Problem 1:

(1) Bob's public-private key pair is (e_b, d_b).

So, $e_b.db \equiv (1 \bmod \phi(n))$…….(1).

From equation 1,

$e_b.d_b = t.\ \phi(n) + 1$ [here t is a positive integer]
$\Rightarrow t.\ \phi(n) = e_b.d_b - 1$
so, $V = t.\ \phi(n) = e_b.d_b - 1$

So, Bob can use his public-private key pair **(eb, db)** to obtain a multiple of **ϕ(n)**. Let us denote that integer by **V**.

Problem 2:

(1) Brute-Force Cryptanalytic Attack: In cryptography, a **brute-force attack** consists of an attacker submitting many passwords or passphrases with the hope of eventually guessing a combination correctly. The conventional encryption algorithms use key lengths ranging from 40 to 168 bits. So the attackers have to use keys 2^{40} to 2^{60} which is not computationally feasible and exhaustive search of the key space for a conventional encryption algorithm.

(2) Known Plaintext Dictionary Attack: This attack is defeated the same way the earlier attack is addressed. Since there are so many different sets of keys available, the size of the dictionary required would be too large to be created. For small key sizes, it could be possible to hack into after a certain amount of time, resources and money. However, for larger key sizes, specifically 128-bit, it could take a very long time.

(3) Replay Attack: Use of TLS handshake process. Both client and server have asymmetric keys of their own. Server public key is used to verify client's identity.

(4) Man-in-the-Middle Attack: TLS prevents the mam in the middle attack by using Certificate Authority(CA). A Certificate Authority is an entity that issues digital certificates. Digital certificates certify the public key of the owner of the certificate (known as the subject), and that the owner controls the domain being secured by the certificate. A CA therefore acts as a trusted third party that gives clients (known as relying parties) assurance they are connecting to a server operated by a validated entity.

(5) Password Sniffing: TLS uses HMAC for MAC. Every message is transmitted with MAC, so no message can't be altered.

(6) IP Spoofing: If the server requests client authentication, the SSL protocol requires that the client create a digital signature by creating a one-way hash from randomly generated

data during the handshake and known only to the client and server. The hash data is encrypted with the client's private key that corresponds to the public key in the certificate received by the server.

(7) IP Hijacking: When establishing the SSL/TLS connection, the two parties agree on a master secret, which is used to produce shared encryption keys and MAC secrets. While the purpose of encryption is to ensure the confidentiality of the conversation, message integrity is protected by a Message Authentication Code (MAC). The attacker wouldn't have the right MAC key to impersonate the legitimate host.

(8) SYN flooding: SYN flooding is also handled by TLS in that the source of the message has to be authenticated before a response is generated. The messages that are continuously sent, can be removed if the source of the requests are considered invalid.