Q1: **Digital signatures and time-stamping**

Q1.1 [3/30] Describe the basic characteristics of the DSS approach to digital signing. What is the advantage of using two pairs of keys for each signature?

Q1.2 [3/30] Alice, Bob and Charlie have made a written agreement and now need to digitally sign it, and to attach a secure time-stamp to each signature. Describe what type of infrastructure they need and a sequence of steps for accomplishing their task.

Q2: **Cryptographic hashing functions**

Q2.1 [2/30] Describe the requirements to be met by a cryptographic hashing function.

Q2.2 [2/30] Describe the Merkle–Damgård construction for hashing a message longer than just one block.

Q2.3 [3/30] Discuss the security of hashing $k|m$, $m|k$, $k|m|k$, where m is a message, k is a secret key and | a symbol denoting concatenation; assume the hashing function built using the Merkle–Damgård construction.

Q3: **Rock-paper-scissors game**

Alice and Bob play a Rock-paper-scissors match. In a single match the two party simultaneously form one of the shapes and the winner is established by the simple chain of circular rules *rock beats scissors*, *scissor beats paper* and *paper beats rocks*. The two players use the following protocol:

[Alice and Bob choose their shapes a and b, where h is a known cryptographic hash function]

A→B: $h(a)$

B→A: b

A→B: a

[Bob checks $h(a)$; then both Alice and Bob know the winner of the game]

Q3.1 [3/30] Discuss possible weaknesses of the protocol, with respect to possible fraudulent behaviors from Alice and/or Bob, both ready to cheat in order to win the game.

Q3.2 [3/30] Fix the weaknesses (small changes!), without introducing third parties or public-key cryptography.

Q4: **Firewall**

Q4.1 [2/30] Illustrate the most relevant characteristics of `iptables`, employed as a firewall.

Q4.2 [2/30] What rules would you set for a mail server accepting connections for EMSTP (port 465) and IMAP (port 993) having a network interface `eth1` exposed to the Internet and another network interface `eth2` exposed to the corporate network?

Q5: **Miscellaneous**

Provide *short* answers to the following questions.

Q5.1 [2/30] What is the existential forgery attack?

Q5.2 [3/30] What is the Optimal Asymmetric Encryption Padding (OAEP) and why does it

Q5: **Miscellaneous**

Provide *short* answers to the following questions.

Q5.1 [2/30] What is the Optimal Asymmetric Encryption Padding (OAEP) and why does it provide "all-or-nothing" security ?

Q5.2 [3/30] Determine the multiplicative inverse of 47 mod 64.

Q5.3 [3/30] Given the two primes 23 and 11 find integer $\alpha > 1$ such that $\alpha^{11} = 1 \bmod 23$

1.1 DSS RELIES ON THE DSA, WHICH USES ASYMMETRIC CRYPTOGRAPHY TOGETHER WITH A SECURE HASH FUNCTION. TO SIGN A MESSAGE, THE SIGNER COMPUTES A HASH OF THE MESSAGE AND THEN GENERATES A SIGNATURE USING BOTH THEIR LONGTERM PRIVATE KEY AND A FRESH RANDOM EPHEMERAL KEY K. THE CORRESPONDING PUB KEY IS USED FOR VERIFICATION.
WE USE TWO PAIRS OF KEYS IN EACH SIGNATURE (LONGTERM KEY PAIR AND THE EPHEMERAL KEY PAIR) BECAUSE, EVEN IF ONE SIGNATURE IS EXPOSED, THE PRIV KEY IS STILL PROTECTED AND THE EPHEMERAL KEY PREVENTS REUSE ATTACKS AND ENSURES THAT EACH SIGNATURE IS UNIQUE.

1.2 THEY NEED A PUBLIC KEY INFRASTRUCTURE (PKI) WITH A TRUSTED CA TO ISSUE AND MANAGE THEIR DIGITAL CERTIFICATES, AND A TIME STAMPING AUTHORITY (TSA) THAT PROVIDES SIGNED TIME STAMPS
EACH SIGNER GENERATES A HASH OF THE AGREEMENT, SIGNS IT WITH THEIR PRIV KEY, AND OBTAINS A DIGITAL SIGNATURE THAT CAN BE VERIFIED WITH THE PUB KEY CERTIFICATED BY THE CA. TO PROVE WHEN THE SIGNATURE WAS CREATED, THE SIGNER THEN SENDS THE HASH OF THE SIGNED DOCUMENT TO THE TSA, WHICH RETURNS A DIGITALLY SIGNED TIMESTAMP TOKEN THAT BINDS THE SIGNATURE TO A PRECISE TIME.
THE DIGITAL SIGNATURE WITH THE TSA PROVIDE AUTHENTICITY, INTEGRITY, NON REPUDIATION AND VERIFIABLE TIMING.

2.1 IT MUST BE PREIMAGE RESISTANT, MEANING THAT GIVEN AN OUTPUT h IT'S INFEASIBLE TO FIND x SUCH THAT $h = H(x)$. IT MUST BE SECOND PREIMAGE RESISTANT, MEANING THAT GIVEN AN INPUT x IT'S INFEASIBLE TO FIND $x' \neq x$ SUCH THAT $H(x) = H(x')$. IT MUST BE COLLISION RESISTANT, SO THAT FINDING TWO DIFFERENT INPUTS THAT PRODUCES THE SAME DIGEST REQUIRE $2^{n/2}$ WORK WITH AN n-BIT HASH. IT SHOULD HAVE THE AVALANCHE EFFECT, WHERE A ONE BIT CHANGE IN THE INPUT CHANGE THE ENTIRE OUTPUT (INTEGRITY).

2.2 IT EXTENDS A FIXED LENGTH COMPRESSION FUNCTION TO HASH ARBITRARILY LONG MESSAGES. THE MESSAGE IS PADDED SO ITS LENGTH IS A MULTIPLE OF THE BLOCK SIZE, AND THE LENGTH OF THE ORIGINAL MESSAGE IS APPENDED TO PREVENT EXTENSION AMBIGUITIES. THE COMPUTATION START FROM A FIXED IV. THE PADDED MESSAGE IS DIVIDED INTO BLOCKS, AND EACH BLOCK IS PROCESSED IN SEQUENCE BY THE COMPRESSION FUNCTION, WHICH TAKES AS INPUT THE PREVIOUS CHAINING VALUE AND THE CURRENT BLOCK, PRODUCING A NEW CHAINING VALUE THIS ITERATIVE PROCESS CONTINUES UNTIL ALL BLOCKS ARE PROCESSED, AND THE FINAL CHAINING VALUE IS THE OUTPUT HASH.

2.3 IF WE USE $H(K \| m)$, THE CONSTRUCTION IS VULNERABLE TO LENGTH EXTENSION ATTACKS: AN ATTACKER WHO KNOWS $H(K \| m)$ BUT NOT K CAN COMPUTE $H(K \| m \| m')$ FOR SOME EXTENSION m'.
IF WE USE $H(m \| K)$, THE KEY IS AT THE END AND THE INTERNAL STATE AFTER PROCESSING m IS EXPOSED, SO IT MAY BE EASIER TO MOUNT ATTACKS.
IF WE USE $H(K \| m \| K)$, SECURITY IMPROVES BECAUSE THE KEY APPEARS BOTH AT THE START AND AT THE END, MAKING HARDER TO MANIPULATE THE HASH.
HOWEVER, THE BEST SOLUTION IS HMAC, WHICH MIXES THE KEY WITH INNER AND OUTER PADS TO AVOID STRUCTURAL WEAKNESS OF MERKLE-DAMGARD.

$$HMAC_K(m) = H((K \oplus OPAD) \| H((K \oplus IPAD) \| m))$$

3.1 THIS PROTOCOL IS INSECURE. FIRST, THE COMMITMENT IS ONE SIDE ONLY, A COMMITS BY SENDING $h(a)$, BUT B NO. AFTER RECEIVING $h(a)$, BOB IS FREE TO CHOOSE b ADAPTIVELY, SO HE CAN COMPUTE $h(\text{ROCK})$, $h(\text{SCISSORS})$, $h(\text{PAPER})$ AND IMMEDIATELY LEARN a, CHOOSING THE BEST MOVE.
SECOND, SINCE ONLY A COMMITTED, SHE COULD ALSO TRY TO CHEAT BY ABORTING AFTER LEARNING b.

3.2 TO FIX THIS, BOTH PLAYERS MUST COMMIT, THEN REVEAL USING A LARGE, RANDOM NONCE: A SENDS $H(a \| r_A)$ AND B SENDS $H(b \| r_B)$, THEN THEY REVEAL a, r_A AND b, r_B. THIS HIDES THE CHOICES, BINDS EACH PLAYER TO THEIR CHOISE AND RESTORE FAIRNESS.

4.1 IPTABLES IS A LINUX TOOL THAT IMPLEMENTS PACKET FILTERING AND FIREWALLING. IT'S STATEFUL, MEANING THAT RULES CAN CHECK THE CONNECTION STATE (NEW - ESTABLISHED - RELATED). RULES ARE ORGANIZED INTO CHAINS (INPUT - OUTPUT - FORWARD - PREROUTING - POSTROUTING). POLICIES CAN BE SET TO ACCEPT OR DROP BY DEFAULT, ALLOWING WHITELIST OR BLACKLIST CONFIGURATIONS. IPTABLES ALLOWS MATCHING BY MANY CRITERIA SUCH AS PROTOCOL, S/D IP, PORT NUMBERS AND INTERFACES.

4.2
```
IPTABLES -P INPUT DROP
IPTABLES -P FORWARD DROP
IPTABLES -P OUTPUT ACCEPT

IPTABLES -A INPUT -i lo -j ACCEPT
IPTABLES -A INPUT -m CONNTRACK --CTSTATE ESTABLISHED,RELATED -j ACCEPT

IPTABLES -A INPUT -i eth2 -j ACCEPT

IPTABLES -A INPUT -i eth1 -p TCP --DPORT 465 -m CONNTRACK --CTSTATE
NEW -j ACCEPT
IPTABLES -A INPUT -i eth1 -p TCP --DPORT 993 -m CONNTRACK --CTSTATE
NEW -j ACCEPT
```

5.1 IT'S THE ABILITY OF AN ADVERSARY TO PRODUCE AT LEAST ONE VALID DIGITAL SIGNATURE OR MAC ON SOME MESSAGE, WHITOUT KNOWING THE SECRET SIGNING KEY. THE FACT THAT A VALID FORGERY EXISTS DEMOSTRATES A WEAKNESS OF THE SIGNATURE OR MAC SCHEME

5.2 OAEP IS A PADDING SCHEME FOR RSA THAT MAKES ENCRYPTION SECURE AND PROBABILISTIC. BEFORE APPLYING RSA, THE PLAINTEXT IS COMBINED WITH RANDOM PADDING AND THEN PROCESSED BY TWO HASH BASED MASKING STEPS, BECAUSE THE SAME MESSAGE ENCRYPTED TWICE PRODUCES DIFFERENT CIPHERTEXTS, PREVENTING DETERMINISTIC ATTACKS.

IT PROVIDES ALL-OR-NOTHING SECURITY BECAUSE THE MESSAGE AND THE RANDOM SEED ARE COMBINED IN SUCH A WAY THAT NO PARTIAL INFORMATION ABOUT THE PLAINTEXT CAN BE OPTAINED.

5.3 $47 \bmod 64 \longrightarrow 47x = 1 \bmod 64$

$64 = 1 \cdot 47 + 17 \quad (17 = 64 - 47)$

$47 = 2 \cdot 17 + 13 \quad (13 = 47 - 2 \cdot 17)$

$17 = 1 \cdot 13 + 4 \quad (4 = 17 - 13)$

$13 = 3 \cdot 4 + 1 \quad (1 = 13 - 3 \cdot 4)$

$\downarrow$

$1 = 13 - 3 \cdot 4 = 13 - 3(17 - 13) = 4 \cdot 13 - 3 \cdot 17 =$

$= 4(47 - 2 \cdot 17) - 3\ 17 = 4 \cdot 47 - 11 \cdot 17 =$

$= 4 \cdot 47 - 11(64 - 47) = 15 \cdot 47 - 11 \cdot 64$

$15\ 47 = 1 \bmod 64 \longrightarrow 47^{-1} = 15 \bmod 64$

5.4 $a^{11} = 1 \bmod 23$

$a^{22} = 1 \bmod 23 \longrightarrow a = 2$

$2^5 = 32 \equiv 9 \bmod 23$

$2^{10} = 9^2 = 81 \equiv 12 \bmod 23$

$2^{11} = 12 \cdot 2 = 24 \equiv 1 \bmod 23$