Microsoft SEAL

GitHub.com/Microsoft/SEAL

Microsoft SEAL — powered by open-source homomorphic encryption technology — provides a set of encryption libraries that allow computations to be performed directly on encrypted data. This enables software engineers to build end-to-end encrypted data storage and computation services where the customer never needs to share their key with the service.

For Developers

Supported platforms:

Written in C++17

Full .NET Standard wrappers

No required dependencies

State-of-the-art HE schemes:

BFV[1], CKKS[2]

✓ Security standard[3]

Bootstrapping

CHET compiler[4]

For Enterprises

Distributed

Hardware Acceleration

Embedded

Code Example

```
[ 1] EncryptionParameters parms(scheme_type::CKKS);
[ 2] parms.set_poly_modulus_degree(8192);
[ 3] parms.set_coeff_modulus(
        CoeffModulus::Create(8192, {60, 40, 40, 60}));
[ 4] auto context = SEALContext::Create(parms);
[ 5] KeyGenerator keygen(context);
[ 6] Encryptor encryptor(context, keygen.public_key());
[ 7] encryptor.encrypt(ptxt1, ctxt1);

[ 8] Evaluator evaluator(context);
[ 9] RelinKeys relin_keys = keygen.relin_keys();
[10] GaloisKeys galois_keys = keygen.galois_keys();
[11] evaluator.multiply_inplace(ctxt1, ctxt2);
[12] evaluator.relinearize_inplace(ctxt1, relin_keys);
[13] evaluator.rescale_to_next_inplace(ctxt1);
[14] evaluator.add(ctxt1, ctxt3, ctxt_res);
[15] evaluator.rotate_vector_inplace(
         ctxt_res, 2, galois_keys);

[16] Decryptor decryptor(context, keygen.secret_key());
[17] decryptor.decrypt(ctxt_res, plain_res);
```

References

1. J. Fan, F. Vercauteren: Somewhat Practical Fully Homomorphic Encryption. IACR Cryptology ePrint Archive 2012: 144.
2. J. H. Cheon, A. Kim, M. Kim, Y. S. Song: Homomorphic Encryption for Arithmetic of Approximate Numbers. ASIACRYPT 2017.
3. HomomorphicEncryption.org
4. R. Dathathri et al.: CHET: an optimizing compiler for fully-homomorphic neural-network inferencing. PLDI 2019.