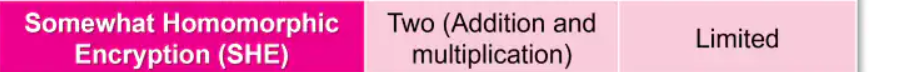
1. **How strong the algorithms are to the attacks in homomorphic encryption Vs. AES**

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

* Somewhat homomorphic encryption
* Compared to completely homomorphic encryption (FHE), which permits unrestricted computations**, SWHE only enables a restricted set of operations.** This is why the term "somewhat" is used.



* The security of certain mathematical issues, including lattice-based problems, determines the security of homomorphic encryption, including SWHE. These systems' security is typically predicated on how hard it is to solve mathematical puzzles that are thought to be challenging, such the Learning With Errors (LWE) issue.
* The particular encryption **technique employed, the parameters selected, and the security presumptions made all affect how strong SWHE is against assaults**.

Brute Force Attacks: Making a systematic attempt to use every key in order to decrypt the data. Brute force assaults are computationally impractical due to the complexity of the mathematical issues that underpin the security of SWHE.

Cryptanalysis is the process of examining an encryption scheme's mathematical structure in order to identify flaws or vulnerabilities. Based on the presumption that some mathematical problems are difficult to solve, SWHE is secure.

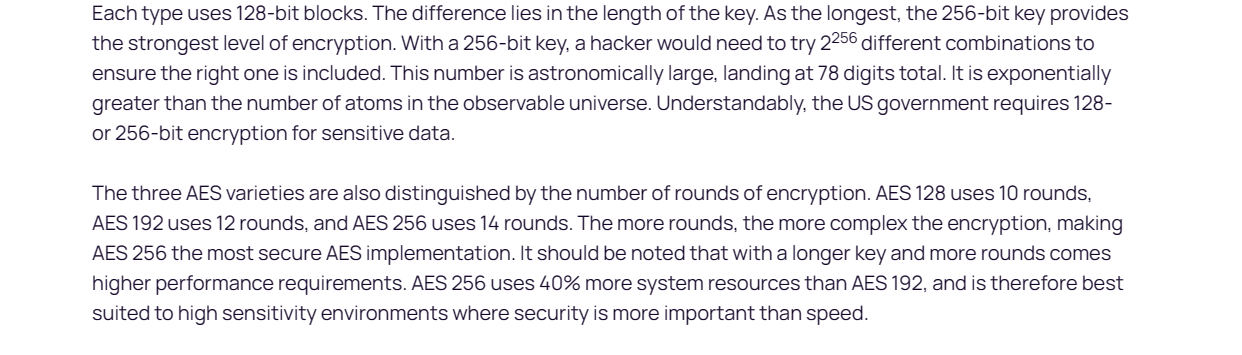
* 256 bit AES

**256-bit keys are thought to be extremely safe and are frequently used to encrypt sensitive data.** The difficulty of specific mathematical operations, such as the difficulty of executing a known plaintext assault or brute force attack, is the foundation for the security of AES.

**strength of AES with a 256-bit key:**

* given the large number of potential keys (2^256) that a 256-bit key size offers, brute force (or exhaustive key search) assaults are not viable given the state of technology today.
* there are no practical cryptanalytic attacks against AES with a 256-bit key.
* AES with a 256-bit key would probably continue to provide good security even in the event of unanticipated advances in cryptography, thanks to the large margin of safety offered by the 256-bit key size.

[Advanced Encryption Standard: Understanding AES 256 - N-able](https://www.n-able.com/blog/aes-256-encryption-algorithm)



AES 256 is virtually impenetrable using brute-force methods. While a 56-bit DES key can be cracked in less than a day, AES would take billions of years to break using current computing technology. Hackers would be foolish to even attempt this type of attack.

1. **Scalability in SWHE Vs. AES**

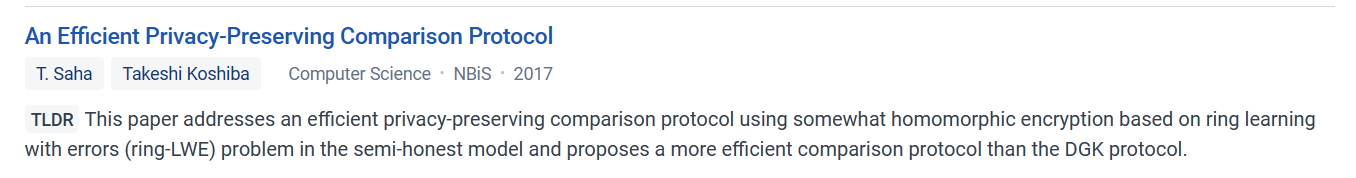
|  |  |  |
| --- | --- | --- |
| **Aspects** | **SWHE** | **AES 256bit** |
| **Memory Usage** | Memory usage may increase, particularly for certain SHE implementations and large datasets. Memory requirements are typically higher compared to AES. | Generally lower memory usage, especially for basic encryption and decryption tasks, providing a more efficient use of memory. |
| **Processing Time** | Processing time may increase, especially for more complex computations or larger data sizes. The increase in processing time is more noticeable than AES for certain operations. | Known for speed and efficiency, with relatively low processing time for standard cryptographic operations. |
| **Parallelization** | SHE algorithms may have limited parallelization capabilities, impacting scalability in parallel processing environments. | AES can benefit from parallelization, allowing efficient use of multiple processing units or cores, enhancing scalability. |
| **Data Size Scalability** | Scalability may be impacted as data sizes increase due to the computational complexity of SHE. | Generally scalable with data size, with efficient processing for various data volumes. AES maintains better scalability for larger datasets. |
| **Computation Depth Scalability** | The impact of increasing computation depth may be significant, affecting the scalability of SHE. | Less affected by computation depth, making it more scalable for a wide range of cryptographic operations. AES typically exhibits more consistent performance across various computation depths. |
| **Resource Utilization** | SHE may utilize resources such as CPU and memory more intensively, affecting overall resource utilization. | Efficient utilization of resources such as CPU, memory, and network bandwidth, contributing to scalability. |
| **Security Trade-offs** | The scalability improvements in SHE may come with trade-offs in terms of security guarantees. | Maintains a balance between security and efficiency without compromising fundamental security guarantees. |
| **Real-world Applications** | Applicable in scenarios prioritizing data privacy where computations on sensitive data are crucial. | Commonly used when a balance between security and performance is required in various real-world applications. |

The limit on somewhat homomorphic encryption comes when a ciphertext generates too much noise in the data. When there’s more noise from a ciphertext, there’s more computational overhead and the somewhat homomorphic encryption scheme functions more slowly.

Another limit of a somewhat homomorphic encryption scheme is multiplicative depth. Multiplicative depth is the maximum number of multiplications a somewhat homomorphic encryption scheme was built to perform. Even with these limits, somewhat homomorphic encryption can still be effectively used in physics applications involving [coherent states](https://journals.aps.org/pra/abstract/10.1103/PhysRevA.97.042308).

1. **Different methods that use to implement somewhat homomorphic encryption Vs AES 256 bit**

[Practical Packing Method in Somewhat Homomorphic Encryption | Semantic Scholar](https://www.semanticscholar.org/paper/Practical-Packing-Method-in-Somewhat-Homomorphic-Yasuda-Shimoyama/d7611ac2d6ff6a7cc6ea9fcb222cca7bdbc3e1cc)

****

**TWO COMMONLY USED METHODS**

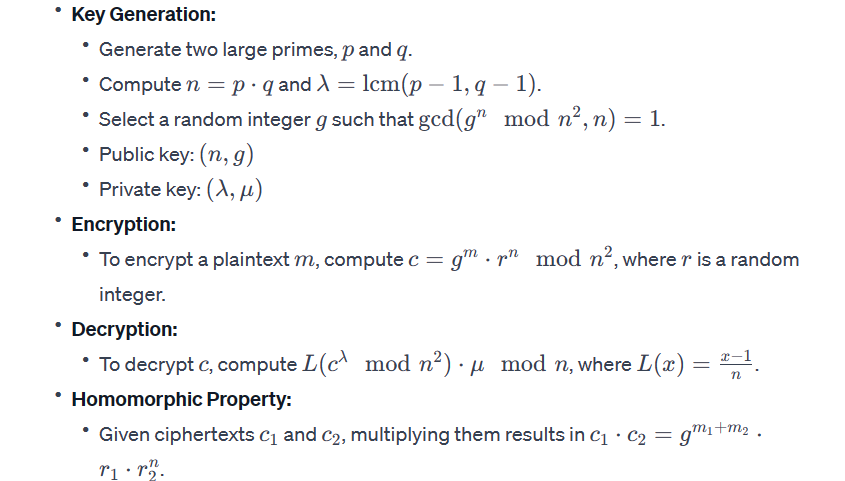
**Paillier Cryptosystem:**

Proposed in 1999 by Paillier (1999), the Paillier cryptosystem is based on the problem that computing *n*th residue classes is computationally intensive. The nature of the algorithm allows for homomorphic addition operations to produce the current answer once decrypted.

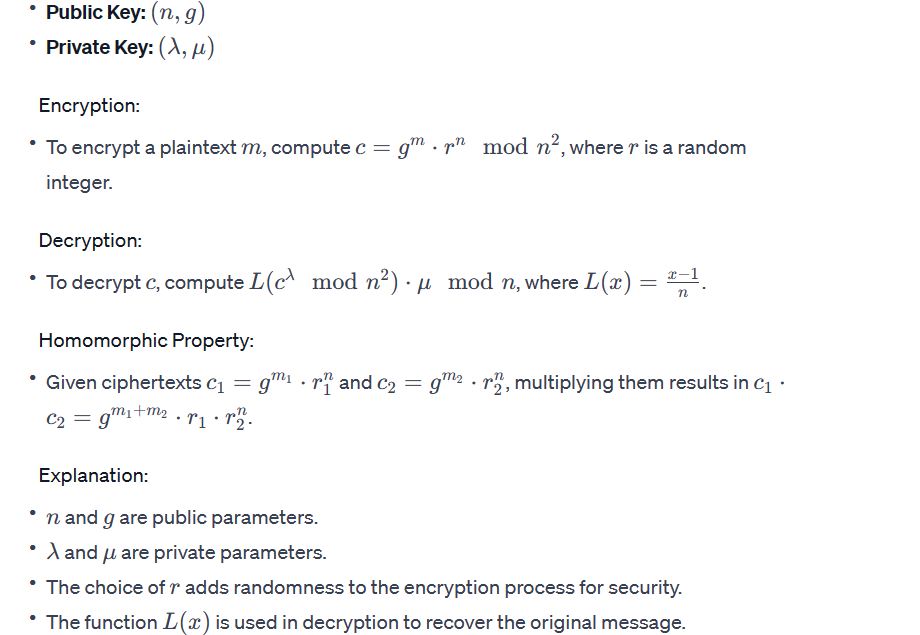
As a public key encryption scheme, Paillier cryptosystem has three stages:

* Generate public-private key pair
* Encryption with public key
* Decryption with private key

[Paillier Cryptosystem - an overview | ScienceDirect Topics](https://www.sciencedirect.com/topics/computer-science/paillier-cryptosystem)

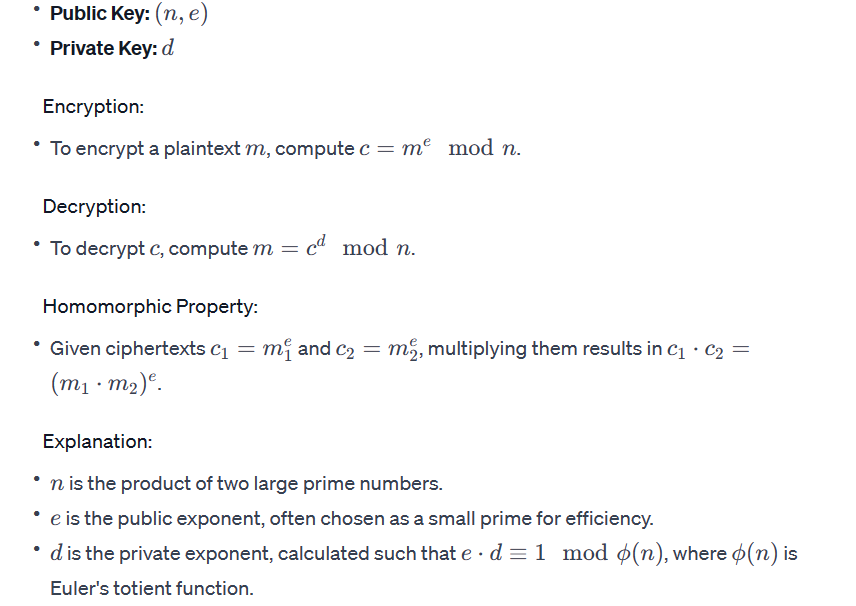


OR ELSE



Paillier is often used in scenarios where privacy-preserving additive computations are required, such as in secure voting systems or privacy-preserving data aggregation.

**RSA Cryptosystem:**

****

**In addition**

**Some somewhat homomorphic encryption schemes are based on lattice problems, such as Learning With Errors (LWE) or Ring-LWE.**

**Protocols..**

**Secure Multi-Party Computation (SMPC) Protocols**

**Protocol for Secure Function Evaluation (PSI)**

**Private Set Intersection (PSI) Protocols**

**Oblivious Transfer-based PSI**

**Homomorphic Encryption in Cloud Computing Protocols**

**Privacy-Preserving Machine Learning Protocols**

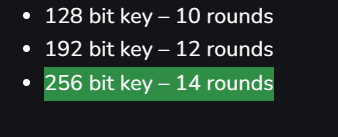
**Secure Aggregation Protocols**

**Cryptographic Accumulators**

**Functional Encryption Protocols**

**AES 256 bit**

applied to AES or symmetric encryption in general, it's important to note that symmetric encryption, like AES, is typically used for direct encryption and decryption, not for performing computations on encrypted data.

****

[Advanced Encryption Standard (AES) - GeeksforGeeks](https://www.geeksforgeeks.org/advanced-encryption-standard-aes/)

* encryption key used in AES 256 encryption is 256 bits long.
* The 256-bit key size offers a vast number of possible combinations, making it extremely difficult for an attacker to guess or crack the key.
* AES 256 encryption is a symmetric encryption algorithm, which means that the same key is used for both encryption and decryption.
* The AES 256 encryption process involves multiple rounds of substitution, permutation, and mixing of data. These rounds ensure that the encryption is highly secure and resistant to various cryptographic attacks. The strength of AES 256 lies in the complexity of its encryption key and the number of encryption rounds performed.
* During each round of the AES 256 encryption process, the plaintext data is divided into blocks, typically 128 bits in size. Each block then undergoes a series of transformations, including substitution with values from a pre-defined lookup table, permutation of the bits within the block, and mixing of the bits using bitwise operations.
* These transformations are repeated for multiple rounds, with the number of rounds depending on the key size. In the case of AES 256, there are 14 rounds, each adding an additional layer of security to the encrypted data. The more rounds performed, the harder it becomes for an attacker to reverse-engineer the encryption and recover the original plaintext.

[Ultimate Guide to AES 256 Encryption: Strengthening Data Protection for Unbreakable Security (kiteworks.com)](https://www.kiteworks.com/secure-file-sharing/ultimate-guide-to-aes-256-encryption/)

**Homomorphic Message Authentication Code (HMAC):**

* While not directly related to homomorphic encryption, HMAC is a method of adding a signature to a message using a secret key. It allows the recipient to verify both the integrity and the authenticity of the message. HMAC is commonly used with symmetric key algorithms like AES.

[What is the Advanced Encryption Standard (AES)? Definition from SearchSecurity (techtarget.com)](https://www.techtarget.com/searchsecurity/definition/Advanced-Encryption-Standard)

1. **Error tolerance in SWHE Vs AES**

**SWHE**

1. **Error Accumulation:**
   * In somewhat homomorphic encryption schemes, particularly those based on lattice problems (e.g., Learning With Errors or Ring-LWE), the repeated application of homomorphic operations can introduce errors.

### Lattice Problems:

### Given a lattice, find the shortest non-zero vector in the lattice.

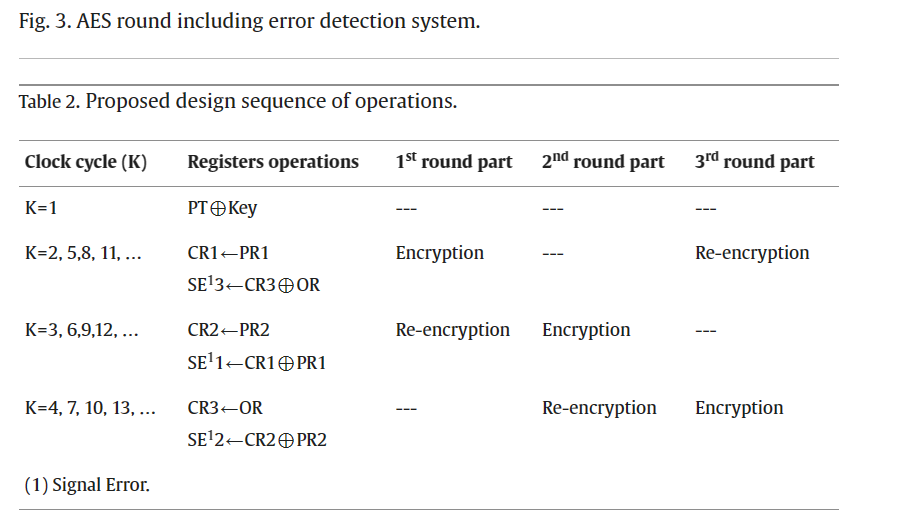
### [StephenHarrigan2017LWE.pdf (uottawa.ca)](https://mysite.science.uottawa.ca/mnevins/papers/StephenHarrigan2017LWE.pdf)

1. **Error Mitigation Techniques:**
   * Schemes often incorporate error-mitigation techniques to manage and minimize the impact of errors during homomorphic operations.
2. **Modulus Switching:**
   * Modulus switching is a technique used to manage errors by periodically "refreshing" the ciphertext, effectively resetting the noise to a tolerable level.
3. **Parameter Selection:**
   * The choice of parameters, such as the modulus and key sizes, can influence the level of error tolerance in a somewhat homomorphic encryption scheme.

### AES-256 (Symmetric Encryption):

1. **No Inherent Error Tolerance:**
   * Unlike homomorphic encryption, symmetric encryption algorithms like AES-256 do not inherently deal with errors or noise introduced during encryption.
2. **Deterministic Decryption:**
   * The decryption process in symmetric encryption is deterministic. If there are errors or modifications in the ciphertext, it typically results in completely incorrect decryption rather than a partially correct result.
3. **Data Integrity Considerations:**
   * While symmetric encryption primarily focuses on confidentiality, ensuring data integrity is typically addressed separately through techniques like message authentication codes (MACs) or cryptographic hashes.

[An improvement of both security and reliability for AES implementations - ScienceDirect](https://www.sciencedirect.com/science/article/pii/S1319157821003578)

****

1. **Cost Aspect for SWHE Vs AES 256 bit**

| **Cost Aspect** | **Somewhat Homomorphic Encryption (SHE)** | **AES-256** |
| --- | --- | --- |
| **Computational Cost** | High due to homomorphic operations. | Low; efficient symmetric encryption. |
| **Key Management** | High; public-key infrastructure, large key sizes. | Low; symmetric key management. |
| **Implementation** | Medium to High; specialized libraries and parameter tuning. | Low; well-established libraries. |
| **Quantum Resistance** | High; considered post-quantum secure. | Low; not inherently quantum-resistant. |
| **Error Tolerance** | Medium to High; error-mitigation techniques. | Low; deterministic, limited error tolerance. |
| **Performance** | Medium; speed of homomorphic operations. | High; fast symmetric encryption/decryption. |
| **Hardware Requirements** | Medium to High; depends on scheme and computations. | Low to Medium; well-supported on modern hardware. |
| **Security Assurance** | High; relies on mathematical hardness, ongoing research. | High; well-understood and extensively analyzed. |
| **Applicability** | Specific use cases requiring computations on encrypted data. | General-purpose data confidentiality. |
| **Overall Cost Assessment** | Medium to High, depending on use case and performance requirements. | Low to Medium; widely adopted and efficient. |