Privacy-Preserving Data Analytics:

What are the financial costs associated with the use of Homomorphic Encryption for privacy-preserving data analytics ?

Responsible Professor: Dr.Zeki Erkin
Supervisor: Dr.Roland Kromes
Examiner: Dr.Xucong Zhang

Iván Moreno Sarriés

Introduction

- In the digital age, data is one of the most valuable assets for both public and private organizations. Healthcare , finance , and government agencies increasingly rely on data analytics to improve decisions and optimize operations.
- To address these privacy risks, a variety of privacy-preserving technologies have been developed . Among them, Homomorphic Encryption (HE) has emerged as one of the most promising. Homomorphic encryption allows computations to be performed directly on encrypted data, without the need to decrypt it first.
- Despite its theoretical appeal, HE has seen limited adoption outside of research prototypes. The main barriers are its significant computational overhead, and critically, its significant fiancial burden.
- We explore the financial viability of HE in realistic cloud environments, using publicly available cloud pricing and salary data to model total expenses.

Total Cost of Ownership(TCO)

- **Final cost**: cloud infrastructure + expert salary = **$125,558.066/year**
- Excludes:
  - Electricity for local encryption
  - Performance-related delays
  - Opportunity cost from staff/resource allocation

- While the cloud hardware and software components are relatively inexpensive, approximately $1,145.4 annually, the specialized cryptography skill set required to use FHE, commands over 98% of the estimated total cost. This underscores that HE is not merely computationally demanding, it also requires specialized human expertise, which presents a major barrier to adoption.

Future work

- Investigate real-world applications of HE in areas with strict privacy requirements, such as healthcare and finance, where privacy benefits justify the overhead.

- Invest in tooling and training to reduce the knowledge barrier, and enable a wider adoption across disciplines and industries. Focusing on methods that balance performance and security guarantees.

- Develop collaborative ecosystems between academia, industry, and government to share best practices, open-source tooling, and reproducible benchmarks.

Homomorphic Encryption (HE)

- **First gen HE (PHE)**: Supported only a single operation type per circuit (addition or multiplication). These schemes suffered from uncontrolled noise growth with each operation, decryption became incorrect when noise exceeded a certain threshold.
- **Second gen HE (SWHE)**: Was characterized by much better techniques for controlling the noise, enabling a limited number of both additive and multiplicative operations simultaneously.
- **Third gen HE (FHE)**: Enabled unlimited computation on encrypted data through a breakthrough technique called bootstrapping, introduced by Craig Gentry in 2009.

Partially Homomorphic Encryption (PHE)

- PHE schemes allow only one type of operation, namely addition (XOR) or multiplication (AND). Many schemes have shaped the evolution of PHE: Rivest et al. (1978) introduced RSA, Goldwasser and Micali (GM) (1982), El-Gamal (1985) , Benaloh (1994) , and Paillier (1999).

Somewhat Homomorphic Encryption(SWHE)

- SWHE schemes support both addition and multiplication operations, but only for a limited number of computations or circuit depth due to noise accumulation in ciphertexts. Most FHE schemes are built upon SWHE schemes with additional properties that enable full homomorphism.

Fully Homomorphic Encryption (FHE)

- An encryption scheme is called fully homomorphic if it supports an unlimited number of homomorphic evaluation operations on encrypted data, with the results remaining within the ciphertext space.

Squashing

- Gentry's bootstrapping technique works only when the decryption algorithm has a small circuit depth. To enable bootstrapping, he introduced a technique called squashing, which reduces the complexity of the decryption circuit.

Bootstrapping

- A scheme is called bootstrappable if it can evaluate its own decryption algorithm circuit.
- Performs decryption of the ciphertext before the noise exceeds a threshold: this removes the noise with respect to the first encryption key. Then, encryption is performed again using a new key. The new encryption key should be chosen carefully to obtain a lower noise level than the removed one

Modern FHE Schemes

BFV

- The **Brakerski-Fan-Vercauteren (BFV)** (also known as FV) scheme is one of the cornerstone protocols in the FHE landscape. Its significance lies in its ability to enable **exact arithmetic over encrypted data**, making it ideal for applications that require precise integer calculations.

CKKS

- The **Cheon-Kim-Kim-Song (CKKS)** scheme is pivotal for homomorphic encryption because it allows to perform **approximate additions and multiplications of ciphertexts**, where its plaintexts can be vectors of real and complex values. Unlike BFV, which guarantees exact integer results, CKKS is tailored for applications that tolerate a small error margin.
- Homomorphic encryption schemes that support approximate arithmetic, such as CKKS, are particularly well-suited for computations on real-valued data.

Performance Evaluation

Data Analytics

- This section reviews the paper from Lo et al."Practical Considerations of Fully Homomorphic Encryption in Privacy-Preserving Machine Learning"

- While FHE can be successfully applied to machine learning tasks, it imposes significant performance overhead compared to plaintext computation. Encryption and decryption dominate the run time, making FHE impractical for large scale analytics workloads nonetheless, in domains where privacy is paramount, such as healthcare or finance, these overheads may be justified.

Real-Life Example

- This section reviews the paper from Guillot et al. "A Performance and Cost Evaluation of Homomorphic Encryption in the Cloud"

- Experimental findings reaffirm that encryption alone, accounts for over 80% of the total processing time in typical workloads, and that performance degrades significantly as encryption parameters (e.g., polynomial modulus degree) increase. The hardware architecture also plays pivotal roles in achieving acceptable performance. These findings emphasize that while FHE is viable for small to medium datasets, it demands significant expertise and resources for larger, more complex applications, a crucial consideration for researchers and businesses alike.