Solutions to the 2022-03-18 exam

Question 1

a) The binary expansion of 41 is $2^0 + 2^3 + 2^5$. By successively squaring the number 3 modulo 7, we find the following table.

n	0	1	2	3	4	5
3^{2^n}	3	2	4	2	4	2

Hence $3^{41} = 3^{2^0} \cdot 3^{2^3} \cdot 3^{2^5} \equiv 3 \cdot 2 \cdot 2 \equiv 5 \pmod 7$.

b) Using that $2^{-1} \equiv 3 \pmod 5$, the system

$$\begin{cases} 2x & \equiv 1 \pmod 5, \\ x & \equiv 2 \pmod 8, \\ x - 1 & \equiv 4 \pmod 7, \end{cases}$$

can be rewritten as

$$\begin{cases} x & \equiv 3 \pmod 5, \\ x & \equiv 2 \pmod 8, \\ x & \equiv 5 \pmod 7. \end{cases}$$

Writing $x = 3 + 5k$, where $k \in \mathbb{Z}$, we obtain that

$$3 + 5k \equiv 2 \pmod 8.$$

Since $5 \cdot 5 = 25 \equiv 1 \pmod 8$, we see that $k \equiv 3 \pmod 8$. Writing $k = 3 + 8l$ for some $l \in \mathbb{Z}$, we see that

$$x = 3 + 5(3 + 8l) \equiv 4 + 5l \equiv 5 \pmod 7.$$

Since $5^{-1} \equiv 3 \pmod 7$, it follows that $l \equiv 3 \pmod 7$. In particular, we find that

$$x = 3 + 5(3 + 8(3 + 7m)) = 3 + 15 + 40(3 + 7m) = 18 + 120 + 280m = 138 + 280m$$

is a solution to the system of congruences above. Moreover, since the moduli 5, 8 and 7 are pairwise relative prime, by the Chinese remainder theorem all solutions must be of the form $x = 138 + 280m$.

c) An element $x \in \mathbb{Z}/20$ is a unit if and only if $\gcd(x, 20) = 1$. Since $20 = 2^2 \cdot 5$, this amounts to not being divisible by 2 or 5. In particular, the units of $\mathbb{Z}/20$ are (the classes represented by) 1, 3, 7, 9, 11, 13, 17 and 19.

Question 2

a) Björn obtained c by computing m^e modulo N, where m is Björn's plaintext.

If Agnetha has not done so yet, she needs to compute her private decryption key d, which is the multiplicative inverse of e modulo $(p-1)(q-1)$. This can be done using the extended Euclidean algorithm. She then recovers Björn's plaintext by computing $c^d \equiv m \pmod{N}$.

The reason that this calculation works is that, when $\gcd(e,(p-1)(q-1)) = 1$, a congruence of the form $x^e \equiv y \pmod{N}$ always has a unique solution in x given by y^d, where d is a multiplicative inverse of e modulo $(p-1)(q-1)$.

b) Since $(x+p)(x+q) = x^2 + (p+q)x + pq$, one can compute p and q if one knows the values $p+q$ and $N = pq$ by finding the roots of this polynomial. Since $(p-1)(q-1) = pq - p - q + 1$, anyone who knows all the public information and the value of $(p-1)(q-1)$ can determine the value of $p+q$ and hence of p and q.

c) Suppose Anni-Frid's RSA keys consist of a modulus N_A, a public encryption exponent e_A, and a private decryption exponent d_A. Elton first intercepts Anni-Frid's public keys N_A and e_A and prevents Benny from obtaining them. Next, Elton creates his own set of RSA keys N_E, e_E, and d_E, and sends N_E and e_E to Benny, who believes they came from Anni-Frid.

Suppose now that Benny wants to send a plaintext m to Anni-Frid. He computes the ciphertext $c = m^{e_E}$, thinking Anni-Frid and only her possesses the correct decryption exponent, and sends c to Anni-Frid. Now Elton intercepts c and computes $m = c^{d_E}$. Elton can now read Benny's message, may choose to alter it to another message m', encrypts it to $c' = m'^{e_A}$ using Anni-Frid's public key, and sends c' to her. Now Anni-Frid finds the plaintext $m' = c'^{d_A}$ using her secret key, thinking the message came from Benny and not knowing Elton was able to read and potentially tamper with it.

Question 3

a) The Miller-Rabin primality test is described in Table 3.2 of [HPS14]:

Input. Integer n to be tested, integer a as potential witness.

1. If n is even or $1 < \gcd(a, n) < n$, return **Composite**.
2. Write $n - 1 = 2^k q$ with q odd.
3. Set $a = a^q \pmod{n}$.
4. If $a \equiv 1 \pmod{n}$, return **Test Fails**.
5. Loop $i = 0, 1, 2, \ldots, k - 1$
   6. If $a \equiv -1 \pmod{n}$, return **Test Fails**.
   7. Set $a = a^2 \bmod n$.
8. End i loop.
9. Return **Composite**.

Table 3.2: Miller–Rabin test for composite numbers

b) We first need to check whether n is even and compute $\gcd(a, n)$, which can both be done in polynomial time. Writing $n - 1$ as $2^k q$ with q odd can be done by repeatedly dividing $n - 1$ by 2, which is also polynomial in the number of digits of n since every division reduced the number of digits of $n - 1$ by one. One then needs to compute a^q modulo n, which can also be done in polynomial time using fast-powering. One then needs to square this number a^q at most k times. Since k can't be greater than the number of digits of n, this can also be performed in polynomial time. We conclude that the complexity Miller-Rabin test is polynomial in the number of digits of n.

c) For a random composite number n, at least 75% of the numbers smaller than n are Miller-Rabin witnesses. One might now be inclined to think that the probability that n is prime if the Miller-Rabin test fails a times is at least $1 - 4^{-a}$. This would yield $p/100 \leq 1 - 4^{-a}$ and hence that $-a \leq \log_4(1 - p/100)$, so Alice would need to check $a \geq \log_4(1 - p/100)$ values she wants to be at least p% sure that the integer n is prime.

However, this is not entirely correct: the probability that a randomly chosen number is prime should also be taken into account. The probability that a randomly chosen number n is prime is roughly $1/\log(n)$ by the prime number theorem. By Bayes's formula, we find that

$$\Pr(n \text{ is prime} \mid \text{test fails } a \text{ times})$$

is equal to

$$\frac{\mathrm{P}(n \text{ is prime})}{\Pr(n \text{ is prime}) + \Pr(\text{test fails } a \text{ times} \mid n \text{ is composite})\Pr(n \text{ is composite})}.$$

Using that $\Pr(\text{test fails } a \text{ times} \mid n \text{ is composite}) \leq 4^{-a}$, we find that

$$\Pr(n \text{ is prime} \mid \text{test fails } a \text{ times})$$

must be greater than or equal to

$$\frac{1/\log(n)}{1/\log(n) + 4^{-a} \cdot (1 - 1/\log(n))} = \frac{1}{1 + (\log(n) - 1) \cdot 4^{-a}}$$

This yields the inequality

$$\frac{p}{100} \leq \frac{1}{1 + (\log(n) - 1) \cdot 4^{-a}},$$

hence

$$4^a \geq \frac{p(\log(n) - 1)}{100 - p}$$

In particular, to be p% sure, Alice needs to test

$$a \geq \log_4 \left(\frac{p(\log(n) - 1)}{100 - p} \right)$$

values.

d) In the Miller-Rabin test, for each composite number at least 75 % of the natural numbers smaller than it are witnesses, while in the Fermat primality test, there exist composite numbers that have no witness. However, the Fermat primality test is faster, so it could be useful to first run the Fermat primality test before running the Miller-Rabin primality test when determining whether a given number is prime.

Question 4

a) Shank's baby-step giant-step algorithm is described in Proposition 2.21 of [HPS14]:

Proposition 2.21 (Shanks's Babystep–Giantstep Algorithm). *Let G be a group and let $g \in G$ be an element of order $N \geq 2$. The following algorithm solves the discrete logarithm problem $g^x = h$ in $\mathcal{O}(\sqrt{N} \cdot \log N)$ steps using $\mathcal{O}(\sqrt{N})$ storage.*

(1) *Let $n = 1 + \lfloor \sqrt{N} \rfloor$, so in particular, $n > \sqrt{N}$.*

(2) *Create two lists,*

$$\begin{aligned} &\text{List 1:} \quad e, g, g^2, g^3, \ldots, g^n, \\ &\text{List 2:} \quad h, h \cdot g^{-n}, h \cdot g^{-2n}, h \cdot g^{-3n}, \ldots, h \cdot g^{-n^2}. \end{aligned}$$

(3) *Find a match between the two lists, say $g^i = hg^{-jn}$.*

(4) *Then $x = i + jn$ is a solution to $g^x = h$.*

b) Suppose the discrete logarithm problem $g^x = h$ in given group G has a solution and let $n = 1 + \lfloor \sqrt{N} \rfloor$, where $N = \mathrm{ord}_G(g)$. The fact that a solution exists implies that $g^x = h$ for some $0 \leq x < N$. Using division with remainder, we can write $x = an + b$ with $0 \leq b < n$. Since $x < N < n^2$, we moreover see that $a < n$ as well. This means that $g^b = hg^{-an}$. Since g^b is in the first list and hg^{-an} in the second list, one is guaranteed to find a solution.

c) This is essentially the first version of the Pohlig-Hellman algorithm (see Theorem 2.31 of [HPS14]).

Since g is a primitive root $\mathbb{F}_p$, we see that $g^x = h$ has a unique solution x in $\mathbb{Z}/(p-1)$. By the Chinese Remainder Theorem, we only need to determine the values of x modulo the three primes q_1, q_2 and q_3 separately. Moreover, note that the element $g^{q_2q_3}$ has order q_1 and that

$$(g^{q_2q_3})^x = h^{q_2q_3},$$

so if y_1 is any solution to

$$(g^{q_2q_3})^{y_1} = h^{q_2q_3},$$

then $x \equiv y_1 \pmod{q_1}$. Similarly, if y_2 and y_3 are solutions to

$$(g^{q_1q_3})^{y_2} = h^{q_1q_3} \quad \text{and} \quad (g^{q_1q_2})^{y_3} = h^{q_1q_2},$$

then $x \equiv y_2 \pmod{q_2}$ and $x \equiv y_3 \pmod{q_3}$, hence the Chinese Remainder Theorem can be used to determine the value of x if one knows such solutions y_1, y_2 and y_3. In particular, solving the original discrete logarithm problem can be reduced to solving three discrete logarithm problems for elements of order q_1, q_2 and q_3, respectively, which is generally much easier.

d) Shank's algorithm requires a lot of storage (namely $\mathcal{O}(\sqrt{N})$, where N is the order of g), making it unusable if g has a large order. On the other hand, Pollard's ρ method does not require much storage since one does not have to store the lists.

Question 5

a) The ElGamal public key cryptosystem is described in Table 2.3 of [HPS14]:

Public parameter creation	
A trusted party chooses and publishes a large prime p and an element g modulo p of large (prime) order.	
Alice	**Bob**
Key creation	
Choose private key $1 \leq a \leq p-1$. 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

b) Let us say Samantha and Victor are using a digital signature scheme. Given a document D, the purpose of a digital signature scheme is to be able to ensure Victor that Samantha approves of this document. This is done by letting Samantha create an extra piece of information D^{sign} based on this document D, such that there is an easy way of verifying that D^{sign} is a "digital signature" of D using a public key that Samantha published, while it is very hard to create (or "forge") such a digital signature D^{sign} without knowing Samantha's private key.

c) The following is roughly the definition of a Hash function given at the start of section 8.1 of [HPS14]:

A hash function is a function Hash that takes arbitrarily large document D as input and that returns a short string of bits $H = \mathsf{Hash}(D)$. It should satisfy the following properties

(i) Computing $\mathsf{Hash}(D)$ should be fast and easy (i.e. linear time).

(ii) Given a possible hash value H, it should be difficult to find a document D such that $\mathsf{Hash}(D) = H$ (i.e. exponential time).

(iii) It should be hard to find two different document D_1 and D_2 such that $\mathsf{Hash}(D_1) = \mathsf{Hash}(D_2)$.

Question 6

a) If B is very small, then it is harder to find numbers a such that $a^2 - N$ is B-smooth. One the other hand, if B is very large, then one will need to find a lot of B-smooth numbers in order to perform the elimination step.

b) Note that $N = 1569929 \equiv 9 \equiv 4 \pmod 5$. In particular, a number of the form $T^2 - N$ is divisible by 5 if and only if $T^2 \equiv 4 \pmod 5$. The solutions to this congruence are $T \equiv 2, 3 \pmod 5$, so a number of the form $F(T)$ is divisible by 5 if and only if $T \equiv 2, 3 \pmod 5$. These are the numbers $F(2492), F(2493, F(2497), F(2498), \ldots$.

c) The following matrix equation in $\mathbb{F}_2$ can be used to find solutions to the equation $x^2 \equiv y^2 \pmod N$:

$$\begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Question 7

a) $y^2 = x^3 + 1$ defines an elliptic curve over $\mathbb{F}_7$ since $4 \cdot 0^3 + 27 \cdot 1^2 = 273 \not\equiv 0 \pmod 7$.

b) We see that $0^3 + 1 \equiv 1 \pmod 7$ and $6^3 + 1 \equiv (-1)^3 + 1 \equiv 0 \pmod 7$, so P and Q are points on E. The points $3P = 2P + P$, $2Q$ and $P + Q$ can be computed using Theorem 6.6 of [HPS14]. This yields $3P = \mathcal{O}$, $2Q = \mathcal{O}$ and $P + Q = (2, 4)$.

Note. The computations should be included in your solution.

c) We see that

$$\begin{aligned} 0^2 &\equiv 0 \pmod 7 \\ 1^2, 6^2 &\equiv 1 \pmod 7 \\ 2^2, 5^2 &\equiv 4 \pmod 7 \\ 3^2, 4^2 &\equiv 2 \pmod 7 \end{aligned}$$

By filling in all possible values $x \in \mathbb{F}_7$ in $x^3 + 1$, we find the points

$$(0,1), (0,6), (1,3), (1,4), (2,3), (2,4), (3,0), (4,3), (4,4), (5,0), (6,0)$$

Including the point $\mathcal{O}$ at infinity, we see that E has 12 points over $\mathbb{F}_7$.

d) Note that a point $P = (x, y)$ on an elliptic curve has order 2 if and only if $y = 0$. In particular, the group $E(\mathbb{F}_7)$ has three points of order 2, which is not possible in a cyclic group.

Question 8

a) The Hasse bound states that the absolute value of $p + 1 - \#E(\mathbb{F}_p)$ is smaller than or equal to $2\sqrt{p}$.

b) There is no obvious way to encode messages as points on elliptic curves.

c) The Diffie-Hellman key exchange is described in Table 6.5 of [HPS14]:

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.	Chooses a secret integer n_B.
Computes the point $Q_A = n_A P$.	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

The algorithms involved in making the actual key exchange are the Elliptic Curve Addition Algorithm and the Double-and-Add Algorithm (also called the Fast Powering Algorithm). These are both polynomial in the size of the input.

d) Let a composite number N be given. In Lenstra's factorization algorithm, one chooses an "elliptic curve E modulo N", or more precisely, an equation of the form $y^2 = x^3 + Ax + B$, together with a point (a, b) on E modulo N. One then computes the multiples $2!P, 3!P, 4!P, \ldots$ up to a specific bound using the Elliptic Curve Addition Algorithm and the Double-and-Add Algorithm. Since N is not a prime number, the Elliptic Curve Addition Algorithm might fail since one might need to invert a number module N which is not relatively prime to N. However, in this case one finds a nontrivial factor of N by computing the greatest common divisor of N and this number. If this does not happen while computing

$2!P, 3!P, \ldots$, then one chooses a new elliptic curve with a new point and repeats the same procedure.

Lenstra's factorization algorithm works well if there exists a prime factor p of N such that $\#E(\mathbb{F}_p)$ is a product of small primes, whereas Pollard's $p-1$ algorithm works well if $\#\mathbb{F}_p^\times$ is a product of small primes for some prime factor p of N. This means that Pollard's $p-1$ algorithm will be very slow if such a prime factor of N does not exist. However, if in the case of Lenstra's algorithm there exists no prime factor p of N such that $\#E(\mathbb{F}_p)$ is a product of small prime numbers, then one can simply run the algorithm again with a new "elliptic curve modulo N" which is likely to have a different number of points.

References

[HPS14] J. Hoffstein, J. Pipher, and J. H. Silverman. *An introduction to mathematical cryptography*. Second. Undergraduate Texts in Mathematics. Springer, New York, 2014, pp. xviii+538. ISBN: 978-1-4939-1710-5.