How to select your privacy-preserving computation technology? A case study in the music industry

Yulu Wang, Charlotte van de Velde, Sabine Oechsner, and Jaap Gordijn

Vrije Universiteit Amsterdam, Amsterdam, the Netherlands
y.wang4@vu.nl, c.s.d.vander.velden@student.vu.nl,
s.a.oechsner@vu.nl, j.gordijn@vu.nl

Abstract. Digital Business Ecosystems (DBEs) increasingly rely on the sharing of sensitive data between stakeholders to foster collaboration. However, to restrict access to this data, traditional security mechanisms are often not sufficient. This paper investigates one such case, part of the Horizon Europe MUSIC360 project, where policymakers want to know the economic value of music at the industry level. We propose a solution design approach that systematically links scenario-specific requirements to technical features of Privacy-Preserving Computation (PPC). A proof-of-concept experiment using the Prio+ protocol demonstrates the usability of our approach by showing that the selected implementation meets both the functional and security requirements.

Keywords: Secure multi-party computation · Privacy-preserving computation · Music digital business ecosystem · Security requirement solution.

1 Introduction

A digital business ecosystem (DBE) is a system of economic actors that depend on each other for their economic well-being and survival [42]. Over the years, DBEs have become complex and decentralized, where numerous stakeholders exchange things of economic value, and to do so, share and transfer data.

Data sharing, especially in the case of valuable or privacy-sensitive data, comes with the requirement to restrict access to data to selected parties [15]. Often, data can be protected by traditional and well-known security mechanisms, like access control systems following the principles of Role-based Access Control (RBAC), Attribute-based Access Control (ABAC), or Access Control Lists (ACLs), enforced by the data owner itself. However, there exist use cases where the well-known access control solutions are not sufficient. Take for example one of the cases in the Horizon Europe MUSIC360 project, where we aim to understand the value of music better. A particular stakeholder, namely the EU policymaker, wants to know the average revenue of all EU rightholders (performers, text- & songwriters) per genre. The MUSIC360 platform collects and stores the data required to answer this question in Europe, but in a highly decentralized way by using a set of data stores, where each store is under control by the data owner (or a party that represents the data owner). The platform follows the idea that stakeholders should stay in control of their own data and decide for themselves how and with whom they want to share their data, e.g., their revenue. Rightholders are willing to contribute to answering the question of policymakers, but for privacy reasons, they do not want to share data about their individual revenue. There is also no trusted third party that could collect the data about the revenue of all right-holders, calculate the revenue, and disclose the calculated average revenue to the policymakers.

Such use cases can be solved by advanced privacy-preserving computation (PPC) techniques. However, PPC as a field is highly technical and requires a thorough understanding of cryptography and mathematics. What is lacking is a comprehensive framework to guide the selection and integration of these technologies for concrete use cases, which can be applied by practitioners. In contrast, existing PPC research focuses on the techniques themselves or narrow application domains at best, leaving practitioners without actionable guidance. A systematic approach to evaluating PPC techniques for DBEs is absent, complicating decisions for industry professionals. This paper addresses this gap by proposing a solution design approach for these hard-to-solve security scenarios and an early-stage mapping framework that bridges technical PPC features with real-world requirements, validated through a case study in the music industry.

This paper is structured as follows. Sec. 2 summarizes relevant related work. Our research approach is outlined in Sec. 3. We propose a solution design approach to deal with complex security requirements (Sec. 4), and show how that works out in a concrete case in the EU music industry (Sec. 5). In Sec. 6 we discuss the results and provide an outlook. Sec. 7 presents conclusions.

2 Related Work

Suppose participants in a DBE need to do operations (e.g., the scenario to calculate the average for a set of revenue numbers) on a data set, while ensuring the confidentiality of the participants' data [20,35,39]. Without the aid of a trusted third party, such a scenario can not be solved by traditional security approaches. However PPC techniques can satisfy the requirements of such a scenario.

Currently, PPC techniques include (1) Secure Multi-Party Computation (SMPC), (2) Homomorphic Encryption (HE), and (3) Differential Privacy (DP). SMPC has evolved to support secure collaborative computations without exposing individual inputs since Yao's work on garbled circuits [45] and the subsequent generalizations by Goldreich et al. [28]. More recent studies have focused on scaling and improving SMPC protocols by combinations of methods such as secret sharing, zero-knowledge proofs, garbled circuits, and homomorphic encryption [27, 47]. HE is a cryptographic technique to do computing operations directly on ciphertext without decrypting the data. The decrypted result is consistent with the same operation on the plaintext. According to the types of operations supported, homomorphic encryption can be divided into Partially Homomorphic Encryption (PHE), Somewhat Homomorphic Encryption (SWHE), and Fully Homomorphic Encryption (FHE) [2]. However, due to the high computational complexity, efficiency issues still limit its widespread deployment in practical applications. There are complementary approaches which reduce the impact of these issues [23]. DP offers protection for personal data in statistical databases by injecting controlled noise into the output [14]. It is often used in privacy computing in addition to other techniques such as SMPC or HE. Although DP is applied to large-scale data analysis and machine learning systems, it still involves a careful trade-off between privacy guarantees and data accuracy.

Although the literature has conducted in-depth research and comparisons on various privacy-preserving computing technologies [5, 11, 16, 26], there is still a lack of a unified and systematic approach for how to select appropriate technologies based on specific DBE scenarios and privacy computing requirements. Establishing such a mapping and decision-making approach will help narrow the gap between theory and practice, especially for non-professionals, to better design and test PPC solutions when certain security requirements arise.

3 Research Approach

We answer our research question (RQ) by Design Science Research (DSR) [22], and more specifically, a Technical Action Research (TAR) approach [43].
RQ*: How to systematically derive relevant security solutions to satisfy complex security requirements in DBEs, e.g. while keeping in control of own valuable data, even if that data is needed by others, e.g. to compute other data?*
We conduct our work with practitioners who (re)design and (re)develop the MUSIC360 platform to better understand the value of music. The simplified engineering cycle that we will follow is described below and shown in Fig. 1.

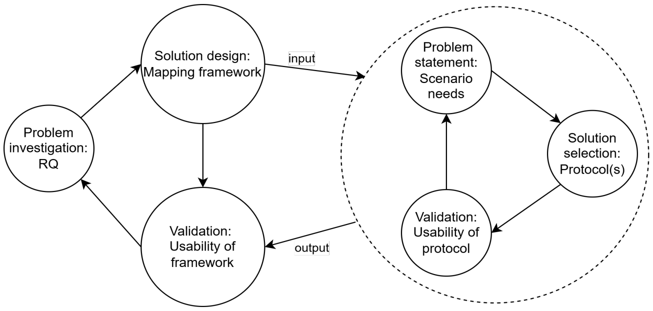

Solution design: Mapping framework
input
Problem statement: Scenario needs
Problem investigation: RQ
Solution selection: Protocol(s)
Validation: Usability of framework
output
Validation: Usability of protocol

Fig. 1. Research design cycle

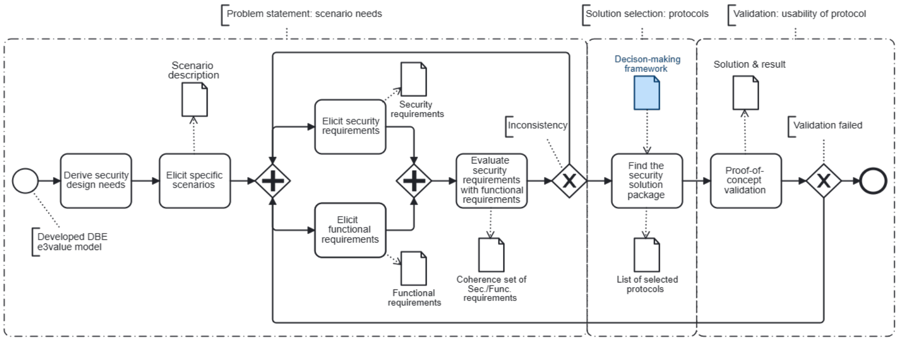
The left part of Fig. 1 shows the engineering cycle for the mapping framework, which means that we will execute the cycle a number of times; this paper reports on one such cycle. The above RQ stems from a problem investigation phase and a literature review in the fields of security requirements techniques, access controls, and PPC. The 'solution design: mapping framework' represents a framework on how to identify the security problem statement, scenarios, and security needs for a specific DBE, and how to choose the most appropriate PPC technology, and is our primary research artifact. Obviously, the question is whether the framework is useful. One of the criteria that can be used to assess usefulness is to validate whether the suggested PPC technique(s) meet the requirements. Of course, there are other criteria that must hold to conclude that your framework is useful, but in this paper we restrict ourselves to checking wether the selected PPC techniques indeed satisfy the stated security requirements. The answer is not trivial, since most PPC techniques are in an early state of development, without many cases of usage in practice. Therefore, for the selected PPC technique by our framework, we employ a prototyping approach to to develop a Mininum Viable Product (MVP) that illustrates that the selected PPC technique satifies the requirements (or not). We consider this as a (partly) validation of the framework.

The right part of Fig. 1 presents how a specific DBE, here MUSIC360, will be explored. The general steps involved in this engineering cycle are: (1) describing the problem statement, scenarios, and corresponding requirements of a particular case, (2) selecting technical PPC solutions & protocols for the identified security and functional requirements via the proposed mapping framework, and (3) validating whether the used protocols satisfy the requirements

4 Solution Design: Mapping Framework

4.1 Mapping framework process

Fig. 2 presents the solution design process to be executed, cf. the BPMN modeling language [41]. We explain the process below:



Problem statement: scenario needs
Solution selection: protocols
Validation: usability of protocol
Scenario description
Derive security design needs
Elicit specific scenarios
Developed DBE e3value model
Elicit security requirements
Security requirements
Elicit functional requirements
Functional requirements
Evaluate security requirements with functional requirements
Coherence set of Sec./Func. requirements
Inconsistency
Decison-making framework
Find the security solution package
List of selected protocols
Solution & result
Proof-of-concept validation
Validation failed

Fig. 2. Mapping framework process

Problem statement: Scenario needs. The process begins with an e^3*value* model [19] of the envisioned DBE (MUSIC360 in this paper). An e^3*value* model shows the objects of economic value that actors exchange, and because these objects are of value, they are vulnerable to attack. The e^3*value* model is not presented in this paper. Based on the value objects, we elicit scenarios with their functional and security requirements. These are analyzed for coherence and consistency, resulting in a final set of requirements, usually after a few iterations. Security requirements that can be solved by well-known techniques, such as traditional access controls, will not be further dealt with. These requirements should be satisfied for a DBE, but it is not our topic of interest.
Solution selection: Protocol(s). We then select the appropriate PPC techniques for the remaining requirements, using feature extraction of known PPC techniques. Given the maturity of the field of PPC, this extraction needs to be updated regularly by experts.
Validation: Usability of protocol. A proof-of-concept prototype implementation of the target scenario through the selected PPC technique(s) will be conducted for validation in terms of requirement satisfaction.

We emphasize that this paper is the first attempt to conduct the whole process of the proposed research design cycle in Fig. 1 and the solution design approach in Fig. 2. More engineering cycle iterations will be required to arrive at a more comprehensive and mature framework.

4.2 Requirements that PPC can satisfy

Based on the literature (e.g., [27]) and experience with security design in the MUSIC360 DBE, we present four security requirement types **[RQTs]** that PPC can satisfy. We aim to extend and organize these requirements into a taxonomy of security requirements, to allow for selection by practitioners.

- **Data privacy protection [RQT1].** Parties want to collaborate with each other without disclosing their data, while still allowing the use of that data. This is crucial when collaborating with competitors.
- **Secure peer-to-peer collaboration [RQT2].** Parties want to collaborate without the need for a trusted third party (TTP) either because the TTP does not exist or can not easily be created.
- **Distributed data ownership [RQT3].** Data is not owned by a single actor, but by multiple actors, and can only be used if all owners consent.
- **Prevention of collusion [RQT4].** A well-known approach to cheating is that two or more actors collude. A requirement can be to prevent collusion.

4.3 Features of PPC techniques.

Different PPC techniques have different sets of features that characterize the technique at hand. Understanding these features is important to match security requirements with the appropriate technique(s).

The supposed [TSM] threat and security model. One consideration in selecting a PPC technique is the threats-and-security model that defines the adversary's capabilities in the presence of certain types of attacks. *Honest* participants adhere strictly to the prescribed technique, refraining from any attempts to extract information beyond their designated input data or the intended computational output. In contrast, *adversaries* potentially compromise the integrity or confidentiality of PPC technology, e.g., by manipulating the computation to produce incorrect or inaccessible results [40]. Based on the behaviour of the adversary, security models can be divided [1,47] into (1) the *semi-honest* adversary model, which supposes that actors exactly follow the prespecified technique, but they will try to learn as much as possible from the information observed; (2) the *malicious* adversary model, where actors cannot obtain any information other than their own input or the intended output; and (3) the *covert* adversary model which balances security and efficiency by allowing adversaries to act maliciously while risking detection.

The [CM] computing model. The computing model focuses on how calculations are done using the PPC technique, how specific computing tasks are transformed into a form suitable for secure processing, and the encryption technologies and security assumptions it relies on. Different PPC approaches leverage distinct cryptographic foundations, protocols including those based on homomorphic encryption (HE), and forms of SMPC such as secret sharing (SS), oblivious transfer (OT) and garbled circuit (GC) [10, 21, 29, 33, 38, 46]. These approaches always come with trade-offs in computational complexity, communication efficiency, and suitability for various types of operations. Additionally, the communication overhead and computational complexity usually increase significantly as the level of the security model increases. (1) *HE*-based solutions provide strong security by allowing computations on encrypted data but suffer from high computational costs, making it suitable for low-interaction settings [18, 37]. (2) *SS*-based achieves efficiency in linear operations through additive secret sharing but incurs high communication overhead due to frequent interaction [3,12,13,36]. (3) *OT*-based approaches offer a balanced trade-off, with moderate communication volume and computational complexity, making them practical for scenarios requiring secure lookups or non-interactive key exchange. However, pure OT-based SMPC without extensions has limitations in performing linear operations efficiently [8, 9, 31]. (4) *GC*-based provides constant communication rounds and is highly efficient for boolean logic, but requires significant pre-processing and high communication volume. [17, 30].

The [DM] deployment model. The deployment model focuses on how the protocol is implemented, that is, how the participants are distributed, how they communicate, and whether auxiliary parties or servers are introduced. The IEEE standards association has classified the SMPC deployment model into three types [1]: (1) *Server-side* MPC: Data providers upload encrypted data to a set of non-colluding servers that perform computations. This ensures data confidentiality but relies on the trustworthiness of the servers; (2) *Peer-to-peer* MPC: Venues run MPC computations themselves, reducing trust dependencies but increasing computational costs; and (3) *Server-aided* MPC: A hybrid approach where some computation nodes are maintained by data providers, while others are external servers that assist in intensive computations like Beaver triple generation.

The [SF] supported function. The technical basis of different PPC techniques determines the scope of functions and the number of participants they can efficiently support. Any function that can be represented by a circuit can be computed in a privacy-preserving way. The efficiency of computing different classes of functions, for example in terms of computation or communication cost, depends on the technique.

5 Case Study: MUSIC360

The Horizon Europe MUSIC360 ecosystem is one example of an innovative digital business ecosystem [1], which aims at providing insights into the value of music to creatives (music performers and authors), venues (restaurants, retail shops, offices), and policymakers (EU officials, national authorities and lobbyists). The MUSIC360 ecosystem provides data about the value of music. One kind of data collected by the MUSIC360 DBE is the economic effect of music played, for example, in terms of increased revenue for the venue that plays the music. Venues (shops, bars, etc.) do so to improve the well-being of customers, create a brand identity, and eventually increase revenue.
In this section, we present the security solution design and validation for one the scenarios of the MUSIC360 DBE. In particular, we execute the right part of Fig. 1, which is further detailed in Fig. 2. The 'Requirement Types' in Sec. 4.2 and the 'Features of PPC techniques' in Sec. 4.3 of the mapping framework are the input for this case study.

5.1 Problem statement

We present how we identify a specific hard-to-solve security scenario in MUSIC360 and state the problem by detailed scenario-specific requirements.

Elicit hard-to-solve security scenarios. Following our approach as illustrated in Fig. 2, we start to elicit security-focused scenarios, after a comprehensive analysis of the MUSIC360 ecosystem in terms of an e^3*value* model to understand the economic objects, which can be under attack. Many valuable objects are data objects too, which have been further explored using UML class modelling. Both the e^3*value* DBE model and the associated class model are out of scope for this paper. The protection of most of the identified data objects can be achieved by using traditional security approaches like authentication, authorization and detailed access control policies. However, there are a few scenarios which can not be satisfied easily. One specific hard-to solve security scenario we found in the MUSIC360 ecosystem is called 'Average increased revenue'.

[1] https://www.music-360.eu

Scenario description: 'Average increased revenue'. In the MUSIC360 DBE, a policymaker would like to know the average increase in revenue of venues (located in a specific region) as a result of playing music in these venues. In order to use music, venues have to pay a fee to Collective Management Organizations (CMOs). Every country has at least one CMO, but usually there are more of them, each representing different kind of rightholders and intellectual property rights. CMOs have a mandate to collect fees for their rightholders. Revenue data of venues is confidential information, as it is competitive and sensitive data. E.g. if the CMOs can obtain the precise revenue increase number associated with music played at each venue separately, they may consider adjusting the venue's licensing fee accordingly. This could have a negative impact on venues. Therefore, only the venue itself should have access to its respective revenue increase.

Scenario requirements. For the scenario 'Average revenue of venues', the following requirements are identified:

Functional: A party (e.g., policymakers) wants to know the influence (in terms of average revenue increase for venues) of playing music at a number of venues:

1. The ability for venues to provide their data to compute the average revenue for a set of venues.
2. The ability to compute the average over the input parties' data.
3. The computation should be performed within an acceptable amount of time (minutes is ok, hours not).

Security: The average should be calculated without disclosing the individual revenue data from venues, which can be detailed as follows:

1. Policymakers as final result receivers should only have *READ* access to the final result **[RQT1]**. Venues that did provide input data should only have *READ* access to the final result and *all access* to their own input data.
2. *Input* parties deliver their input without disclosing their full and readable input to a single party. Parties interested in the *result* of the calculation obtain the complete result without obtaining the full result of a single party **[RQT1,2]**. This implies that no single party knows the input of the individual venue, except the venue itself, and also no single party knows the result, except the receiver (the policymaker).
3. Individual revenue values of venues must remain confidential and should not be inferable from the final output or any intermediate data **[RQT1]**.
4. The deployed solution must be resilient to collusion, e.g., a coalition of venues and computation servers should not be able to reconstruct venue-specific data **[RQT4]**.
5. The technology should include mechanisms to prevent malicious inputs to guarantee the correctness and effectiveness of the result, such as zero or extreme outliers that could distort the average, without requiring disclosure of actual input values.
6. There is no trusted third party who can do the calculation on behalf of the party interested in the result **[RQT2]**.

5.2 Solution selection

The requirements mentioned above cannot be solved by traditional security approaches tied to a single actor. Therefore, we analyze the 'Average increased revenue' scenario in the SMPC context.

Involved parties. For the involved parties, we distinguish *input* parties, *computation* parties and *result* parties. Input parties provide the data needed for the computation, whereas the computational party performs the computations. Result parties are the intended recipients of the computation. Fig. 3 explains how the MUSIC360 parties play the role of input, computation and result party.

1. **Venues are input parties:** Each venue keeps its own increased revenue data. This input data is secret.
2. **CMOs are computation parties:** CMOs receive and process the parts of the revenue data without being able to reconstruct individual revenues.
3. **Policymakers are result parties:** Policymakers want to get the result: the average revenue increase.

Using the mapping framework. We have derived specific requirements for the scenario 'Average increased revenue'. We classify them into **[F]** - functional requirements factor, **[S]** - security requirements factor, and **[O]** - other requirements. We derive and generalize the factors below from the evaluated refined scenario-specific requirements and map them to certain key technology features we characterized in Sec.4.3.

1. **Scenario factor: [F] Required calculation type & function.**
   - *Requirement:* Type: sum (addition), mean (division). Function:$R_m = \frac{\sum(R_i)}{n}$, $(R_i = R_i\_before - R_i\_after)^{2345}$.
   - *Reasoning:* Protocols supporting linear computation, especially an efficient addition operation, is needed.
   - *Related PPC feature:* **[SF]** Supported function: linear, addition.
2. **Scenario factor: [S] Data ownership & confidentiality in multi-party nature [RQT1].**
   - *Requirement:* Venues must retain their revenue data ownership and ensure this confidential data will not be disclosed.
   - *Reasoning:* Secret-sharing-based SMPC ensures raw revenue values are never reconstructed. Data is split into shares distributed across multiple servers, where no single server (or minority coalition) can infer private values.
   - *Related PPC feature:* **[CM]** Computing model: secret-sharing-based SMPC.
3. **Scenario factor: [S] Collusion resistance [RQT4].**
   - *Requirement:* The main collusion risk exists between servers or venues and needs to be prevented.

[2] R$_i\_before$: The revenue in a fixed period when not playing any music.

[3] R_i_after: The revenue in a fixed period while playing music.

[4] R_i: Each venue i has a confidential increased revenue data.

[5] R_m: Average (mean) of increased revenue.

  - *Reasoning:* (1) In a contract-regulated business environment, contractual obligations between venues and CMOs can deter malicious behavior (e.g., submitting false data) to some extent. Hence, at least a semi-honest adversary model will be needed. (2) The primary risk is a passive inference of individual revenues, not active attack while secret sharing inherently mitigates this by splitting data into shares. (3) A server-side SMPC model creates a separation between computation servers and venues (clients), reducing collusion incentives to some extent.
  - *Related PPC feature:* **[TSM]** Threat security model: at least a semi-honest adversary model; **[CM]** computing model: secret-sharing-based SMPC; **[DM]** Deployment model: server-side SMPC model.
4. **Scenario factor: [O] Resource-constrained.**
   - *Requirement:* Venues may lack computational resources to perform intensive cryptographic operations and communications locally.
   - *Reasoning:* (1) A server-side SMPC model offloads computation to dedicated servers. This eliminates the need for peer-to-peer coordination between venues, reducing local computational burdens and enabling the possibility of more participating venues. (2) We will prefer semi-honest rather than malicious-secure protocols to avoid incurring prohibitive overhead for frequent computations and communications.
   - *Related PPC feature:* **[DM]** Deployment model: server-side SMPC model; **[TSM]** threat security model: semi-honest adversary model.
5. **Scenario factor: [O] Efficiency in computation task**
   - *Requirement*: The computation of average increased revenue relies primarily on addition and mean operations.
   - *Reasoning*: (1) Secret-sharing protocols optimize arithmetic operations, outperforming Boolean-centric alternatives (e.g., garbled circuits) on most linear operations. (2) Server-side deployment usually minimizes latency for large-scale deployments.
   - *Related PPC feature:*: **[DM]** Deployment model: server-side SMPC model; **[CM]** Computing model: secret-sharing-based SMPC.

   For all reasoning holds that a TTP is not available (**[RTQ2]**).

In summary, our solution selection will be protocols supporting linear or addition operations, having a server-side deployment, using a secret-sharing-based approach, and under a semi-honest adversarial model.

Protocol selection. The protocol selection is based on the scenario-specific technology feature preferences we reasoned from the mapping framework. As the first attempt, due to the time limit and page limit, we evaluate six protocols that are from active open source projects or have been widely discussed in the PPC community. Table 1 below shows how well these popular protocols meet these technical features and corresponding requirements.

EasySMPC [44] and Prio+ [3] align well with all the SMPC technology feature preferences we reasoned from the mapping framework. Their open-source nature also shows their advantages in protocol selection. However, the feasibility of the experiment (even the real business analysis) also relies on the deployment

SMPC technology	SMPC technology feature preferences			
	[SF] linear, addition	**[TSM]** semi-honest	**[DM]** server-side	**[CM]** secret sharing
Prio+ [3]	●	●	●	●
Overdrive [25]	●	○	●	○
FRESCO (BGW) [7]	●	●	○	●
Mascot [24]	●	○	○	○
Whisper [32]	●	○	●	●
EasySMPC [44]	●	●	●	●

Table 1. The degree of satisfaction of different protocols for scenario-specific technical feature preferences

difficulty and ease of use. The EasySMPC method utilizes email as the main digital communication channel, which may introduce new security problems and reduce reliability. Also, the data input approach in EasySMPC is more suitable for a certain amount of data that is well structured in CSV format, the approach is too heavy for our scenario needs. The Prio+ protocol is suitable for a proof-of-concept experiment because we can set different parameters according to a specific scenario to test its usability as a basic validation, and it ticks all the boxes. Hence, the Prio+ [3] protocol was selected as a promising security solution for the scenario 'Average increased revenue'.

5.3 Validation: Usability of protocol

To evaluate the applicability and usability of SMPC protocols selected according to our proposed framework and approach based on our scenario analysis, we conducted a proof-of-concept experiment utilizing the selected protocol prototype [6]: *Prio+*. This section outlines the protocol execution procedure, presents the experimental test results, clarifies the usability, and offers reflections from the perspective of this first attempt.

Protocol execution procedure. Based on the above analysis, we listed relative results as below: (1) the selected PPC protocol: *Prio+*; (2) the required calculation formula: $R_m = \frac{\sum(R_i)}{n}$ ($R_i = R_i\_before - R_i\_after$); (3) the involved parties: input parties (venues), computation parties (CMOs), output parties (policymakers). All involved participants should acknowledge the specific computation task and the procedure of the security scenario, 'Average increased revenue'. The conceptual security model by the SMPC protocol for this scenario is shown in Fig. 3. We explain it as below:

1. *Input submission:* Each venue i has confidential revenue data: R_i and act as an input client to submit it. We assume that all venues will honestly submit their data since the selected protocol supports semi-honesty.

[6] https://github.com/KuraTheDog/Prio-plus

2. *Data validation: Prio+* enables the input verification using share conversion to make sure the provided revenue increase data are within acceptable bounds (e.g., non-negative values or not excessively large).
3. *Computation factor generation & aggregation:* Each input client generates multiple secret shares of its increased revenue data R_i. These shares are distributed to the aggregators.
4. *Computation execution:* The computing servers collect the data shares from all the involved venues and execute computing tasks on their computing nodes according to the SMPC protocol.
5. *Reconstruction & output:* The final sum result is sent back to the policymakers' clients and then reconstructed. The servers do not learn the individual inputs or the final result unless they collude. In *Prio+*, in order to compute the mean, the computation servers are allowed to modestly leak the number of involved venues and then clients can locally compute the mean.

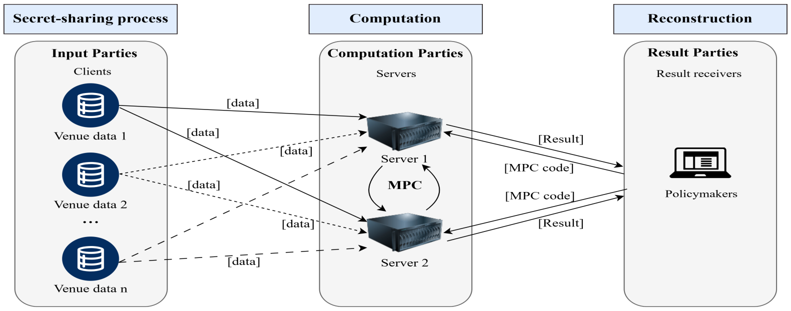


Fig. 3. SMPC model for scenario 'Average increased revenue': [data] and [Result] are secret shares divided by SMPC protocol. Different line styles are used solely for visual clarity, without implying any semantic distinction.

Experiment setting & result. For the scenario proof-of-concept computation experiment, our basic container deployment using the *Prio+* protocol via Docker is conducted through simulating two server instances and multiple clients (operates in a cluster of individual clients). We used the supported function: INT_SUM because of the required computation type and function. We tried different values in two parameters: (1) max_bits (data size can be input): $2^{12}/2^{16}$; (2) num_inputs (number of venues): 10/100/1000. These two parameter settings are related to the actual scenario nature. We look into two variables - total sent bits and total time - under the different settings of the two parameters we mentioned before. These two variables can help to measure whether the basic execution (time, data volume) of the selected protocol is within an acceptable range.

Validation discussion. Following our solution design approach and the mapping framework, the Prio+ [3] protocol was selected as a promising security

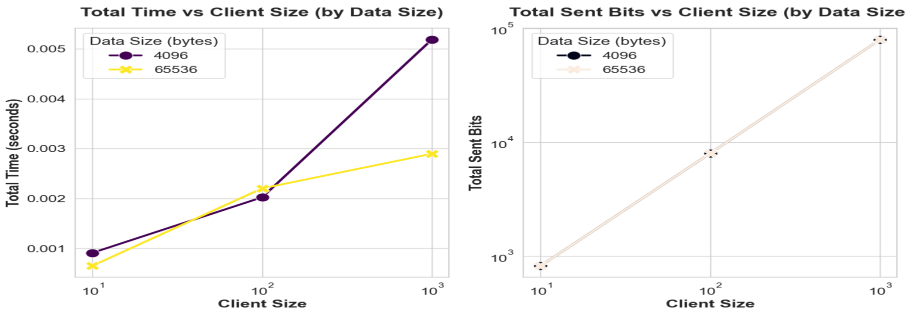


Fig. 4. Experimental results obtained using the different parameter settings

solution to satisfy the scenario's needs. Its technology features align well with all the security and functional requirements (see Sec. 5.2) and meet practical constraints. We demonstrate the matching with the requirements in Table 2. The experiment results shown in Fig. 4 suggest the usability concerning efficiency of the selected protocol in this specific 'Average increased revenue' scenario.

6 Discussion

6.1 Usability of solution design: mapping framework

The reasonable result we have obtained in this proof-of-concept experiment shows the feasibility and usability of our proposed design process and framework in finding an appropriate SMPC technology for a specific scenario. By analysing its execution, we derive critical insights into how technical features and scenario-specific requirements interact, and how these interactions can guide future experiments and framework development. The scenario 'Average revenue of venues' serves as a case study to refine our approach and inform the design of a more comprehensive and rigorous mapping framework.

Identify unsolved security scenario. For each case, how to identify and reason when and why we need advanced PPC methods as the solution for complex security scenarios will be an interesting problem. As the first step of the solution design approach in Fig. 2, it can be investigated through the taxonomy study of security requirements [6, 34], the most related issues of security requirements in DBEs can be generally divided into data breaches, cyberattacks, and privacy violations [4]. Privacy violations, especially when value analysis involves data from multiple parties, often lead to complex solution design. Identifying value objects and transfers early can support security requirement elicitation and help uncover complex security scenarios.

Practical constraints as another dimension. Through practical construction and experimentation using the early-stage mapping framework, we find that practical constraints (e.g., cost, trust level, limited resources) play a key role in choosing SMPC deployment models. These constraints often influence

Requirements Index		Matching evidence in *Prio+*
Functional	1	*Prio+* allows multiple clients' input, which can be easily set by the parameter: 'num_inputs'.
	2	*Prio+* supports the sum computation type by protocol option 'INT_SUM'. In order to compute the average, the protocol allows the servers to modestly leak the number of venues. Then clients can locally compute the mean.
	3	Our experiment result in Fig.4 confirmed this.
Security	1	*Prio+* built on 'secret-sharing' as a cryptographic tool to keep the data private.
	2	*Prio+* ensures raw revenue values are never reconstructed.
	3	In *Prio+*, data is split into shares distributed across multiple servers, where no single server (or minority coalition) can infer private values.
	4	*Prio+* follows the server-side deployment, which creates a separation between computation servers and venues (clients), reducing collusion incentives.
	5	*Prio+* enables the input verification using share conversion to make sure the provided revenue increase data are within acceptable bounds.

Table 2. The matching degree of *Prio+* to scenario requirements

the trade-off between stronger security guarantees and limited computational or communication resources.

Priority weighting between factors/features. By introducing additional assumptions to the mapping framework, priority can be assessed more effectively. For example, in certain scenarios, performance may take priority if low-latency is required (e.g., real-time analytics). One example principle can be: for resource-limited parties (e.g., small venues), prefer lightweight protocols like Prio+ over more complex, resource-intensive ones.

6.2 Limitations

The potential risks and limitations of this research are as follows: (1) Deployment complexity remains difficult to estimate and should be incorporated as an evaluation metric in the proposed framework. (2) Scenarios involving multiple concurrent privacy requirements may introduce protocol-switching needs or trade-offs, necessitating more comprehensive reasoning and compatibility considerations. (3) As the study is centered on a specific use case in the music DBE domain, broader validation is required to support generalizability across other contexts.

6.3 Future Work

SMPC technique dataset. Our experience in music DBE scenarios shows that protocol mapping and selection must carefully consider key technical features. To support this, we plan to create a structured dataset as an appendix to the mapping framework. This dataset will classify methods by features such as threat model, computing model, deployment model, support functions, and available benchmark results. It will help efficiently map to DBE requirements and support reproducible research.

More case studies. We will expand case studies to other scenarios in MUSIC360 or other DBEs, extending the scope to other privacy-preserving methods as needed. We aim to refine our solution design approach and the mapping framework, supported by continuous systematic literature review and practical experience.

7 Conclusion

This study highlights the feasibility of privacy-preserving computation technologies, especially secure multi-party computation, as a security solution for digital business ecosystems. By analyzing the MUSIC360 case, we show how to find suitable SMPC technologies for specific security scenarios to achieve secure aggregation of sensitive data (e.g., venue revenue) while meeting security and functional requirements and practical constraints. The proposed mapping framework successfully links scenario-specific requirements factors (e.g., computation type, anti-collusion, resource constraints) with the technical features of SMPC (e.g., secret sharing and semi-honest adversarial model). Experimental verification confirms the usability of the highly satisfied protocols found according to the mapping framework in specific DBE security scenarios.

This research emphasizes that SMPC's strength lies in its ability to keep privacy in a limited mutual trust environment without sacrificing data utility. It is a critical advantage for DBEs where stakeholders must collaborate without exposing proprietary information. By combining the theoretical capabilities of SMPC with the real-world DBE needs, this work advances the application of PPC techniques in multi-party settings. It provides practitioners with a foundational approach to systematically evaluate and implement SMPC.

Acknowledgements

Funded by the European Union under project no. 101094872. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

References

1. Ieee recommended practice for secure multi-party computation. IEEE Std 2842-2021 pp. 1–30 (2021). https://doi.org/10.1109/IEEESTD.2021.9604029
2. Acar, A., Aksu, H., Uluagac, A.S., Conti, M.: A survey on homomorphic encryption schemes: Theory and implementation. ACM Computing Surveys (Csur) **51**(4), 1–35 (2018)
3. Addanki, S., Garbe, K., Jaffe, E., Ostrovsky, R., Polychroniadou, A.: Prio+: Privacy preserving aggregate statistics via boolean shares. In: International Conference on Security and Cryptography for Networks. pp. 516–539. Springer (2022)
4. Aksoy, C.: Digital business ecosystems: An environment of collaboration, innovation, and value creation in the digital age. Journal of Business and Trade **4**(2), 156–180 (2023)
5. Almagrabi, A.O., Bashir, A.K.: A classification-based privacy-preserving decision-making for secure data sharing in internet of things assisted applications. Digital Communications and Networks **8**(4), 436–445 (2022)
6. Alqassem, I., Svetinovic, D.: A taxonomy of security and privacy requirements for the internet of things (iot). In: 2014 IEEE International Conference on Industrial Engineering and Engineering Management. pp. 1244–1248. IEEE (2014)
7. Amini, R., Fesq, L., Mackey, R., Mirza, F., Rasmussen, R., Troesch, M., Kolcio, K.: Fresco: A framework for spacecraft systems autonomy. In: 2021 IEEE Aerospace Conference (50100). pp. 1–18. IEEE (2021)
8. Asharov, G., Lindell, Y., Schneider, T., Zohner, M.: More efficient oblivious transfer extensions with security for malicious adversaries. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 673–701. Springer (2015)
9. Asharov, G., Lindell, Y., Schneider, T., Zohner, M.: More efficient oblivious transfer extensions. Journal of Cryptology **30**, 805–858 (2017)
10. Bian, S., Jiang, W., Sato, T.: Privacy-preserving medical image segmentation via hybrid trusted execution environment. In: 2021 58th ACM/IEEE Design Automation Conference (DAC). pp. 1347–1350. IEEE (2021)
11. Boulemtafes, A., Derhab, A., Challal, Y.: A review of privacy-preserving techniques for deep learning. Neurocomputing **384**, 21–45 (2020)
12. Corrigan-Gibbs, H., Boneh, D.: Prio: Private, robust, and scalable computation of aggregate statistics. In: 14th USENIX symposium on networked systems design and implementation (NSDI 17). pp. 259–282 (2017)
13. Cramer, R., Damgård, I., Maurer, U.: General secure multi-party computation from any linear secret-sharing scheme. In: International Conference on the Theory and Applications of Cryptographic Techniques. pp. 316–334. Springer (2000)
14. Dwork, C.: Differential privacy. In: International colloquium on automata, languages, and programming. pp. 1–12. Springer (2006)
15. Fabian, B., Gürses, S., Heisel, M., Santen, T., Schmidt, H.: A comparison of security requirements engineering methods. Requirements engineering **15**, 7–40 (2010)
16. Feng, J., Yang, L.T., Gati, N.J., Xie, X., Gavuna, B.S.: Privacy-preserving computation in cyber-physical-social systems: A survey of the state-of-the-art and perspectives. Information Sciences **527**, 341–355 (2020)
17. Gascón, A., Schoppmann, P., Balle, B., Raykova, M., Doerner, J., Zahur, S., Evans, D.: Privacy-preserving distributed linear regression on high-dimensional data. Cryptology ePrint Archive (2016)

18. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Proceedings of the forty-first annual ACM symposium on Theory of computing. pp. 169–178 (2009)
19. Gordijn, J., Wieringa, R.: E3value User Guide - Designing Your Ecosystem in a Digital World. The Value Engineers, 1st edn. (2021)
20. Gordijn, J., Wieringa, R.: The business model of digital ecosystems: Why and how you should do it. In: Guerreiro, C.G.M.J.S. (ed.) Advances in Enterprise Engineering XVI. Lecture Notes in Business Information Processing, Springer-Verlag, Germany (2023)
21. HAN Wei-Li, SONG Lu-shan, R.W.q.L.G.p.W.Z.x.: Secure multi-party learning: From secure computation to secure learning. CHINESE JOURNAL OF COMPUTERS **46**(7), 1494–1512 (2023)
22. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MIS Q. **28**(1), 75–105 (Mar 2004)
23. Jin, W., Yao, Y., Han, S., Gu, J., Joe-Wong, C., Ravi, S., Avestimehr, S., He, C.: Fedml-he: An efficient homomorphic-encryption-based privacy-preserving federated learning system. arXiv preprint arXiv:2303.10837 (2023)
24. Keller, M., Orsini, E., Scholl, P.: Mascot: faster malicious arithmetic secure computation with oblivious transfer. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. pp. 830–842 (2016)
25. Keller, M., Pastro, V., Rotaru, D.: Overdrive: Making spdz great again. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 158–189. Springer (2018)
26. Lindell, Y.: Secure multiparty computation for privacy preserving data mining. In: Encyclopedia of Data Warehousing and Mining, pp. 1005–1009. IGI global (2005)
27. Lindell, Y.: Secure multiparty computation. Communications of the ACM **64**(1), 86–96 (2020)
28. Micali, S., Goldreich, O., Wigderson, A.: How to play any mental game. In: Proceedings of the Nineteenth ACM Symp. on Theory of Computing, STOC. pp. 218–229. ACM New York (1987)
29. Mohassel, P., Rindal, P.: Aby3: A mixed protocol framework for machine learning. In: Proceedings of the 2018 ACM SIGSAC conference on computer and communications security. pp. 35–52 (2018)
30. Mohassel, P., Rosulek, M., Zhang, Y.: Fast and secure three-party computation: The garbled circuit approach. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. pp. 591–602 (2015)
31. Naor, M., Pinkas, B.: Efficient oblivious transfer protocols. In: SODA. vol. 1, pp. 448–457 (2001)
32. Rathee, M., Zhang, Y., Corrigan-Gibbs, H., Popa, R.A.: Private analytics via streaming, sketching, and silently verifiable proofs. In: 2024 IEEE Symposium on Security and Privacy (SP). pp. 3072–3090. IEEE (2024)
33. Riazi, M., Weinert, C., Tkachenko, O., Songhori, E., Schneider, T., Chameleon, F.: A hybrid secure computation framework for machine learning applications. In: Proceedings of the ACM Asia Conference on Computer and Communications Security (AsiaCCS) (2017)
34. Rjaibi, N., Rabai, L.B.A.: Developing a novel holistic taxonomy of security requirements. Procedia Computer Science **62**, 213–220 (2015)
35. Senyo, P.K., Liu, K., Effah, J.: Digital business ecosystem: Literature review and a framework for future research. International journal of information management **47**, 52–64 (2019)

36. Shamir, A.: How to share a secret. Communications of the ACM **22**(11), 612–613 (1979)
37. Smart, N.P., Vercauteren, F.: Fully homomorphic encryption with relatively small key and ciphertext sizes. In: International Workshop on Public Key Cryptography. pp. 420–443. Springer (2010)
38. Song, L., Lin, G., Wang, J., Wu, H., Ruan, W., Han, W.: Sok: Training machine learning models over multiple sources with privacy preservation. arXiv preprint arXiv:2012.03386 (2020)
39. Suuronen, S., Ukko, J., Eskola, R., Semken, R.S., Rantanen, H.: A systematic literature review for digital business ecosystems in the manufacturing industry: Prerequisites, challenges, and benefits. CIRP Journal of Manufacturing Science and Technology **37**, 414–426 (2022)
40. Wang, Z., Cheung, S.C.S., Luo, Y.: Information-theoretic secure multi-party computation with collusion deterrence. IEEE Transactions on Information Forensics and Security **12**(4), 980–995 (2016)
41. White, S.A., Miers, D.: BPMN modeling and reference guide: understanding and using BPMN. Future Strategies Inc. (2008)
42. Wieringa, R., Gordijn, J.: Digital Business Ecosystems. TVE Press, Soest, The Netherlands (2023), https://www.thevalueengineers.nl/digital-business-ecosystems-book?page=digital-business-ecosystems-book
43. Wieringa, R., Moralı, A.: Technical action research as a validation method in information systems design science. In: International Conference on Design Science Research in Information Systems. pp. 220–238. Springer (2012)
44. Wirth, F.N., Kussel, T., Müller, A., Hamacher, K., Prasser, F.: Easysmpc: a simple but powerful no-code tool for practical secure multiparty computation. BMC bioinformatics **23**(1), 531 (2022)
45. Yao, A.C.C.: How to generate and exchange secrets. In: 27th annual symposium on foundations of computer science (Sfcs 1986). pp. 162–167. IEEE (1986)
46. Zhang, Q., Xin, C., Wu, H.: Privacy-preserving deep learning based on multiparty secure computation: A survey. IEEE Internet of Things Journal **8**(13), 10412–10429 (2021)
47. Zhao, C., Zhao, S., Zhao, M., Chen, Z., Gao, C.Z., Li, H., Tan, Y.a.: Secure multiparty computation: theory, practice and applications. Information Sciences **476**, 357–372 (2019)