Alan Palayil

A20447935

ECE 543

Homework #4

Due Date: 4/9/2023

6.3) d is the multiplicative inverse of e mod $(p-1)(q-1)$, and is its unique up to multiples of $(p-1)(q-1)$.

6.4) $\phi(n) = (p-1)(q-1) = pq - p - q + 1 \approx n - 2\sqrt{n}$

6.6) Based on Homework Problem 6.4, the likelihood of a number not being relatively prime to n is approximately $2/\sqrt{n}$. If this number is not a multiple of n, it could rapidly facilitate the factorization of n. Applying Euclid's algorithm to the number and n would yield either p or q, effectively compromising the security of RSA for n.

6.8) $(m_i^j)^d \bmod n = (m_i^d)^j \bmod n$. Thus, in order to compute the signature on $m_i^j \bmod n$, Fred just raises the signature on m_i to the j^{th} power, mod n.
$(\bmod_i^{-1})^d \bmod n = (m_i^d)^{-1} \bmod n$, so to compute the signature on m_i^{-1} mod n, Fred just computes the inverse mod n of the signature on m_i.
$(m_1 \cdot m_2)^d \bmod n = m_1^d \cdot m_2^d \bmod n$, so to compute the signature on $m_1 \cdot m_2$ mod n, Fred just multiplies the signature on m_1 by your signature on m_2 mod n.

6.10) In ElGamal, the signature X is computed using the message digest of m | T_m (d_m), the secret number (S_m), and the signer's private key (s). The message digest function is public, so d_m can be calculated. If S_m is known, then $d_m S \bmod (p-1)$ can be computed by subtracting S_m from X. If $q = (p-1)/2$ is prime, then $S \bmod q$ can be obtained by multiplying $d_m S \bmod q$ by $d_m^{-1} \bmod q$. Then, $S \bmod (p-1)$ is either $S \bmod q$ or $S \bmod q + p$, and we can determine which by verifying which gives T when used as the exponent of g.
If two different signature X_1 and X_2 use the same S_m, their difference $X_1 - X_2$ can be used to compute $S \bmod (p-1)$. Specifically, $(d_1 - d_2)^{-1}$

mod q and $(d_1 - d_2)^{-1}(X_1 - X_2) \bmod q$ can be calculated, which gives $S \bmod q$. Finally, $S \bmod (p-1)$ can be obtained by checking whether $S \bmod q$ or $S \bmod q+p$ gives T when used as the exponent of g.

15.1) B/Y/Z/A and A/C.

15.2) If a subtree A or A/B is renamed, relative name cross-certs remain valid but absolute name cross-certs must be reissued. If A/B/C is renamed to something not beginning with A/B, then the relative name cross-cert is invalid but the absolute name cross-cert remains valid. Absolute name cross-certs remain valid if only the issuer's name changes but the target's doesn't, while relative name cross-certs change. If both entities change in the same way, then both absolute and relative name cross-cert must be reissued.

15.4) Downloading Bob's key from an IP address is efficient in terms of computation but insecure without authentication. Looking up Bob's key in a directory via an unauthenticated interaction is also efficient in terms of computation but insecure without authentication. Having an authenticated conversation to the directory is secure and having the directory sign the information you request is more secure but requires more computation. Storing and retrieving certificates from the directory is secure and flexible but requires more bandwidth and computation. Overall, the best scheme depends on the specific use case and the trade-offs between security, efficiency, flexibility, and ease of management.

15.5) A CRL must be reissued periodically to update the date and time of the CRL, as well as to remove any expired entries from the list of revoked certificates.

15.6) Certificates need an expiration date to limit the duration of their validity, in case the revocation mechanism fails or is not used for some reason.

15.8) Keeping hashes of valid certificates ensures that the full certificate is checked, preventing attacks based on fake certificates with valid serial numbers.