Privacy-preserving computing in the music industry

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

Vrije Universiteit 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.

Key words: secure multi-party computation, privacy-preserving computation, music industry, requirement-solution matching

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 [44]. 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 [13]. Often, data can be protected by traditional security practices, like various ways of access control. However, there exist use cases where these well-known solutions are not sufficient. One of these cases is discussed in the paper, namely the need to calculate the average revenue of a set of companies, without disclosing the revenue of each company to the party(ies) that calculate(s) the average revenue.

In the Horizon Europe MUSIC360 project, we aim to understand the value of music. For example, an EU policymaker wants to know the average revenue of all EU rightholders (performers, lyricists, and composers) by genre. The MUSIC360 platform collects this data across Europe and stores it in a decentralized way, controlled by the data owner or their representative. The platform is designed such that stakeholders keep control of their own data and decide if and how to disclose their revenue information. Rightholders are willing to help answer policymakers' questions but, for privacy reasons, they do not want to share the individual revenue data. Also, there is no trusted third party who can collect all rightholders' revenue data, calculate the average, and share it with policymakers.

Such use cases can be solved by advanced privacy-preserving computation (PPC) techniques, such as Secure Multi Party Computing (SMPC), Homomorphic Encryption (HE), or Differential Privacy (DP). However, PPC is highly technical and requires a thorough understanding of cryptography and mathematics. What is lacking is a comprehensive approach 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. Also, PPC technologies such as secure multi-party computation (SMPC) are often costly and complex. This raises another core question: under what security requirement scenarios is their use truly justified? To answer this, our work also integrates a DBE security requirement-driven view into the solution design procedure.

Our research question, therefore is: *'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?'*. This paper addresses this question by an early-stage solution design procedure, and a decision-making framework that identifies hard-to-solve security scenarios and 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 procedure 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 reflections. Sec. 7 presents conclusions.

2 Related Work

Suppose participants in a DBE need to perform operations such as computing the average of revenue data of all parties while ensuring that individual data remains confidential [18, 35, 40]. Without a trusted third party, such scenarios cannot be handled by traditional security methods. However, privacy-preserving computation (PPC) techniques can address this challenge.

PPC techniques commonly include (1) Secure Multi-Party Computation (SMPC), (2) Homomorphic Encryption (HE), and (3) Differential Privacy (DP). SMPC enables secure collaborative computation without exposing individual inputs, starting from Yao's garbled circuits [47] and its generalizations [29], with further developments combining secret sharing, zero-knowledge proofs, and HE [27, 49]. HE supports operations on encrypted data and ensures that decrypted results match those in plaintext, but its high computational cost limits scalability [2,22]. DP protects privacy by adding noise to statistical outputs [12], and is often used alongside SMPC or HE. While effective for large-scale data analysis, DP involves a trade-off between utility and privacy guarantees.

Although prior work has compared these techniques in depth from a technical perspective [4, 9, 14, 26], to the best of our knowledge, there remains a lack of systematic methods to help practitioners select appropriate PPC techniques based on specific DBE scenarios and security requirements. A well-structured mapping or decision-making framework is needed to support practitioners particularly non-experts—in designing effective and practical PPC-based solutions.

3 Research Approach

We answer our research question (RQ) by Design Science Research (DSR) [20], and more specifically, a Technical Action Research (TAR) approach [45].
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. These practitioners are experts on intellectual property clearing, including the supporting processes. They are not knowledgeable on cryptography. The simplified engineering cycle that we will follow is described below and shown in Fig. 1.

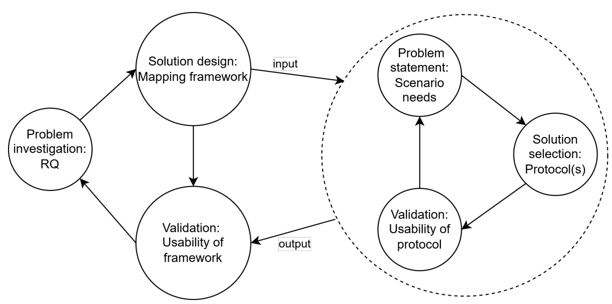

Fig. 1. Research design cycle

The left part of Fig. 1 shows the engineering cycle for the solution design process, meaning the cycle will be repeated several times. This paper reports on one such cycle. The research question comes from problem investigation and a literature review about security needs and privacy-preserving computation (PPC). The 'solution design: mapping framework' describes how to justify advanced PPC technologies in a specific DBE, identify the related security problem, and scenarios, and choose the best PPC technology. The key question is whether this procedure is useful. One way to assess usefulness is to check if the suggested PPC techniques at least meet the requirements. There are other criteria as well, but this paper focuses on verifying if the selected PPC techniques satisfy the stated security requirements. This is not straightforward, since most PPC techniques are still in early development and lack many practical use cases. Therefore, for the selected PPC technique, we use a prototyping approach to develop a Minimum Viable Product (MVP) to show whether the technique meets the requirements. We consider this as a partial validation of the whole procedure.

The right part of Fig. 1 presents how a specific DBE, here MUSIC360, will be explored. The steps are: (1) describing the problem statement, scenarios, and requirements, (2) selecting technical PPC solutions & protocols for the identified security- and functional requirements via the proposed decision-making framework, and (3) validating whether the used protocols satisfy the requirements.

4 Solution Design: Mapping Framework

4.1 Solution design procedure

Fig. 2 presents the solution design procedure to be executed, cf. the BPMN modeling language [43]. We explain the procedure below:

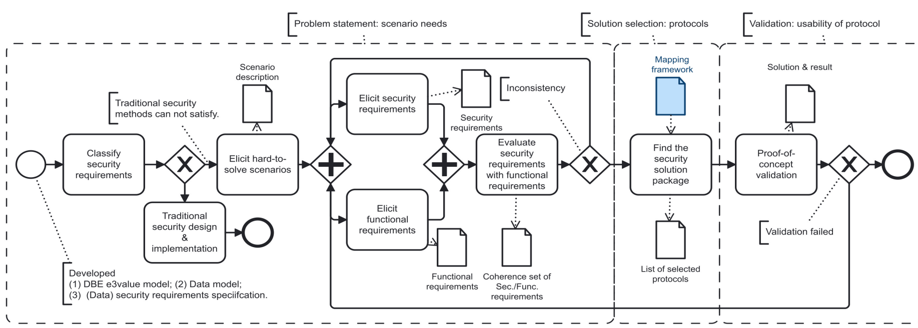

Fig. 2. Solution design procedure

Problem statement: Scenario needs. The solution design procedure starts with requirements elicitation. For DBEs, we develop a value-oriented model of the DBE at hand, cf. the e^3*value* modeling language [17]. We do so because e^3*value* shows the valuable objects that the actors in the ecosystem exchange with each other, which may subject to attack. Therefore, it is important to know these objects. From a security point of view, it is important to know the data objects associated with e^3*value* objects, because it is likely that these data objects are under attack too. And these data objects may be privacy sensitive, which results in an additional attack vector. Therefore, we also develop a UML class model to represent the structure of the data. Finally, we elicit security requirements directly related to the data in a series of workshops. These three requirement elicitation steps are reported in [41] in detail, but are not discussed in this paper due to lack of space. Based on these artefacts, we: (1) introduce an assessment step to cluster security requirements and identify hard-to-solve security scenarios. Security requirements that can easily be solved with standard security technology, such as access controls, are taken out of the process and addressed in their own right. This is outside the scope of this paper. This step acknowledges that not all requirements can be addressed by traditional security methods. It helps us to justify the necessity of applying expensive and complex PPC technologies in DBEs; (2) elicit these specific security 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.

Solution selection: Protocol(s). We then select the PPC techniques for implementing these security scenarios, using our decision-making framework (marked blue): consisting of *feature* extraction of known PPC techniques and *requirement factors*. Ideally, there would be a mapping from requirements factors onto PPC features beforehand. This mapping should be created by PPC experts, can be reused, and given the state-of-the-art of PPC techniques, should be updated regularly. In this paper, we perform the mapping ourselves.

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 iteration to conduct the whole procedure of the proposed research design cycle in Fig. 1 and the solution design procedure in Fig. 2. More engineering cycle iterations will be required to arrive at a more comprehensive and mature framework. In the following subsections, we explain some of the core steps illustrated in Fig. 2.

4.2 Security requirement analysis for DBE

DBE-specific security frames and requirement types. In DBEs, security requirements vary widely in complexity and urgency. Most of these requirements can be addressed by well-known technology. But sometimes there are hard-to-satify requirements. Below, we identify a number of these requirement types, as security problem frames, loosely based on the idea of problem framing [21]. Our problem frames serve not as an exhaustive list of all security concerns, but as a perspective for identifying when DBE-specific demand crosses the threshold into the domain of PPC technologies, and thus can be seen as a theory of complex security problems. Below are six representative security frames **[Sec-frame]** and their related requirement types **[RQTs]** observed in our case and validated through prior DBE literature [25, 27, 35, 39].

1. **[Sec-frame 1] Secure multi-party data collaboration**
   - **[RQT1]** Confidential multi-stakeholder computation: Stakeholders can engage in joint computations without disclosing raw input values to each other, preserving the confidentiality of sensitive data.

- **[RQT2]** Trust-limited collaboration: The computation framework must avoid reliance on centralized or trusted third parties, ensuring decentralized execution without exposing data.
- **[RQT3]** Decentralized data ownership: The security design must ensure that data can only be included in computations when explicitly authorized by each independent stakeholder, reinforcing legal compliance.

2. **[Sec-frame 2] Privacy-preserving aggregation and results**
   - **[RQT4]** Protection against aggregation leakage: When publishing aggregate statistics, the security design must ensure that individual contributions cannot be inferred, even with auxiliary external knowledge.
   - **[RQT5]** Controlled exposure of computation results: Published results must be processed to prevent reverse-engineering of private inputs.
3. **[Sec-frame 3] Encrypted data processing**
   - **[RQT6]** Computation on processed inputs: The security design must support secure computation over processed (e.g., secret-shared, encrypted) data during processing, ensuring data protected end-to-end.
   - **[RQT7]** Persistent encryption across lifecycle: Sensitive data must remain encrypted during storage, transit, and use, avoiding plaintext exposure.
4. **[Sec-frame 4] Collusion-resistant computation**
   - **[RQT8]** Collusion-resistant protocol design: The security design must prevent coalitions of dishonest participants from inferring private data or manipulating computation outcomes.
   - **[RQT9]** Resilience to insider threats: Computation must remain secure even when some participants are compromised or act maliciously, without degrading global guarantees.
5. **[Sec-frame 5] Verifiability without disclosure**
   - **[RQT10]** Input/result verifiability without data disclosure: Security design should provide verification for the precise of computation results without gaining access to intermediate states or private inputs.
6. **[Sec-frame 6] Dynamic and resource-constrained security**
   - **[RQT11]** Resource-constrained environments: The security design must remain functional on devices or infrastructures with limited resources, using lightweight cryptographic techniques where necessary.
   - **[RQT12]** Compliance with evolving policies: The security design must adapt to varying or evolving privacy regulations across areas.

Assessing when PPC application is justified. The security frames and security requirements types above often appear in interdependent forms. For example, [RQT1] - Confidential multi-stakeholder computation coupled with the [RQT2] - Trust-limited collaboration makes PPC not only beneficial but necessary. When one or more of these compound conditions are observed, conventional methods such as Role Based Access Controls (RBACs), encrypted data-at-rest, or trusted logging infrastructures are insufficient. Our solution design procedure starts from leveraging this logic to assess when and why PPC (particularly SMPC) should be considered during security design. In general, the emergence of these independent/compound requirement types forms the decision boundary that justifies the adoption of advanced privacy-preserving computation.

4.3 Decision-making framework

We construct the decision-making framework (marked blue in Fig. 2) by two main components: (1) Security frames **[Sec-frame]** and their related requirement types **[RQTs]**, which can be linked to scenario-specific requirements items for reasoning; (2) PPC techniques features, which we explain below.

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

1. **The supposed *[TSM]* threat and security model:** One feature is the threats-and-security model, which defines adversary behavior during attacks. **Honest** parties follow the protocol and only access their own inputs and outputs. In contrast, **adversaries** attempt to break security or correctness, such as manipulating computation results [42]. They are typically classified as follows [1, 49]: (1) **Semi-honest**: follow the protocol but try to infer additional information; (2) **Malicious**: behave arbitrarily to compromise security; (3) **Covert**: may misbehave if they believe they won't be caught.
2. **The *[CM]* computing model:** The computing model describes how PPC techniques perform secure computations, including task transformation, encryption methods, and security assumptions [8, 19, 30, 34, 38, 48]. Different approaches offer trade-offs in computational and communication cost, and support different operations. **(1) HE-based** approaches enable computation on encrypted data with strong security, but incur high computational overhead, suitable for low-interaction settings [16, 37]. **(2) SS-based** methods efficiently support linear operations using additive secret sharing, but require frequent communication [3, 10, 11, 36]. **(3) OT-based** techniques balance communication and computation, ideal for secure lookups or key exchange, but less efficient for arithmetic operations [6, 7, 32]. **(4) GC-based** methods use constant communication rounds and suit boolean logic, but demand heavy pre-processing and bandwidth [15, 31]. [15, 31].
3. **The *[DM]* deployment model:** This defines how SMPC protocols are implemented, including participant roles and communication setup [1]. (1) **Server-side**: Data is distributed among multiple non-colluding servers for computation in a way that conceals the data from each server individually. This approach is efficient but still relies on trusting that the servers do not collude. (2) **Peer-to-peer**: Participants compute directly, reducing trust but increasing cost. (3) **Server-aided**: A hybrid model where some nodes are controlled by data providers, and others are external servers that help particularly heavy parts of the computation.

4. **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[1] is an innovative digital business ecosystem that aims to provide insights into the value of music for creatives (performers and authors), venues (restaurants, shops, offices), and policymakers (EU officials, national authorities, and lobbyists). The MUSIC360 ecosystem collects data about the value of music. One type of data collected is the economic impact of music, such as the increase in revenue for a venue that plays music. Venues like shops and bars play music to improve customer well-being, build their brand identity, and ultimately boost their revenue.
In the following, we use a scenario from the MUSIC360 DBE—'Average increased revenue'— to show one iteration of the security solution design and validation. In particular, we execute the right part of Fig. 1, which is further detailed in Fig. 2. The decision-making framework in Sec. 4.3 is 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.

Identify and elicit specific security scenarios. Following our procedure as illustrated in Fig. 2, we start from the MUSIC360 e^3*value* model, UML data model (based on [28]) and (data) security requirements specifications resulting from a series of workshop sessions. During lack of space, this step is not presented here. During these sessions, we have identified a number of hard-to-solve security requirements, one of them the secure calculation of the average increased revenue, further referred to as 'Average increased revenue'.

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 kinds of rightholders and intellectual property rights. CMOs have a mandate to collect fees for their rights-holders. Revenue data of venues is confidential information, as it is competitive and sensitive. E.g. if the CMOs can obtain the precise revenue increase number associated with

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

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. **[RQT1]**
2. The ability to compute average over the input parties' data.**[RQT1, RQT2]**
3. The computation should be performed within an acceptable amount of time (seconds preferred, minutes acceptable, hours not). **[RQT11]**
4. The underlying PPC protocol used for the computations should be open source for the feasibility of the proof-of-concept experiment.
5. Complex privacy computational burdens should not be placed on venues due to typically limited computational resources. **[RQT11]**

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

1. Policymakers as result receivers should only have *READ* access to the final result. Venues that did provide input data should only have *READ* access to the final result and *all access* to their own input data. **[RQT5, RQT10]**
2. *Venues* deliver their input without disclosing their full and readable input to a party. Parties interested in the *result* of the calculation obtain the calculated average result without obtaining the individual result or each party. 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 receivers (the policymaker and possibly venues). **[RQT1, RQT2, RQT4]**
3. Individual revenue values of venues must remain confidential and should not be inferable from the final output or any intermediate data. **[RQT4, RQT5]**
4. The solution must be resilient to collusion, e.g., a coalition of venues and CMOs should not be able to reconstruct venue-specific data. **[RQT8,RQT9]**
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. **[RQT10]**
6. There is no trusted third party who can do the calculation on behalf of the party interested in the result. **[RQT2]**

In summary, this security scenario needs to introduce the advanced PPC technologies as it involves multiple RQTs we identified.

5.2 Solution selection

Involved parties. We distinguish *input* parties, *computation* parties and *result* parties. Fig. 3 shows how the MUSIC360 parties play their role. (1) Venues as input parties: Each venue keeps its own increased revenue data (secret). (2) Servers as computation parties: Servers receive and process the parts of the revenue data without being able to reconstruct individual revenues. (3) Policymakers as result parties: Policymakers want to get the average revenue increase.

Using the decision-making framework. We apply the decision-making framework (in Sec.4.3) to the security scenario 'Average increased revenue' analysis results. We linked the requirement types **[RQTs]** to the evaluated refined scenario-specific factors (by combining strong-related scenario requirement items) and mapped them to certain key PPC technology features via reasoning.

1. **Scenario factor: Required calculation type & function.**
   - *Requirement:* Type: sum (addition), mean (division). Function:$R_m = \frac{\sum(R_i)}{n}$, $(R_i = R_i\_before - R_i\_after)^2$.
   - *Reasoning:* Protocols supporting linear computation, especially an efficient addition operation, is needed.
   - *Related PPC feature:* **[SF]**: linear, addition.
2. **Scenario factor: Data ownership & confidentiality in multi-party nature [RQT1, RQT2, RQT3, RQT4, RQT5].**
   - *Requirement:* Venues must retain their revenue data ownership and ensure this confidential data will not be disclosed among computations, holds the fact that a TTP is not available.
   - *Reasoning:* Secret-sharing-based SMPC ensures raw data 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]**: secret-sharing-based.
3. **Scenario factor: [S] Collusion resistance [RQT8, RQT9].**
   - *Requirement:* The main collusion risk exists between servers or venues and needs to be prevented.
   - *Reasoning:*(1) Contracts between venues and CMOs help prevent dishonest behavior like faking data, so a semi-honest adversary model is appropriate. (2) The main risk is others inferring individual revenue, not active attacks. Secret sharing reduces this risk by splitting the data. (3) Using server-side SMPC separates venues from the servers doing the computation, which helps reduce chances of collusion.
   - *Related PPC feature:* **[TSM]**: at least semi-honest; **[CM]**: secret-sharing-based; **[DM]** Deployment model: server-side.
4. **Scenario factor: Resource-constrained [RQT11].**

[2] R_i_before: The revenue in a fixed period when not playing any music; R_i_after: The revenue in a fixed period while playing music;R_i: Each venue i has confidential increased revenue data; R_m: Average (mean) of increased revenue.

  - *Requirement:* Venues may lack computational resources to perform intensive cryptographic operations and communications locally.
  - *Reasoning:* (1) Server-side SMPC shifts the computation to dedicated servers, avoiding peer-to-peer coordination. This lowers load on venues and allows more participants. (2) We choose semi-honest over malicious ones to avoid the high cost of frequent computation and communication.
  - *Related PPC feature:* **[DM]**: server-side; **[TSM]**: semi-honest.
5. **Scenario factor: Efficiency/effectiveness in computation task [RQT10].**
  - *Requirement*: The computation should be performed within an acceptable amount of time (minutes is ok, hours not). Mechanisms to prevent malicious inputs to guarantee the correctness and effectiveness of the result without requiring disclosure of actual input values.
  - *Reasoning*: (1) Secret-sharing protocols outperform boolean-centric alternatives (e.g., GC) on most linear operations. (2) Server-side deployment usually minimizes latency for large-scale deployments.
  - *Related PPC feature:*: **[DM]**: server-side; **[CM]**: secret-sharing-based.

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 process. 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 [46] and Prio+ [3] match well with the SMPC features identified in our mapping. Their open-source nature is also a plus. However, practical use depends on how easy they are to deploy and operate. EasySMPC uses email for communication, which may raise security concerns and reduce reliability. Its data input method (CSV files) works for small, structured data but is too heavy for our needs. Prio+ is more suitable for a proof-of-concept. It allows scenario-specific settings, supports basic validation, and meets all our requirements. Therefore, we chose Prio+ as the security solution for the "average increased revenue" scenario.

5.3 Validation: Usability of protocol

To evaluate the applicability and usability of SMPC protocols selected according to our proposed decision-making framework, we conducted a proof-of-concept experiment utilizing the selected protocol prototype [3]: *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

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

SMPC technology	SMPC technology feature preferences			
	[SF] linear, addition	**[TSM]** semi-honest	**[DM]** server-side	**[CM]** secret sharing
Prio+ [3]	●	●	●	●
Overdrive [24]	●	○	●	○
FRESCO (BGW) [5]	●	●	○	●
Mascot [23]	●	○	○	○
Whisper [33]	●	○	●	●
EasySMPC [46]	●	●	●	●

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

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 submits its confidential revenue R_i as input client. We assume honest submissions due to the semi-honest security model.
2. *Data validation:* $Prio+$ checks that inputs are valid (e.g., non-negative, reasonable) using share conversion.
3. *Computation factor generation & aggregation:* Each venue splits its data into secret shares and sends them to the aggregators.
4. *Computation execution:* The computing servers perform secure computations on the received shares following the protocol.
5. *Reconstruction & output:* The final sum is sent to policymakers and reconstructed locally. Aggregators don't learn individual inputs or results unless they collude. To compute the average, $Prio+$ allows revealing the number of participants.

Experiment setting & result. 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. 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 in terms of the requirements.

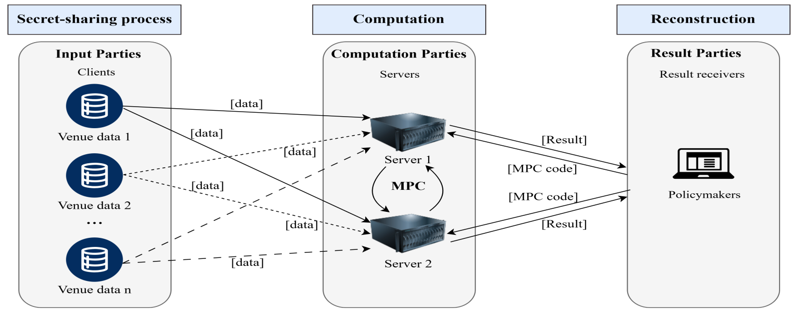

Fig. 3. SMPC model for scenario 'Average increased revenue': [data] and [result] are secret shares divided by SMPC protocol. Different line styles are used solely for visual clarity, without implying any semantic distinction.

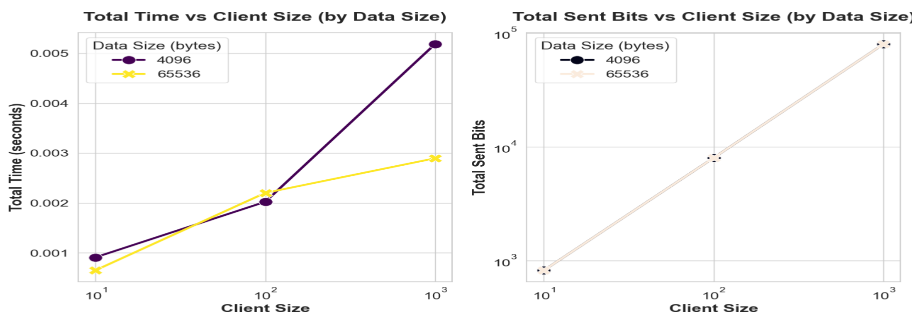

Fig. 4. Experimental results obtained using the different parameter settings

Validation discussion. The Prio+ [3] protocol was selected as a promising security 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 part of (due to page limit) the matching in Table 2. The experiment results suggest the usability of the selected protocol in this specific 'Average increased revenue' scenario.

6 Discussion

6.1 Usability of solution design: mapping framework

The reasonable result in this proof-of-concept experiment shows the feasibility and usability of our proposed design procedure 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 procedure and inform the design of a more comprehensive and rigorous decision-making framework.

Requirements Index		Matching evidence in *Prio+*
Functional	1	*Prio+* allows multiple clients' input, which can be easily set by the parameter: 'num_inputs'.
	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.
	3	In *Prio+*, data is split into shares distributed across multiple servers, where no single server (or minority coalition) can infer private values.
	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

6.2 Reflections

Throughout the construction and evaluation of the early-stage solution design process and decision-making framework in the music DBE context, several key insights have emerged that highlight the limitations of our current approach.

Practical constraints as another dimension. Practical constraints (e.g., cost, trust assumptions, computational resources) have a decisive impact on the PPC deployment strategy. In real-world use cases like MUSIC360, some participants cannot afford high-latency or high-overhead protocols. Our framework allows trade-offs between security and performance, but does not yet formally model or quantify these trade-offs. We suggest incorporating a resource-evaluation layer that measures protocol feasibility in constrained environments.

Priority weighting between factors/features. The use of scenario factors (e.g., 'resource-constrained', 'collusion-resistant') allows qualitative prioritization. However, these mappings are currently performed manually. The framework lacks formal weighting mechanisms to balance competing requirements (e.g., performance vs. cost). A possible extension is to support multi-criteria decision analysis (MCDA) for feature ranking and protocol scoring.

Gaps in scenario-to-requirement mapping. Although our framework maps functional and security requirements to scenario-specific factors, our experiment exposed cases where requirement types (RQTs) were underspecified or unmapped: (1) The scenario factor "required computation type" was difficult to trace to a specific RQT. This weakens traceability and hinders automatic reasoning. (2) The scenario mandated open-source protocol availability, yet this constraint was not represented in the framework. This suggests the need to add realistic, non-functional requirement types such as transparency or maintainability. These gaps indicate that the current framework may benefit from a non-functional requirement module for context-specific factors.

Deployment complexity as an overlooked metric.. Our experience shows that while many PPC protocols align well with security and functional requirements in theory, deployment complexity often becomes the real challenge in practice. Even relatively practical options like Prio+ demand container orchestration and manual client-server coordination if needed. This indicates that deployment feasibility is multifaceted, involving aspects such as technical setup burden, integration cost, tooling maturity, and operational risk. We believe that future versions of the framework should include a Deployment Complexity Evaluation (DCE) component to ensure protocol selection is guided not only by security alignment, but also by practical feasibility, especially in resource-constrained DBEs where technical capacity is limited.

Need for broader types of RQTs.. The current list of 12 RQTs (grouped under six security frames) was sufficient for the "average increased revenue" scenario. However, more complex scenarios in DBEs may have more security concerns. The MUSIC360 project itself includes other data flows (e.g., artist metadata, cross-country licensing) that raise such concerns. This suggests that our framework needs a broader and evolving taxonomy of requirement types, which can ideally be expanded with new domain contexts.

7 Conclusion

This study highlights the feasibility of privacy-preserving computation technologies, especially secure multi-party computation as a security solution for several hard-to-solve requirement types in DBEs. By analyzing the MUSIC360 case, we show how to find suitable PPC 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 decision-making framework successfully links scenario-specific requirements factors (e.g., computation type, anti-collusion, resource constraints) with the technical features of PPC (e.g., secret sharing and semi-honest adversarial model). Experimental verification confirms the usability of the highly satisfied protocols found according to the decision-making 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 PPC 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. 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)
5. 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)
6. 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)
7. Asharov, G., Lindell, Y., Schneider, T., Zohner, M.: More efficient oblivious transfer extensions. Journal of Cryptology **30**, 805–858 (2017)
8. 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)
9. Boulemtafes, A., Derhab, A., Challal, Y.: A review of privacy-preserving techniques for deep learning. Neurocomputing **384**, 21–45 (2020)
10. 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)
11. 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)
12. Dwork, C.: Differential privacy. In: International colloquium on automata, languages, and programming. pp. 1–12. Springer (2006)
13. Fabian, B., Gürses, S., Heisel, M., Santen, T., Schmidt, H.: A comparison of security requirements engineering methods. Requirements engineering **15**, 7–40 (2010)
14. 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)
15. 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)
16. 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)
17. Gordijn, J., Wieringa, R.: E3value User Guide - Designing Your Ecosystem in a Digital World. The Value Engineers, 1st edn. (2021)

18. 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)
19. 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)
20. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MIS Q. **28**(1), 75–105 (Mar 2004)
21. Jackson, M.: Problem frames: analyzing and structuring software development problems. Addison-Wesley Longman Publishing Co., Inc., USA (2000)
22. 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)
23. 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)
24. 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)
25. Lenkenhoff, K., Wilkens, U., Zheng, M., Süße, T., Kuhlenkötter, B., Ming, X.: Key challenges of digital business ecosystem development and how to cope with them. Procedia Cirp **73**, 167–172 (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. McLaren, A., Wang, Y., van de Velde, C., Green, E., Wieringa, R., Gordijn, J.: Value-based security requirements in a highly decentralized digital ecosystem: the music360 case. In: Hacks, S., Roelens, B. (eds.) PoEM-Companion 2024 PoEM 2024 Forum, M4S, FACETE, AEM, Tools and Demos. pp. 1–15. CEUR Workshop Proceedings, CEUR-WS (2024), publisher Copyright: © 2024 Copyright for this paper by its authors.; 17th IFIP WG 8.1 Working Conference on the Practice of Enterprise Modeling Forum, 1st International Workshop on Models for Simulation, M4S 2024, 2nd International Workshop on the Foundations and Applications of Capabilities in Enterprises, Transformations, and ESG Initiatives, FACETE 2024, Workshop on Advancing Enterprise Modelling through Digital Transformation, FAIR Data Management, and Blockchain Integration, AEM 2024, Tools and Demos, PoEM-Companion 2024 ; Conference date: 03-12-2024 Through 05-12-2024
29. 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)
30. 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)
31. 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)
32. Naor, M., Pinkas, B.: Efficient oblivious transfer protocols. In: SODA. vol. 1, pp. 448–457 (2001)

33. 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)
34. 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)
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. Susanto, H., Yie, L.F., Setiana, D., Asih, Y., Yoganingrum, A., Riyanto, S., Saputra, F.A.: Digital ecosystem security issues for organizations and governments: Digital ethics and privacy. In: Web 2.0 and cloud technologies for implementing connected government, pp. 204–228. IGI Global (2021)
40. 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)
41. Wang, Y., Green, E., Wieringa, R., Gordijn, J.: An exercise in concurrent software engineering: The music360 digital ecosystem. In: Proceedings of the IEEE Conference on Business Informatics. CBI 2025. IEEE (2025)
42. 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)
43. White, S.A., Miers, D.: BPMN modeling and reference guide: understanding and using BPMN. Future Strategies Inc. (2008)
44. 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`
45. 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)
46. 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)
47. 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)
48. 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)

49. Zhao, C., Zhao, S., Zhao, M., Chen, Z., Gao, C.Z., Li, H., Tan, Y.a.: Secure multi-party computation: theory, practice and applications. Information Sciences **476**, 357–372 (2019)