CS162 – Section 11

True/False

1. Public key cryptography requires participants to distribute a secret keys

   False.

2. A digital certificate is an encrypted binding between the user's identity and user's public key using a certification authority's (e.g., Verisign) *public key*.

   False.

3. "Delay checking" of the password is an effective way to make it harder to crack a password, assuming the attacker doesn't have access to /etc/passwd

   True.

4. Checking the size of every argument before copying it in the buffer can avoid buffer overflow attacks.

   True.

5. Typically, the number of hosts infected by a worm increases linearly.

   False.

Short Answer

1. What are three common ways of compromising passwords?

   password guessing, dictionary attack, dumpster diving

2. What are four security requirements, explain them:

   Authentication: Ensures that a user is who is claiming to be.
   Data integrity: Ensure that data is not changed from source to destination or after being written on a storage device.
   Confidentiality: Ensures that data is read only by authorized users.
   Non-repudiation:
   a). Sender/client cannot later claim didn't send/write data;
   b). Receiver/server can't claim didn't receive/write data.

3. What do DES, and AES stand for? Are they symmetric key encryption?

   DES: Data Encryption Standard
   AES: Advanced Encryption Standard
   Yes.

4. Does the following mutual authentication work? Why? If not, please provide a working version. Alice's public key Pub_A, private key Pri_A.

   Bob's public key Pub_B, private key Pri_B.
   Alice and Bob know all each other public keys.
   Alice: Send E(E(N_x, Pri_A), Pub_B)
   Bob: Receive msg from Alice. Send back E(E(N_x, Pri_B), Pub_A)
   Alice: Receive msg from Bob. Start to send real message E(E(N_x, Pri_A) + msg, Pub_B) N_x is a random message generated by Alice.

   No. Bob can not be sure that he is talking with Alice.
   Alice: Send E(E(N_x, Pri_A), Pub_B)
   Bob: Receive msg from Alice. Send back E(E(N_x + N_y, Pri_B), Pub_A)
   Alice: Receive msg from Bob. Start to send real message E(E(N_x+N_y, Pri_A) + msg, Pub_B) N_x is a random message generated by Alice.
   N_y is a random message generated by Bob.

Long Answer
For this problem, assume that Alice wants to send a single message M to Bob. To do so, Alice and Bob can potentially use a number of different approaches and cryptographic technologies, which we will describe using the following terminology:

M	Plaintext for a single message
$A \| B$	Concatenation of A with B. Assume the receipient can unambiguously decompose this back into the original values of A and B.
K_A	Alice's public key
K_A^{-1}	Alice's corresponding private key
K_B	Bob's public key
K_B^{-1}	Bob's corresponding private key
E_K	Public-key encryption using RSA with the public key K
$\text{Sign}_{K^{-1}}$	Public-key signing using RSA with the private half of K.
s_k	Symmetric cryptography key
AES_{s_k}	Symmetric-key encryption using AES-256 in CBC mode, with the key s_k
AES-EMAC_{s_k}	Keyed MAC function presented in lecture, using the key s_k
PRNG_{s_k}	Bit-stream from a cryptographically strong pseudo-random number generator, seeded with s_k
IV	An Initialization Vector randomly generated for each use
SHA	SHA-256 hash function

(b) Alice sends to Bob: $E_{K_A}(M \parallel \text{Sign}_{K_A^{-1}}(\text{SHA}(M)))$

> **Solution: Broken**—to decrypt with this scheme, Bob needs to possess Alice's private key.

(c) Alice sends to Bob: $E_{K_B}(M \parallel \text{Sign}_{K_B^{-1}}(\text{SHA}(M)))$

> **Solution: Broken**—this scheme requires Alice to possess Bob's private key for the signing operation.

(d) Alice sends to Bob: $E_{K_A}(M)$, $\text{Sign}_{K_B^{-1}}(\text{SHA}(M))$

> **Solution: Broken**—to decrypt with this scheme, Bob needs to possess Alice's private key. Alice also needs to possess Bob's private key for the signing operation.

(e) Alice sends to Bob: $E_{K_B}(M)$, $\text{Sign}_{K_A^{-1}}(\text{SHA}(M))$

> **Solution:** Provides all of *Confidentiality* (via the encryption using Bob's public key), *Integrity* (via the digital signature over the hash of the message), *Authentication* (likewise) and *Non-Repudiation* (via Alice using her private key for the digital signature).
>
> It's valid to note that Eve can exploit this structure to conduct a *confirmation* attack, because the using of signing allows Eve to determine whether M had a given value. That means the approach would no longer have full *Confidentiality*.

(f) Alice generates a new symmetric key s_k and sends to Bob:
$E_{K_A}(s_k)$, $E_{K_B}(s_k)$, $\text{AES}_{s_k}(M)$

Solution: Only provides *Confidentiality*. While Bob cannot recover s_k from $E_{K_A}(s_k)$ (because Bob lacks Alice's private key), he can do so from $E_{K_B}(s_k)$. By itself, AES does not provide integrity or authentication, so this scheme only provides *Confidentiality*, and because Alice does not sign her message, it also lacks non-repudiation.