1. Time complexity of homomorphic encryption Vs. AES

**(SwHE: somewhat homomorphic encryption and AES 256bit standard.)**

**Time Complexity**

* **Encryption Time**
* Somewhat homomorphic encryption
* Compare with Fully Homomorphic Encryption (FHE), with the SwHE faster than FHE because of simpler operations and less complex algorithms. FHE significantly slower than SHE because of bootstrapping process for arbitrary computations. However, encryption time depends on data size, chosen scheme and operation type and desired level of security.

Typically, encryption time for smaller data in SwHE algorithm is from milliseconds to seconds of data range. But FHE range in several seconds to minutes for small data. Both FHE and SWHE encryption time increasing with larger datasets.

* 256-bit AES
* Compare with AES bit 128 & AES 192 bit, AES 256 bit get slowest encryption time. But AES 256 bits offers highest of security against brute-force attacks, currently considered unbreakable for practical purposes. AES 128 bit occurs fasted encryption time with less security.

So, when we are comparing SwHE and 256-bit AES, SwHE encryption time significantly slower than AES-256 bits. As the Encryption time comparison,

* AES-256: Significantly faster than any SHE schemes due to its simpler design and focus on efficiency.
* SHE: Varies depending on the specific scheme and operation type. Generally, it's at least 1000 times slower than AES-256, even for simple operations like addition or multiplication.
* **Decryption Time**

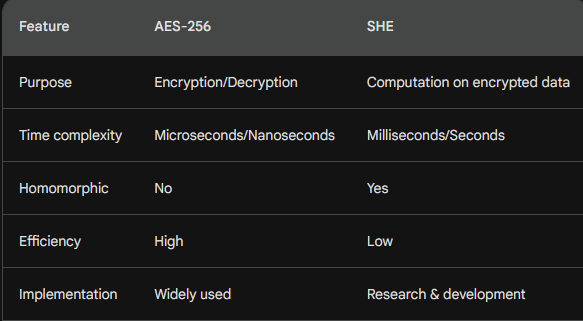
When we consider the decryption, there are two types of decryptions. One is time takes to decrypt the result of computations performed on encrypted data and other one is time takes to recover the original data from encrypted data without performing any computations.

Let’s consider the time takes to recover the original data from encrypted data without any computation performed.

* Somewhat homomorphic encryption
* When we consider SwHE with FHE two take essentially same time for the encryption. Both SHE and FHE recovering the original data requires decrypting the cyphertext using correct secret key. That doesn’t involve any computations on encrypted data.

Also, decryption time depending on Data size, Hardware and software implementation and key size.

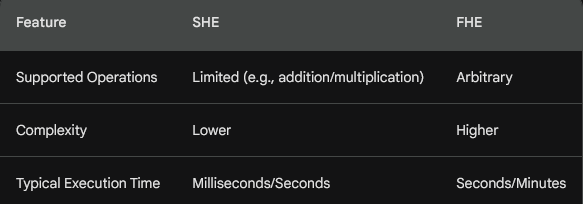
* 256-bit AES
* Comparing with AES-128 bits and AES-192 bits, same as the encryption AES-256 has slowest decryption time but with highest level of security.
* Typical Decryption Times:
  + - * **AES-128:** Microseconds to nanoseconds on modern hardware.
      * **AES-192:** Slightly slower than AES-128, typically in the microsecond range.
      * **AES-256:** Can be several times slower than AES-128, depending on implementation, ranging from microseconds to tens of microseconds.

When comparing SwHE vs 256-bit AES decryption time, it’s pretty like encryption time. AES-256 bits extremely fast and with modern hardware it’s measured in microseconds or nanoseconds. But, in the SHE it has slow decryption time. It measured in milliseconds or even seconds. So, compare with 256-bit AES SwHE significantly slower than AES.

* **Execution time**
* Somewhat homomorphic encryption

Comparing SwHE with FHE execution time it’s affect some factors for execution time. There are algorithm and optimization, Operation type, Data size and Hardware as well.

SwHE has limited operations with less complex mathematical structures and operations compare to the FHE. So, it leading SwHE to faster execution time. SwHE has faster execution time compare with FHE.



* 256-bit AES

When comparing AES-256 with AES-128 and AES-192, the key length doesn’t directly affect the speed of the algorithm. Key length primarily affects the security of the encryption. Not for the execution time.

So, considering that all three key lengths get extremely fast execution times, typically in measured in microseconds or even nano seconds in modern processors.

Let’s consider execution time with SwHE and 256-bits AES. SwHE execution time typically in a millisecond to second range. Even very simple operations like addition or multifaction. Also, it can significantly be slower for complex computation or some large datasets. But AES-256 offers extremely fast execution time. Also, it in microsecond or nanosecond range with modern hardware as well.

Finally, AES-256 significantly faster than any SwHE operation.

1. Space complexity of homomorphic encryption Vs. AES

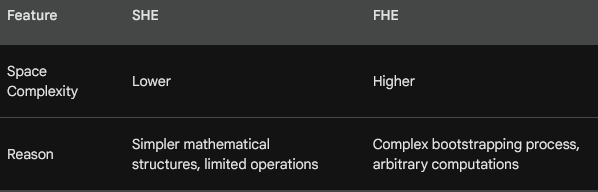
**Space Complexity**

* Somewhat homomorphic encryption

When considering space complexity, there are some factors affecting space complexity. Scheme design, operation type and security level are the main factors of that. SwHE support only limited operations like addition and multifaction. So, it can utilize simpler mathematical structures.

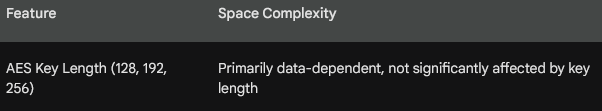
FHE needs to perform a complex process called bootstrapping. This process involves refreshing the ciphertext after a certain number of operations to ensure correctness.

So, SwHE generally has lower space complexity compared to FHE. Usually, FHE has higher space complexity due to its ability to support arbitrary computations.

****

* 256-bit AES

Space complexity of AES, regardless of key length. It is primarily independent of key length. The space complexity of AES is primarily determined b the data size being encrypted or decrypted. Overall space complexity of AES practically the same size for all three key lengths in most scenarios.



When considering the space complexity between SwHE and AES-256, SwHE utilize complex mathematical structures to support computations on encrypted data. So, SwHE require more space to represent encrypted data and intermediate results compared to similar structures used in AES-256.

While the disparity in key size (AES utilizing 256 bits compared to a variable size in SHE) may be a consideration, its impact is often overshadowed by the intricacies of the underlying mechanisms in SHE.

So, in generally SwHE has higher space complexity compared to AES-256. AES-256 has relatively low space complexity.

1. Efficiency of regular implementation of homomorphic encryption Vs. AES

**Efficiency of regular implementation**

* Somewhat homomorphic encryption

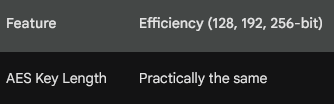
Compare SwHE with FHE, there are more factors affecting efficiency. However, SwHE generally considered more efficient than FHE. Because of faster computation speeds for supported operations, require less memory to store encrypted data and perform operations and easier to implement due to simpler mathematical structure. On the other hand FHE typically less efficient than SwHE.

****

* 256-bit AES

The efficiency of regular implementations of AES is practically the same for all three key lengths (128, 192 and 256). AES algorithm itself is designed for efficiency and offers extremely fast execution times for both encryption and decryption, regardless of the key length.

Therefore, we can expect the efficiency of AES to be essentially the same for 128-bit, 192-bit, and 256-bit key lengths in most real-world applications.



Let’s consider the Efficiency of regular implementation of homomorphic encryption Vs. AES-256. Here, 256 bits AES offers higher efficiency than SwHE. AES performs encryption and decryption in microseconds or nanoseconds, while SHE operations can take milliseconds or even seconds. Not only that but also, when considering space complexity AES requires less memory to store encrypted data and perform operations compared to SwHE.

Simply, 256 bits AES offer higher efficiency when compare with SwHE.

1. Making the environment for the integrated files

**Making the environment for the integrated files**

256-bit AES and Somewhat Homomorphic Encryption (SHE) environment for integrated files depends heavily on the specific purpose and requirements.

|  |  |  |
| --- | --- | --- |
|  | **256-bit AES** | **SwHE** |
| Functionality | Primarily focuses on securely storing and transmitting data in an encrypted format. It cannot perform computations on the encrypted data itself. | Allows to perform limited computations on encrypted data without decrypting it. This is advantageous for tasks like searching encrypted data or performing specific calculations while maintaining privacy. |
| Security | offer strong security guarantees for their respective purposes. AES is a well-established and widely trusted encryption standard. | offer strong security guarantees for their respective purposes. SwHE utilizes advanced cryptographic techniques to ensure the confidentiality and integrity of data during computations. |
| Performance | Offers significantly faster encryption, decryption, and data access compared to SHE. This is crucial for real-time applications or environments handling large amounts of data. | Involves complex mathematical operations, leading to slower performance for both encryption/decryption and computations on encrypted data. The execution time can vary based on the specific SHE schemes, type of operation, and data size. |
| Resource Requirements | Generally, requires fewer computational resources compared to SHE. This makes it suitable for resource-constrained environments or devices with limited processing power. | Due to the complex computations involved, SHE implementations can have higher resource requirements in terms of memory and processing power. |

1. Usable tools of homomorphic encryption Vs. AES

**Usable tools**

* Somewhat homomorphic encryption

While SHE is an evolving field, there are still a limited number of user-friendly tools available compared to AES.

1. **HElib**

****

* An open-source C++ library for implementing SHE operations.
* Supports various SHE schemes and offers a rich set of functionalities.
* Requires programming expertise in C++ for use.

1. **SEAL (Microsoft SEAL)**

A logo with a white and blue background

Description automatically generated with medium confidence

* An open-source library for homomorphic encryption from Microsoft.
* Supports several SHE schemes and offers a user-friendly C++ API.
* Requires programming expertise in C++ for use.

1. **homomorphic.js**

* A JavaScript library for implementing SwHE operations in web browsers.
* Enables basic homomorphic operations on encrypted data directly in web browsers.
* Offers limited functionalities compared to HElib and SEAL but provides a more accessible option for web-based applications.
* 256-bit AES

There are numerous tools available for implementing 256-bit AES encryption and decryption.

1. **OpenSSL**

**A black and red logo

Description automatically generated**

* A free and open-source software library that provides various cryptographic functions, including AES encryption and decryption.
* Available on various platforms, including Windows, macOS, and Linux.
* Offers a command-line interface (CLI) and APIs for integration into applications.

1. **A blue text on a black background

   Description automatically generatedGnuPG (Gnu Privacy Guard)**

* A free and open-source tool for encrypting and signing data and communications.
* Supports AES encryption, including 256-bit key length.
* Available on various platforms, including Windows, macOS, and Linux.
* Offers a CLI and graphical user interface (GUI) for ease of use.

1. **Microsoft BitLocker (Windows)**

**A close-up of a lock and keys

Description automatically generated**

* A built-in feature on Windows devices that provides full-disk encryption using AES.
* Transparent to users, automatically encrypting and decrypting the entire disk.
* Offers strong security and easy manageability for Windows systems.

1. **A silver house with a dial

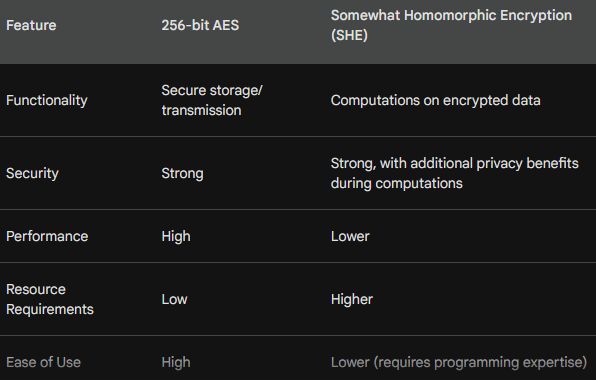
   Description automatically generatedApple FileVault (macOS)**

* A built-in feature on macOS devices that encrypts the entire disk volume.
* Uses AES-XTS with 256-bit keys for strong encryption.
* Transparent to users, automatically encrypting and decrypting the disk.
* Provides robust security for macOS systems.

1. Benefits of homomorphic encryption Vs. AES

**Benefits**

|  |  |
| --- | --- |
| SwHE | 256-bits AES |
| **Computation on Encrypted Data**  Allows performing specific calculations on encrypted data without decrypting it, preserving data privacy during computations. | **High Speed and Efficiency**  Offers extremely fast encryption and decryption times, making it suitable for real-time applications and large data volumes. |
| **Enhanced Security**  Offers an additional layer of security by keeping data encrypted even during computations, potentially mitigating risks associated with data breaches. | **Widely Adopted and Supported**  A well-established and widely used standard, making it compatible with various platforms, tools, and libraries. |
| **Privacy-Preserving Analytics**    Enables secure analysis of sensitive data without compromising confidentiality, facilitating valuable insights while protecting privacy | **Strong Security**  Provides robust protection against unauthorized access to data, with 256-bit key length offering exceptional security against brute-force attacks. |
| **Futureproofing**  Holds potential for enabling more complex computations on encrypted data in the future, offering possibilities for advanced privacy-preserving applications. | **Low Resource Requirements**  Requires minimal computational resources, making it ideal for resource-constrained environments or devices with limited processing power. |
|  | **Ease of Use**  Numerous user-friendly tools and libraries are available for implementing AES, making it accessible to users with varying technical expertise. |

****