HOMOMORPHIC ENCRYPTIONS

Course Project, CS6530, July-Nov 2025

Sri Sai Shanmukh raj Malireddi
CS22B029

Vekateshwara reddy Nemali
CS22B027

INTRODUCTION

- HE is an encryption method that allows computations to be performed directly on encrypted data. The concept of performing operations on encrypted data dates back to the late 1970s. In 1978, Rivest, Adleman, and Dertouzos first introduced the idea of privacy homomorphisms, laying the foundation for what would later evolve into homomorphic encryption. Their proposal highlighted the potential for computations to be carried out on ciphertexts without needing decryption, though practical implementations at the time were highly limited.

- Over the years, researchers developed partially homomorphic encryption (PHE) schemes, which allowed either addition or multiplication on encrypted data, but not both. Notable examples include the RSA cryptosystem (multiplicative homomorphism) and the Paillier cryptosystem (additive homomorphism). While these provided important building blocks, they could not support arbitrary computations.

- A major breakthrough came in 2009, when Craig Gentry introduced the first fully homomorphic encryption (FHE) scheme. His work demonstrated that it was theoretically possible to perform arbitrary computations (both addition and multiplication) on encrypted data, making FHE a milestone in cryptography. However, Gentry's scheme was computationally expensive and impractical for large-scale use.

- Since then, significant progress has been made in improving the efficiency of HE. Multiple FHE schemes, such as BGV, BFV, CKKS, and TFHE, have been developed, each tailored for different types of computations (e.g., exact arithmetic, approximate arithmetic, or faster bootstrapping). Advances in algorithms and hardware have further pushed HE closer to real-world applications.

- Today, homomorphic encryption is viewed as one of the most promising tools for privacy-preserving computation, particularly in cloud services, artificial intelligence, healthcare analytics, and finance. It continues to evolve, balancing security, efficiency, and scalability.

FORMALISING HOMOMORPHIC PROPERTY

- HE is an encryption method that allows **computations to be performed directly on encrypted data.**
- After decryption, the result is the same as if operations were performed on the original plaintext.

Formal Property:

For encryption function ***Encryption*** and decryption function ***Decryption:***

$$Decryption(Encryption(m1) \boxplus Encryption(m2)) = m1 \oplus m2$$

Here, $\oplus$ = operation on plaintexts, and $\boxplus$ = corresponding operation on ciphertexts.

PAILIER CRYPTOSYSTEM

- Public-key cryptography traditionally allowed either encryption/decryption or digital signatures but did not directly support computations over encrypted data.
- Early schemes like RSA offered multiplicative properties, but an efficient additive homomorphic system was lacking. This limitation motivated the search for cryptosystems capable of supporting privacy preserving computations such as secure voting, private aggregation of sensitive information, and confidential benchmarking of data.
- Against this backdrop, Paillier proposed a new encryption scheme in 1999 based on the composite residuosity class problem, which allowed secure additions of encrypted integers. The Paillier cryptosystem was introduced by Pascal Paillier in 1999 at EUROCRYPT .
- It represents a milestone in public-key cryptography by being one of the first practical additive homomorphic schemes. Its construction relies on the decisional composite residuosity assumption (DCRA), which ensures that distinguishing certain composite residues modulo n^2 is computationally infeasible.
- Since its publication, Paillier's scheme has become a foundational building block for privacy-preserving systems such as e-voting, private information retrieval, and data aggregation in distributed systems.
- Paillier's additive homomorphism makes it ideal for secure voting, privacy-preserving data aggregation (e.g., summing encrypted sensor data), and financial protocols where summation of private values is required without exposing the underlying data.

RSA (HOMORPHIC VARIANT)

- RSA, introduced by Rivest, Shamir, and Adleman in 1977 is one of the most widely used public-key cryptosystems.
- Originally designed for secure communication and digital signatures, RSA is based on the mathematical difficulty of factoring large composite integers.
- Although RSA was not explicitly developed as a homomorphic encryption scheme, it naturally exhibits a multiplicative homomorphic property.
- This means that the product of two ciphertexts corresponds to the encryption of the product of the underlying plaintexts.
- This property makes RSA suitable for specific privacy-preserving computations, such as delegated multiplications or certainsecure protocols, though it lacks additive homomorphism and full homomorphism like Paillier or Gentry's scheme.
- While it does not support additions like Paillier, it remains useful for protocols where multiplying encrypted values is required. Because RSA lacks semantic security under chosen-plaintext attacks when used naively, padding schemes such as OAEP are normally used in practice to strengthen its security.

GENTRY'S FHE SCHEME

- In 2009, Craig Gentry introduced the first construction of a fully homomorphic encryption (FHE) scheme. Before Gentry's work, cryptosystems like Paillier or RSA only supported partial homomorphisms (additive or multiplicative).

- Gentry's breakthrough enabled both addition and multiplication on ciphertexts, allowing arbitrarydepth computations on encrypted data. This capability opened the door to performing secure computations in untrusted environments, such as cloud computing, without ever revealing sensitive plaintexts. Gentry's scheme relies on the hardness of lattice-based problems and introduces the concept of bootstrapping to manage the noise that accumulates during homomorphic operations.

- Gentry's FHE scheme laid the foundation for later practical FHE constructions such as BFV and CKKS. It is particularly useful in privacy-preserving cloud computing, encrypted machine learning, and secure multi-party computation, where arbitrary operations on encrypted data are required without revealing sensitive information.

- By allowing both addition and multiplication operations on ciphertexts, Gentry's scheme enables the evaluation of any computable function over encrypted inputs, a capability that was previously impossible with traditional partially homomorphic cryptosystems. This property opens the door to fully secure outsourced computations, such as encrypted database queries, confidential medical data analysis, and private financial modeling, without exposing the underlying sensitive data.

- Furthermore, the concept of bootstrapping introduced by Gentry not only solved the problem of noise growth in ciphertexts but also inspired numerous optimizations in subsequent FHE schemes, making fully homomorphic encryption increasingly practical for real world applications.

- Despite the initial computational overhead, Gentry's FHE serves as a theoretical and practical cornerstone for secure computation research, and its principles continue to guide advancements in efficient, privacy-preserving cryptography for cloud services, federated learning, and blockchainbased confidential computation.

BFV (BRAKERSKI FAN-VERCAUTEREN)

- The BFV scheme, independently proposed by Brakerski and Fan and Vercauteren, is a lattice-based fully homomorphic encryption scheme that improves the efficiency of Gentry's original FHE construction. BFV supports both addition and multiplication on encrypted integers modulo a plaintext modulus, while managing the noise growth efficiently without frequent bootstrapping.
- It is based on the Ring Learning With Errors (RLWE) problem, which is believed to be hard even for quantum computers.
- BFV is widely used in privacy-preserving computations, such as secure aggregation, encrypted machine learning, and confidential data analysis in cloud environments.
- BFV is particularly suited for applications requiring exact arithmetic on encrypted integers, including encrypted voting, privacy preserving statistics, and secure financial computations.
- It is also used in secure data aggregation from IoT devices and confidential benchmarking of organizational data, where maintaining exact numerical integrity is crucial. More over, BFV supports efficient batch operations using SIMD techniques, allowing multiple encrypted values to be processed

CKKS (CHEON-KIM-KIM-SONG)

- The CKKS scheme, proposed by Cheon, Kim, Kim, and Song in 2017, is a lattice-based homomorphic encryption scheme designed for approximate arithmetic on real or complex numbers.
- Unlike BFV, which handles exact integers, CKKS allows efficient computation on encrypted floating point data with controlled approximation errors.
- It leverages the Ring Learning With Errors (RLWE) problem for security and uses rescaling techniques to manage noise growth during repeated homomorphic multiplications.
- CKKS is widely applied in privacy-preserving machine learning, encrypted signal processing, and computations on sensitive numerical data in cloud environments.
- CKKS has become the standard for encrypted machine learning and data analytics, enabling operations on sensitive real-world data without exposing the raw values.