# **Homomorphic Encryption Method**

**Author:** Bill Buchanan, Nick Pitropakis

Homomorphic encryption can either be partial homomorphic encryption (PHE) or fully homomorphic encryption (FHE). With FHE we have a full range of mathematical operations. With this we convert our input data values into integers or floating-point values. In some homomorphic methods, we only deal with integers, while others (such as CKKS) can operate on real numbers. For each method, we generate a unique key pair for the homomorphic encryption. This is different from a key pair used in SSI (and which uses a traditional public key pair). We then encrypt with the public key, and can then process using a mathematical operation, and without revealing the original data. The private key is then used to decrypt the data, and reveal the result of the mathematical operation (Figure 1).

Diagram

Description automatically generated

**Figure 1:** Outline of homomorphic encryption

One method that can be used in TRUSTEE is to define a data processing control. If we have data providers of Bob, Alice and Peggy, and a data consumer of Wendy, then the contract can bind the provision of the data to the consumer. For this, we can define a contract which will define the usage of the data, the data schema, and define the homomorphic circuit to use. We then end up with an instance of the contract, and which will create a unique homomorphic key pair for the instance of the contract. The private key for this, can then be stored in a secure enclave, and where the private key is never exposed anywhere outside the enclave. Each identity involved in the data processing infrastructure also have their own public key pair for SSI, and where data can be signed-for with their private key, and then checked against the public key on the blockchain (Figure 2). Figure 3 then shows a possible architecture, and where a secure enclave generates the homormophic key pair, and stores the private key. This private key is never used outside the secure enclave, and only a key identifier is used to define the key to be used. Along with this, TRUSTEE can sign the data result within the enclave, so that the result and the signature can only be used within the enclave.

Graphical user interface

Description automatically generated

**Figure 2:** Defining a data binding contract

Diagram

Description automatically generated

**Figure 3:** Outline of the process