**Part (i):**

Alice wants to send message M1, to make sure that her message is confidential when sent to Bob, Alice should encrypt it with Bob’s public key. That way, only Bob who has the private key can decrypt it.

So Alice encrypts the message using his Public key: EPu(B)(M) this message can now be reffered to as M1.

Now Bob wants to decipher the message M1, so he is going to use his private key as it is the only key able to decrypt something encrypted with his public key.

DPr(B)(EPu(B)(M)) = The original message M.

**Part (ii):**

If Alice wants to send the message M2 without the need for it to be confidential, but still maintain the authenticity, she should use a digital signature. This means she is going to hash the message and then encrypt the hash with her private key. Computational efficienct can be a concert here when large messages (data) is transferred.

She computes the hash of the message: H(M2)

She encrpyts it with her private key: EPr(A)(H(M2))

She sends both the message and the digital signature: M2, EPr(A)(H(M2))

Now Bob needs to verify the message did in fact come from Alice and hasn’t been altered on the way.

He decrypts the received hash using Alice’s public key to get the hash of the original message: DPu(A)(EPr(A)(H(M2)))

He computes the hash of the received message: M2: H(M2)

He compares the hash he computes with the decrypted hash from Alice. If they match then it confirms that the message is indeed from Alice and ensures its integrity.