Problems

13.1. In this exercise, we want to analyze some variants of key derivation. In practice, one *masterkey* k_{MK} is exchanged in a secure way (e.g. certificate-based DHKE) between the involved parties. Afterwards, the session keys are regularly updated by use of key derivation. For this purpose, three different methods are at our disposal:

(1) $k_0 = k_{MK}$; $k_{i+1} = k_i + 1$
(2) $k_0 = h(k_{MK})$; $k_{i+1} = h(k_i)$
(3) $k_0 = h(k_{MK})$; $k_{i+1} = h(k_{MK}||i||k_i)$

where $h()$ marks a (secure) hash function, and k_i is the ith session key.

1. What are the main differences between these three methods?
2. Which method provides *Perfect Forward Secrecy*?
3. Assume Oscar obtains the nth session key (e.g., via brute-force). Which sessions can he now decrypt (depending on the chosen method)?
4. Which method remains secure if the masterkey k_{MK} is compromised? Give a rationale!

13.2. Imagine a peer-to-peer network where 1000 users want to communicate in an authenticated and confidential way without a central Trusted Third Party (TTP).

1. How many keys are collectively needed, if symmetric algorithms are deployed?
2. How are these numbers changed, if we bring in a central instance (Key Distribution Center, KDC)?
3. What is the main advantage of a KDC against the scenario without a KDC?
4. How many keys are necessary if we make use of asymmetric algorithms?

Also differentiate between keys which *every* user has to store and keys which are collectively necessary.

13.3. You have to choose the cryptographic algorithms for a KDC where two different classes of encryption occur:

- $e_{k_{U,KDC}}()$, where U denotes an arbitrary network node (user),
- $e_{k_{ses}}()$ for the communication between two users.

You have the choice between two different algorithms, DES and 3DES (Triple-DES), and you are advised to use distinct algorithms for both encryption classes. Which algorithm do you use for which class? Justify your answer including aspects of security as well as celerity.

13.4. This exercise considers the security of key establishment with the aid of a KDC. Assume that a hacker performs a successful attack against the KDC at the point of time t_x, where all keys are compromised. The attack is detected.

1. Which (practical) measures have to be taken in order to prevent decryption of future communication between the network nodes?

2. Which steps did the attacker have to take in order to decipher data transmissions which occurred at an earlier time ($t < t_x$)? Does such a KDC system provide Perfect Forward Secrecy (PFS) or not?

13.5. We will now analyze an improved KDC system. In contrast to the previous problem, all keys $e_{k_{U,KDC}}()$ are now refreshed in relatively short intervals:

- The KDC generates a new (random) key: $k_{U,KDC}^{(i+1)}$
- The KDC transmits the new key to user U, encrypted with the old one: $e_{k_{U,KDC}^{(i)}}(k_{U,KDC}^{(i+1)})$

Which decryptions are possible, if a staff member of the KDC is corruptible and "sells" all recent keys $e_{k_{U,KDC}^{(i)}}$ of the KDC at the point of time t_x? We assume that this circumstance is not detected until the point of time t_y which could be much later, e.g., one year.

13.6. Show a key confirmation attack against the basic KDC protocol introduced in Sect. 13.2.1. Describe each step of the attack. Your drawing should look similar to the one showing a key confirmation attack against the second (modified) KDC-based protocol.

13.7. Show that PFS is in fact not given in the simplified Kerberos protocol. Show how Oscar can decrypt past and future communications if:

1. Alice's KEK k_A becomes compromised
2. Bob's KEK k_B becomes compromised

13.8. Extend the Kerberos protocol such that a mutual authentication between Alice and Bob is performed. Give a rationale that your solution is secure.

13.9. People at your new job are deeply impressed that you worked through this book. As the first job assignment you are asked to design a digital pay-TV system which uses encryption to prevent service theft through wire tapping. As key exchange protocol, a strong Diffie–Hellman with, e.g., 2048-bit modulus is being used. However, since your company wants to use cheap legacy hardware, only DES is available for data encryption algorithm. You decide to use the following key derivation approach:

$$K^{(i)} = f(K_{AB} \parallel i). \tag{13.1}$$

where f is an irreversible function.

1. First we have to determine whether the attacker can store an entire movie with reasonable effort (in particular, cost). Assume the data rate for the TV link is 1 Mbit/s, and that the longest movies we want to protect are 2 hours long. How many Gbytes (where $1M = 10^6$ and $1G = 10^9$) of data must be stored for a 2-hour film (don't mix up bit and byte here)? Is this realistic?

2. We assume that an attacker will be able to find a DES key in 10 minutes using a brute-force attack. Note that this is a somewhat optimistic assumption from an attacker's point of view, but we want to provide some medium-term security by assuming increasingly faster key searches in the future.
   How frequently must a key be derived if the goal is to prevent an offline decryption of a 2-hour movie in less than 30 days?

13.10. We consider a system in which a key k_{AB} is established using the Diffie–Hellman key exchange protocol, and the encryption keys $k^{(i)}$ are then derived by computing:

$$k^{(i)} = h(k_{AB} \parallel i) \tag{13.2}$$

where i is just an integer counter, represented as a 32-bit variable. The values of i are public (e.g., the encrypting party always indicates which value for i was used in a header that precedes each ciphertext block). The derived keys are used for the actual data encryption with a symmetric algorithm. New keys are derived every 60 sec during the communication session.

1. Assume the Diffie–Hellman key exchange is done with a 512-bit prime, and the encryption algorithm is AES. Why doesn't it make cryptographic sense to use the key derivation protocol described above? Describe the attack that would require the least computational effort from Oscar.
2. Assume now that the Diffie–Hellman key exchange is done with a 2048-bit prime, and the encryption algorithm is DES. Describe in detail what the advantages are that the key derivation scheme offers compared to a system that just uses the Diffie–Hellman key for DES.

13.11. We reconsider the Diffie–Hellman key exchange protocol. Assume now that Oscar runs an active man-in-the-middle attack against the key exchange as explained in Sect. 13.3.1. For the Diffie–Hellman key exchange, use the parameters $p = 467$, $\alpha = 2$, and $a = 228$, $b = 57$ for Alice and Bob, respectively. Oscar uses the value $o = 16$. Compute the key pairs k_{AO} and k_{BO} (i) the way Oscar computes them, and (ii) the way Alice and Bob compute them.

13.12. We consider the Diffie–Hellman key exchange scheme with certificates. We have a system with the three users Alice, Bob and Charley. The Diffie–Hellman algorithm uses $p = 61$ and $\alpha = 18$. The three secret keys are $a = 11$, $b = 22$ and $c = 33$. The three IDs are ID(A)=1, ID(B)=2 and ID(C)=3.

For signature generation, the Elgamal signature scheme is used. We apply the system parameters $p' = 467$, $d' = 127$, $\alpha' = 2$ and β. The CA uses the ephemeral keys $k_E = 213$, 215 and 217 for Alice's, Bob's and Charley's signatures, respectively. (In practice, the CA should use a better pseudorandom generator to obtain the k_E values.)

To obtain the certificates, the CA computes $x_i = 4 \times b_i + \mathrm{ID}(i)$ and uses this value as input for the signature algorithm. (Given x_i, $\mathrm{ID}(i)$ follows then from $\mathrm{ID}(i) \equiv x_i \bmod 4$.)

1. Compute three certificates $Cert_A$, $Cert_B$ and $Cert_C$.

2. Verify all three certificates.
3. Compute the three session keys k_{AB}, k_{AC} and k_{BC}.

13.13. Assume Oscar attempts to use an active (substitution) attack against the Diffie–Hellman key exchange with certificates in the following ways:

1. Alice wants to communicate with Bob. When Alice obtains C(B) from Bob, Oscar replaces it with (a valid!) C(O). How will this forgery be detected?
2. Same scenario: Oscar tries now to replace only Bob's public key b_B with his own public key b_O. How will this forgery be detected?

13.14. We consider certificate generation with CA-generated keys. Assume the second transmission of $(\text{Cert}_A, k_{pr,A})$ takes place over an authenticated but insecure channel, i.e., Oscar can read this message.

1. Show how he can decrypt traffic which is encrypted by means of a Diffie–Hellman key that Alice and Bob generated.
2. Can he also impersonate Alice such that he computes a DH key with Bob without Bob noticing?

13.15. Given is a user domain in which users share the Diffie–Hellman parameters α and p. Each user's public Diffie–Hellman key is certified by a CA. Users communicate securely by performing a Diffie–Hellman key exchange and then encrypting/decrypting messages with a symmetric algorithm such as AES.

Assume Oscar gets hold of the CA's signature algorithm (and especially its private key), which was used to generate certificates. Can he now decrypt old ciphertexts which were exchanged between two users before the CA signature algorithm was compromised, and which Oscar had stored? Explain your answer.

13.16. Another problem in certificate systems is the authenticated distribution of the CA's public key which is needed for certificate verification. Assume Oscar has full control over all of Bob's communications, that is, he can alter all messages to and from Bob. Oscar now replaces the CA's public key with his own (note that Bob has no means to authenticate the key that he receives, so he thinks that he received the CA public key.)

1. (Certificate issuing) Bob requests a certificate by sending a request containing (1) Bob's ID $ID(B)$ and (2) Bob's public key B from the CA. Describe exactly what Oscar has to do so that Bob doesn't find out that he has the wrong public CA key.
2. (Protocol execution) Describe what Oscar has to do to establish a session key with Bob using the authenticated Diffie–Hellman key exchange, such that Bob thinks he is executing the protocol with Alice.

13.17. Draw a diagram that shows a key transport protocol shown in Fig. 6.5 from Sect. 6.1, in which RSA encryption is used.

13.18. We consider RSA encryption with certificates in which Bob has the RSA keys. Oscar manages to send Alice a verification key $k_{pr,CA}$ which is, in fact, Oscar's key. Show an active attack in which he can decipher encrypted messages that Alice sends to Bob. Should Oscar run a MIM attack or should he set up a session only between himself and Alice?

13.19. Pretty Good Privacy (PGP) is a widespread scheme for electronic mail security to provide authentication and confidentiality. PGP does not necessarily require the use of certificate authorities. Describe the trust model of PGP and how the public-key management works in practice.