Q1: Define Euler's Totient Function of a positive integer n with words.

A1: The number of non-negative integers less than n which are coprime to n.

Q2: Precisely describe how to calculate Euler's Totient Function of a positive integer n.

A2: First find the prime factorisation of n, call it $n = {p_1}^{e_1} \cdot {p_2}^{e_2} \cdots {p_k}^{e_k}$. Next calculate the following to find $\phi(n)$:

$$\prod_{i=1}^{k} \left({p_i}^{e_i - 1} \cdot (p_i - 1) \right)$$

Q3: State Euler's Theorem.

A3: If $\gcd(a, n) = 1$, then $a^{\phi(n)} \equiv 1 \pmod{n}$.

Q4: State Fermat's Little Theorem.

A4: If a is a positive integer and p is prime, then $a^p \equiv 1 \pmod{p}$.

Q5: What does it mean for a sett of integers to be pairwise coprime?

A5: The greatest common divisor of all the elements is 1.

Q6: State the Chinese Remainder Theorem.

A6: if $m_1, m_2, \cdots, m_r$ are pairwise coprime positive integers and $a_1, a_2, \cdots, a_r$ are integers, then the system of congruences

$$\begin{aligned} x &\equiv a_1 \pmod{m_1} \\ x &\equiv a_2 \pmod{m_2} \\ &\vdots \\ x &\equiv a_r \pmod{m_r} \end{aligned}$$

has a unique solution modulo $M := m_1 \cdot m_2 \cdots m_r$ which is given by

$$x = \sum_{i=1}^{r} a_i M_i y_i \pmod{M}$$

where $M_i := M/m_i$ and $y_i :\equiv M_i^{-1} \pmod{m_i}$ for $1 \leq i \leq r$.

Q7: What is Kerckhoff's Assumption?

A7: Everything about a cryptographic system is public knowledge except for the key. In other words, the enemy knows the system.

Q8: Briefly explain the four levels of attacks on a cryptosystem, in ascending order of strength.

A8:

1. **Ciphertext only:** Opponent has access to some ciphertext and might use statistical information to determine the corresponding plaintext.
2. **Known plaintext:** Opponent knows some plaintext-ciphertext pairs and uses it to gain more knowledge of the key.
3. **Chosen plaintext:** Opponent is temporarily able to encrypt to build a collection of known plaintext-ciphertext pairs.
4. **Chosen plaintext-ciphertext:** Opponent is temporarily able to encrypt and decrypt.

Q9: What is the plaintext and ciphertext space of a shift cipher?

A9: Both are the non-negative integers less than 26.

Q10: Is a shift cipher a public or private system? What is/are the key(s)?

A10: Private. Key is the shift amount.

Q11: How do you encrypt a plaintext message with a shift cipher?

A11: Add the private key to the message modulo 26. In other words, rotate the letter by the key amount.

Q12: How do you decrypt a ciphertext message encrypted with a shift cipher?

A12: Subtract the private key from the message modulo 26. In other words, rotate the letter backwards by the key amount.

Q13: What are the best attacks for a shift cipher?

A13: Brute force (try every key) or analyse letter frequencies to determine the key.

Q14: What is the plaintext and ciphertext space for an affine cipher?

A14: Both are non-negative integers less than 26.

Q15: Is an affine cipher a public or private system? What is/are the key(s)?

A15: Private. The key is a pair of non-negative integers less than 26 where one of them is coprime to 26.

Q16: How do you encrypt a plaintext message with an affine cipher?

A16: Suppose the key is (a, b) where a is coprime with 26 and the message is x. To encrypt the message, calculate:

$$y \equiv ax + b \pmod{26}$$

Q17: How do you decrypt a ciphertext message encrypted with an affine cipher?

A17: Suppose the key is (a, b) where a is coprime with 26 and the encrypted message is y. To decrypt the message, calculate:

$$x \equiv a^{-1}(y - b) \pmod{26}$$

Q18: What are the best attacks for an affine cipher?

A18: Obtain two plaintext-ciphertext pairs and solve the resulting linear congruences for the key. Could also brute force. In otherwords, given x_1, y_1, x_2, y_2, solve the following for a and b.

$$y_1 \equiv ax_1 + b \pmod{26}$$
$$y_2 \equiv ax_2 + b \pmod{26}$$

You can also just try all combinations of a and b.

Q19: What is the plaintext and ciphertext space for a mixed alphabet cipher?

A19: Both are non-negative integers less than 26.

Q20: Is a mixed alphabet cipher a public or private system? What is/are the key(s)?

A20: Private. The key is a permutation of the set of non-negative integers less than 26.

Q21: How do you encrypt a plaintext message with a mixed alphabet cipher?

A21: Suppose the key is a permutation of the set of non-negative integers less than 26, π and the message is x. To encrypt the message, calculate:

$$y = \pi(x).$$

Q22: How do you decrypt a ciphertext message encrypted with a mixed alphabet cipher?

A22: Suppose the key is a permutation of the set of non-negative integers less than 26, π and the encrypted message is y. To decrypt the message, calculate:

$$x = \pi^{-1}(y).$$

Q23: What are the best attacks for a mixed alphabet cipher?

A23: Use statistical analysis on the known language's letter frequencies. Can also search for common digrams and trigrams in the ciphertext which will correspond to things like "th", "he", "in", or "the", "ing", "and", etc. Piece together the key bit by bit.

Q24: What is the plaintext and ciphertext space for a Vigenère cipher?

A24: Both are strings of non-negative integers less than 26.

Q25: Is a Vigenère cipher a public or private system? What is/are the key(s)?

A25: Private. The key is a string of non-negative integers less than 26.

Q26: How do you encrypt a plaintext message with a Vigenère cipher?

A26: Suppose the key is a string of m non-negative integers less than 26, $k_1, k_2, \cdots k_m$. Further suppose the message is a string of m non-negative integers less than 26, $x_1, x_2, \cdots x_m$. To encrypt the message, calculate:

$$\begin{aligned} y_1 &\equiv x_1 + k_1 \pmod{26} \\ y_2 &\equiv x_2 + k_2 \pmod{26} \\ &\vdots \\ y_m &\equiv x_m + k_m \pmod{26} \end{aligned}$$

In other words, treat each element of the key and message as its own shift cipher and encrypt it accordingly.

Q27: How do you decrypt a ciphertext message encrypted with a Vigenère cipher?

A27: Suppose the key is a string of m non-negative integers less than 26, $k_1, k_2, \cdots k_m$. Further suppose the encrypted message is a string of m non-negative integers less than 26, $y_1, y_2, \cdots y_m$. To encrypt the message, calculate:

$$\begin{aligned} x_1 &\equiv y_1 - k_1 \pmod{26} \\ x_2 &\equiv y_2 - k_2 \pmod{26} \\ &\vdots \\ x_m &\equiv y_m - k_m \pmod{26} \end{aligned}$$

In other words, treat each element of the key and encrypted message as its own shift cipher and decrypt it accordingly.

Q28: What are the best attacks for a Vigenère cipher?

A28: Determine the key length, m, and break m shift ciphers independently using frequency analysis.

Q29: Name and explain one method to determine the key length of a Vigenère cipher.

A29: Kasiski's test. Find repeated trigrams in the ciphertext. If the key length is less than the number of occurrences of a trigram, then its corresponding plaintext must have been encrypted with the same shift more than once (by the Pigeonhole Principle).
We can then extract a common factor from the distances between the starting index of each repeated trigram to estimate the key length.

Q30: What is the index of coincidence of a language? How is it denoted?

A30: The probability of drawing two matching letters through random selection from a text in the given language. It is denoted with the Greek letter φ (phi).

Q31: How do you calculate the index of coincidence for a language with n letters where the ith letter has probability of occurring p_i?

A31:

$$\sum_{i=1}^{n} p_i^2$$

Q32: What is the index of coincidence approximately equal to for English?

A32: 0.0667.

Q33: What is the index of coincidence equal to for a random language?

A33: $1/26 \approx 0.0384$.

Q34: Name and explain one method to determine the key length of a Vigenère cipher.

A34: Friedman's first method. Suppose your guess for the key length is m. Extract every mth letter from the ciphertext and calculate the index of coincidence of the extracted text. If it's close to the index of coincidence for English, it's probably the correct key length.

Q35: Name and explain one method to determine the key length of a Vigenère cipher.

A35: Friedman's second method. Calculate the index of coincidence for the entire ciphertext (φ_T) and use the measure of how "flat" is it compared with English (φ_L) and random text (φ_0) to estimate the key length.

$$m \approx \frac{\varphi_L - \varphi_0}{\varphi_T - \varphi_0}$$

Q36: What is the plaintext and ciphertext space for a Hill cipher?

A36: Strings of non-negative integers less than 26.

Q37: Is a Hill cipher a public or private system? What is/are the key(s)?

A37: Private. The key is a invertible matrix of non-negative integers less than 26.

Q38: How do you encrypt a plaintext message with a Hill cipher?

A38: Suppose the key is an invertible $m \times m$ matrix K and x is a message of length m. To encrypt the message, calculate:

$$y \equiv Kx \pmod{26}$$

Q39: How do you decrypt a ciphertext message encrypted with a Hill cipher?

A39: Suppose the key is an invertible $m \times m$ matrix K and y is an encrypted message of length m. To decrypt the message, calculate:

$$x \equiv K^{-1}y \pmod{26}$$

Q40: What are the best attacks for a Hill cipher?

A40: Obtain m plaintext-ciphertext pairs each of length m. Solve the resulting linear congruences for K:

$$\begin{aligned} y_1 &\equiv Kx_1 \pmod{26} \\ y_2 &\equiv Kx_2 \pmod{26} \\ &\vdots \\ y_m &\equiv Kx_m \pmod{26} \end{aligned}$$

Q41: How can we easily calculate Euler's Totient function of n where n is the product of two primes? In other words, how can we arrive at a simple expression for $\phi(pq)$?.

A41: We note that the prime factorisation of n is p^1q^1 so:

$$\phi(n) = \left[p^{1-1}(p-1)\right] \cdot \left[q^{1-1}(q-1)\right] = (p-1)(q-1)$$

Q42: What is the plaintext and ciphertext space for RSA?

A42: Both are non-negative integers less than the product of two primes.

Q43: Is RSA a public or private system? What is/are the key(s)?

A43: Public.

- The **public** key is (n, b) where n is the product of two **distinct** large primes p and q and b is a randomly chosen positive integer less than $\phi(n)$ (ϕ is Euler's Totient function) which is invertible modulo $\phi(n)$.
- The **private** key is (a, p, q) where a is $b^{-1} \pmod{\phi(n)}$ and p and q are the primes whose product makes n.

Q44: How do you encrypt a plaintext message with RSA?

A44: Suppose our public key is (n, b) and the message is x. To encrypt the message, calculate:

$$y \equiv x^b \pmod{n}.$$

Q45: How do you decrypt a ciphertext message with RSA?

A45: Suppose our private key is (a, p, q) so $n = pq$ and our encrypted message is y. To decrypt the message, caclulate:

$$x \equiv y^a \pmod{n}.$$

Q46: What are the best attacks for RSA?

A46:

- If our private key has one prime much smaller than the other, we can easily factorise n.
- If our private key has primes roughly the same size, we can also easily factorise n.
- If the message to encrypt, x, is so small that $y := x^b \pmod{n}$ is less than n, then you can simply take the bth root of y over the reals to get back x.
- If the message to encrypt is predictable, simply try encrypting guesses and see if the resulting ciphertext matches.
- If the public exponent, b is so small that the same plaintext message is likely to be encrypted with b or more different keys, you can recover the plaintext using the Chinese Remainder Theorem.

Q47: Why is it important to sign and verify signatures of messages encrypted with public key cryptography?

A47: Because the public key of every sender is known, anybody can use the public key to encrypt any message and send to anybody. You need some claim of authenticity in the messages you send and receive.

Q48: How can a sender sign their RSA-encrypted message?

A48: Attach a signature to their encrypted message, which is a hashed plaintext encrypted with their private key rather than the public one. In other words, if the hashed message is $f(x)$ and the private exponent is a, the attaches the following signature to the ciphertext:

$$s \equiv (f(x))^a \pmod{n}$$

Q49: How can a receiver verify the signature of an RSA-encrypted message?

A49: Decrypt the signature, s, with the intended sender's public exponent, b, and see if it results in the hashed decrypted message $f(x)$. In other words, verify the following holds:

$$s^b \equiv f(x) \pmod{n}$$

Q50: How could a malicious sender send an encrypted message with RSA claiming to be somebody they're not.

A50: They obtain the public intended encrypted message and the signature and look for a collision in the hash function. They can then send their malicious encrypted message along with the original sender's signature and nobody will be able to detect foul play.

Q51: What is the plaintext and ciphertext space for ElGamal?

A51: Both are non-negative integers less than some large prime.

Q52: Is ElGamal a public or private system? What is/are the key(s)?

A52: Public.

- The **public** key is a set of three integers, (p, α, β) where:
  - p is a large prime,
  - α is a positive integer less than p, which, when repeatedly squared, covers the entire set of positive integers less than p,
  - $\beta \equiv \alpha^a \pmod{p}$ where a is a random integer between 2 and $p-2$.
- The **private** key is a which is the random integer between 2 and $p-2$ from the definition of β in the public key.

Q53: How do you encrypt a plaintext message with ElGamal?

A53: Suppose our public key is p, α, β and the message to send is x. To encrypt the message, choose a random integer d between 2 and $p-2$ and calculate:

$$(c_1, c_2) \equiv \left(\alpha^d,\, \beta^d x\right) \pmod{p}$$

Q54: How do you decrypt a ciphertext message with ElGamal?

A54: Suppose our private key is a and the encrypted message is (c_1, c_2). To decrypt the message, calculate:

$$x \equiv (c_1^a)^{-1} c_2 \pmod{p}$$

Q55: What are the best attacks for ElGamal?

A55: If we obtain one plaintext-ciphertext pair and another ciphertext which was encrypted using the same random exponent d, we can recover the plaintext corresponding to the second ciphertext.

Suppose we know x_1 encrypts to (y_1, y_2) and we wish to find the value of some x_2 that maps to (c_1, c_2).

We first note the value of β^d as follows:

$$\beta^d \equiv y_2 x_1^{-1} \pmod{p}$$

Next we use this expression to calculate the value of x_2 as follows:

$$\begin{aligned} x_2 &\equiv \left(\beta^d\right)^{-1} c_2 \pmod{p} \\ &\equiv \left(y_2 x_1^{-1}\right)^{-1} c_2 \pmod{p} \\ &\equiv y_2^{-1} x_1 c_2 \pmod{p} \end{aligned}$$

Q56: Explain how a man-in-the-middle attack works where the sender, opponent, and receiver are named Alice, Eve, and Bob respectively.

A56:

- Alice wants to encrypt a message and send it to Bob.

- Eve can listen to Alice's requests to the public key directory and inject her own communications to Alice and Bob.
- Alice requests Bob's public key from the directory.
- Eve silently sends Alice **her** public key instead of Bob's.
- Alice encrypts her message with Eve's public key instead of Bob's and sends it through Eve with intent of arriving to Bob.
- Eve decrypts the message with her private key and reads it.
- Eve then re-encrypts the message with Bob's public key and sends it to Bob.
- Neither Alice nor Bob know Eve has read the message.

Q57: What does PKI stand for and what does it aim to achieve?

A57: Public Key Infrastructure. It provides a means of trusting the authenticity of senders and receivers of publicly encrypted messages.

Q58: What is one method of certification in PKI? Roughly how does it work?

A58: Certificate Authorities. A publicly trusted certificate authority (CA) signs and publishes people's public keys verifying they are who they say they are. If you trust the CA, you can trust a public key belongs to somebody if it's been signed by that CA.

Q59: What is one method of certification in PKI? Roughly how does it work?

A59: Web of Trust. There is no centralised certificate authority, but instead, end users are all encouraged to meet each other and personally verify each other's identity and sign their public keys in person.

Q60: What is the difference between a synchronous and asynchronous stream cipher?

A60:

- **Synchronous:** The keystream is independent of the plaintext.

- **Asynchronous:** The keystream uses the plaintext to generate the element of the stream.

Q61: What does it mean for a stream cipher to be periodic with period d?

A61: The keystream repeats itself after d elements.

Q62: What is the plaintext and ciphertext space for the Autokey cipher?

A62: Both are non-negative integers less than 26.

Q63: Is the Autokey cipher a public or private system? What is/are the key(s)?

A63: Private. The key is a non-negative integer less than 26 is called the "seed" and the rest of the keystream continues to take on the values from the plaintext message but does not include the last element of the plaintext message.

Q64: How do you encrypt a plaintext message with the Autokey cipher?

A64: Add the key stream to the message modulo 26. In other words, add the seed to the first letter of the message and then add the first letter of the message to the next letter of the message and so on until you add the second last letter of the message to the last letter of the message

Q65: How do you decrypt a plaintext message with the Autokey cipher?

A65: Subtract the key stream from the message modulo 26.

Q66: Is the Autokey cipher synchronous or asynchronous?

A66: Synchronous.

Q67: Is the Autokey cipher periodic? If so, what is the period?

A67: It's non-periodic.

Q68: What are the best attacks for the Autokey cipher?

A68: There are only 26 initial seeds. Just try them all and look for something that looks like English when you decrypt using that seed.

Q69: What does it mean for a linear feedback shift register to be m-stage?

A69: The seed must include a pair of bitstrings of length m where the neither is ending with a 0.

Q70: What is the plaintext and ciphertext space for a linear feedback shift register?

A70: Both are binary digits (bits).

Q71: Is an m-stage linear feedback shift register a public or private system? What is/are the key(s)?

A71: Private. The key is a pair of strings of m bits which is called the "seed". The next element of the keystream is a linear combination of the previous m bits of the keystream. For example, if the seed consists of the following two bit strings:

$$\begin{aligned} l_1, l_2, \cdots, l_m \\ c_0, c_1, \cdots, c_{m-1} \end{aligned}$$

then the $(m+i)$th element of the keystream is given by:

$$l_{m+i} = c_0 l_i + c_1 l_{i+1} + \cdots + c_{m-1} l_{i+m-1}$$

Q72: How do you encrypt a plaintext message with a linear feedback shift register?

A72: Add the keystream to the message.

Q73: How do you decrypt a plaintext message with a linear feedback shift register?

A73: Subtract the keystream from the message.

Q74: Is a linear feedback shift register synchronous or asynchronous?

A74: Synchronous.

Q75: Is an m-stage linear feedback shift register periodic? If so, what is the period?

A75: Yes. A good choice of starting seed can give a period of $2^m - 1$.

Q76: What are the best attacks for an m-stage linear feedback shift register?

A76: If you have $2m$ plaintext-ciphertext pairs, you can easily compute the first $2m$ keystream bits by adding each plaintext-ciphertext pair (i.e. $l_i \equiv x_i + y_i \pmod 2$).

From there, solve m linear equations for $c_0, c_1, \cdots c_{m-1}$.

$$\begin{aligned} l_{m+1} =& c_0 l_1 + c_1 l_2 + \cdots + c_{m-1} l_m \\ l_{m+2} =& c_0 l_2 + c_2 l_3 + \cdots + c_{m-1} l_{m+1} \\ & \vdots \\ l_{2m} =& c_0 l_m + c_2 l_{m+1} + \cdots + c_{m-1} l_{2m-1} \end{aligned}$$

And then we will have obtained the seed (i.e. the key) to the cipher.

Q77: Name and label each component of Enigma.

A77:

- The keyboard
- The plugboard S
- The rotors L, M, N
- The reversing drum R

- The glowlamps

Q78: What does the plugboard do in Enigma?

A78: Allows up to 6 pairs of letters to be swapped as configured by the user.

Q79: What do the three rotors do in Enigma? How do they work?

A79:

- Each rotor arbitrarily permutes the letters coming in. Not swapping, you can't simply run a letter through the rotors twice to get back the same thing.
- Each rotor has its permutation hardwired.
- The first rotor spins after every key press.
- The second rotor spins every time the first rotor has made a full revolution (i.e. every 26 key presses).
- The third rotor spins every time the second rotor has made a full revolution (i.e. every 26×26 key presses).

Q80: What does reversing drum do in Enigma?

A80: Arranges the 26 letters into 13 pairs and swaps each pair of letters so that the input is never equal to the output. This swapping permutation is hardwired into the drum.

Q81: Describe the path of a signal from its initial position on the keyboard through to its final position on the glowlamps.

A81:

1. Signal runs through the switch board to potentially get swapped to a signal representing a different letter.
2. Signal runs through each rotor, being arbitrarily permuted as it does.
3. Signal runs through the reversing drum, being swapped with a different letter (not itself).

4. Signal runs back through each rotor again in reverse, being arbitrarily permuted again as it does.
5. Signal runs back through the plugboard to potentially swap whichever letter comes in with a different letter.

Q82: What are two properties that a good pseudo-random number generator have? Explain what is meant by each.

A82: Should be:

- **Unpredictable:** If we have so far generated $x_1, x_2, \cdots x_k$, there is no polynomial-time algorithm that can predict x_{k+1} with probability greater than 1/2.
- **Deterministic/reproducible:** The same stream of numbers can be generated with the same small starting seed.

Q83: What is the Avalanche Effect of good cipher design?

A83: Small parts of the input affect large parts of the output.

Q84: What is the Completeness Effect of good cipher design?

A84: Small parts of the output depend on large parts of the input.

Q85: What is the plaintext and ciphertext space of AES-128?

A85: Both are 128-bit blocks.

Q86: Is a AES-128 a public or private system? What is/are the key(s)?

A86: Private. The key consists of a master key which is used to deterministically generate a sequence of "round" keys i.e. subkeys.

Q87: How do you encrypt a plaintext message with AES-128 with r rounds (high level steps only)?

A87:

1. Add round key K_0 to the message
2. Pack the result into a 4×4 array of bytes one column at a time so that the first 4 bytes of make up the first column of the array read top to bottom.
3. For each round except the last, $i = 1, 2, \cdots, r-1$:
   (a) Substitute bytes
   (b) Shift rows
   (c) Mix columns
   (d) Add round key K_i
4. For the final round, r:
   (a) Substitute bytes
   (b) Shift rows
   (c) Add round key K_r

Q88: In AES-128, explain the "shift rows" stage.

A88: The first row of the 4×4 array stays fixed. The second row is cycled 1 step to the left. The third row is cycled 2 steps to the left. The fourth row is cycled 3 steps to the left.

Q89: In AES-128, briefly explain the "substitute bytes" stage.

A89: Replace each byte of the array using a large S-box.

Q90: In AES-128, briefly explain the "mix columns" stage.

A90: Left multiply each column of the array by a fixed matrix of polynomials in $GF(2^8)$. You will need to convert the bytes in the column to polynomials, do the multiplication, and then convert back to bytes.

Q91: How do you decrypt a ciphertext message with AES-128?

A91: It is largely the same as encryption, except for a few minor tweaks:

- Use the inverse S-box for "substitute bytes"
- Use the inverse matrix for "mix columns"
- Cycle to the right instead of to the left for "shift rows"
- The subkeys are used in the reverse order and need to be transformed by the inverse "mix columns" matrix.

Q92: What does a Feistel cipher take as input and output?

A92: Both are blocks of even length.

Q93: How does a single round of a Feistel cipher work?

A93:

1. A block of even length goes in with the round key.
2. The right half of the block is encrypted with the round key and added to the left half of the block.
3. The right half of the block becomes the left half of the input to the next round.
4. The left half of the block becomes the right half of the input to the next round.

Q94: What is the only difference in a Feistel cipher between encryption and decryption?

A94: Feed the subkeys in the reverse order for each round.

Q95: What is the most important aspect of a Feistel cipher in terms of security?

A95: The half-block encryption function must be highly non-linear.

Q96: What is the bias of a random binary variable X?

A96: If the probability that X is 0 is $\frac{1}{2} + \epsilon$ and the probability that x is 1 is $\frac{1}{2} - \epsilon$ then the bias is ϵ.

Q97: What is the Piling Up Lemma?

A97: If $X_1, X_2, \cdots, X_n$ are random binary variables with biases $\epsilon_1, \epsilon_2, \cdots, \epsilon_n$, then $X_1 + X_2 + \cdots + X_n$ has bias $2^{n-1} \prod_{i=1}^{n} \epsilon_i$.

Q98: How do you calculate the bias of a linear relationship of the form $X_i + Y_j$ within an S-box?

A98: Draw up a table of all the possible X-values in the left columns and their corresponding Y-values in the next columns. Calculate their sum in the proceeding columns and find the percentage all the rows where that sum is 0. This is the bias for that linear relationship.

Q99: What properties would a linear relationship within an S-box have to be considered "useful"?

A99: The bias should be far from 0 and/or it should use few input/output bits.

Q100: Question about linear cryptanalysis.

A100: ...

Q101: Question about differential cryptanalysis.

A101: ...

Q102: Using words, what is the probability of x given y? How is it denoted?

A102: The probability that x happens already given that y has happened. $\mathbb{P}(x \mid y)$.

Q103: What is a useful formula for calculating the probability of x given y?

A103:

$$\mathbb{P}(x \mid y) = \frac{\mathbb{P}(x \text{ and } y)}{\mathbb{P}(y)}$$

Q104: Give two equivalent statements to "x and y are probabilistically independent".

A104:

- $\mathbb{P}(x \mid y) = \mathbb{P}(x)$
- $\mathbb{P}(x \text{ and } y) = \mathbb{P}(x) \cdot \mathbb{P}(y)$

Q105: What is Bayes' Theorem?

A105:

$$\mathbb{P}(y \mid x) = \frac{\mathbb{P}(x \mid y) \cdot \mathbb{P}(y)}{\mathbb{P}(x)}$$

Q106: How do you compute the probability of a ciphertext y occurring in a particular cipherspace $\mathcal{C}$?

A106: For all keys K which can be used to obtain y, calculate the probability of K being chosen multiplied by the probability of the plaintext which encrypts to y under K being chosen. Sum these calculations together. In other words:

$$\mathbb{P}(y) = \sum_{\{K \mid y \in C(K)\}} \mathbb{P}(K) \cdot \mathbb{P}(d_K(y))$$

Q107: How do you compute the probability of a ciphertext y occurring given a particular plaintext x has occurred? In other words, give an expression for $\mathbb{P}(y \mid x)$.

A107: The sum of the probabilities of the occurrence of a key which encrypts x to y.

$$\mathbb{P}(y \mid x) = \sum_{\{K \mid e_K(x) = y\}} \mathbb{P}(K)$$

Q108: Using Bayes' Theorem, how do you compute the probability of a plaintext x occurring given a particular ciphertext y has occurred? In other words, give an expression for $\mathbb{P}(x \mid y)$.

A108: Calculate the probability of y occurring given x has occurred. Then calculate the probability of x and y occurring independently. Then apply Bayes' Theorem, i.e.

$$\mathbb{P}(x \mid y) = \frac{\mathbb{P}(y \mid x) \cdot \mathbb{P}(x)}{\mathbb{P}(y)}$$

Q109: What is the easiest way to remember which order the individual probabilities on the top and bottom go in Bayes' Theorem?

A109: Consider the case where x and y are independent so that $\mathbb{P}(x \mid y) = \mathbb{P}(x)$ and $\mathbb{P}(y \mid x) = \mathbb{P}(y)$. Plug this into the equation and see if it holds true.

Q110: What does a cipher designer want so their cipher achieves perfect secrecy?

A110: The plaintext and ciphertext are independent. In other words, for all plaintexts x and ciphertexts y:

$$\mathbb{P}(x \mid y) = \mathbb{P}(x)$$

Again, in other words, the ciphertext y gives **no knowledge** of the plaintext.

Q111: If every ciphertext in a cipher can be obtained from **some** plaintext-key pair (i.e. the cipherspace contains no unnecessary elements), then what does perfect secrecy require?

A111: The keyspace is at least as big as the ciphertext space which is at least as big as the plaintext space. In other words:

$$|\mathcal{K}| \geq |\mathcal{C}| \geq |\mathcal{P}|$$

Q112: State Shannon's Theorem.

A112: If the keyspace, ciphertext space, and plaintext are all the same size, the cipher obtains perfect secrecy if and only if:

- each key is chosen with equal probability; and
- for each plaintext x and ciphertext y there is a unique key which encrypts x to y.

Q113: What is the plaintext and ciphertext space of Vernam's one-time pad?

A113: Both are strings of bits.

Q114: Is Vernam's one-time pad a public or private system? What is/are the key(s)?

A114: Private. Key is a string of random bits the same length as the plaintext / ciphertext.

Q115: How do you encrypt a plaintext message with Vernam's one-time pad?

A115: Add the private key to the message modulo 2

Q116: How do you decrypt a ciphertext message encrypted with Vernam's one-time pad?

A116: It's the same as encryption, i.e. add the private key to the message modulo 2.

Q117: What are the best attacks for Vernam's one-time pad?

A117: There are none really. It achieves perfect secrecy.

Q118: What are some disadvantages of using Vernam's one-time pad?

A118:

- The key is as long as the message itself so sharing the key is just as difficult as sharing the message itself.

- Generating random bits is hard. You would have to toss a coin for every bit of the message you want to send if you want to do it properly.

Q119: What is the plaintext and ciphertext space for Elliptic Curve Cryptography?

A119: Points on a elliptic curve of the form $y^2 = x^3 + ax + b$.

Q120: What is the additive inverse of an element in an elliptic curve group?

A120: Flip the point about the x-axis.

Q121: How do you calculate $p + q$ in an elliptic curve group?

A121: "Draw" a line which intersects p and q.

- If the line also intersects a third point on the curve, flip that third point about the x-axis to arrive at the result.
- Otherwise, $p + q = 0$.

Q122: Is Elliptic Curve Cryptography a public or private system? What is/are the key(s)?

A122: Public.

- The **public** key is (G, α, β) where:
  - G is an elliptic curve group,
  - α is a generator for G, i.e. an element which, when repeatedly "added" to itself, covers the entire elliptic curve,
  - $\beta = \alpha \cdot a$ where a is a random non-zero point on the curve.
- The **private** key is a which is the random point from the definition of β in the public key.

Q123: How do you encrypt a plaintext message with Elliptic Curve Cryptography?

A123: Suppose our public key is G, α, β and the message to send is x. To encrypt the message, choose a random non-zero point on the curve, d and calculate:

$$(c_1, c_2) = (\alpha \cdot d,\ \beta \cdot d + x)$$

Q124: How do you decrypt a ciphertext message with Elliptic Curve Cryptography?

A124: Suppose our private key is a and the encrypted message is (c_1, c_2). To decrypt the message, calculate:

$$x = -c_1 \cdot a + c_2$$

Q125: What is one advantage of Elliptic Curve Cryptography?

A125: The key size is small compared with RSA which providing approximately the same level of security

Q126: What is one disadvantage of Elliptic Curve Cryptography?

A126: The plaintext space is points on an elliptic curve which does not have an obvious meaningful relation to English letters.

Q127: What the hell is D'oh?

A127: Just say D'oh!

Q128:

A128: