Digital Signatures

1. Possible reasons are:
   a) The message was modified (error or changed by an adversary).
   b) The signature value is faulty (error or changed by an adversary).
   c) A computational error occurred during signature generation or verification.
   d) The wrong private or public key was used.
2. The new scheme is insecure since different messages have the same signature value. An adversary who knows the signature of (m_1, m_2) automatically gets a valid signature of $(m_1 \oplus m_0,\ m_2 \oplus m_0)$ for any message m_0 of the same length as m_1 and m_2.
   The scheme should be changed: sign $m_1 \| m_2$ instead of $m_1 \oplus m_2$.
3. Only signature b) is valid.
   An existential forgery for a chosen signature value s is $m = s^e \bmod N$. For $s = 99$, we get $m = 17$.
4. The prime factors of N are $p = 127$ and $q = 227$. Then $\varphi(N) = (p-1)(q-1) = 28476$. Compute $d = (5 \bmod 28476)^{-1}$ using the Euclidean Algorithm on input 28476 and 5. This gives $d = -5695 \equiv 22781 \bmod 28476$.
   Signature: $s = 11111^{22781} \equiv 7003 \bmod 28829$.
   Verification: $s^e = 7003^5 \equiv 11111 \bmod 28829$.
5. It is not sufficient to check a given hash value. The verifier might otherwise check the integrity of another document or a fabricated hash value (existential forgery). The verifier needs the received message and has to compute the hash value himself.
6. It is very hard to generate a valid RSA-PSS signature without the private key. An adversary can choose s, but it is very unlikely that the encoded message $EM = s^e \bmod N$ is valid, i.e., that $maskedDB$ and H' correspond to any message m.
7. Then the signature generation is deterministic, but the security of RSA-PSS is not substantially reduced. During verification, DB and the salt value is computed from $maskedDB$ and H'. The result should be compared to the expected salt value.
8. In this experiment, the adversary is given a message m, which is generated uniformly at random, and the public RSA parameters e and N. They need to find a signature value s such that $s^e = m \bmod N$. This is equivalent to plain RSA decryption of m. The RSA assumption says that there is a key generation algorithm such that for *uniform* messages the probability of success is negligible in terms of the key size. Hence the plain RSA signature is secure under these conditions. Note the difference to the original signature forgery experiment: the adversary wins if they create a valid signature of any new message.
9. a) Mallory can strip off Alice's signature and sign the ciphertext using his private key. Bob verifies the signature (using Mallory's public key) and decrypts the ciphertext (using his private key). He might think that the message originates from Mallory.
   b) Alice cannot deny her signature of the *ciphertext*. However, it is difficult for a third party to prove that Alice signed the corresponding plaintext. Bob would

have to reveal his private key to allow the verification of the plaintext, unless encryption is deterministic so that each encryption gives the same ciphertext.

10. a) Suppose that (r, s) is an ElGamal signature of m, i.e.,

$$r \equiv g^k \bmod p \text{ and } s \equiv k^{-1}(H(m) - ar) \bmod q.$$

Then:

$$A^r r^s = g^{ar} g^{ks} = g^{ar} g^{H(m)-ar} = g^{H(m)} \bmod p.$$

This shows that the verification of the ElGamal signature is correct.
Remark: For DSA, r and s are reduced modulo q and verification is done differently. Furthermore, the private key is $x = -a$ and the public key is $A = g^x \bmod p$. The definition of s gives

$$k = s^{-1}(H(m) - ar) = s^{-1}(H(m) + xr) \bmod q$$

and therefore

$$r = g^k = g^{s^{-1}H(m)} g^{s^{-1}xr} = g^{s^{-1}H(m)} A^{s^{-1}r} \bmod p.$$

This is used for verification, i.e., the DSA signature is valid if

$$r \bmod q = \left(g^{s^{-1}H(m)} A^{s^{-1}r} \bmod p \right) \bmod q.$$

b) $A = 4^{20} \equiv 17 \bmod 59$.
c) $r = 4^5 \equiv 21 \bmod 59$, $k^{-1} \equiv 6 \bmod 59$ and $s = 6 \cdot (8 - 20 \cdot 21) \equiv 22 \bmod 29$. The ElGamal signature is $(r, s) = (21, 22)$.
d) $A^r r^s = 17^{21} \cdot 21^{22} \bmod 59 \equiv 46$ is equal to $g^{H(m)} = 4^8 \bmod 59 \equiv 46$.
e) If a signature (r, s), hash $H(m)$ and k is known, then an adversary can easily compute the secret key a:

$$sk = H(m) - ar \bmod q \Rightarrow a = r^{-1}(H(m) - sk) \bmod q$$

If the same secret parameter k is used to generate the signatures s_1 and s_2 of m_1 and m_2, then we have:

$$k = s_1^{-1}(H(m_1) - ar) = s_2^{-1}(H(m_2) - ar) \bmod q.$$

Rearranging the last equation gives a.