Homomorphic Encryption (HE)

Return to Glossary

Homomorphic Encryption (HE) is a method of encryption that allows any data to remain encrypted while it's being processed and manipulated. It enables you or a third party (such as a cloud provider) to apply functions on encrypted data without needing to reveal the values of the data. A homomorphic cryptosystem is like other forms of public encryption in that it uses a public key to encrypt data and allows only the individual with the matching private key to access its unencrypted data (though there are also examples of symmetric key **Homomorphic Encryption** as well). However, what sets it apart from other forms of encryption is that it uses an algebraic system to allow you or others to perform a variety of computations (or operations) on the encrypted data.

In practice, most **Homomorphic Encryption** schemes work best with data represented as integers and while using addition and multiplication as the operational functions. This means that the encrypted data can be manipulated and analyzed as though it's in plaintext format without actually being decrypted. In other words, HE can enable your employees (or a third party) to work with and use the encrypted data without having access to or knowing the contents of the decrypted data. They can compute and process the encrypted data to get an encrypted answer, but only you can decrypt the ciphertext and understand what it means. **Homomorphic Encryption** requires few rounds of interactions and uses arithmetic circuits (which focus on additions and multiplication, allowing you to add and multiply numbers) rather than Boolean circuits like other methods of secure computation (such as two-party computation [2PC] or general Multi-Party Computation (MPC)).

Source: https://www.thesslstore.com/blog/what-is-homomorphic-encryption/