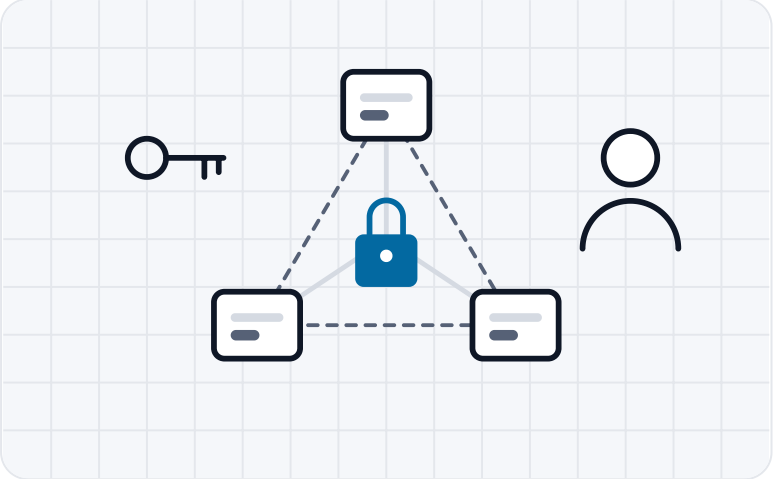
A-SIT Technology
E-Government Innovation Center

25 August 2023 Concept · Privacy & Crypto · Transparency · 13 min read

Access Control for Distributed Data Analysis using Secure Multi-party Computation

Stefan More



Authors Stefan More
Source https://technology2.a-sit.at/en/publications/access-control-for-distributed-data-analysis-with-mpc/

Key findings

- Without access control, a compromised data catalogue could trigger arbitrary computations and see results without users' consent.
- The architecture extends MPC-based data catalogues with data usage policies that restrict buyers and permitted computations, including their parameters.
- Each MPC node evaluates the policies and the buyer's credentials before decrypting its data shares and computing.
- The result shares are encrypted with the buyer's public key; only the authorised buyer can view the result.
- Policy enforcement is guaranteed as long as at least one computation node is honest; the data catalogue does not need to be trusted.

Multi-party computation (MPC) is a cryptographic method for secure distributed execution of computations. In an MPC system, multiple participants collaboratively work on a computation, such as data analysis. Each participant contributes a portion of the data, with this data remaining confidential even during the computation. Hence, no participant learns anything about the data of the other participants.

A special case of MPC applications is computation on data from data catalogues, where users provide encrypted data for an MPC system. In this process, a user's data is divided using cryptographic techniques (secret sharing) and encrypted for individual nodes of the MPC system. A user can then perform a computation on this data, while the MPC system learns nothing about the data except the result of the computation. The user providing the data must trust the system to only perform authorized computations on the data.

The objective of this project is to enhance such an MPC system with an access control mechanism, allowing users to retain control over their data. Additionally, it aims to prevent the MPC system from learning anything about the computation's result, ensuring that only authorized users can access it.

1. Introduction

Personal data is an attractive source of insights for a diverse field of research and business. While our data is highly valuable, it is often privacy-sensitive. Thus, regulations like the GDPR restrict what data can be legally published, and what a buyer may do with this sensitive data. While personal data must be protected, we can still sell some insights gathered from our data that do not hurt our privacy. The major challenge such a data catalogue faces is balancing between offering valuable insights into data while preserving privacy requirements. One approach to solve this challenge is to apply cryptographic techniques to enable privacy-preserving computations on personal data. Multi-Party Computation (MPC) is one possible technique to achieve this. MPC computations allow for calculating statistics or training machine learning models on personal data without accessing the data in plain.

However, the subject of the data cannot restrict who can buy or what type of computation the data is allowed.

Control computations on personal data In this report, we discuss a flexible access control architecture for MPC-based data catalogues, which can be applied to existing data markets. Our architecture enables data sellers (or subjects) to define detailed policies restricting who can buy their data. Furthermore, a seller can control what computation a specific buyer can purchase on the data, and make constraints on its parameters to mitigate privacy breaches. The computation system then enforces the policies before initiating a computation.

2. Background: Multi Party Communication

Multi-party computation (MPC) are distributed protocols that allow a group of parties to jointly compute a function over their inputs while keeping those inputs private [1]. Thus, no party learns anything about the inputs of the other parties beyond what can be inferred from their own input and the output of the function.

Practically efficient protocols have been demonstrated in wide range of MPC frameworks and allow MPC to be deployed in real-world scenarios and products. Especially programs with integer arithmetic can be computed highly efficiently using secret-sharing based techniques.

- In Multi-Party Computation (MPC), the computation is performed distributed on several nodes, and each node only receives a part of the user's data.
- The data is distributed in opaque shares [2] to several nodes for computation.
- Only the final assembly of all the output shares discloses the result to the computation buyer.

Practically efficient protocols have been demonstrated in wide range of MPC frameworks and allow MPC to be deployed in real-world scenarios and products. Especially programs with integer arithmetic can be computed highly efficiently using secret-sharing based techniques.

3. Background: Data Catalogues

In order to enhance the exploitation of personal data sets, available data must be efficiently brokered to relevant consumers. Data catalogues (or marketplaces) take on this brokerage task. They are an online platform that brings together the producers of personal data with relevant consumers. However, the collected personal data is highly sensitive, and legislators tend to protect it well.

Private data catalogues try to mitigate these issues by using modern privacy-enhancing technologies. These technologies enable computation on personal data without revealing the data itself. To do so, MPC can be used to preserve users' privacy. The data is distributed in opaque shares to several nodes for computation. Only the final assembly of all the output shares discloses the result to the computation buyer.

4. Distributed Access Control Concept

A MPC system that performs privacy-preserving computations on the data of a data catalogue represents the baseline for our work. In this chapter, we first describe the architecture and privacyassumptions of such a system. We also show how this design prevents users from controlling computations on their data. In the next section, we describe our solution to this problem by adding access control mechanisms to the distributed computation system.

4.1 Baseline: Architecture

In general, the considered computation system consists of the following components:

Data Subject (or Seller) : The actor who produces data and wants to offer it to other entities. To host this data, the data seller uses some public cloud storage. Some models subdivide the data seller further into separate roles, i.e., the data producer/generator, the data subject, and the data provider.

Data Buyer: The actor that wants to buy computations on the data of several data sellers. They select one or multiple data products from a catalogue and decide which computations to execute. The data buyer is sometimes referred to as data consumer.

Data Catalogue and Broker System: The online platform which acts as a broker to connect data sellers with relevant data buyers and enables the data trade. The catalogue provides a list of data products to which a data seller can add their data records. In addition, the catalogue helps the data buyer to find data products of their liking and sells the utilization of the data on its computation infrastructure. Additional tasks the catalogue offers are out of the scope of this report, e.g., payment processing.

In this report, we use a privacy-preserving computation system to perform the computation requested by a data buyer. The number of **computation nodes** N involved in this computation depends on the cryptographic technique applied by the catalogue. In Multi-Party Computation (MPC), the computation is performed distributed on several nodes (N - 1), and each node only receives a part of the user's data. Alternative techniques we did not consider in this report are Functional Encryption (FE) and full homomorphic encryption (FHE) are performed on a single node (N = 1).

Conceptually, the brokerage and computation of some data involves the following steps:

1. Data seller selects some data they would like to sell and performs a cryptographic function (i.e., secret sharing [2]) on the data. It then encrypts each "data share" for one of the nodes of the computation system.
2. Data seller uploads all data shares to the data catalogue system.
3. Data seller adds some metadata to the uploaded data to create a listing (i.e., data product) on the data catalogue system.
4. Data buyer browses the data catalogue and selects a set of data products they are interested in, and chooses a computation they would like to perform on the data (e.g., by selecting from a set of standard computations, or by defining some formula).
5. Data catalogue sends the query and data of the selected data products to all nodes of the computation system. Each node only receives the share of the data products that was encrypted to their key material, because they can only open this share. In the end, each node decrypts the shares specified for them and therefore only receives a share of the data product's data.
6. Computation system nodes collaborate with a MPC protocol to execute the ordered computation function. After finishing the computation, each node sends their share of the result to the data catalogue.
7. Data catalogue combines all shares of the computation into the final result and sends the result to the data buyer.

4.2 Baseline: Privacy-Assumptions

By using cryptographic secret sharing on the data products, each node receives only a share of the data. Additionally, by encrypting this shares for the individual node, other nodes don't have access to the share of a specific node. Additionally, the catalogue system itself has no access to any data and only sees the metadata (i.e., product listing on the catalogue).

However, the stated approach has several disadvantages.

Since the MPC computation system receives computation requests from the catalogue and executes it, a malicious (or compromised) catalogue system could initiate any computation request on the data without the users consent or even without the user learning about this request. This could, for example, we used to recover the encrypted data in plain text by requesting an identity function on the data.

As an additional problem, even the result of a legitimate computation initiated by an legitimate data buyer is send back to the data catalogue, so the data catalogue learns about the computation results. The catalogue could then, for example store it, and combine results from different computations to learn more about the data, or sell the results to other entities.

Further, the stated approach relies on the data catalogue to enforce which computation functions or other computation parameters can be applied to some data product. This means that a data seller has to rely on the catalogue to enforce their privacy expectations and requirements. This is a problem in case of a malicious marketplace, or if the data stored at the catalogue leaks and malicious actors use it to directly initiate computations.

4.3 Privacy and Security Goals

In the section above we discuss the downsides and risks of MPC based computation systems. As goal of this report we therefore try to mitigate this issues. In specific, we aim to enable users to define their own rules on who can perform computations on their data and what kind of computations. Additionally, we prevent a curious catalogue from learning about the results of a computation.

By doing so, we remove the need for trust in the data catalogue and move it to the distributed MPC computation system. In doing so, we use the same trust assumptions that already hold for the MPC computations correctness itself: if at least one of the computation nodes behaves in an honest way, the computation result is guaranteed to be correct. The same assumption is true for our access control architecture.

4.4 Adding Access Control

To achieve the stated goals, we introduce an architecture for an extension of MPC-based data catalogues. This extension features a flexible access control mechanism based on a policy system. In the resulting catalogue system, data sellers can define expressive policies to control the usage of their data. Those policies are attached to data products offered on a catalogue. When a buyer purchases a computation on some data products, they are asked to provide a set of credentials certifying their identity and other attributes. The catalogue's computation system then verifies if the credentials fulfil the policy for the selected data. Additionally, the system uses the policy to check if the (now-authenticated) buyer is qualified to execute the requested computation. Only then the system proceeds and executes the computation.

Further, the buyer's credentials are used to encrypt the result of a computation. This ensures that only the legitimate buyer can access the result.

Our design focuses on private data catalogues which allow a computation on user's data without the user's involvement (non-interactive). Thus, we don't consider systems where the user participates in computation on their data (which does not require this type of policy system).

To achieve this goal, we add a Policy Interpreter to the architecture of the computation system. The catalogue uses this policy interpreter software component to decide if a particular buyer is qualified to acquire (a computation on) some data records. As an input the interpreter takes a data usage policy defined by the seller for their data, as well as a set of credentials from the buyer, alongside some metadata about the requested computation.

The resulting architecture is visualized in Figure 1.

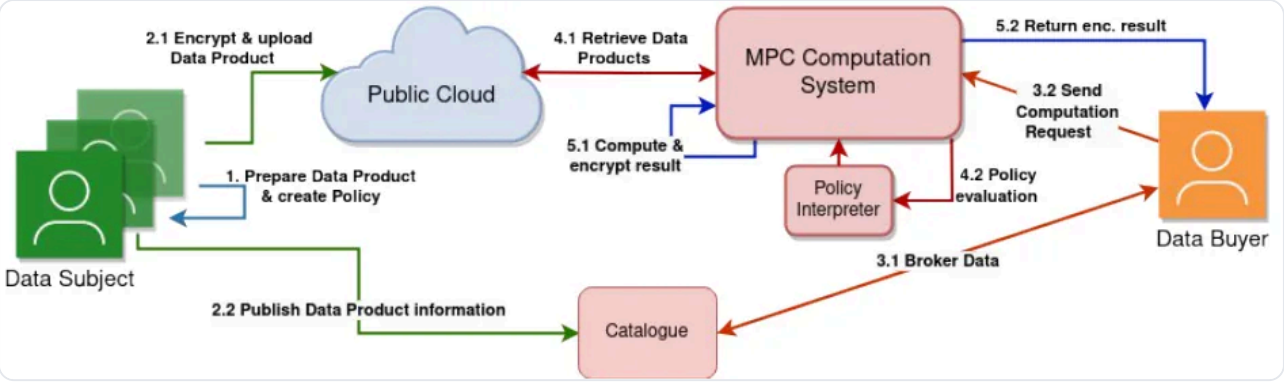


Figure 1: Architecture and data flow of a MPC computation system with policy-based access control

Our architecture extend the flow described in Section 4.1 above. In specific, we add the following steps to the data catalogue and computation system.

The data subject calculates cryptographic shares and encrypts them for each MPC node as before. In addition to the encrypted shares, the subject also appends their data usage policy to the data product and uploads both items to a public cloud (or the catalogue directly). In this policy, the data subject (acting as data seller) defines rules about their data.

On one hand, the data seller can define who can perform computations on their data. They can limit the set of data buyer to specific entity, or to a type of entity. For example, the policy can contain rules about the identity (credential) of the data buyer, or about which trust scheme (e.g., set of issuers) the seller considers trustworthy.

On the other hand, the data seller defines rules on the computation itself. For example, they can restrict which function can be performed on their data, and what parameters can be used to this function. Additionally, they can define that at least a certain amount of other products needs to be part of the computation as well. This can prevent data recovery, e.g., if a data buyer performs a computation on a single product.

After the data buyer selected some data products, the data catalogue sends a computation request to the MPC computation nodes. In addition to the computation request, the data buyer also provides a set of identity credentials to the system. The details of the computation flow are visualized in Figure 2.

Each computation node of the MPC system receives the shares encrypted for them and the usage policies for all involved data product. Before performing any computations, the computation node evaluates all involved policies. By doing so, the node checks if the buyer is eligible to perform the computation (by verifying the buyer's credentials), and if the computation query complies to the specific computation

restrictions. Only if all policies pass, the node decrypts their data shares, allows the computation and initiates the MPC service. After all nodes concludes that the policy is OK and imitated the MPC service, the MPC protocol proceeds as in the baseline case.

When the MPC process finishes, each node is in possession of a share of the computation's result. Each node then encrypts this share with the public key of the buyer (retrieved from the credentials) and sends the encrypted result back to the catalogue. By doing so, we ensure that only the (previously authorized) buyer can access the result.

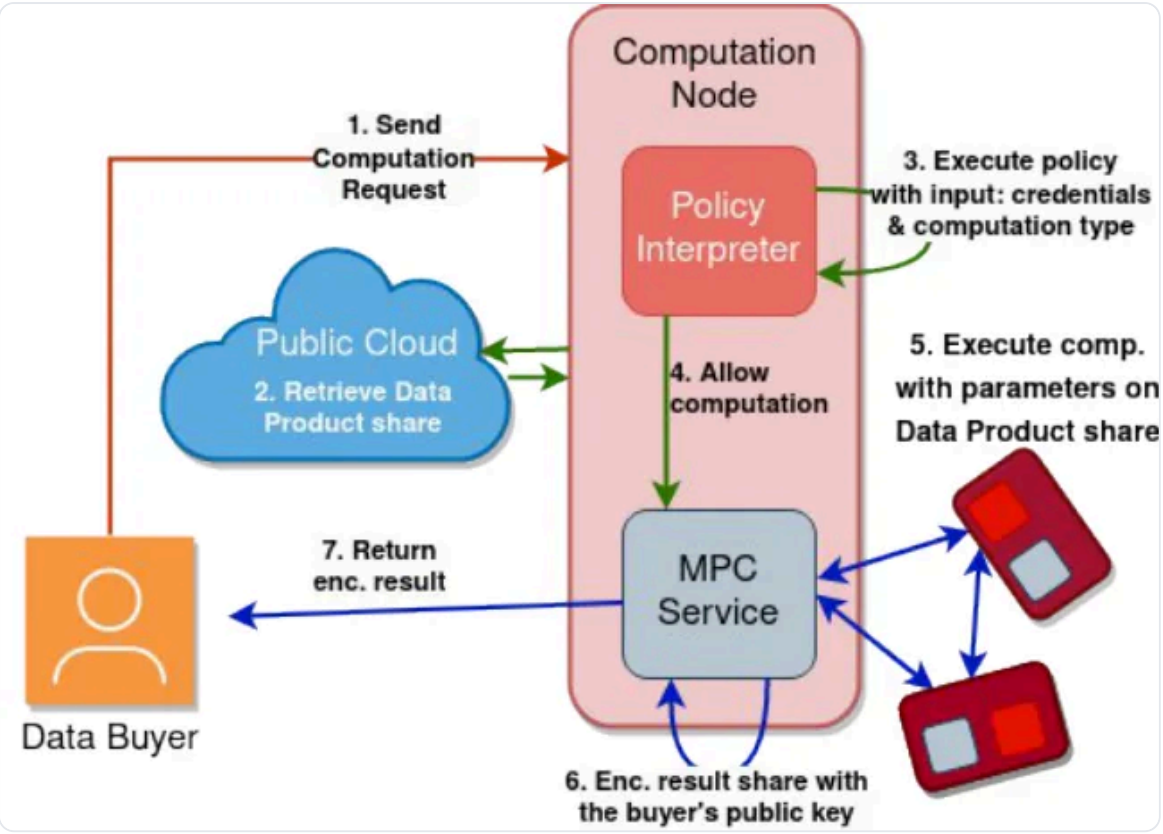


Figure 2: Architecture and data flow of a computation node

5. Conclusions

In Multi-Party Computation (MPC), the computation is performed distributed on several nodes, and each node only receives a part of the user's data. The data is distributed in opaque shares to several nodes for computation. Only the final assembly of all the output shares discloses the result to the computation buyer.

In this report, we discuss a strategy on how to extend a MPC-based data catalogue with access control features. In specific, we allow users to attach data usage policies to their data. This data usage policies are then enforced collectively by the computation nodes. By doing so, we ensure that no computation on any data is possible if at least one computation node enforces the policy in an honest way.

References

[1] Andrew Chi-Chih Yao. Protocols for Secure Computations. In 23rd FOCS. IEEE Computer Society Press, 1982.

[2] Adi Shamir. How to Share a Secret. In Communications of the Association for Computing Machinery, 1979.