COMPSCI 4CR3 Textbook Problems

Dev Mody

December 2024

Contents

1 Chapter 1

1.1 Question 2

We have received the following ciphertext which was encoded with a shift cipher: xultpaajcxitltlxaarpjhtiwtgxktghidhipxciwtvgtpilpitghlxiwiwtxgqadds.

1.1.1 Perform an attack against the cipher based on a letter frequency count: How many letters do you have to identify through a frequency count to recover the key? What is the cleartext?

To decrypt the ciphertext, we only need one letter since this was done by a shift cipher. The code in my life outlines that the most frequent letter was t and we know that e is the most frequent letter in the English language. To solve this riddle, we need to make 11 steps ahead in the alphabet for each letter. The decrypted plaintext is `if we all unite we will cause the rivers to stain the great waters with their blood`

1.1.2 Who wrote this message?

The person who wrote this message was Tecumseh, a Shawnee chief and warrior who promoted resistance to the expansion of the United States onto Native American lands.

1.2 Question 3

We consider the long-term security of the Advanced Encryption Standard (AES) with a key length of 128 bits with respect to exhaustive key-search attacks. AES is perhaps the most widely used symmetric cipher at this time

1.2.1 Part 1

Assume that an attacker has special-purpose hardware chips (ASICs) that check $5 \cdot 10^8$ keys per second and has a budget of \$1 million. One ASIC costs \$50, and we assume 100% overhead for integrating it. How many ASICs can we run in parallel given the budget? How long does an average key search take? Relate this to the age of the universe, 10^{10} years.

One search engine costs \$100 including the overhead. Thus, \$1 million buys us 10,000 search engines. This means if we run all 10,000 search engines in parallel, we have $5 \cdot 10^8 \cdot 10^4 = 5 \cdot 10^{12}$ key searches per second. Since we're searching for AES 128 bit keys, on average we check 2^{127} keys. Thus, if we divide the number of keys by the speed at which they're searched, we get a time of $1.08 \cdot 10^{18}$ years, which is 100 million times longer than the age of the universe.

1.2.2 Part 2

We try now to take advances in computers into account. The estimate usually applied is Moore's Law which states that the computing power doubles every 18 months while the costs of integrated circuits stays constant. How many years do we have to wait until a key search machine can be built to break 128 bit AES with an average search time of 24 hours? Assume a budget of $1 million.

Let i be the number of Moore iterations needed. $2^i = 1.08 \cdot 10^{18} \cdot 365 \implies i = 68.42 \approx 69$. Thus we have to wait for $1.5 \cdot 69 = 103.5$ years.

1.3 Question 4

We now consider the relations between passwords and their size. For this purpose, we consider a cryptosystem where the user enters a key in the form of a password.

1.3.1 Part 1

Assume a password consisting of 8 letters where each letter is encoded with ASCII code. What is the size of the key space which can be constructed by such passwords?

Since each letter is an ASCII, there are 7 bits per character meaning $2^7 = 128$ possible characters per letter. We have 8 such letters, which means the size of the key space is $(2^7)^8 = 2^{56}$ bits.

1.3.2 Part 2

Assume that most users only use 26 lowercase letters from the alphabet instead of the full 7 bits of the ASCII-encoding. What is the corresponding key length in bits in this case?

In this case, say we had 26 possible characters per letter. Then $2^b = 26$. If we solve for b, we get $b = \log_2(26) \approx 5$ bits per letter. If we have 8 such letters in our password, the key length would be $8 \times 5 = 40$ bits.

1.3.3 Part 3

At least how many characters are required for a password to generate a key length of 128 bits in the case of 7-bit characters or 26 lowercase letters from the alphabet?

For the 7-bit characters, we need at least one character for the password. In the case of the 26 lowercase letters. We know each letter is represented by 5 bits. If we divide $128/5 \approx 26$ letters. Thus we need at least 26 letters to have at least 128 bits for a key length.

1.4 Question 6

In this problem we consider the difference between end-to-end encryption (E2EE) and more classical approaches to encrypting when communicating over a channel that consists of multiple parts. E2EE is widely used, e.g., in instant messaging services such as WhatsApp or Signal. The idea behind this is that encryption and decryption are performed by the two users who communicate and all parties eavesdropping on the communication link cannot read (or meaningfully manipulate) the message.
In the following we assume that each individual encryption with the cipher e() is secure, i.e., the cryptographic algorithm cannot be broken by an adversary. First we look at the communication between two smartphones without end-to-end encryption, shown in Figure 1.7. Encryption and/or decryption happen three times in this setting: Between Alice and base station A (air link), between base stations A and B (through the internet), and between base station B and Bob (again, air link). Describe which of the following attackers can read (and meaningfully manipulate) messages.

A hacker who can listen to (and alter) messages on the air link between Alice and her base station can definitely alter the encrypted message Alice sends. They cannot read the plaintext but they can tamper with the encrypted message thus sending a different message to the base station which would ultimately mean a different message for Bob.
The Mobile Operator that runs and controls Base Station A can definitely read the plaintext and tamper with it because the message from Alice is decrypted and another encrypted message is sent.
Since the Base Stations have the power to read and tamper with encrypted messages, a National Law Enforcement Agency controlling these stations have the power to do so as well.
An intelligence agency of a foreign country that can wiretap any internet communication can only tamper with encrypted messages sent from one base station to the other.

1.5 Question 13

This problem deals with the affine cipher where the key is given as a = 7 and b=22.

1.5.1 Decrypt the text: falszztysyjzyjkywjrztyjztyynaryjkyswarztyegyyj

The decrypted text is FIRST THE SENTENCE AND THEN THE EVIDENCE SAID THE QUEEN

1.5.2 Who wrote the line?

This line was referenced in Alice in Wonderland.

1.6 Question 15

We consider an attack scenario where the adversary Oscar manages to provide Alice with a few pieces of plaintext that she encrypts. Show how Oscar can break the affine cipher by using two pairs of plaintext–ciphertext, (x_1, y_1) and (x_2, y_2). What is the condition for choosing x_1 and x_2?

Oscar can definitely break the Affine Cipher using two pairs of plaintext-ciphertext. This is because the Affine Cipher is analagous to the Caesar Cipher where each letter is shifted the same amount of times. Thus, if two entirely different $x_1 \neq x_2$ plaintexts were encrypted and sent over, Oscar would definitely have two distinct $y_1 \neq y_2$. As a result, it would be possible to do some frequency or analytical attack where Oscar would need to find out the common shifts in letters.

1.7 Question 16

An obvious way to increase the security of a symmetric algorithm is to apply the same cipher twice $y = e_{k2}(e_{k1}(x))$. As is often the case, things can be tricky and the results are often different from the expected ones. In this problem we show that a double encryption with the affine cipher is only as secure as single encryption. Assume $e_{k1} \equiv a_1 x + b_1 \mod 26$ and $e_{k2} \equiv a_2 x + b_2 \mod 26$.

1.7.1 Show that there exists a single cipher $e_{k3} \equiv a_3 x + b_3 \mod 26$ which performs exactly the same as $e_{k2}(e_{k1}(x))$

We can substitute the values to show this holds. $e_{k2}(e_{k1}(x)) \equiv a_2(a_1 x + b_1) + b_2 \mod 26 \equiv a_2 a_1 x + a_2 b_1 + b_2 \mod 26 \equiv a_3 x + b_3 \mod 26 \equiv e_{k3}$

1.7.2 Briefly describe what happens if an exhaustive key-search attack is applied to a double-encrypted affine ciphertext. Is the effective key space increased?

An exhaustive key-search attack on a double-encrypted affine ciphertext does not increase the effective key space because the double encryption can be reduced to a single affine cipher with combined coefficients. Therefore, the key space remains the same as for a single affine cipher, offering no additional security.

2 Chapter 2

2.1 Question 1

The Stream Cipher described in Definition 2.1.1 can be generalized to work in alphabets other than the binary one. For manual encryption, an especially useful one is a stream cipher that works on letters.

2.1.1 Develop a scheme which operates with the letters A, B,. . ., Z, represented by the numbers 0,1,...,25. What does the key (stream) look like? What are the encryption and decryption functions?

In the original stream cipher, the entire concept was XORing each bit of the plaintext to the same corresponding position of the key. In a way if we denote each bit value of the plaintext x_i, each bit value of the key k_i and each bit value of the ciphertext y_i, then we can define the encryption and decryption processes as the following for each bit. $y_i = e(x_i) \equiv x_i + k_i \mod 2$ and $x_i = d(y_i) = y_i - k_i \mod 2$. Using this knowledge, we can say that if we each letter spanning 26 different values, for a string consisting of n different letters, the encryption and decryption processes are similar, just over $\mathbb{Z}_{26}$ instead of $\mathbb{Z}_2$. As a result, $y_i \equiv e(x_i) \equiv x_i + k_i \mod 26$ and $x_i = d(y_i) = y_i - k_i \mod 26$. I have created the following code in the cipher on the GitHub.

2.1.2 Example Decryption

The ciphertext is bsaspp kkuosr, the key is rsidpy dkawoa. Using the code, I was able to find out that the plaintext was kaspar hauser.

2.2 Question 2

Assume we store a one-time key on a DVD with a capacity of 1 Gbyte. Discuss the real-life implications of a one-time pad (OTP) system. Address issues such as the life cycle of the key, storage of the key during the life cycle/after the life cycle, key distribution, generation of the key, etc.

First of all, since the key is generated on this DVD, the distribution of the key must be physical from one person to another. To address security, the one time key on the DVD can be vaulted by another form of encryption. However, this key distribution is very impractical in the real world. The generation of the key must be truly randomized; if a PRNG generated, this system is susceptible to many forms of attacks, mainly known-plaintext attacks. The life cycle of the key is simple. It remains on the DVD but after its usage, it must be properly disposed.

2.3 Question 3

Assume an OTP-like encryption with a short key of 128 bits. This key is then used periodically to encrypt large volumes of data. Describe how an attack works that breaks this scheme.

Since this cryptosystem is OTP-like, I'm assuming that the keys of 128 bit length are randomly generated by PRNGs, thus making them susceptible to attacks because of their deterministic behaviour. First of all, since the key space is quite short, it is possible for the attacker to somehow perform an exhaustive key search for all 2^{128} possible keys to see what the plaintext is by doing a simple XOR. However, this would be computationally infeasible to do. Another way of breaking this cipher would be a known plaintext attack. Since keys are 128 bits long, then the plaintext would be 128 bits long. Therefore, it is possible to obtain some part of the plaintext and since the attacker knows the ciphertext, it is possible to XOR them together to obtain some part of the key. If the attacker obtains another key instance from another example, they would have the ability to observe overlappings that would help obtain the full plaintext by retrieving the full key. Another way to break this scheme would be to perform a side-channel attack where the attacker exploits the leakage of PRNGs.

2.4 Question 4

At first glance it seems as though an exhaustive key search is possible against an OTP system. Given is a short message, let's say 5 ASCII characters represented by 40 bits, which was encrypted using a 40-bit OTP. Explain exactly why an exhaustive key search will not succeed even though sufficient computational resources are available. This is a paradox since we know that the OTP is unconditionally secure. That is, explain why a brute-force attack does not work.

From the above description, we know that in this OTP system, key lengths are 40 bits and they are randomly generated by TRNGs. Thus, the algorithm used to generate keys is not deterministic, making distinct ciphertexts for the same plaintext. Although sufficient computational resources are available to perform the exhaustive key search of 40^5 possible keys, obtaining a key would help understand how one plaintext helps generate the corresponding ciphertext but for future encryptions, this would irrelevant since this is an OTP.

2.5 Question 6

The OTP offers provable security. Describe two major drawbacks of the OTP that render it impractical for most applications such as encryption of emails or instant messaging.

First off, since a key is generated by a TRNG, it must be sent to both people of the cryptosystem to be useful; if you generate two random keys, chances are that they won't be the same thus making it useful for the decryption after the encryption. For the encryption of emails and instant messaging, this would make it a lot more work for no reason because sending the OTP key over an insecure channel like messaging is useless, so it must be sent physically, which is bound for other types of attacks to occur. At the same time, after this key is used, since people can keep track of messaging over these channels, the key must be disposed immediately after usage because the attackers could leverage this property to find out about your conversation after it has occurred.

2.6 Question 7

We will now analyze a PRNG generated by an LFSR of degree 3 characterized by $(p_2 = 1, p_1 = 0, p_0 = 1)$

2.6.1 What is the sequence generated from the seed $(s_2 = 1, s_1 = 0, s_0 = 0)$?

The output sequence is 0, $(0, 1, 1, 1, 0, 1) = 011101$

2.7 Question 8

Assume we have a stream cipher whose period is quite short. We happen to know that the period is 150–200 bits in length. We assume that we do not know anything else about the internals of the stream cipher. In particular, we should not assume that it is a simple LFSR. For simplicity, assume that English text in ASCII format is being encrypted. Describe in detail how such a cipher can be attacked. Specify exactly what Oscar has to know in terms of plaintext/ciphertext, and how he can decrypt all ciphertext.

If a stream cipher has a short period of 150-200 bits, and the attacker knows that English text in ASCII format is being encrypted, the cipher can be attacked using a repetition attack with knowing the plaintext properties. For this, Oscar needs to know a long sample of the ciphertext to capture multiple repetitions of the key stream, and also needs the partial known plaintext alongside the frequencies. Here is what he should do after obtaining this information. He can estimate the period by analyzing the ciphertext for repeating patterns. Once that occurs, he can XOR the plaintext with the ciphertext to reveal part of the key stream and use the periodic nature to construct the entire key stream. Once that is done, he could decrypt the entire ciphertext.

2.8 Question 11

Given is a stream cipher based on a single LFSR as key stream generator. The LFSR has a degree of 256.

2.8.1 How many plaintext/ciphertext bit pairs are needed to launch a successful attack?

For this, since we want to leverage the periodicity of the LFSR, we would need to have at least pairs of $2 \cdot (2^{256} - 1)$ bits. The attack is similar to Question 8, where the key is obtained by XORing the known plaintext and ciphertext.

2.9 Question 12

We conduct a known-plaintext attack on an LFSR-based stream cipher. We know that the plaintext sent out was: 1001 0010 0110 1101 1001 0010 0110 and the ciphertext was: 1011 1100 0011 0001 0010 1011 0001

If we XOR the plaintext with the ciphertext, we get the key stream to be 0010111 0010111 0010111 0010111. As a result, the degree is $7 = 2^m - 1 \implies m = 3$. The initial seed is 001.

3 Chapter 3

3.1 Question 1

One important property that ensures that DES is secure is that S-Boxes are nonlinear. We verify this property by computing the output for S_1 for several pairs of inputs. Show that $S_1(x_1) \oplus S_1(x_2) \neq S_1(x_1 \oplus x_2)$ for:

1. $x_1 = 000000, x_2 = 000001$ To work with S-Boxes like S_1, we need to treat the first four bits of the 6 as the column number to work with. Thus, $x_1 = 000000$ meaning we refer to Column 0 and Row 0. Similarly for $x_2 = 000001$, we refer to Column 0 and Row 1 of the S_1 S-box. Here, $S_1(x_1) = (14)_{10} = (1110)_2$ and $S_1(x_2) = (00)_{10} = (0000)_2$. If we calculate $x_1 \oplus x_2 = 000001 = 0001$. We can also see that $S_1(x_1) \oplus S_1(x_2) = 1110_2$. Since $S_1(x_1) \oplus S_1(x_2) \neq S_1(x_1 \oplus x_2)$, we have shown S_1 is nonlinear for this case.

3.2 Question 2

The S-Box S_4 has special properties. Show that the first row can be computed from the 0th row with the help of the following mapping: $(y_1, y_2, y_3, y_4) \rightarrow (y_2, y_1, y_4, y_3) \oplus (0, 1, 1, 0)$ where (y_1, y_2, y_3, y_4) denote the binary output of the S-box. It is sufficient to show the mapping for the first five entries of the row.

The first five entries of the 0th row of S_4 are $07, 13, 14, 03, 00, 06$ respectively for each of the columns. For each entry, we will convert it to its binary form and the form above, and then perform the calculation to make sure this property holds:

1. $x = (07)_{10} = (00000111)_2 = (0111)_2$ where $y_1 = 0, y_2 = y_3 = y_4 = 1$. Then $(y_2, y_1, y_4, y_3) = 1011$. If we do $1011 \oplus 0110$, we get $1101 = (13)_{10}$, where this is the corresponding value for Row 1 and Column 0.
2. We continue this for each of the five entries.

3.3 Question 3

We want to verify that $IP(\cdot)$ and $IP^{-1}(\cdot)$ are truly inverse operations. Consider a vector $x = (x_1, x_2, \ldots, x_{64})$ of 64 bits. Show that $IP^{-1}(IP(x)) = x$ holds for the first five bits of x. In the 8×8 matrix of IP and IP^{-1}, we can say that the first five bits of the vector x are $(x_1, x_2, x_3, x_4, x_5)$ are $1, 2, 3, 4, 5$. The mapping after IP transforms x to be the positions $58, 50, 42, 34, 26$. After applying IP^{-1}, we get the positions $x = (1, 2, 3, 4, 5)$ again. Thus, hence proven.

3.4 Question 4

What is the output of the first round of the DES algorithm when the plaintext and the key are both all zeros?

Left $= 00000000000000000000000000000000$ and Right $= 11011000110110001101101110111100$

3.5 Question 5

What is the output of the first round of the DES algorithm when the plaintext and the key are both all ones?

Left = 11111111111111111111111111111111 and Right = 00100111001001110010010001000011

3.6 Question 6

Remember that it is desirable for good block ciphers that a change in one input bit affects many output bits, a property that is called diffusion or the avalanche effect. We try now to get a feeling for the avalanche property of DES. We apply an input word that has a "1" at bit position 57 and all other bits as well as the key are zero. (Note that the input word has to run through the initial permutation.)

1. How many S-Boxes get a different input compared to the case when all-zero plaintext is provided? Since permutations are considered diffusion operations, after performing the initial permutation we could see a minor differences in the first round of the S-Box Substitution step after Key Expansion. We have a single S-Box contain 1 as an input in the bitstring whereas the rest of the S-Boxes have 0s as inputs. However, as we get to more and more rounds, we start to see more changes.

2. The first round will have the minimum number of output bits that will change because we're considering only one bit flip in the input. This is where only one output bit will change.

3.7 Question 7

An avalanche effect is also desirable for the key: A one-bit change in a key should result in a dramatically different ciphertext if the plaintext is unchanged.

1. Assume an encryption with a given key. Now assume the key bit at position 1 (prior to PC–1) is flipped. Which S-boxes in which rounds are affected by the bit flip during DES encryption? Since the round keys are used in all 16 rounds of encryption, a change in the key bit will alter all 16 rounds keys. As a result, since the round keys are XORed with the expanded key before entering the S-Boxes, changes will propagate to all S-Box inputs. The S-Boxes used in all 16 rounds of DES during encryption

2. Which S-boxes in which DES rounds are affected by this bit flip during DES decryption? Since the same operation is performed just with a reversed key stream, a single key flip will propagate the same kind of changes to the S-Box inputs and outputs as in encryption.

3.8 Question 8

In this problem we look at the relationship between the DES round keys and the original key. It turns out that each of the 48 bits of every round key $k_1, \ldots, k_16$ is a direct map of one bit of the original 64 bit input key k

1. Determine which of the bits of k form the first two bits of the round key k_1 The first two bits of the round key k_1 come from the 56 bits that we chosen from 64 bits of k. From the Permutation Choice Table 2, we can see that the first two bits come from the 14th and 17th positions of the 64 bit key.

2. Determine which of the bits of k form the first two bits of the round key k_2 The first two bits of k_2 come from the 56 bits that we augmented from Round 1. This was when 56 bits were chosen from the 64 original bits by removing parity bits and then performing left shifts a certain number of times on the right and left sides of the key and sending them over to Round 2 and then to Permutation Choice Table 2.

3.9 Question 9

A DES Key K_w is called a weak key if encryption and decryption are identical operations

1. Describe the relationship of the subkeys in the encryption and decryption algorithm that is required so this property is fulfilled. For $DES_{K_w}(x) = DES^{-1}_{K_w}(x)$ to be true, we must have $k_1 = k_{16}, k_2 = k_{15}, \ldots$. Thus, this can only occur when all subkeys are identical to their inverse counterparts.

2. What are the four weak DES keys? The first is $0x0000000000000000$, the second is $0xFFFFFFFFFFFFFFFF$, the third is $0x0101010101010101$ and the last is $0xFEFEFEFEFEFEFEFE$.

3. The chance of having a weak key is $4/2^{56}$.

3.10 Question 11

Assume we perform a known-plaintext attack against DES with one pair of plaintext and ciphertext. How many keys do we have to test in a worst-case scenario if we apply an exhaustive key search in a straightforward way? How many on average?

In the worst case scenario in an exhaustive, we would need to test 2^{56} keys since out of the 64 bits, keys are constructed using 56 bits. In this case, the worst case is when the last key is correct for the attack. On average, the correct key in an exhaustive search should come halfway, thus narrowing our search by 2. This means that we would test 2^{55} on average for an exhaustive search.

4 Chapter 4

4.1 Question 3

Generate the multiplication table for the extension field $GF(2^3)$ for the case that the irreducible polynomial is $P(x) = x^3 + x + 1$.

The possible values of $GF(2^3)$ is $000 \equiv 0, 001 \equiv 1, 010 \equiv x, 011 \equiv x+1, 100 \equiv x^2, 101 \equiv x^2+1, 110 \equiv x^2+x, 111 \equiv x^2+x+1$.

×	000	001	010	011	100	101	110	111
000	000	000	000	000	000	000	000	000
001	000	001	010	011	100	101	110	111
010	000	010	100	101	011	001	111	101
011	000	011	110	101	111	100	001	010
100	000	100	011	111	110	010	101	001
101	000	101	001	100	010	111	011	110
110	000	110	111	001	101	011	010	100
111	000	111	101	010	001	110	100	011

4.2 Question 5

The Multiplication is on $GF(2^4)$ and the Irreducible Polynomial is $P(x) = x^4 + x + 1$

$A(x) = x^2 + 1, B(x) = x^3 + x^2 + 1 \implies A(x) \cdot B(x) \mod P(x) \equiv x^3 + x^2$
$A(x) = x^2 + 1, B(x) = x + 1 \implies A(x) \cdot B(x) \mod P(x) \equiv x^3 + x^2 + x + 1$

6 Chapter 6

6.1 Question 1

As we have seen in this chapter, public-key cryptography can be used for encryption and key exchange. Furthermore, it has some properties (such as non-repudiation) that are not offered by secret-key cryptography. So why do we still use symmetric-key cryptography in most applications in practice?

From a theoretical point of view, public key cryptography is seen as a replacement for symmetric cryptography. However, in practical applications, symmetric ciphers are faster than public key schemes. Thus, we use them for bulk data encryption and when the transfer of keys is done in a secure manner like from Diffie-Hellman Key Exchange.

6.2 Question 2

In this problem, we want to compare the computational performance of symmetric and asymmetric algorithms. Assume a fast public-key library such as OpenSSL that can decrypt data at a rate of 2.5 MByte/sec using the RSA algorithm on a modern PC. On the same machine, AES can decrypt at a rate of 2.5 GByte/sec. Assume we want to decrypt a movie stored on a Blu-ray disc. The movie requires 50 GByte of storage. How long does decryption take with each algorithm?

For OpenSSL, it would take 20000 seconds meaning nearly 5 and a half hours. For AES, it would take 20 seconds.

6.3 Question 3

Assume a (small) company with 120 employees. A new security policy demands that all email communication between all employees must be encrypted with a symmetric cipher. How many keys are required if there should be a unique secure channel between every possible pair of communicating parties?

In terms of combinations, there are exactly 7140 pairs of employees in the company, where each pair must have a unique key. Thus, since two of the same keys need to be dispersed per pair, there would be exactly 14,280 keys required for this to work.

6.4 Question 5

Using the basic form of the Euclidean algorithm, compute the greatest common divisor of

1. 7469 and 2464. The GCD is 77. I used the code on the GitHub

2. 2689, 4001. The GCD is 1

3. 286875 and 333200. The GCD is 425

7 Chapter 7

7.1 Question 1

Let the two primes $p = 41$ and $q = 17$ be the setup parameters for RSA.

7.1.1 Which of $e_1 = 32$ and $e_2 = 49$ is a valid RSA exponent? Justify your choice.

First of all, since the two primes are given to us, we can simply calculate our public modulus $n = p \cdot q = 697$. Using `sympy` I can calculate $\Phi(n) = 640$. I run a simple for-loop for each number in the Integer Ring $\mathbb{Z}_{\Phi(n)}$ to see if e_1 and e_2 have modular inverses. I found out that e_1 has no modular inverse while e_2 has the modular inverse 209 under $\Phi(n)$. Thus, I choose e_2 to be my public exponent so I can find $d = 209$.

7.2 Question 3

Encrypt and decrypt by means of the RSA algorithm with the following system parameters

- $p = 3, q = 11, d = 7, x = 5 \implies n = 33, d = 7$. This means we have to find e via finding the modular inverse of d with respect to $\Phi(n) = 20$. We found e using the Extended Euclidean Algorithm to be 3. By encrypting $y \equiv x^e \mod n \equiv 26$, we used the private key d to decrypt $x \equiv y^d \mod n \equiv 5$.

7.3 Question 8

Popular RSA modulus sizes are 2048, 3072 and 4092 bits.

1. How many random odd integers do we have to test on average until we expect to find one that is a prime?
   To find how many random odd integers we have to test on average until we find a prime, we know that the probability for a random odd prime number p is approximately $2/\ln(p)$. For a 2048 bit modulus n, the primes p and q should each have a length of about 1024 bits. Thus the probability of each being prime is $2/\ln(2^{1024}) = 2/1024\ln(2) \approx 1/354$. Thus, we have to perform 354 random odd numbers in 1024 bits until we find one that is a prime. Similarly, for a 3072 bit modulus, we consider the individual primes to have bit lengths $3072/2 = 1536$. In this case, we apply $2/1536\ln(2)$ to show that we would need approximately 532 tests. Finally for 4092 bit modulus, we would need 709 tests.

2. Derive a simple formula for any arbitrary RSA modulus size.
   Given the modulus size n, we know that they must be split into two primes p and q such that their lengths are half of n. We know that the expected number of trials is the reciprocal of the probability of a number being a prime. We know that for a given modulus n of bit length b, the probability of p of b bits being prime is $4/b\ln(2)$. As a result, we know that the expected number of trials is approximately $b\ln(2)/4$.

7.4 Question 9

One of the most attractive applications of public-key algorithms is the establishment of a secure session key for a private key algorithm such as AES over an insecure channel. Assume Bob has a pair of public/private keys for the RSA cryptosystem. Develop a simple protocol using RSA which allows the two parties Alice and Bob to agree on a shared secret key. Who determines the key in this protocol, Alice, Bob or both?

We could use the Diffie-Hellman Key Exchange but using the RSA protocol. In this case, Bob has his public key (n, e) and private key d. Bob sends over his public key to Alice over the insecure channel. On Alice's side, she generates a secret session key using Bob's public key. Alice encrypts this session key using Bob's public key and sends this cipher over to Bob. Bob uses his private key to decrypt the cipher to obtain the shared session key. This session key could be a symmetric encryption key that could be used between the two individuals. In this case, Alice was the one who determined this session key whereas Bob was the person who enabled the communication between the two.

7.5 Question 10

In practice, it is sometimes desirable that both communication parties influence the selection of the session key. For instance, this prevents the other party from choosing a key which is a weak key for a symmetric algorithm. Some block ciphers such as DES and IDEA have weak keys. Messages encrypted with weak keys can be recovered relatively easily from the ciphertext. Develop a protocol similar to the one above in which both parties influence the key. Assume that both Alice and Bob have a pair of public/private keys for the RSA cryptosystem. Please note that there are several valid approaches to this problem. Show just one.

The solution to this problem is literally the Diffie-Hellman Key Exchange with large modulus values to make the Discrete Logarithmic Problem computationally infeasible to break.

7.6 Question 11

In this exercise, you are asked to attack an RSA-encrypted message. You are the attacker and you obtain the ciphertext $y = 1141$ by eavesdropping on the channel. The public key is $k_{pub} = (n, e) = (2623, 2111)$.

1. Consider the encryption formula. All variables except the plaintext x are known. Why can't you simply solve the equation for x?

   This is a hard problem to do because in the Discrete case, finding the Discrete Logarithm of x with respect to a large modulus n is quite difficult.

2. In order to determine the private key d, we need to calculate $d \equiv e^{-1} \mod \Phi(n)$. There is an efficient expression for calculating $\Phi(n)$. Can we use it here?

   Since n is not a prime number, we must resort to finding the prime factorization of n to determine $\Phi(n)$ instead of letting $\Phi(n) = n - 1$ if n was prime.

3. $n = 43 \times 61$. Thus, $\Phi(n) = (43 - 1) \times (61 - 1) = 2520$. We must now calculate d via finding the modular inverse of $e = 2111$ with respect to 2520. We use the extended euclidean algorithm to derive $d = 191$. We can now calculate x via decrypting with d by doing $y^d \mod n$. Thus, $x = 1088$.

7.7 Question 12

We now show how an attack with chosen ciphertext can be used to break an RSA encryption.

1. Show that the multiplicative property holds for RSA. Show that the product of two ciphertexts y_1 and y_2 is equal to the encryption of the product of the two respective plaintexts x_1 and x_2.

   Given the plaintexts x_1 and x_2 were encrypted with the public key n, e. We can say that $y_1 \equiv x_1^e \mod n$ and $y_2 \equiv x_2^e \mod n$. We can say that $y_1 \cdot y_2 \equiv (x_1^e \mod n) \cdot (x_2^e \mod n) \mod n$. By the multiplication property of modular arithmetic, this is equivalent to $y_1 \cdot y_2 \equiv (x_1 \cdot x_2)^e \mod n$. Thus this property holds.

2. Assume Oscar eavesdrops and obtains a ciphertext y_1, which is the encrypted version of a message x_1 that was setn from Alice to Bob. Oscar wants to know x_1. Oscar computes t which he encrypts. We assume that he can obtain the decryption of one ciphertext that he sends to Bob by having access to Bob's computer at a certain point in time. Show how Oscar can construct ciphertext y in such a way that he can use its decryption for computing x_1.

   We use the multiplicative property of RSA to show how this attack works. We know that $y_1 \equiv x_1^e \mod n$ for some public key (n, e) for which there is a corresponding private key d. Since Oscar encrypts his t in the same way, he sends $t^e \mod n$. By the multiplicative property, he can craft a single message using y and his encrypted message to form $y \equiv (y_1 \cdot t)^e \mod n$. Since he has access to Bob's computer, Oscar knows d. Thus he knows $x_1 \times t \mod n$. He can apply the modular inverse of t with respect to n using the Extended Euclidean Algorithm to find x_1.

7.8 Question 13

Alice wants to send a message to Bob encrypted with his public key (n, e). She decides to use the ASCII table to assign a number to each character and to encrypt them separately.

7.8.1 Oscar eavesdrops on the transferred ciphertext. Describe how he can successfully decrypt the message by exploiting the non probabilistic property of RSA.

If the message is long enough, and letters are repeated, Oscar can notice that some ciphertexts are the same due to the non-probabilistic property of RSA. By doing a letter frequency attack, he could obtain the exact message of the cipher without having to go through the long computation process.

7.8.2 Bob's public key is $(3763, 11)$. Decrypt the ciphertext $y = 2514, 1125, 333, 3696, 2514, 2929, 3368, 2514$. Assume Alice only chose capital letters $A, \ldots, Z$.

The decrypted text is SIMPSONS.

7.8.3 Is the attack still possible if we use OAEP padding?

The simple answer is no. By applying the padding and hashing, it is no longer computationally feasible to see the deterministic or non-probabilistic property of RSA, thus not making this possible.

8 Chapter 8

8.1 Question 1

Determine the order of all elements of the multiplicative groups of:

1. $\mathbb{Z}_5^\times$. Element 1 has an order of 1, elements 2 and 3 has an order of 4 and element 4 has an order of 2. This group has 4 elements. Not all the orders divide the number of elements. The primitive elements are $2 and 3$.
2. The other parts are done exactly like this

8.2 Question 2

Consider group $\mathbb{Z}_{53}^\times$. What are the possible element orders? How many elements exists for each order?

There are 6 possible orders: 52, 1, 26, 13, 4 and 2. $\mathbb{Z}_{53}^\times$ is a subgroup of order 52. The elements $1, 2, 3, 5, 8, 12, 14, 18, 19, 20, 21, 22, 26$ are of order 1. The elements $, 6, 7, 9, 11, 17, 25, 29, 37, 38, 40, 43, 2, 3, 5, 8, 12, 14, 18, 19, 20, 21, 22, 26, 27, 31, 32, 33, 34, 35, 39, 41, 45, 4$ are of order 26, and so on.

8.3 Question 4

You are given a prime $p = 4969$ and the corresponding $\mathbb{Z}_{4969}^\times$.

8.3.1 Determine how many generators exist.

There are 1584 generators

8.3.2 What is the probability of a randomly chosen element being a generator?

The probability is roughly 32%.

8.4 Question 5

Compute the two public keys and the joint key k_{AB} for the DHKE scheme with parameters $p = 467, \alpha = 2, a = 400, b = 134$

I generate two public keys, one for each person by doing $A \equiv \alpha^a \mod p$ and $B \equiv \alpha^b \mod p$. Thus I get $A = 137$ and $B = 84$. By raising both public keys to the opposite private keys and reducing it with respect to p, we get $k_{AB} = 90$.

8.5 Question 6

Prime $p = 467$ and $\alpha = 4$ where α has order 233 in the group, meaning it generates a subgroup with 233 elements. For $a = 400, b = 134$ and $a = 167, b = 134$, why are the shared session keys of DHKE the same?

The reason why the two shared keys are both 161 is because 400×134 and 167×134 are congruent modulo 233.

8.6 Question 7

In the DHKE protocol, the private keys are chose from the set $\{2, \ldots, p-2\}$. Why are the values 1 and p - 1 are excluded?

If private key $x = 1$ was chosen. Then for any generator α and prime modulus p, $X \equiv \alpha^1 \mod p \equiv \alpha$. Since the generator is already public in the system, an attacker would find out that the public key and private key are the same. That means DLP is solved immediately. If $x = p - 1$ was chosen as the private key of one end of the system, then $X \equiv \alpha^{p-1} \mod p$, which by Fermat's Little Theorem, means that $X \equiv 1 \mod p$. This would not make a useful shared key as the DLP would be easily solvable. If both parties use $p - 1$ as their private keys, they would have the same public keys.

8.7 Question 9

This problem demonstrates what can go wrong if a generator is chosen that has certain undesirable properties.

8.7.1 Show that the order of an element $a \in \mathbb{Z}_p$ with $a = p - 1$ is always 2.

In the context of DHKE, we are considering the finite cyclic group $\mathbb{Z}_p^\times$ where $a = p - 1$. We calculate the order of a manually by computing such a k such that $a^k \mod p \equiv 1$. $a^1 \equiv p - 1 \mod p$. $a^2 \equiv (p-1)^2 \mod p \equiv p^2 - 2p + 1 \mod p \equiv 1 \mod p$. Thus, the order is 2.

8.7.2 What subgroup is generated by a?

Any subgroup of order 2 is generated by a. Primarily, the subgroup $\{1, p - 1\}$ is generated by $a = p - 1$.

8.8 Describe a simple attack on DHKE which exploits this property

Say in DHKE, we have some generator α and prime modulus p. If we take some public key $A \equiv \alpha^{p-1} \mod p$. Then we know by Fermat's Little Theorem that $A \equiv 1 \mod p$. Then the shared key is dependent on the other person's public key.

8.9 Question 11

We know that the Diffie-Hellman Problem is equally as hard as the Discrete Logarithm Problem in the group $\mathbb{Z}_p^\times$. However, this only holds for passive attacks where Oscar is only capable of eavesdropping. If Oscar can manipulate messages between Alice and Bob, the key agreement protocol can easily be broken. Develop an active attack against the protocol with Oscar being the man in the middle.

The man-in-the-middle attack is pretty straightforward; it is about Oscar trying to impersonate both of the senders. This requires Oscar to be able to intercept messages and manipulate them instead of just eavesdropping. Say Alice sends her public key to Bob. Oscar intercepts it and generates his own private key, and since the generator and prime modulus are public, Oscar sends his public key back to Alice to simulate Bob. He can do the same for Bob. Thus, he doesn't need to obtain the shared key because the communication is already sabotaged. However, signature schemes come to the rescue in terms of ensuring integrity and validity of the message.

8.10 Question 13

Say $p = 467$ and $\alpha = 2$. We use Elgamal Encryption Scheme where the private key $d = 105$ and $i = 213$ and the plaintext is $x = 33$. Show the Encryption and Decryption Method

Say Bob has the private key $d = 105$. He encrypts his message $K_B \equiv \alpha^d \mod p \equiv 444$. Alice's Public Key is computing via $K_A \equiv \alpha^i \mod p \equiv 29$. The shared keys are both calculated $K_{AB} \equiv K_B^i \mod p$ or $K_{AB} \equiv K_A^d \mod p \equiv 292$. Alice encrypts her messaged with the shared key to get $y \equiv K_{AB} \cdot x \mod p \equiv 296$. Bob decrypts the message to get $x \equiv K_{AB}^{-1} \cdot x \mod p$.

8.11 Question 17

The Elgamal scheme is nondeterministic: A given plaintext x has many valid ciphertexts, e.g., both $x = 33$ and $x = 248$ have the same ciphertext in the problem.

8.11.1 Why is the Elgamal encryption scheme nondeterministic?

By choosing a different secret exponent i, the ciphertext y of the same plaintext x is different every time.

8.11.2 How many valid ciphertexts exist for each message x (general expression)? How many are there for the system in Problem 8.13 (numerical answer)?

In general, there are $\#\{2, 3, \dots, p-2\} = p - 3$ different valid ciphertexts for one plaintext

8.11.3 Is the schoolbook RSA cryptographic system nondeterministic once the public key has been chosen?

The plain RSA cryptosystem is deterministic. A specific plaintext always yields the same ciphertext assuming the same public parameters.

9 Chapter 9

9.1 Question 1

Show the condition $4a^3 + 27b^2 \neq 0 mod p$ is fulfilled for the curve $y^2 \equiv x^3 + 2x + 2 \mod 17$

Here $a = 2$ and $b = 2$. We check $4 \times 8 + 27 \times 4 \mod 17 \neq 0$. As a result, we have shown this property holds.

9.2 Question 2

Perform the Point Addition $(13, 7) + (6, 3)$ in the curve $y^2 \equiv x^3 + 2x + 2 \mod 17$

$(13, 7) + (6, 3) = (7, 11)$

9.3 Question 3

In this chapter, the elliptic curve $y^2 \equiv x^3+2x+2 \mod 17$ is given with $\#E = 19$. Verify Hasse's Theorem for this curve.

Hasse's Theorem states that given an elliptic curve E modulo p, the number of points on the curve $p + 1 - 2\sqrt{p} \leq \#E \leq p + 1 + 2\sqrt{p}$. In this case, $p = 17$. We can verify that $18 - 2\sqrt{(17)} \leq 19 \leq 18 + 2\sqrt{17}$.

9.4 Question 4

The curve $y^2 \equiv x^3 + 2x + 2 \mod 17$ has all of its points being generators. Why is this the case?

Since the number of points on the curve $\#E = 19$ is a prime number, the group of points on the Elliptic Curve form a cyclic group of prime order. In such cases, any element is a primitive element.

9.5 Question 5

Let $E : y^2 \equiv x^3 + 3x + 2 \mod 7$ be an Elliptic Curve.

9.5.1 Compute all the points on E

The points on E are: $[(Infinity), (0, 3), (0, 4), (2, 3), (2, 4), (4, 1), (4, 6), (5, 3), (5, 4)]$

9.5.2 What is the order of the group?

The order of the group is the total number of points, namely 9

9.5.3 Is $\alpha = (0, 3)$ a primitive element?

Since the order of $\alpha = (0, 3)$ is 9 which is the order of the group, it is a primitive element.

9.6 Question 6

Given are three curves. $E_1 : y^2 \equiv x^3+5x+4 \mod 11$**,** $E_2 : y_2 \equiv x^3+15x+29 \mod 28$ **and** $E_3 : y^2 \equiv x^3+12x+11 \mod 13$**. Which one is suited for the cryptosystem?**

First of all, since the modulus for E_2 is not a prime number, it is immediately not suitable. Next, we check the a and b values for E_1 and E_3. For E_1, since $4 \times 125 + 27 \times 16 = 932 \neg \equiv 0 \mod 11$, E_1 is valid. For E_3, since $4 \times 1728 + 27 \times 121 \equiv 0 \mod 13$, it doesn't resemble a proper elliptic curve. Thus E_1 is the most suitable curve.

9.7 Question 7

Given an elliptic curve with group order 16. Show that one can determine the order of any element α **with at most three point doublings.**

The goal is to determine such a k such that $k \cdot \alpha = Infinity$. A key insight is to understand that the order of any element must divide 16. The divisors of 16 are $1, 2, 4, 8, 16$. We must choose the smallest k such that $k \cdot \alpha = Infinity$. Computing $2 \cdot \alpha, 4 \cdot \alpha$ and $8 \cdot \alpha$ requires at most 3 point doublings.

9.8 Question 9

Given is the curve $E : y^2 \equiv x^3 + 9x + 1 \mod 11$**. Determine all points on the curve and their orders. Compute** $2P$ **and** $3P$ **for** $P = (2, 4)$

The points on the curve with their corresponding orders are: $\{(None, None) : 1, (0, 1) : 4, (0, 10) : 4, (1, 0) : 2, (2, 4) : 4, (2, 7) : 4, (3, 0) : 2, (7, 0) : 2\}$
Given $P = (2, 4)$, we can say that $2P = (1, 0)$ and $3P = (2, 7)$

10 Chapter 10

10.1 Question 1

In Section 10.1.3 we state that sender (or message) authentication always implies data integrity. Why? Is the opposite true too, i.e., does data integrity imply sender authentication? Does confidentiality always guarantee integrity?

Let's recall that for a security system to ensure ensure message authentication, the sender of the message must be authentic themselves. Also for a security service to ensure integrity, the messages should not have been modified in the transfer of the message. If a message from Alice to Bob is authentic, it is assured that a modification makes the attacker the originator. The opposite cannot be true because the message can still be unaltered but message authenticity is not given. Confidentiality cannot ensure integrity because the information is secret from all parties.

10.2 Question 2

A painter comes up with a new business idea: He wants to offer custom paintings from photos. Both the photos and paintings will be transmitted in digital form via the internet. One concern that he has is discretion towards his customers, since potentially embarrassing photos, e.g., nude pictures, might be sent to him. Thus, the photo data should not be accessible to third parties during transmission. The painter needs multiple weeks for the creation of a painting, and hence he wants to ensure that he cannot be fooled by someone who sends in a photo assuming a false name. He also wants to be assured that the painting will definitely be accepted by the customer and that she cannot deny the order.

1. Which of the four basic security services are needed for the communication between the customers and the painter.
   Integrity, Message Authentication and Non-Repudiation are needed.
2. Which cryptographic primitives (e.g., symmetric encryption) can be utilized to achieve the security services? Assume that several megabytes of data have to be transmitted for every photo.
   Digital Signatures can be provided.

10.3 Question 4

Compute the RSA signatures for the message $x = 55$ below with the public key (n,e) = (2183, 97) and the private key (d) = (409). Check your results by verifying the signatures.

We know $k_{pub} = (n, e) = (2183, 97)$ and $k_{pr} = d = 409$. Given $x = 55$, we must compute the signature s by performing $s \equiv x^d \mod n$ which is $s = 2178$. Afterwards to verify, we must compute $x' \equiv s^e \mod n$ which is $x' = 55$. Since $x' = x$, we can say this is valid.

10.4 Question 5

Given an RSA Signature Scheme with the public key $(n, e) = (9797, 131)$, which of the following signatures are valid?

- (123, 6292). Valid.
- (4333, 4768). Invalid.
- (4333, 1424). Valid.

10.5 Question 6

Given an RSA signature scheme with a given public key, show how Oscar can perform an existential forgery attack by providing an example.

Oscar can perform an existential forgery attack on a basic RSA Signature Scheme by generating his own $s \in \mathbb{Z}_n$ and computes $x \equiv s^e \mod n$. He basically forges his own signature and message to impersonate Bob.

10.6 Question 8

Given is an RSA signature scheme with EMSA-PSS padding as shown in Section 10.2.3. Describe step-by-step the verification process that has to be performed by the receiver of a signature that was EMSA-PSS encoded.

The signature generation process works like the following. The sender generates some random salt value. We concatenate a new string of using the hash of the original message, the salt and a fixed padding. We compute the hash of this new string, and we concatenate it with another padding and salt to form a data block. We apply a mask generating function to the hash of the changed string to compute the mask of the datablock. We XOR the mask with the datablock to compute the masked block. The signature is the masked block, the hash value, fixed padding and the salt. The verification process proceeds in a similar way. The receiver uses the salt, padding1 and padding 2 values with the received message M to recreate EM and to check whether the EMSA-PSS encoding is correct.

10.7 Question 10

We consider the Elgamal Signature Scheme. Given are Bob's private key $K_{pr} = d = 67$ and $K_{pub} = (p, \alpha, \beta) = (97, 23, 15)$

10.7.1 Calculate the signature (r, s) and verification for the following messages $x = 17, k_E = 31$

We compute $r \equiv \alpha^{k_E} \mod p \equiv 87$. We then compute $s \equiv (x - d \cdot r) \cdot k_E^{-1} \mod (p-1) \equiv 20$.
For the verification, we compute $t \equiv \beta^r \cdot r^s \mod p \equiv 68$ and $\alpha^x \mod p \equiv 68$. Since they're the same, this is true

10.7.2 Compare the RSA signature scbeme with the Elgamal signature scheme. Name some advantages and drawbacks of each?

RSA is preferred for applications requiring fast operations and compact signatures, especially in systems with constrained resources. Elgamal offers stronger security guarantees through randomness but at the cost of efficiency and signature size. It is often replaced in practice by elliptic curve-based variants like ECDSA, which are more efficient than standard Elgamal.

10.8 Question 12

Given is an Elgamal signature scheme with given public parameters. Show how Oscar can perform an existential forgery attack by providing an example of a valid signature.

Given public parameters p, α, β, Oscar chooses two integers i, j such that $gcd(j, p-1) = 1$ and computes his own r, s using those values. He also computes his own message to impersonate Bob.

10.9 Question 13

Given is an Elgamal signature scheme with the public parameters $p, \alpha \in \mathbb{Z}_p^\times$ and an unknown private key d. Due to an incorrect software implementation, there is the following dependency between two consecutive ephemeral keys: $k_{E_{i+1}} = k_{E_i} + 1$. Assume two consecutive signatures for the plaintexts x_1 and x_2 are given (r_1, s_1) and (r_2, s_2). Explain how an attacker is able to calculate the private key with those values.

Recall that the encryption process is by computing $r \equiv \alpha^{k_E} \mod p$ and $s \equiv (x - d \cdot r) \cdot k_E^{-1} \mod (p-1)$. We're given a pair of signatures (r_1, s_1) and (r_2, s_2) alongside p and α. Since $k_{E_{i+1}} = k_{E_i} + 1$, $s_1 \cdot k_{E_1} \equiv (x_1 - d \cdot r_1) \mod (p-1)$ and $s_2 \cdot k_{E_2} \equiv (x_2 - d \cdot r_2) \mod (p-1)$ and this means $s_2 \cdot k_{E_2} \equiv (x_2 - d \cdot r_2) \mod (p-1)$. We can create a system of linear equations and solve for d.

10.10 Question 14

The parameters of a DSA Scheme are given by $p = 59$, $q = 29$, $\alpha = 3$ and Bob's private key is $d = 23$. Show the signing procedure and verification procedure for $h(x) = 17$ and $k_E = 25$

Signing Procedure: Since we have $p = 59, q = 29, \alpha = 3$ and private key $d = 23$, we must first compute $\beta \equiv \alpha^d \mod p$, which gives us $\beta = 45$. Next, we move to the signing procedure where since we're given $k_E = 25$ and $h(x) = 17$, we can compute $r \equiv (\alpha^{k_E} \mod p) \mod q \equiv 22$ and $s \equiv (h(x) + d \times r) \times k_E^{-1} \mod q \equiv 26$. Thus our signed message would be $(17, (22, 26))$.

Verification Procedure: We compute $w \equiv s^{-1} \mod q$, $u_1 \equiv (w \times h_x) \mod q$, $u_2 \equiv (w \times r) \mod q$ and $v \equiv (\alpha^{u_1} \times \beta^{u_2} \mod p) \mod q$. For my calculations, I got $w = 25, u_1 = 19, u_2 = 28, v = 22$. We then compute $v \equiv r \mod q$ and since $22 \mod 29 = 22 = r$, the signature is valid.

10.11 Question 15

Show how DSA can be attacked if the same ephemeral key is used to sign two different messages.

Similarly to the attack on Elgamal Signature Scheme, an attacker can use following system of equations $s_1 \equiv (h(x_1) + d \cdot r) \cdot k_E^{-1} \mod q$ and $s_2 \equiv (h(x_2) + d \cdot r) \cdot k_E^{-1} \mod q$ for known s_1, s_2, x_1 and x_2 to first compute the ephemeral key k_E and then the private key d: we represent this as a system of linear equations and solve for d.