Vanilla RSA and Dependent RSA are different only in Encryption and Decryption.

In Vanilla RSA,

Encryption

C = me (mod n) -- (1)

Decryption

M = Cd (mod n) --(2)

For Dependent RSA, we substitute m with ‘(K + 1)’ when compared to equation (1), as we can see below:-

C1 = (K + 1)e (mod n) --(from equation (1))

Because ‘(K + 1)’ replaces m in Dependent RSA, while decryption the value of ‘K’ will be:-

(K + 1) = C1d (mod n) --(from equation (2))

=> K = {C1d (mod n)} – 1

Accordingly, ‘C2’ is the cipher text, which integrates ‘K’ in the encipherment. Due to this, the value of ‘m’ (the decrypted plain text) can be calculated only after we solve for ‘K’:-

C2 = m[Ke (mod n)] ---(encryption)

=> m = [Ke (mod n)]/C2 ---(decryption)

Dependent RSA works as its totally similar to RSA, we just introduce a constant ‘K’, which further decreases the chance of a successful break in the encryption and decryption process. It adds one more sub problem in the encryption and decryption. One will need to compute ‘K’ to formulate the cipher text to plain text. Additionally, ‘K’ is also hidden in ‘C1’, making it all the more difficult. One will need ‘C1’ and ‘C2’ to get to the final deciphered plain text.