

Implement the Paillier encryption scheme in Python and encrypt two integers (e.g., 15 and 25). Print the ciphertexts, perform an addition operation on the encrypted integers without decrypting them, and print the encrypted result. Decrypt the result of the addition and verify it matches the sum of the original integers. Additionally, implement scalar multiplication on encrypted data, where an encrypted integer is multiplied by a scalar (e.g., 15×3) without decrypting it, and verify the result by decryption. Extend the scheme to homomorphically add multiple encrypted integers (e.g., 10, 20, 30) and decrypt the result to verify the sum. Integrate a searchable encryption scheme where a small set of key-value pairs (e.g., {"ID_1": 15, "ID_2": 25, "ID_3": 40}) can be searched for by keyword (e.g., "ID_2") without decrypting the entire set, and verify the encrypted value returned by the search. Implement batch decryption to allow multiple ciphertexts to be decrypted together, and finally, compare the encryption, homomorphic operation, and decryption times for small (e.g., 15) and larger integers (e.g., 1000), analyzing the performance of the Paillier scheme.