Chapter 4: Approaches for Private Computation

Lecture PETs4DS:
Privacy Enhancing Technologies for Data Science

Dr. Benjamin Heitmann and Prof. Dr. Stefan Decker
Informatik 5
Lehrstuhl Prof. Decker

RWTH AACHEN UNIVERSITY

Overview of chapter

- **High-level Concepts**
- **Secure Multi-party Computation**
- **Yao's Garbled Circuits**
- **Limitations**

High-level Concepts

Based on:
Yakoubov, Sophia, et al. "A survey of cryptographic approaches to securing big-data analytics in the cloud." *High Performance Extreme Computing Conference (HPEC),* IEEE, 2014.

Motivation

- Cloud computing enables organisations to outsource computation.
  - Enables storage and analysis of data on shared resources.
  - Easy to handle fluctuations in volume and velocity of data
- Total size of market for public cloud services is ~205 billion USD in 2016
- However, cloud computing introduces many new threats:
  - Cloud infrastructure might be provided by untrusted entities.
  - Tampering with data or computation is possible.
- **How can we address these new threats?**
- **Is private computation possible even in the cloud?**

Common Node Roles for Big Data Analytics in Cloud Computing

Many Big Data analytics applications involve the following node roles:

- **Input node (I)**
  - Supplies raw data
  - Examples: aggregated data from sensors or users of a system
- **Compute node (C)**
  - Performs computation on the data
- **Storage node (S)**
  - Store data between computations: input & temporary & output data
- **Result node (R)**
  - Receives computation result
  - Sends result to client or makes automated decision

Running Example: Genomic Data Analysis

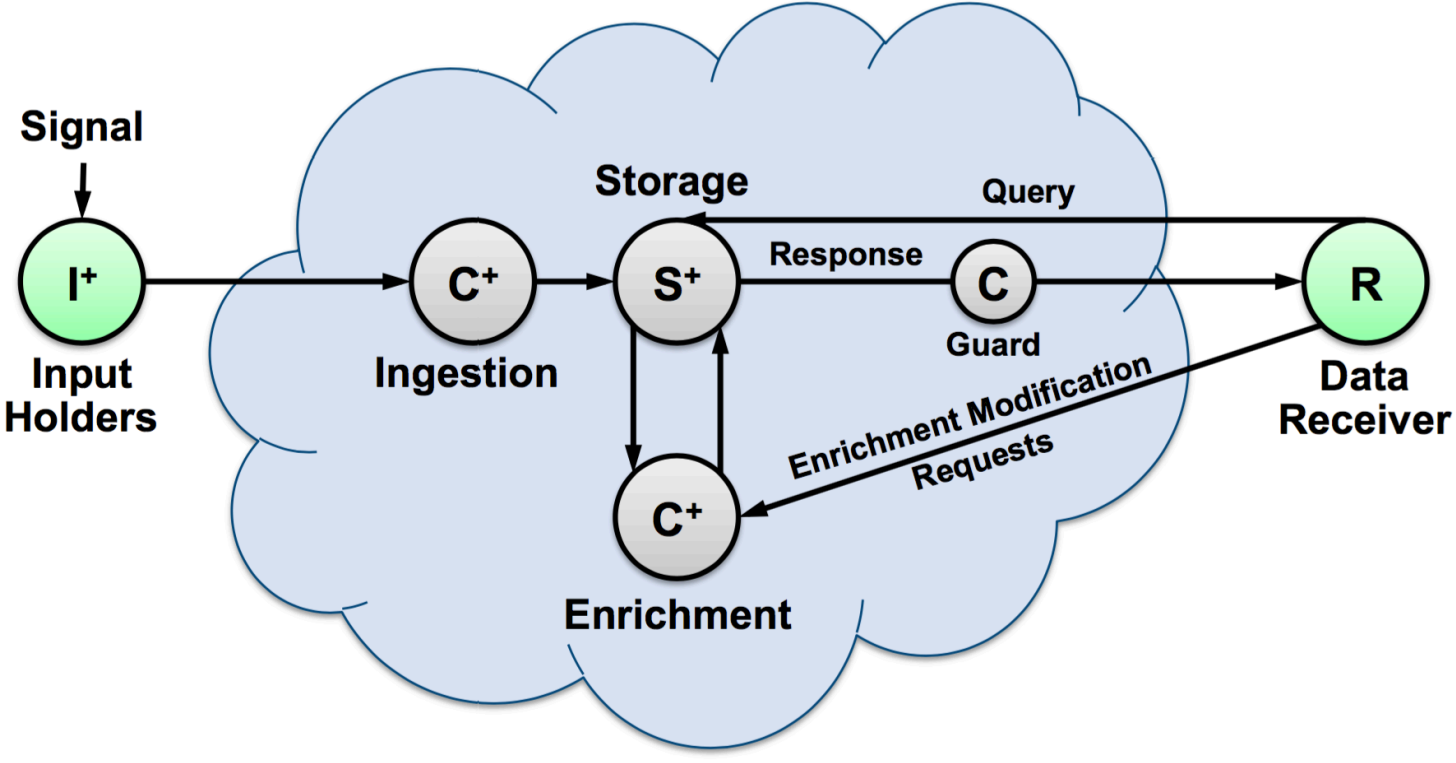
- Sensitive data: reference data sets for genomic sequences.
- Maintained by agencies like National Institutes for Health (NIH) in US

- **Data analytics application:**
  - Allow scientists to check correlation of genetic sequence to reference sequence
  - Correlation can be checked using sequence metadata
  - However, sequence metadata needs to be computed
  - Requires private computation

Running Example: Genomic Data Analysis

- Data is generated by genetic sequencers -> **Input node**
- Genetic sequences need to be stored for use as input / intermediary / output data -> **Storage node**
- The incoming data needs to be pre-processed for organizing it -> **Compute node**
- Meta-data on genetic sequences provides enrichment of data -> **Compute node**
- Output results are presented to scientists -> **Result node**
- Output needs to be checked to control unintended disclosure of data **-> Compute node**

Running Example: Genomic Data Analysis



Desirable Security and Privacy Properties of Cloud Computing

- **Availability**
  - Data owners have access to their data and to computation resources
  - Main goal of cloud infrastructure
  - Already solved by infrastructure
- **Confidentiality**
  - All sensitive data remains secret from adversaries and untrusted entities
  - Corresponds to Disclosure threat in LINDDUN
  - Not guaranteed!
- **Integrity**
  - All outputs of computation are correct.
  - Any unauthorised modification of sensitive data can be traced.
  - Not guaranteed!

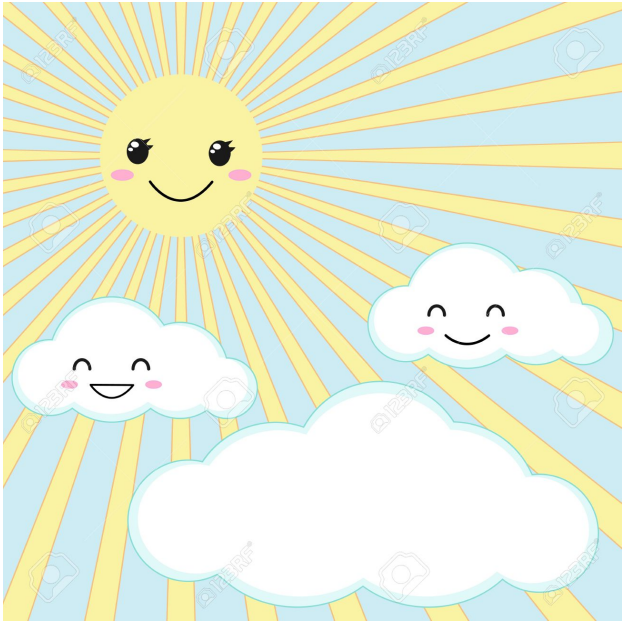
Cloud Computing Scenarios

- **Untrusted Cloud**
  - Cloud provider makes no guarantees
  - Confidentiality or integrity of data might not be maintained by cloud nodes
  - Most common type of cloud infrastructure today, e.g. Amazon Web Services (AWS)
  - Any kind of adversary is possible
  - Corresponds to public cloud deployment model

Cloud Computing Scenarios

- **Trusted Cloud**
  - Cloud is completely controlled by organisation (it is in-house / on-site)
  - Air-gap between cloud and outside network
  - Clients data stays confidential and can not leave cloud
  - Cloud nodes can still be corruputed
  - Adversary can threaten data integrity
  - Very common in government use cases

  - Corresponds to private cloud deployment model



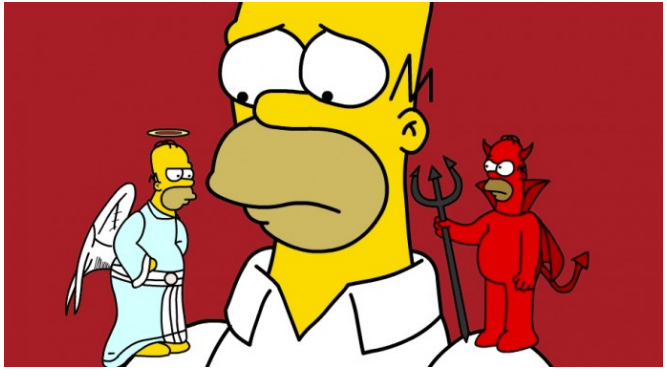
Cloud Computing Scenarios

- **Semi-Trusted Cloud**
  - Client can not fully trust the cloud
  - Not the whole cloud is malicious
  - Parts of cloud could be controlled by adversary
  - Common when cloud provider maintains security, but does not guard against other threats
  - Corresponds to public & private & hybrid cloud deployment model

Adversary types

The two most important types of adversary

- **Honest-but-curious (HBC) adversary**
  - Controlled nodes perform all computation and protocol like honest node
  - Adversary tries to learn additional information
  - Can combine information from multiple controlled nodes
  - Can learn information which normally no single party can learn



- **Malicious adversary**
  - Controlled nodes deviate arbitrarily from computation / protocol.
  - Examples: send malformed messages, incorrect computation, active collusion.
  - Active effort to violate confidentiality or integrity.

- A single adversary could control multiple nodes

Approaches to Achieve Computational Privacy

- **Homomorphic Encryption (HE)**
  - Encrypt data before sending to cloud
- **Verifiable Computation (VC)**
  - Prove that results are correct
- **Secure Multi-Party Computation (SMPC)**
  - Compute results together without sharing private data

Homomorphic Encryption: Use Case and Cloud Scenario

- **Genomic data analytics example**
  - NIH wants to outsource computation to Amazon Elastic Compute Cloud (EC2)
  - Amazon EC2 provides an untrusted cloud
  - No guarantees regarding confidentiality

- **How to enable private computation on untrusted cloud?**
  - Is it possible to use encrypted data for computation without decrypting it?

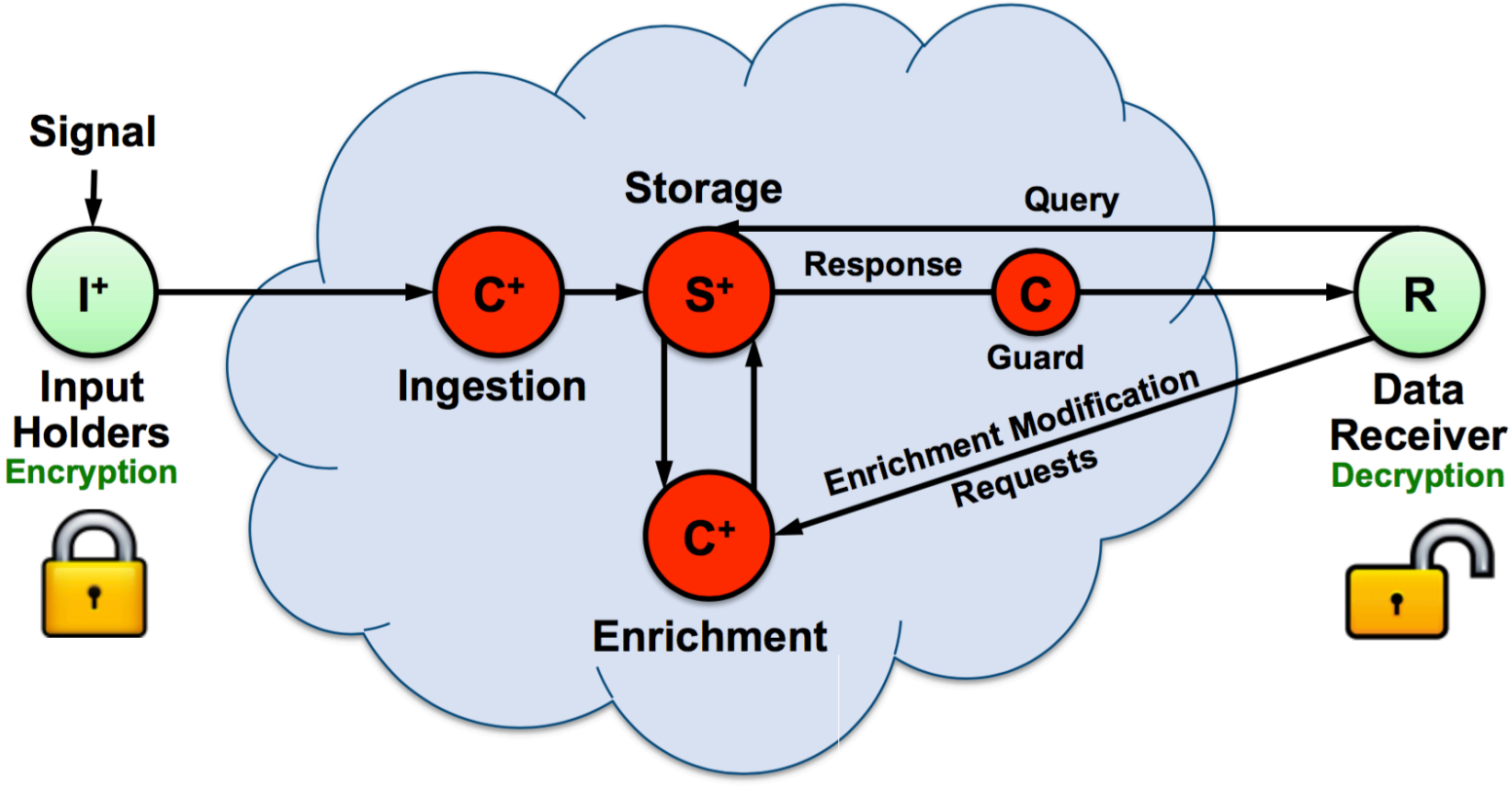
Homomorphic Encryption: Use Case and Cloud Scenario

- **Homomorphic encryption:**
  - Encryption of message m under key k: $E_k(m)$
  - Decryption under key k: $D_k(\,)$
  - **Requirement for homomorphic encryption:**
    For function f there is f' such that $D_k\big(f'(E_k(m))\big) = f(m)$

- Fully Homomorphic Encryption (FHE) enables any kind of computation
- Somewhat Homomorphic Encryption (SHE) enables only some operations, e.g. multiplication

- **Limitations:**
  - Slow: matrix-vector multiplication for 256 integers takes 26 seconds [HElib]
  - Only one key: all input and result nodes have to share the same key

Homomorphic Encryption: Illustration



Homomorphic Encryption: Overview of Properties

- **Confidentiality:** yes
  - Compute and storage nodes do not have access to the unencrypted data
- **Integrity:** no
  - Modification of data is possible
  - Faulty computation is possible
- **Adversary type addressed:** malicious
- **Requires interaction:** no
  - HC is possible with only one party providing data

Verifiable Computation: Use Case and Cloud Scenario

- **Genomic data analytics example**
  - NIH has funding to build its own cloud with no connection to outside networks (air gap)
  - No data can leave this private cloud
  - Confidentiality of data is automatically guaranteed
  - Private cloud can be compromised, e.g. through malware or supply chain attack
  - Integrity of data and computation still needs to be protected
- **Is it possible to guarantee integrity in a private cloud?**
- Solved for data storage, e.g. with checksums and broadcasting of checksums
- Can we guarantee integrity of computation results ?

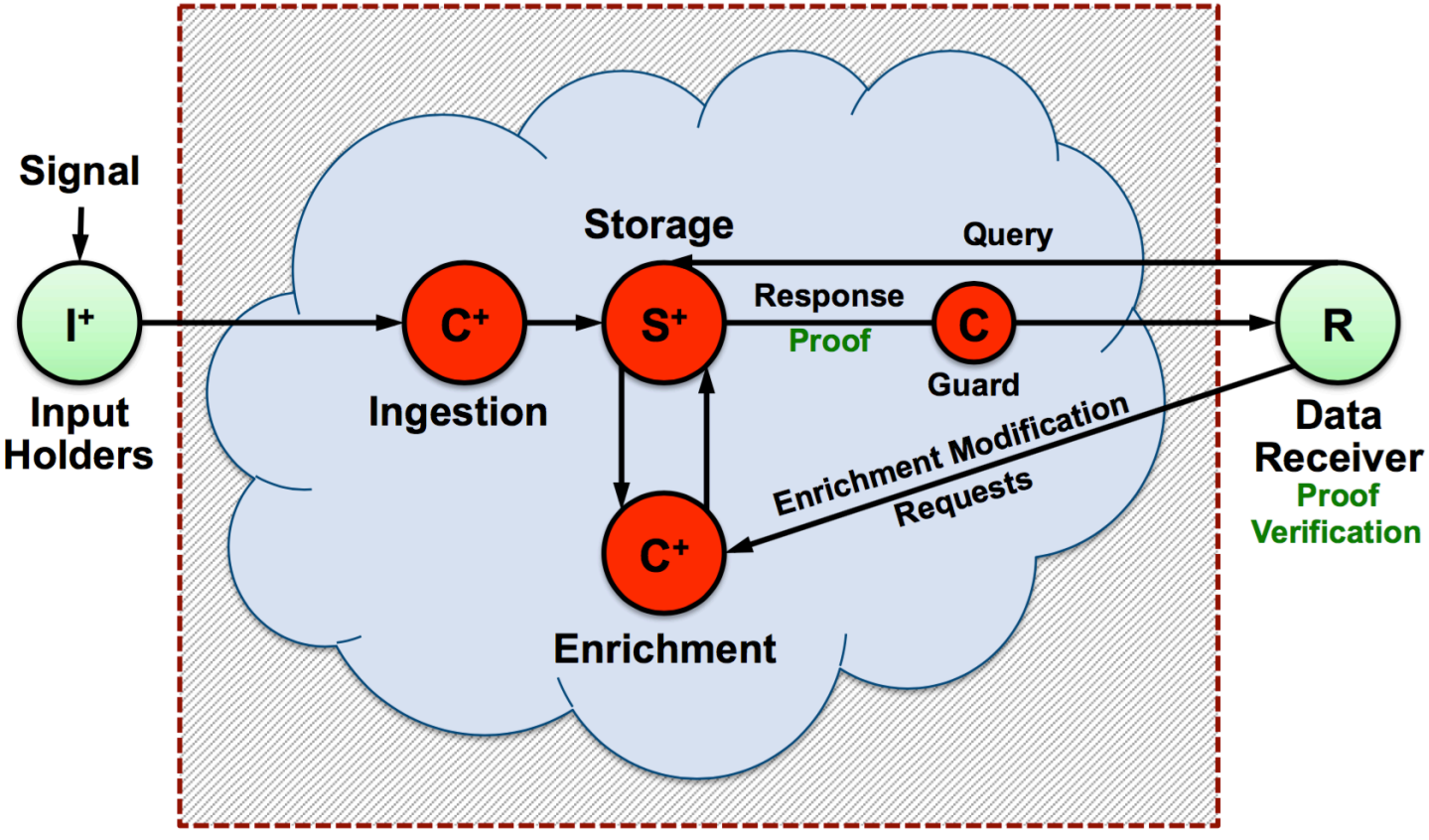
Verifiable Computation: Details

- Verifiable computation allows data owner to check integrity of computation.

- **How does verifiable computation work?**
  - **Data owner** gives data with specification of computation to *prover*
  - **Prover** has more computation resources than data owner
  - Prover performs computation as specified
  - Prover returns result of computation together with proof of correctness.
  - Data owner then verifies proof.
  - VC requires proofs to be easier to verify than the proven computation.
  - Details of such proofs not covered by this lecture.

- **Limitations:**
  - VC does not involve encryption of data.
  - Slow: Computation is usually faster then constructing proof.
  - Pinocchio library: computation with 277.000 multiplications, construction of proof takes 144 sec, verification of proof takes 10 ms.

Verifiable Computation: Illustration



Verifiable Computation: Overview of Properties

- **Confidentiality:** no
  - If VC is used in a private cloud, confidentiality is already guaranteed by isolation of cloud
- **Integrity:** yes
  - Computation nodes can prove correctness of results
- **Adversary type addressed:** malicious
- **Requires interaction:** no

Secure Multi-Party Computation: Use Case and Cloud Scenario

- **Genomic data analytics example**
  - After building a private cloud, the NIH discovers they need even more computing resources
  - Private cloud is expanded with computation nodes from public cloud
  - Result is a semi-trusted cloud.
  - Not all nodes will be controlled by adversary.
  - However, both HBC and malicious adversaries are possible.

- **Is it possible for multiple nodes to collaboratively perform a computation?**
- **If yes, can this be done without actually sharing the secret data?**

Secure Multi-Party Computation: Details

How does SMPC work?

- All nodes split their secret data into shares.
- Shares are distributed to the other nodes.
- Every node then calculates intermediate results and shares them with all other nodes.
- This allows verifying the correctness of intermediate results.
- Each node then can use the public intermediate results with his private shares to determine result.
- All nodes will get the same result.

- No node learns something they don’t already know, except for the result of the computation.

Secure Multi-Party Computation: Limitations

Limitations of SMPC:

- Slow, but not as much overhead incurred as for HE
- Requires more than two parties, but does not scale well with number of parties

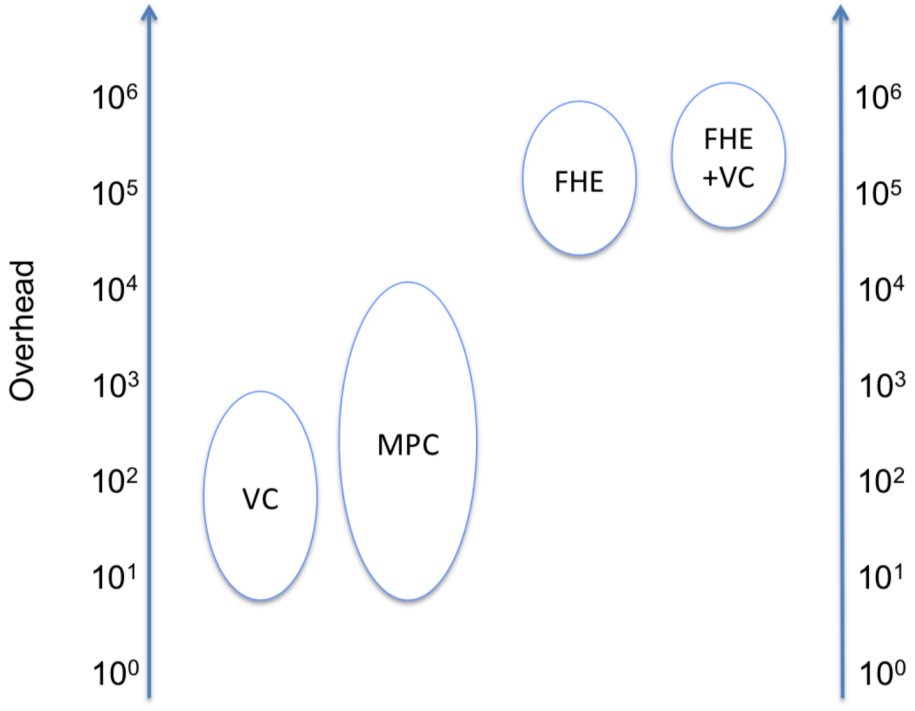
Secure Multi-Party Computation : Overview of Properties

- **Confidentiality:** yes
  - No node learns anything new, besides the result of the computation.
- **Integrity:** yes
  - Intermediate results have to be shared, which allows verification of correctness.
- **Adversary type addressed:** malicious and HBC
  - Most SMPC schemes allow $t < \frac{n}{2}$ HBC adversaries or $t < \frac{n}{3}$ malicious adversaries, but higher bounds also possible.
- **Requires interaction:** yes
  - SMPC requires 3 nodes or more.
  - Confidentiality and integrity of SMPC are enforced through interaction between nodes.

Comparison of Approaches for Private Computation

Approach	Adversary Type	Confidentiality	Integrity	Requires Interaction
Homomorphic Encryption (HE)	Malicious	YES	NO	NO
Verifiable Computation (VC)	Malicious	NO	YES	NO
Secure Multi-Party Computation (SMPC)	Honest-but-curious (HBC) or Malicious	YES	YES	YES

Comparison of Performance Overhead Incurred by Approaches



Graphical depiction of the multiplicative performance overheads over unsecured computation incurred by HE, VC and multi-party computation (MPC).

MPC and SMPC are the same.

Summary

- **Cloud computing** provides shared resources for computation
- Different cloud deployment scenarios have different **confidentiality and integrity requirements**
- **Adversaries** can be honest but curious or malicious.
- Three promising **approaches to address threats** in cloud computing are:
  - Homomorphic encryption (HE)
  - Verifiable computation (VC)
  - Secure Multi-Party Compuation (SMPC)

- Each approach addresses different **requirements**, and provides different **guarantees**
- All approaches incur significant **performance overheads**
- SMPC is the most **promising** approach with the least overhead