Problems

10.1. In Sect. 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? Justify both answers.

10.2. In this exercise, we want to consider some basic aspects of security services.

1. Does privacy always guarantee integrity? Justify your answer.
2. In which order should confidentiality and integrity be assured (should the entire message be encrypted first or last)? Give the rationale for your answer.

10.3. Design a security service that provides data integrity, data confidentiality and nonrepudiation using public-key cryptography in a two-party communication system over an insecure channel. Give a rationale that data integrity, confidentiality and nonrepudiation are achieved by your solution. (Recommendation: Consider the corresponding threats in your argumentation.)

10.4. 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 photos, might be sent to him. Hence, the photo data should not be accessible for third parties during transmission. The painter needs multiple weeks for the creation of a painting, and hence he wants to assure 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. Choose the necessary security services for the transmission of the digitalized photos from the customers to the painter.
2. Which cryptographic elements (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.

10.5. Given an RSA signature scheme with the public key $(n = 9797, e = 131)$, which of the following signatures are valid?

1. $(x = 123, \text{sig}(x) = 6292)$
2. $(x = 4333, \text{sig}(x) = 4768)$
3. $(x = 4333, \text{sig}(x) = 1424)$

10.6. Given an RSA signature scheme with the public key $(n = 9797, e = 131)$, show how Oscar can perform an existential forgery attack by providing an example of such for the parameters of the RSA digital signature scheme.

10.7. In an RSA digital signature scheme, Bob signs messages x_i and sends them together with the signatures s_i and her public key to Alice. Bob's public key is the pair (n, e); her private key is d.

Oscar can perform man-in-the-middle attacks, i.e., he can replace Bob's public key by his own on the channel. His goal is to alter messages and provide these with a digital signature which will check out correctly on Alice's side. Show everything that Oscar must do for a successful attack.

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

10.9. One important aspect of digital signatures is the computational effort required to (i) sign a message, and (ii) to verify a signature. We study the computational complexity of the RSA algorithm used as a digital signature in this problem.

1. How many multiplications do we need, on average, to perform (i) signing of a message with a general exponent, and (ii) verification of a signature with the short exponent $e = 2^{16} + 1$? Assume that n has $l = \lceil \log_2 n \rceil$ bits. Assume the square-and-multiply algorithm is used for both signing and verification. Derive general expressions with l as a variable.
2. Which takes longer, signing or verification?
3. We now derive estimates for the speed of actual software implementation. Use the following timing model for multiplication: The computer operates with 32-bit data structures. Hence, each full-length variable, in particular n and x, is represented by an array with $m = \lceil l/32 \rceil$ elements (with x being the basis of the exponentiation operation). We assume that one multiplication or squaring of two of these variables modulo n takes m^2 time units (a time unit is the clock period times some constant larger than one which depends on the implementation). Note that you never multiply with the exponents d and e. That means, the bit length of the exponent does not influence the time it takes to perform an individual modular squaring or multiplication.
   How long does it take to compute a signature/verify a signature if the time unit on a certain computer is 100 nsec, and n has 512 bits? How long does it take if n has 1024 bit?
4. Smart cards are one very important platform for the use of digital signatures. Smart cards with an 8051 microprocessor kernel are popular in practice. The 8051 is an 8-bit processor. What time unit is required in order to perform one signature generation in 0.5 sec if n has (i) 512 bits and (ii) 1024 bits? Since these processors cannot be clocked at more than, say, 10 MHz, is the required time unit realistic?

10.10. We now consider the Elgamal signature scheme. You are given Bob's private key $K_{pr} = (d) = (67)$ and the corresponding public key $K_{pub} = (p, \alpha, \beta) = (97, 23, 15)$.

1. Calculate the Elgamal signature (r, s) and the corresponding verification for a message from Bob to Alice with the following messages x and ephemeral keys k_E:

   a. $x = 17$ and $k_E = 31$

b. $x = 17$ and $k_E = 49$
c. $x = 85$ and $k_E = 77$

2. You receive two alleged messages x_1, x_2 with their corresponding signatures (r_i, s_i) from Bob. Verify whether the messages $(x_1, r_1, s_1) = (22, 37, 33)$ and $(x_2, r_2, s_2) = (82, 13, 65)$ both originate from Bob.
3. Compare the RSA signature scheme with the Elgamal signature scheme. Where are their relative advantages and drawbacks?

10.11. Given is an Elgamal signature scheme with $p = 31$, $\alpha = 3$ and $\beta = 6$. You receive the message $x = 10$ twice with the signatures (r,s):

$$(i) \quad (17, 5)$$
$$(ii) \quad (13, 15)$$

1. Are both signatures valid?
2. How many valid signatures are there for each message x and the specific parameters chosen above?

10.12. Given is an Elgamal signature scheme with the public parameters $(p = 97, \alpha = 23, \beta = 15)$. Show how Oscar can perform an existential forgery attack by providing an example for a valid signature.

10.13. Given is an Elgamal signature scheme with the public parameters $p, \alpha \in \mathbb{Z}_p^*$ and an unknown private key d. Due to faulty implementation, the following dependency between two consecutive ephemeral keys is fulfilled:

$$k_{E_{i+1}} = k_{E_i} + 1.$$

Furthermore, two consecutive signatures to the plaintexts x_1 and x_2

$$(r_1, s_1)$$
$$\text{and} \quad (r_2, s_2)$$

are given. Explain how an attacker is able to calculate the private key with the given values.

10.14. The parameters of DSA are given by $p = 59, q = 29, \alpha = 3$, and Bob's private key is $d = 23$. Show the process of signing (Bob) and verification (Alice) for following hash values $h(x)$ and ephemeral keys k_E:

1. $h(x) = 17, k_E = 25$
2. $h(x) = 2, k_E = 13$
3. $h(x) = 21, k_E = 8$

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

10.16. The parameters of ECDSA are given by the curve $E : y^2 = x^3 + 2x + 2 \bmod 17$, the point $A = (5,1)$ of order $q = 19$ and Bob's private $d = 10$. Show the process of signing (Bob) and verification (Alice) for following hash values $h(x)$ and ephemeral keys k_E:

1. $h(x) = 12, k_E = 11$
2. $h(x) = 4, k_E = 13$
3. $h(x) = 9, k_E = 8$