alpine:~/Projets/Homo²rphOS/SEAL/native/examples/build# ./bin/sealexamples Microsoft SEAL version: 4.1.2 +---+ | The following examples should be executed while reading | | comments in associated files in native/examples/. | +---+ | Examples | Source Files | +---------------------------+---------------------------+ | 1. BFV Basics | 1_bfv_basics.cpp | | 2. Encoders | 2_encoders.cpp | | 3. Levels | 3_levels.cpp | | 4. BGV Basics | 4_bgv_basics.cpp | | 5. CKKS Basics | 5_ckks_basics.cpp | | 6. Rotation | 6_rotation.cpp | | 7. Serialization | 7_serialization.cpp | | 8. Performance Test | 8_performance.cpp | +---------------------------+---------------------------+ [0 MB] Total allocation from the memory pool > Run example (1 ~ 8) or exit (0): 1 +----------------------------------+ | Example: BFV Basics | +----------------------------------+ Line 132 --> Set encryption parameters and print / | Encryption parameters : | scheme: BFV | poly_modulus_degree: 4096 | coeff_modulus size: 109 (36 + 36 + 37) bits | plain_modulus: 1024 \ Parameter validation (success): valid ~~~~~~ A naive way to calculate 4(x^2+1)(x+1)^2. ~~~~~~ Line 212 --> Express x = 6 as a plaintext polynomial 0x6. Line 223 --> Encrypt x_plain to x_encrypted. + size of freshly encrypted x: 2 + noise budget in freshly encrypted x: 55 bits + decryption of x_encrypted: 0x6 ..... Correct. Line 270 --> Compute x_sq_plus_one (x^2+1). + size of x_sq_plus_one: 3 + noise budget in x_sq_plus_one: 33 bits + decryption of x_sq_plus_one: 0x25 ..... Correct. Line 300 --> Compute x_plus_one_sq ((x+1)^2). + size of x_plus_one_sq: 3 + noise budget in x_plus_one_sq: 33 bits + decryption of x_plus_one_sq: 0x31 ..... Correct. Line 315 --> Compute encrypted_result (4(x^2+1)(x+1)^2). + size of encrypted_result: 5 + noise budget in encrypted_result: 4 bits NOTE: Decryption can be incorrect if noise budget is zero. ~~~~~~ A better way to calculate 4(x^2+1)(x+1)^2. ~~~~~~ Line 353 --> Generate relinearization keys. Line 361 --> Compute and relinearize x_squared (x^2), then compute x_sq_plus_one (x^2+1) + size of x_squared: 3 + size of x_squared (after relinearization): 2 + noise budget in x_sq_plus_one: 33 bits + decryption of x_sq_plus_one: 0x25 ..... Correct. Line 376 --> Compute x_plus_one (x+1), then compute and relinearize x_plus_one_sq ((x+1)^2). + size of x_plus_one_sq: 3 + noise budget in x_plus_one_sq: 33 bits + decryption of x_plus_one_sq: 0x31 ..... Correct. Line 390 --> Compute and relinearize encrypted_result (4(x^2+1)(x+1)^2). + size of encrypted_result: 3 + size of encrypted_result

(after relinearization): 2 + noise budget in encrypted_result: 11 bits NOTE: Notice the increase in remaining noise budget. Line 407 --> Decrypt encrypted_result (4(x^2+1)(x+1)^2). + decryption of 4(x^2+1)(x+1)^2 = 0x54 ..... Correct. Line 425 --> An example of invalid parameters / | Encryption parameters : | scheme: BFV | poly_modulus_degree: 2048 | coeff_modulus size: 109 (36 + 36 + 37) bits | plain_modulus: 1024 \ Parameter validation (failed): parameters are not compliant with HomomorphicEncryption.org security standard +----------
----------+ | The following examples should be executed while reading | | comments in associated files in native/examples/. | +----------+ | Examples | Source Files | +----------+----------+
| 1. BFV Basics | 1_bfv_basics.cpp | | 2. Encoders | 2_encoders.cpp | | 3. Levels | 3_levels.cpp | |
4. BGV Basics | 4_bgv_basics.cpp | | 5. CKKS Basics | 5_ckks_basics.cpp | | 6. Rotation | 6_rotation.cpp |
| 7. Serialization | 7_serialization.cpp | | 8. Performance Test | 8_performance.cpp | +----------+----------+
[8 MB] Total allocation from the memory pool > Run example (1 ~ 8) or exit (0): 2 +----------+ | Example: Encoders | +----------+ +---------
----------+ | Example: Encoders / Batch Encoder | +----------+ / | Encryption parameters : | scheme: BFV | poly_modulus_degree: 8192 | coeff_modulus size: 218 (43 + 43 + 44 + 44 + 44) bits | plain_modulus: 1032193 \ Batching enabled: true Plaintext matrix row size: 4096 Input plaintext matrix: [0, 1, 2, 3, 0, ..., 0, 0, 0, 0, 0] [4, 5, 6, 7, 0, ..., 0, 0, 0, 0, 0] Line 124 --> Encode plaintext matrix: + Decode plaintext matrix ..... Correct. [0, 1, 2, 3, 0, ..., 0, 0, 0, 0, 0] [4, 5, 6, 7, 0, ..., 0, 0, 0, 0, 0] Line 141 --> Encrypt plain_matrix to encrypted_matrix. + Noise budget in encrypted_matrix: 146 bits Second input plaintext matrix: [1, 2, 1, 2, 1, ..., 2, 1, 2, 1, 2] [1, 2, 1, 2, 1, ..., 2, 1, 2, 1, 2] Line 172 --> Sum, square, and relinearize. + Noise budget in result: 114 bits Line 187 --> Decrypt and decode result. + Result plaintext matrix ..... Correct. [1, 9, 9, 25, 1, ..., 4, 1, 4, 1, 4] [25, 49, 49, 81, 1, ..., 4, 1, 4, 1, 4] +----------+ | Example: Encoders / CKKS Encoder | +----------+ / | Encryption parameters : | scheme: CKKS |

poly_modulus_degree: 8192 | coeff_modulus size: 200 (40 + 40 + 40 + 40 + 40) bits \ Number of slots: 4096
Input vector: [0.000, 1.100, 2.200, 3.300]
Line 297 --> Encode input vector. + Decode input vector ..... Correct.
[-0.000, 1.100, 2.200, 3.300, ..., 0.000, -0.000, 0.000, -0.000]
Line 313 --> Encrypt input vector, square, and relinearize. + Scale in squared input: 1.15292e+18 (60 bits) Line 333 --> Decrypt and decode.
+ Result vector ..... Correct. [0.000, 1.210, 4.840, 10.890, ..., -0.000, -0.000, -0.000, 0.000] +----------
| The following examples should be executed while reading |
| comments in associated files in native/examples/. |
+----------
| Examples | Source Files |
+----------+----------+
1. BFV Basics	1_bfv_basics.cpp
2. Encoders	2_encoders.cpp
3. Levels	3_levels.cpp
4. BGV Basics	4_bgv_basics.cpp
5. CKKS Basics	5_ckks_basics.cpp
6. Rotation	6_rotation.cpp
7. Serialization	7_serialization.cpp
8. Performance Test	8_performance.cpp
+----------+----------+
[31 MB] Total allocation from the memory pool
> Run example (1 ~ 8) or exit (0): 0
alpine:~/Projets/Homo^{2}rphOS/SEAL/native/examples/build# pwd