Amherst College
Department of Mathematics and Statistics

MATH 252 — MIDTERM 2 — SPRING 2019

Comment (2023): this exam is from an older version of this course, taught at Brown. There are some differences of style and emphasis compared to Math 252 here. Problems about topics we have not discussed are crossed out in this document.

Also note that **four-function calculators were permitted** on this exam, so it requires some arithmetic that I would not expect you to do by hand.

NAME:____

Read This First!

- Keep cell phones off and out of sight.
- Do not talk during the exam.
- You are allowed one page of notes, front and back.
- You may use a calculator, but **you are expected to use only the four arithmetic functions**, in order to be fair to students with a four-function calculator. Clearly write the calculations you have done on the page.
- You may use any of the blank pages to continue answers if you run out of space. Please clearly indicate on the problem's original page if you do so, so that I know to look for it.
- In order to receive full credit on a problem, solution methods must be complete, logical and understandable.

Grading - For Instructor Use Only

Question:	1	2	3	4	Total
Points:	7	7	7	7	28
Score:					

This page intentionally left blank. You may use it for scratchwork or to continue answers to any question (note clearly on the original page if you do so).

1. [7 points] You may omit; Elliptic curves will not appear on our midterm 2. ~~Consider the elliptic curve over $\mathbb{F}_{11}$ defined by the following congruence.~~

$$Y^2 \equiv X^3 + 7X + 9 \pmod{11}$$

~~The point $P = (2, 3)$ lies on this curve (you do not need to check this). Compute $P \oplus P \oplus P$ on this curve.~~

This page intentionally left blank. You may use it for scratchwork or to continue answers to any question (note clearly on the original page if you do so).

2. [7 points] You are using DSA with the following parameters (see the DSA summary at the back of the exam packet for notation).

$$p = 23 \qquad q = 11 \qquad g = 2$$

Your private key is $a = 3$. You wish to sign the document $d = 4$, and choose the random (ephemeral) element $k = 8$. Compute the signature (S_1, S_2).

This page intentionally left blank. You may use it for scratchwork or to continue answers to any question (note clearly on the original page if you do so).

3. [7 points] Eve has recently succeeded in writing an efficient factoring algorithm, and has decided to use it for nefarious purposes. Her algorithm is written in a function `factor(N)`, which takes an integer $N \geq 2$ as input and returns some prime factor of N.

   Write a function `breakRSA(N, e, c)` that takes Bob's public numbers N and e and a ciphertext c sent to Bob by Alice, and returns Alice's plaintext m (notation as in the summary table at the back of the exam packet). Your function may use Eve's new `factor` function, as well as any built-in functions in Python (such as `pow(a,b,m)`, which efficiently computes $a^b \pmod{m}$). You should write the code for any other helper functions you use that are not built in to Python.

This page intentionally left blank. You may use it for scratchwork or to continue answers to any question (note clearly on the original page if you do so).

4. [7 points] Comment (2023) This problem originally referred to ECDSA, which we have not yet discussed this semester. I've rewritten it slightly to refer to DSA instead. Samantha and Victor agree to the following digital signature scheme. The public parameters and key creation are identical to ECDSA (see the table at the back of the exam packet for details and notation). The verification process is different. As in ECDSA, Victor begins by computing the following two numbers.

$$\begin{aligned} v_1 &= ds_2^{-1} \pmod q \\ v_2 &= s_1 s_2^{-1} \pmod q \end{aligned}$$

Victor considers a signature (s_1, s_2) valid if and only if the following verification equation holds.
$\cancel{x(v_1V \ominus v_2G) \mod q = s_1}$

$$A^{v_1} g^{v_2} \% p \% q = s_1$$

Determine a signing procedure that Samantha can follow to sign a chosen document d for this system.

This page intentionally left blank. You may use it for scratchwork or to continue answers to any question (note clearly on the original page if you do so).

Reference tables from textbook:

Alice	Bob
Public parameter creation	
A trusted party chooses and publishes a (large) prime p and an integer g having large prime order in $\mathbb{F}_p^*$.	
Private computations	
Choose a secret integer a.	Choose a secret integer b.
Compute $A \equiv g^a \pmod{p}$.	Compute $B \equiv g^b \pmod{p}$.
Public exchange of values	
Alice sends A to Bob $\longrightarrow$	A
B	$\longleftarrow$ Bob sends B to Alice
Further private computations	
Compute the number $B^a \pmod{p}$.	Compute the number $A^b \pmod{p}$.
The shared secret value is $B^a \equiv (g^b)^a \equiv g^{ab} \equiv (g^a)^b \equiv A^b \pmod{p}$.	

Table 2.2: Diffie–Hellman key exchange

Alice	Bob
Public parameter creation	
A trusted party chooses and publishes a large prime p and an element g modulo p of large (prime) order.	
Key creation	
Choose private key $1 \le a \le p-1$. Compute $A = g^a \pmod{p}$. Publish the public key A.	
Encryption	
	Choose plaintext m. Choose random element k. Use Alice's public key A to compute $c_1 = g^k \pmod{p}$ and $c_2 = mA^k \pmod{p}$. Send ciphertext (c_1, c_2) to Alice.
Decryption	
Compute $(c_1^a)^{-1} \cdot c_2 \pmod{p}$. This quantity is equal to m.	

Table 2.3: Elgamal key creation, encryption, and decryption

Bob	Alice
Key creation	
Choose secret primes p and q. Choose encryption exponent e with $\gcd(e, (p-1)(q-1)) = 1$. Publish $N = pq$ and e.	
Encryption	
	Choose plaintext m. Use Bob's public key (N, e) to compute $c \equiv m^e \pmod{N}$. Send ciphertext c to Bob.
Decryption	
Compute d satisfying $ed \equiv 1 \pmod{(p-1)(q-1)}$. Compute $m' \equiv c^d \pmod{N}$. Then m' equals the plaintext m.	

Table 3.1: RSA key creation, encryption, and decryption

Samantha	Victor
Key creation	
Choose secret primes p and q. Choose verification exponent e with $\gcd(e, (p-1)(q-1)) = 1$. Publish $N = pq$ and e.	
Signing	
Compute d satisfying $de \equiv 1 \pmod{(p-1)(q-1)}$. Sign document D by computing $S \equiv D^d \pmod{N}$.	
Verification	
	Compute $S^e \bmod N$ and verify that it is equal to D.

Table 4.1: RSA digital signatures

Samantha	Victor
Public parameter creation	
A trusted party chooses and publishes a large prime p and primitive root g modulo p.	
Key creation	
Choose secret signing key $1 \le a \le p-1$. Compute $A = g^a \pmod{p}$. Publish the verification key A.	
Signing	
Choose document D mod p. Choose random element $1 < k < p$ satisfying $\gcd(k, p-1) = 1$. Compute signature $S_1 \equiv g^k \pmod{p}$ and $S_2 \equiv (D - aS_1)k^{-1} \pmod{p-1}$.	
Verification	
	Compute $A^{S_1} S_1^{S_2} \bmod p$. Verify that it is equal to $g^D \bmod p$.

Table 4.2: The Elgamal digital signature algorithm

Samantha	Victor
Public parameter creation	
A trusted party chooses and publishes large primes p and q satisfying $p \equiv 1 \pmod{q}$ and an element g of order q modulo p.	
Key creation	
Choose secret signing key $1 \le a \le q-1$. Compute $A = g^a \pmod{p}$. Publish the verification key A.	
Signing	
Choose document D mod q. Choose random element $1 < k < q$. Compute signature $S_1 \equiv (g^k \bmod p) \bmod q$ and $S_2 \equiv (D + aS_1)k^{-1} \pmod{q}$.	
Verification	
	Compute $V_1 \equiv DS_2^{-1} \pmod{q}$ and $V_2 \equiv S_1 S_2^{-1} \pmod{q}$. Verify that $(g^{V_1} A^{V_2} \bmod p) \bmod q = S_1$.

Table 4.3: The digital signature algorithm (DSA)

Public parameter creation	
A trusted party chooses and publishes a (large) prime p, an elliptic curve E over $\mathbb{F}_p$, and a point P in $E(\mathbb{F}_p)$.	
Private computations	
Alice	**Bob**
Chooses a secret integer n_A. Computes the point $Q_A = n_A P$.	Chooses a secret integer n_B. Computes the point $Q_B = n_B P$.
Public exchange of values	
Alice sends Q_A to Bob $\longrightarrow$	Q_A
Q_B $\longleftarrow$	Bob sends Q_B to Alice
Further private computations	
Alice	**Bob**
Computes the point $n_A Q_B$.	Computes the point $n_B Q_A$.
The shared secret value is $n_A Q_B = n_A(n_B P) = n_B(n_A P) = n_B Q_A$.	

Table 6.5: Diffie–Hellman key exchange using elliptic curves

Public parameter creation	
A trusted party chooses a finite field $\mathbb{F}_p$, an elliptic curve $E/\mathbb{F}_p$, and a point $G \in E(\mathbb{F}_p)$ of large prime order q.	
Samantha	**Victor**
Key creation	
Choose secret signing key $1 < s < q-1$. Compute $V = sG \in E(\mathbb{F}_p)$. Publish the verification key V.	
Signing	
Choose document $d \bmod q$. Choose random element $e \bmod q$. Compute $eG \in E(\mathbb{F}_p)$ and then, $s_1 = x(eG) \bmod q$ and $s_2 \equiv (d + s s_1)e^{-1} \pmod{q}$. Publish the signature (s_1, s_2).	
Verification	
	Compute $v_1 \equiv d s_2^{-1} \pmod{q}$ and $v_2 \equiv s_1 s_2^{-1} \pmod{q}$. Compute $v_1 G + v_2 V \in E(\mathbb{F}_p)$ and verify that $x(v_1 G + v_2 V) \bmod q = s_1$.

Table 6.7: The elliptic curve digital signature algorithm (ECDSA)

You may use the rest of this page for scratchwork or to continue answers to any question (note clearly on the original page if you do so)

This page intentionally left blank. You may use it for scratchwork or to continue answers to any question (note clearly on the original page if you do so).

This page intentionally left blank. You may use it for scratchwork or to continue answers to any question (note clearly on the original page if you do so).