(i).

Alice should encrypt the message M1 using bobs public key Pu(B) because only bobs private key Pr(B) should be able to decrypt it.

Public key cryptography (like RSA) is computationally expensive and much slower because it relies on complex math’s operations that binary computers struggle to solve. Symmetric key cryptography (like AES) is much faster as it uses simpler operations.

If you’re dealing with larger messages, it would be better if you could use less computational resources and encrypt it faster but results in a less secure encryption. You may take a hybrid approach alternatively to increase security further.

(ii).

Alice encrypts the message with her private key, Bob uses Alice’s public key to decrypt it, if the message is decrypts, then Bob knows Alice sent it. This message can also be decrypted by everyone else too but they will also know Alice is the source of the message insuring authenticity.

If you want to prove authenticity but feel encrypting the whole message is unnecessary you can instead create a unique fingerprint (hash) encrypt the fingerprint and attach the unencrypted message with the encrypted fingerprint. You can read the message and check its authenticity then by decrypting the hash with Alice’s public key, if it decrypts correctly, you’ll know it’s from her. This is faster and in this case just as secure.