) + (5, 5) = (3, 2) - (5, 2) = (3, 2) - 2(3, 2) = -(3, 2) = (3, 5).$$

OR

$$k = \frac{y_2 - y_1}{x_2 - x_1} = \frac{5 - 2}{5 - 3} = \frac{3}{2} \equiv 3 \cdot 4 \equiv 5 \equiv -2 \pmod{7} \quad (2 \cdot 4 \equiv 1 \pmod{7})$$

$$x_3 = k^2 - x_1 - x_2 = 4 - 3 - 5 \equiv 3 \pmod{7}$$

$$y_3 = k(x_1 - x_3) - y_1 = -2(3 - 3) - 2 \equiv 5 \pmod{7}$$

Problem 8. (6 points) Answer Only

State decides to launch a communications channel in order to securely communicate with the Green Bay over an insecure channel. John wants to test different schemes in his attempt to attain confidentiality and integrity. Notation:

- m is the message Jason sends to the recipient.
- k_1, k_2, k_3 are secret keys known only to John and the recipient
- ECB, CBC, and CTR represent block cipher encryption modes for a secure block cipher.
- Assume that CBC and CTR mode are called with randomly generated IVs.
- H is a collision-resistant, one-way hash function.
- HMAC is the construction from lecture.

Decide whether each scheme below provides confidentiality, integrity, both, or neither. For all question parts, the ciphertext is the value of c ; s is a temporary value that is not sent as part of the ciphertext.

1) Consider the following scheme:

$$s = \text{CBC}_{k_1}(m), \quad c_1 = \text{ECB}_{k_2}(s), \quad c_2 = \text{HMAC}_{k_3}(s), \quad c = (c_1, c_2).$$

- ☐ Confidentiality only
- ☐ Integrity only
- ☐ Both confidentiality and integrity
- ☐ Neither confidentiality nor integrity

Solution: Both confidentiality and integrity

2) Consider the following scheme:

$$c_1 = \text{ECB}_{k_1}(m), \quad c_2 = H(k_2 \| c_1), \quad c = (c_1, c_2).$$

- ☐ Confidentiality only
- ☐ Integrity only
- ☐ Both confidentiality and integrity
- ☐ Neither confidentiality nor integrity

Solution: Neither confidentiality nor integrity

Problem 9. (8 points)

Alice designs a scheme that generates a single MAC on a set of n messages $\{m_1, m_2, \dots, m_n\}$. (Each entry m_i must be unique. sets of the same elements in different order are considered to be equal.

- Compute HMACs on each individual message. $t_i = \text{HMAC}_k(m_i)$, for $1 \leq i \leq n$.
 - XOR all the HMAC outputs together to get the final output tag $T = t_1 \oplus t_2 \oplus \dots \oplus t_n$.
- 1) Does this scheme require the message length to be less than or equal to the length of the HMAC output?

Solution: No, because input of HMAC can be any length.

2) Alice computes the MAC for two message sets:

- The set $A = \{A_1, A_2, \dots, A_n\}$ has tag T_A .
- The set $B = \{B_1, B_2, \dots, B_m\}$ has tag T_B .

Mallory observes both message lists and both MACs. Mallory does not know k . Mallory wants to compute a valid MAC on some message list that is not A or B . Give a valid (message set, tag) pair that Mallory could compute.

Solution:

message:	$\{A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m\}$
MAC:	$\{T_A \oplus T_B\}$