陳奕嘉 E94115011 資訊115

**Secure Computing Technologies (12/23演講報告)**

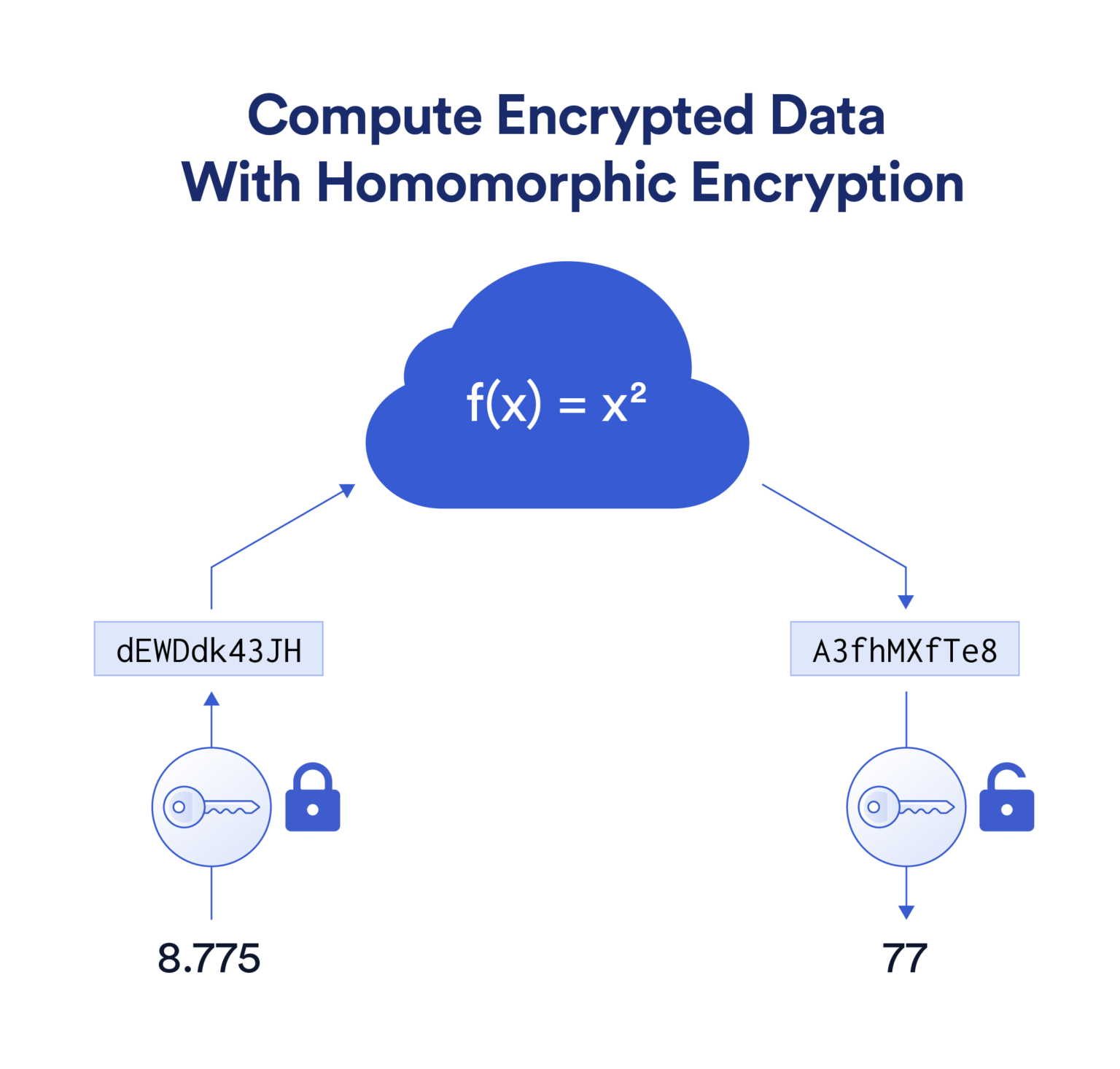
**Table of Contents**

1. Introduction
2. Homomorphic Encryption (HE)
3. Differential Privacy (DP)
4. Secure Multiparty Computation (SMPC)
5. Federated Learning (FL)
6. Interplay Between HE, DP, SMPC, and FL
7. Challenges and Considerations
8. Conclusion
9. References

In today's data-driven world, ensuring the privacy and security of sensitive information is paramount. Technologies such as Homomorphic Encryption (HE), Differential Privacy (DP), Secure Multi-Party Computation (SMPC), and Federated Learning (FL) have emerged as pivotal tools in this endeavor. This report delves into each of these technologies, exploring their functionalities, applications, and the interplay between them.

**Homomorphic Encryption (HE)**

Homomorphic Encryption is a form of encryption that permits computations to be performed on ciphertexts, generating an encrypted result that, when decrypted, matches the outcome of operations performed on the plaintext. This capability is invaluable for processing sensitive data in untrusted environments, such as cloud computing, without exposing the data itself.



img source: https://chain.link/education-hub/homomorphic-encryption

This diagram demonstrates **Homomorphic Encryption (HE)**, a cryptographic method that allows computations to be performed on encrypted data without decrypting it. The diagram shows encrypted inputs (represented by coded strings) being processed by a function f(x) = x^2 in the cloud. The encrypted results are then decrypted to reveal the correct outputs (8.775 and 77) without exposing the original data during the computation process. HE ensures data privacy during cloud-based processing, making it valuable for secure data analytics and machine learning on sensitive datasets.

There are three primary types of HE:

**Partial Homomorphic Encryption (PHE):** Supports either addition or multiplication, but not both, an unlimited number of times.

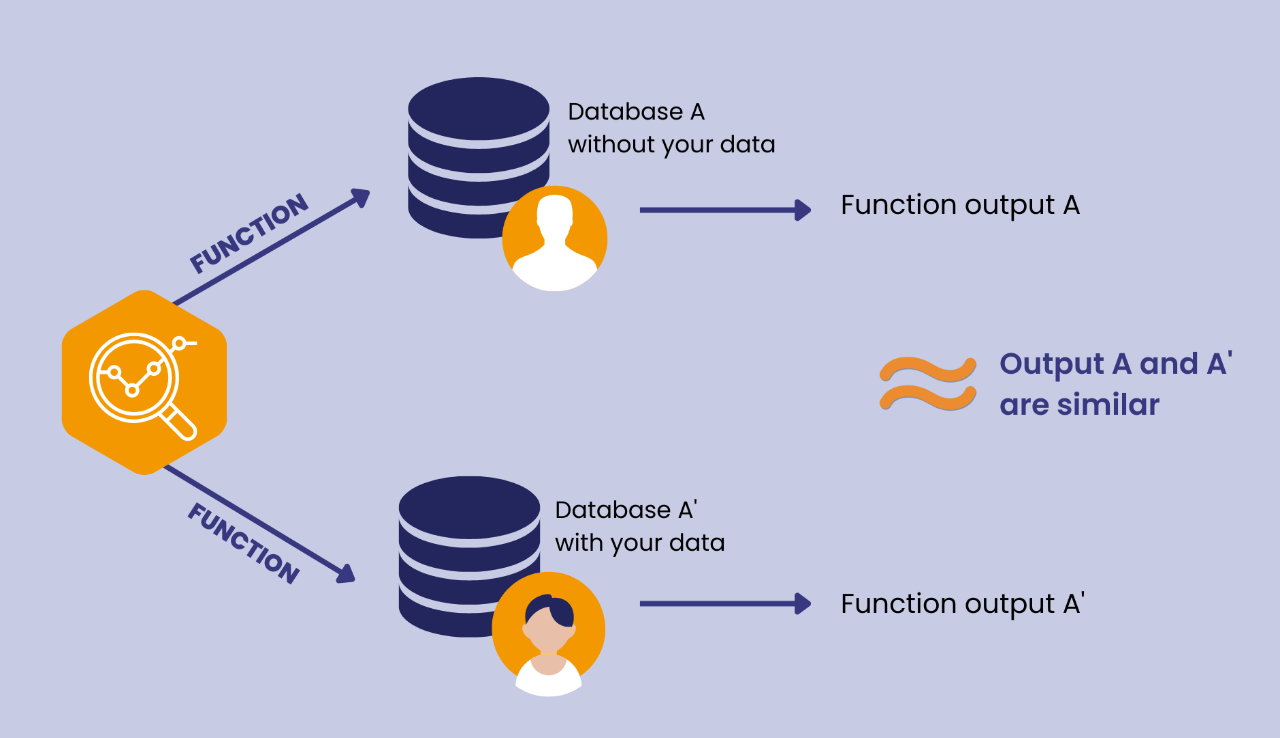
**Somewhat Homomorphic Encryption (SWHE):** Allows both addition and multiplication but with limited operations.

**Fully Homomorphic Encryption (FHE):** Enables unlimited addition and multiplication operations, presenting the most versatile yet complex form.

The ElGamal scheme is discussed as a fundamental example of multiplicative PHE. Through key generation, encryption, and decryption phases, ElGamal encryption ensures secure multiplicative operations on ciphertexts. The document also introduces additive PHE through the Paillier scheme, which facilitates secure addition operations. These methodologies provide foundational building blocks for more sophisticated cryptographic applications.

**Differential Privacy (DP)**

Differential Privacy (DP) is a robust framework designed to protect individual data points while allowing statistical analysis of datasets.



img source: https://www.anonos.com

This diagram explains the concept of **Differential Privacy (DP)**. It compares two databases – **Database A** without an individual's data and **Database A'** with the individual's data. When a function is applied to both databases, the outputs (A and A') remain similar, demonstrating that the presence or absence of a single data point does not significantly affect the overall result. This ensures privacy by preventing the identification of any individual’s data from the output, making Differential Privacy crucial for protecting personal information in data analysis and machine learning.

The presentation introduces the concept by explaining the notion of ε-DP, where ε quantifies the permissible information leakage. A smaller ε signifies stronger privacy guarantees. The concept of ε, δ-DP is also presented, where δ represents the probability of the privacy guarantee failing. This relaxation allows for practical implementations while maintaining a high degree of privacy.

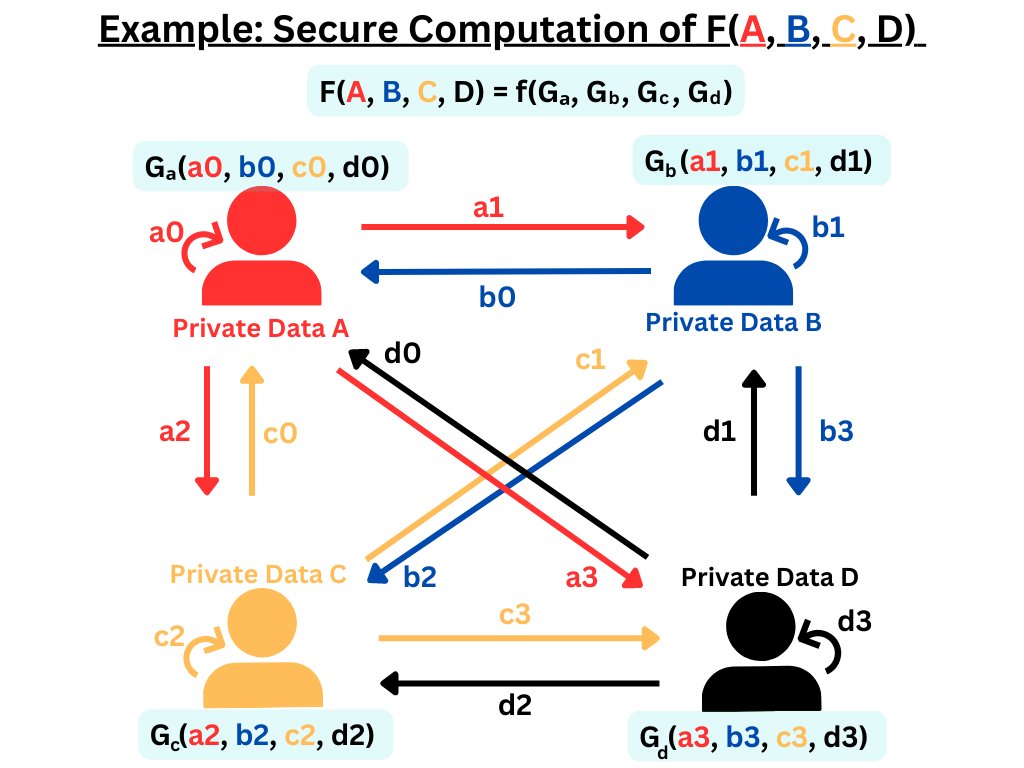
Key techniques for achieving DP include the addition of noise to query outputs. Two primary noise distributions are discussed:

* **Laplace Distribution:** Commonly applied for pure DP, providing noise proportional to the sensitivity of the dataset.
* **Gaussian Distribution:** Utilized for (ε, δ)-DP, balancing privacy and utility by introducing probabilistic guarantees.

Differential Privacy plays a pivotal role in data analytics, machine learning, and public data sharing by mitigating the risks of re-identification and linkage attacks.

**Secure Multiparty Computation (SMPC)**

Secure Multiparty Computation (SMPC) enables collaborative computation on private inputs without revealing them. This cryptographic protocol is crucial for scenarios involving mutually distrustful parties who wish to jointly compute a function.



img source: https://iudx.org.in/secure-multi-party-computation

This diagram illustrates **Secure Multi-Party Computation (SMPC)**, where multiple parties collaboratively compute a function F(A,B,C,D) over their private inputs without revealing those inputs to each other. Each party exchanges encrypted data points (shown by colored arrows) with the others, ensuring that no single participant has full access to the raw data. The final computation produces a result that reflects the collective input while maintaining data confidentiality. SMPC is widely used in fields like healthcare, finance, and research, allowing for secure data analysis across organizations without compromising privacy.

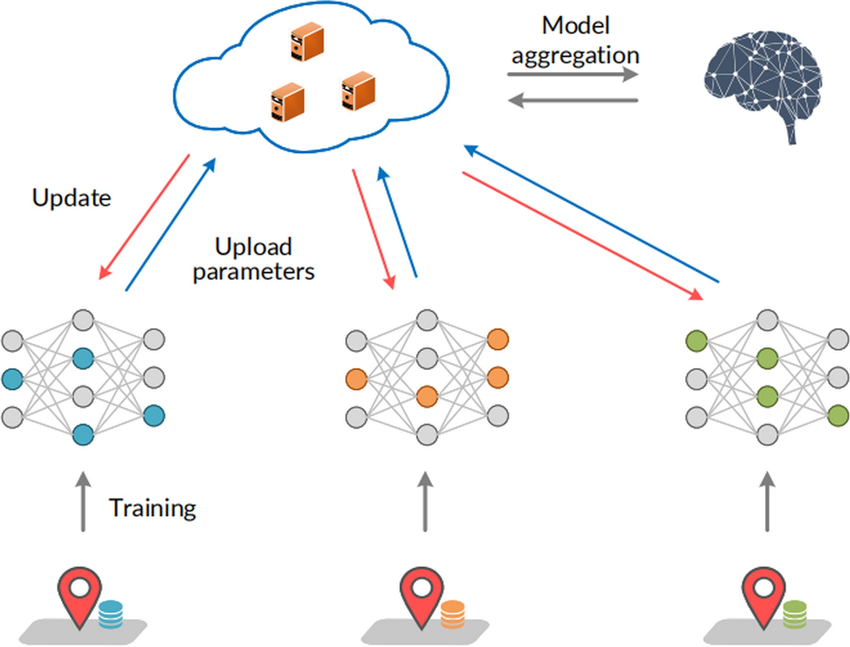
The presentation highlights Yao’s Millionaire Problem as a classic example, illustrating how two parties can determine who is wealthier without disclosing their wealth. Yao’s protocol employs asymmetric encryption and decryption steps to achieve privacy.

A broader application is demonstrated through Secure Multiparty Sum, where multiple parties aggregate their private data securely.

SMPC is foundational for secure voting systems, privacy-preserving analytics, and confidential benchmarking.

**Federated Learning (FL)**

Federated Learning (FL) presents a paradigm shift in machine learning by decentralizing the training process. Rather than aggregating raw data in a central server, FL orchestrates local model training across distributed devices, ensuring data privacy and reducing communication overhead.



img source: https://iudx.org.in/secure-multi-party-computation

This diagram illustrates **Federated Learning (FL)**, a machine learning approach that trains a model across multiple decentralized devices or servers holding local data. Each device trains a local model on its data, and the updated parameters (not the raw data) are uploaded to a central cloud server. The server aggregates these updates to improve the global model, which is then distributed back to the devices for further training. This process repeats iteratively. Federated Learning enhances privacy by ensuring that raw data remains on the local devices, making it valuable for applications in healthcare, finance, and mobile services.

Two primary FL architectures are discussed:

* Server-Orchestrated FL: A central server coordinates model updates from distributed clients.
* Decentralized FL: Clients communicate directly, aggregating models without centralized control.

Key challenges addressed include:

* Data Heterogeneity: Variability in local datasets.
* Computational Constraints: Varying client capabilities.
* Privacy Risks: Potential model inversion and poisoning attacks.

FL's significance lies in its application to sensitive domains such as healthcare, finance, and mobile applications, where data privacy is paramount.

**Interplay Between These Technologies**

The integration of HE, DP, SMPC, and FL can lead to robust privacy-preserving data analysis and machine learning frameworks:

HE and FL: Applying Homomorphic Encryption within Federated Learning allows computations on encrypted local models, enhancing security. For instance, encrypted model updates can be aggregated without decryption, ensuring that individual updates remain confidential.

DP and FL: Incorporating Differential Privacy into Federated Learning ensures that updates from individual devices do not leak sensitive information, even indirectly. This is achieved by adding noise to model updates, balancing privacy and utility.

SMPC and FL: Secure Multi-Party Computation can be employed in Federated Learning to securely aggregate model updates from different clients. This ensures that the central server learns nothing about individual updates beyond the aggregated result, mitigating risks associated with data leakage.

HE and DP: Combining Homomorphic Encryption with Differential Privacy allows computations on encrypted data while ensuring that the outputs do not compromise individual privacy. This dual-layered approach is particularly useful in scenarios requiring both data confidentiality and privacy-preserving data analysis.

**Challenges and Considerations**

While these technologies offer significant advancements in privacy preservation, they also present certain challenges:

Computational Overhead: Techniques like Homomorphic Encryption and Secure Multi-Party Computation can introduce substantial computational complexity, potentially impacting performance.

Accuracy Trade-offs: Implementing Differential Privacy involves adding noise to data, which can affect the accuracy of analytical results or machine learning models.

Scalability**:** Deploying these technologies at scale, especially in resource-constrained environments, requires careful consideration to balance privacy, efficiency, and accuracy.

**Conclusion**

By exploring Homomorphic Encryption, Differential Privacy, Secure Multiparty Computation, and Federated Learning, it equips students with the foundational knowledge necessary to navigate the evolving landscape of cybersecurity. As threats to data privacy and integrity continue to grow, the principles outlined in this document will remain instrumental in shaping the future of secure computing.

**Reference**

https://www.mdpi.com/2079-9292/13/22/4478

https://arxiv.org/abs/2010.05867

https://www.jpmorgan.com/content/dam/jpm/cib/complex/content/technology/ai-research-publications/pdf-9.pdf

https://ieeexplore.ieee.org/document/10320195