Satisfying Complex Data Security Requirements in Digital Business Ecosystems

1st Yulu Wang
Computer Science Department
Vrije Universiteit Amsterdam
Amsterdam, the Netherlands
y.wang4@vu.nl

2nd Charlotte van de Velde
Computer Science Department
Vrije Universiteit Amsterdam
Amsterdam, the Netherlands
c.s.d.vander.velden@student.vu.nl

3rd Sabine Oechsner
Computer Science Department
Vrije Universiteit Amsterdam
Amsterdam, the Netherlands
s.a.oechsner@vu.nl

4th Jaap Gordijn
Computer Science Department
Vrije Universiteit Amsterdam
Amsterdam, the Netherlands
j.gordijn@vu.nl

***Abstract*—Digital Business Ecosystems (DBEs) involve collaboration and sharing of data across various independent parties. Data sharing comes with security requirements, e.g. who may see which data elements. Often, these security requirements can be satisfied by well-known techniques, such as access controls, but sometimes the traditional solutions are not sufficient. For example, in our use case there is a requirement to sum up the revenue of companies by the government to calculate the average revenue for an industry, without disclosing the revenue of each company. To satisfy these kinds of requirements without a trusted third party, advanced Privacy-Preserving Computation (PPC) techniques are needed. However, the field of PPC is technically difficult to understand for most people and is highly specialized. We are not aware of a unified, comprehensive framework that can guide the systematic selection and integration of PPC methods, given the security requirements of a DBE use case. Therefore, our research goal is to establish such a framework. In this paper, two motivating examples are given, taken from the music digital business ecosystem we participate in.**

***Index Terms*—Security requirement, digital business ecosystem, privacy-preserving computation, secure multi-party computation.**

I. Introduction

Digital business ecosystems (DBEs) are systems of economic actors that depend on each other for their survival and well-being [1]. Actors in a DBE exchange things (often data) of economic of value with each other. Because the data is valuable, it is subject to attack. To understand which data is valuable, we take the DBE's business value model, e.g. following the e^3*value* method [2], as point-of-departure for eliciting the security requirements.

Often, traditional security technology such as detailed access control policies, can be used to secure valuable data. However, there are other use cases where the usual security techniques are not sufficient. Take for example our use case, where the government has to calculate the average revenue for a particular industry. Obviously, the revenue of a particular company is interesting for competitors, hence it should be considered as valuable data that should not be made public. The question is then how to calculate the average for an industry, without disclosing the revenue data of each company to the government. One way to do so, is to use advanced techniques such as Privacy-Preserving Computation (PPC). This is in particular relevant if no trusted third party can be identified.

The field of PPC is complex, understood by only a few, and fragmented. Therefore, for practitioners, it is hard to select appropriate PPC techniques, satisfying a set of complex security requirements at hand. To the best of our knowledge, work on PPC techniques and the requirements that they can satisfy, focuses on isolated technique categories and highly specific application areas (federated learning, deep learning, etc.). A unified, comprehensive framework that can guide the systematic selection and integration of multiple PPC methods is lacking [3]–[6]. Additionally, the limited existing work [7], [8], which compares PPC techniques, relies on theoretical descriptions that are inaccessible and incomprehensible to practitioners. This creates a gap in practical guidance for DBE professionals who need to implement privacy computing solutions tailored to their specific security requirements and project contexts. 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 perspective into the whole 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?'*. We are specifically interested in cases where trusted third parties are not an option as part of the solution (e.g., because they do not exist for the case at hand), and security requirements can only be satisfied by complex security technology based on modern cryptography. In this paper, we propose a research approach to address the stated

research question with some preliminary results, to answer that question.

This paper is structured as follows: Sec. II reviews the literature on security requirement engineering and privacy-preserving computation in the DBE environment, introducing characteristics of existing methods and identifying research gaps. Sec. III outlines the adopted research approach and research plan by engineering cycle. Sec. IV detailed the initial solution design process, integrating problem investigation, solution design, and approach validation to show how we plan to construct a comprehensive framework for mapping scenario-specific requirements to PPC solutions. Sec. V overviews the music digital business ecosystem case and contextualizes challenges in finding solutions for complex (data) security requirements. We present two scenarios in the music DBE case and carry out the whole research process in Scenario 1 "Average revenue of venues". Sec. VI will discuss some key findings and reflections for the mapping framework construct and highlight the possible contributions of our work.

II. Related Work

A. Security requirements engineering in DBEs

Security Requirements Engineering (SRE) is an important part of the design of each digital business ecosystem. A DBE is by definition multi-party, and so has different, and sometimes conflicting interests. This holds also for the data that the various parties own.

SRE involves the identification, analysis, specification, and validation of security requirements to ensure that DBEs are robust against threats and vulnerabilities [9]. In this area, Goal-Oriented Requirements Engineering (GORE) methods (such as KAOS, *i** and Secure Tropos) provide a systematic path to derive and refine security goals [10]–[12]. Additionally, the approach based on misuse and abuse cases [13] and the risk-based approaches (e.g., CORAS methodology, OCTAVE) [14] also provide effective means to identify and quantify security threats. To systematically embed security requirements engineering into the system development procedure, tools and methods such as Secure Tropos, SQUARE, and STORE have been proposed [15]–[17]. Maskani et al. emphasized that multiple perspectives, such as users, threats, and goals, must be balanced in security requirements modeling [18].

Although the above methods perform well in traditional security requirements engineering, there is still a research gap in their applicability in requirements capture and solution mapping for complex security requirements about data ownership and privacy-preserving computation in DBEs.

B. Privacy-preserving computation

Different participants often need to collaborate to complete valuable data computing tasks while ensuring the privacy of their respective data in DBEs [19]–[21]. For example, take the case that multiple parties want to sum their respective data and calculate the average value, but only obtain the overall result without disclosing the individual data. Without the existing of a trusted third party, such a requirement can not be satisfied and solved by traditional access control techniques. Therefore, it is necessary to introduce advanced PPC technology. We are specifically interested in use cases and the kind of security requirements that need to be satisfied by complex security technology such as PPC. Not only this technology should be accessible to practitioners who develop DBEs; but the PPC field itself would also benefit from a thorough understanding of use cases, their security requirements and possible solutions to justify this technology.

Currently, the main PPC methods include (1) Secure Multi-Party Computation (SMPC), (2) Homomorphic Encryption (HE), and (3) Differential Privacy (DP). SMPC has emerged as an important branch since Yao's seminal work on garbled circuits [22] and the subsequent generalizations by [23]. SMPC has evolved to support secure collaborative computations without exposing individual inputs. 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 [24]. Based on the current results, practical applications such as secure data analytics and privacy-preserving machine learning are being widely explored [7], [25]. Homomorphic encryption (HE) allows specific operations to be performed directly on ciphertext and ensures that the decrypted result is consistent with the result calculated on the plaintext [26]. Since Gentry proposed the first fully homomorphic encryption scheme [27], despite its high computational complexity, there have been many attempts to optimize computational efficiency in recent years [28]. Finally, differential privacy (DP) protects individual data contributions by injecting noise into statistical outputs [29]. Its application as a supplement to SMP&HE has been extended to large-scale data analysis and machine learning systems, but a careful balance is still needed between privacy protection and data accuracy.

C. Secure multi-party computation in DBEs

In a DBE environment, multi-party collaboration not only requires each party to maintain data ownership, but also requires accurate calculations/analysis under limited trust assumptions. Specifically, our use cases assume that a party, trusted by all other parties in the DBE does not exist. Homomorphic encryption is generally more attractive in cloud processing, but its efficiency issues currently still limit large-scale applications, and although differential privacy has high computational efficiency, it may lead to insufficient precision in the results. In comparison, the decentralized nature of SMPC gives it a clear advantage in such DBE scenario applications [25], although it also faces computing resource limitations, protocol complexity, and scalability issues in actual deployment. Although the existing literature has conducted in-depth research and comparisons on various privacy-preserving computing technologies [3]–[6], there is still a lack of a unified and systematic approach/framework 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 be-

tween theory and practice, especially for non-professionals to better design and test PPC solutions when certain security requirements arises.

III. Research Plan

We emphasize our research question is:
***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?*

For the coming years, we plan to answer this question by taking a Design Science Research (DSR) [30] point of view, and more specifically a Technical Action Research (TAR) approach [31]. This means that we will work with real DBEs in the field, and practitioners who (re)design them. In this (and the following) paper(s), we are involved in developing a DBE in the international music sector. The aim of this DBE is to better understand the value of music. The simplified engineering cycle that we will follow is described below and shown in Fig. 1:

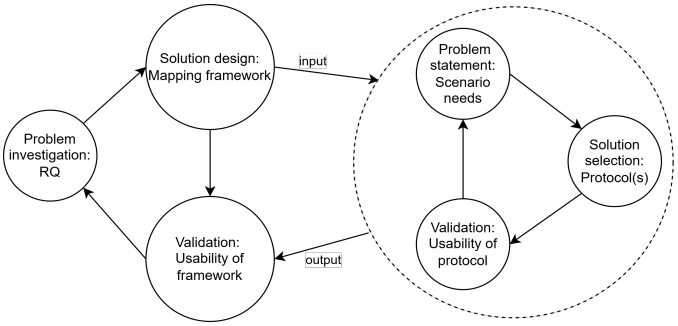

Fig. 1. Research design cycle

The left part of Fig. 1 shows the engineering cycle that we will execute a number of times, namely for a series of different DBEs. 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 a **problem investigation** and a literature review in case security needs and privacy-preserving computation (PPC). To understand the problem of satisfying complex security requirements in a DBE better, we have done an extensive literature review, in the fields of DBE modeling, security requirements engineering, and PPC technology (the result of this review is briefly reported on in Sec. II). At the same time, we have started to work on DBE design for the music ecosystem, which is our first DBE case under study. Doing the literature review and the DBE design concurrently is due to practical considerations, e.g. how project funding, in our case Horizon Europe, works. The result of these activities is the articulated research question (RQ). In addition, we found that many security requirements can be addressed by well-known techniques, such as various forms of access control mechanisms. But we also found that there are some hard-to-satisfy security requirements, regarding data ownership, and not disclosing valuable data. That particular finding resulted in the stated research question, the focus on advanced PPC techniques, and the specific SR types (problems) in DBEs that they can solve.

The **"solution design: mapping framework"** describes how to justify using advanced PPC technologies in a specific DBE, identify the related security problem, scenarios, and needs, 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 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 a partial **validation** of the whole design cycle. The outcome informs the validation task in the left cycle in Fig. 1 in somewhat more detail; to what extent did the proposed solution framework work, and how can it be improved.

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 security scenario, (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. We discuss it in Sec. IV.

IV. The Initial Solution Deisgn Procedure

A. Solution design procedure

Fig. 2 presents the part of the solution design procedure, cf. the BPMN modeling language [32], we carry out for our first use case, to calrify and satisfy hard-to-solve security requirements. We explain the procedure below:

Problem statement: Scenario needs. The procedure begins with the e^3*value* model, the UML data model and (data) security requirements specifications or other existing modeling artefacts 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 data model defines the structure and relationship of data in the DBE, it ensures that the data is organized in a way that supports business needs and shows some security scenarios. The (data) security requirements specification shows different kinds of security considerations and needs in this DBE. Based on these artefacts, we: (1) introduce an assessment step to cluster security requirements accordingly and identify hard-to-solve security scenarios. This step acknowledges that not all requirements can be addressed

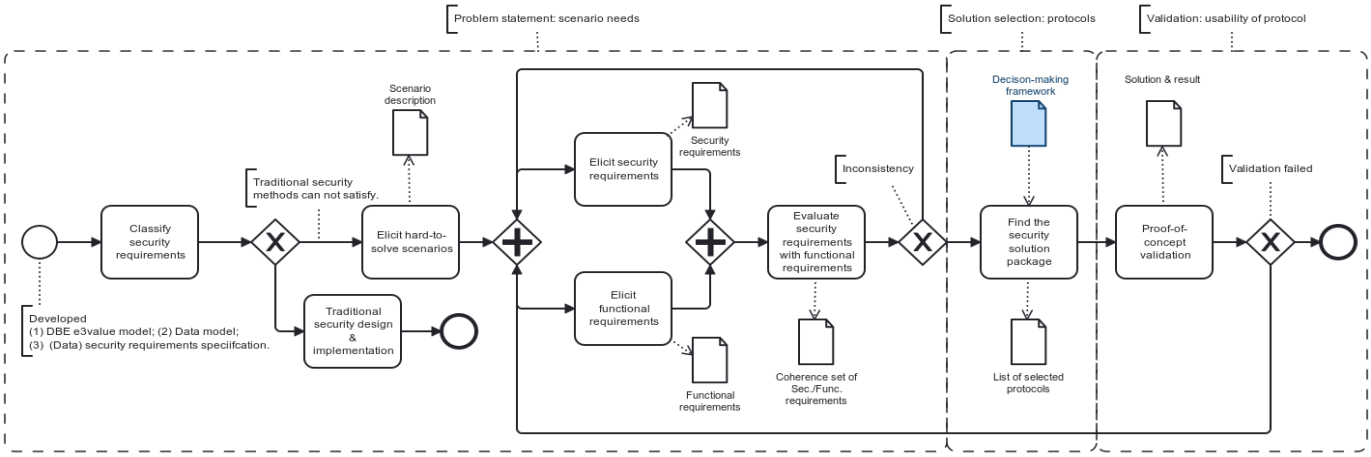

Fig. 2. Solution design procedure

by traditional security methods. It can help us to jutisfy 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 appropriate PPC techniques for implementing these security scenarios, using the proposed decision-making framework (marked blue): constructed by two sides: feature extraction of known PPC techniques and realistic scenario-specific requirement factors. 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 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. We will explain some core steps in Fig. 2 below.

B. Security requirement analysis for DBE

DBE-specific security patterns and requirement types. In DBEs, security requirements differ in complexity and urgency. We identify common requirement areas, covering both standard IT security and DBE-specific needs. On one hand, DBEs face traditional issues like access control, data integrity, authentication, availability, and encryption, which standard mechanisms can handle. On the other hand, DBEs raise new concerns from decentralized governance, trust, cross-border rules, dynamic collaboration, and competition. These often exceed the capabilities of conventional tools, calling for advanced PPC technologies. Our taxonomy in Table. I highlights when DBE-specific needs enter the realm of PPC, based on our case and supported by prior DBE literature [19], [25], [33], [34].

Assessing when PPC application is justified. The security patterns 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 (such as SMPC) not only beneficial but necessary. When one or more of these compound conditions are observed, conventional methods such as RBAC, encrypted data-at-rest, or trusted logging infrastructures are insufficient. Our solution deisgn procedure start 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.

C. Decision-making framework

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

Features of PPC techniques.. While reviewing PPC techniques (see also Sec. II) we found features of PPC technologies that can be used to select the appropriate PPC technology for the use case at hand. These features are:

- The supposed **[TSM]** threat security model: defines adversarial capabilities, distinguishing between semi-honest, malicious, and covert adversaries [7].
- The **[CM]** computing model: categorizes SMPC techniques based on cryptographic foundations, including homomorphic encryption, secret sharing, oblivious transfer, garbled circuits, and hybrid approaches [35]–[37], each with trade-offs in efficiency and security.
- The **[DM]** deployment model: describes how SMPC is implemented in real-world systems, with classifications

Security Pattern	Requirement Types
[Sec-pt 1] Secure multi-party data collaboration	**[RQT1]** Stakeholders compute jointly without revealing raw data. **[RQT2]** No need for central or trusted third parties. **[RQT3]** Data use requires explicit consent from each owner.
[Sec-pt 2] Privacy-preserving aggregation & result publication	**[RQT4]** Prevent inference of individual data from aggregates. **[RQT5]** Results must not allow reverse-engineering of inputs.
[Sec-pt 3] Encrypted data processing	**[RQT6]** Support computation over encrypted or secret-shared data. **[RQT7]** Keep data encrypted throughout its lifecycle.
[Sec-pt 4] Collusion-resistant computation	**[RQT8]** Prevent groups from inferring private data or corrupting results. **[RQT9]** Stay secure even with compromised participants.
[Sec-pt 5] Verifiability without disclosure	**[RQT10]** Enable result verification without exposing private data.
[Sec-pt 6] Dynamic & resource-constrained security	**[RQT11]** Work on devices with limited resources. **[RQT12]** Adapt to changing privacy regulations.

TABLE I
SECURITY PATTERNS AND RELATED REQUIREMENT TYPES IN DBES

such as server-side MPC, peer-to-peer MPC, and server-aided MPC [38].
- The **[SF]** supported crypto functionality (e.g. summing up numbers without disclosing the number itself): highlights the functional suitability for various application scenarios of different SMPC methods [24].

Integrating the above investigation, we developed a preliminary mapping framework that aligns categorized requirements with the appropriate PPC technology features

V. THE MUSIC DBE CASE

The music digital business ecosystem we work on is one example of an innovative digital business ecosystem, 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). One of the main high-level goal of the music ecosystem is to provide data about the value of music widely and transparently, respecting confidentiality requirements, where 'value' can have an economic, societal, or cultural connotation. To do so, the music DBE platform collects data about the music used in venues (e.g., by installing music fingerprinting devices in venues to discover what is actually played), effects of music played (for example increased revenue), and metadata of music such as rightsholders' (performers and authors) information, music work (including recordings), and earnings by rightsholders.

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 decision-making framework which includes 'Requirement Types' in Sec. IV-B and the 'Features of PPC techniques' in Sec. IV-C is the input for this case study. We present how we identify a specific hard-to-solve security scenario in MUSIC360 and state the problem by detailed scenario-specific requirements.

A. Problem statement: 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 (we present in [39]) and (data) security requirements specifications concluded from series of workshops to elicit security scenarios cannot be implemented by using traditional security approaches like authentication, authorization and detailed access control policies. In constract, we need to introduce advanced PPC technology to find a useful security solution. These scenarios are related to the requirement that a calculation needs to be done and that the party doing the calculation may see the result but not the elements that lead to the result. Two of the scenarios are (we omit the specific requirment specifications here and only present the overall insights):

1) S1 - Average revenue of venues
   - **Functional**: A party wants to know the average effects (in terms of revenue increase) of playing music at a number of venues.
   - **Security**: The average should be calculated without disclosing the individual revenue data to the party who wants to know the average.
2) S2 - Average earnings of performers
   - **Functional**: A party wants to know the average earnings of performers (e.g. of a specific genre).
   - **Security**: The average should be calculated without disclosing the individual earnings of each performer.

As these scenarios in terms of requirements are well understood, we can observe that both the functional and security requirements are mutual consistent. For both scenarios, the assumption is that there is no trusted third party who can do the calculation on behalf of the party interested in the result. In both scenarios, several characteristics can be observed: decentralized, privacy-preserving collaboration, multi-party data ownership, and limited trust by contract. These also can be found as common features in DBEs from our initial literature review and security requirement analysis in DBEs. Hence, we argue that secure multi-party computation (SMPC) offers key advantages in most DBEs.

B. S1 - Average revenue of venues: using the decision-making framework

We apply the decision-making framework (in Sec.IV-C) 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:$R_m = \frac{\sum(R_i)}{n}$, $(R_i = R_i\_before - R_i\_after)$[1].
   - *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].**
   - *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 undering a semi-honest adversarial model.

C. S1 - Average revenue of venues: usability validation of selected solution

Following our framework, the Prio+ [40] protocol was selected as a promising solution to satisfy the found requirements. It aligns well with the requirements. Additionally, considering the feasibility of the experiment, its open-source and lightweight features also promote our choice. Our basic container deployment via Docker for scenario testing is two server instances and a cluster of individual clients. We used the supported function: INT_SUM in $Prio+$ protocol [2] and tried different max_bits (data size can be input) and different client size (number of venues): 10/100/1000 in the parameters. Experimental results obtained on total sent bits and total time with different client sizes and data sizes are shown in Fig. 3, initially demonstrating the usability of the selected protocol in this specific "average increased revenue" scenario.

D. S2 - Average earnings of performers: brief analysis

We initially analyzed scenario-related security requirements, certain constraints, and calculation functions of the scenario "Average earnings of performers" as follows. The second-round proof-of-concept validation for our proposed process and framework based on this scenario is ongoing work.

1) **Scenario factor: Data ownership & confidentiality in multi-party nature.** Ensure that each CMO retains ownership and confidentiality of their respective artists' earnings data during the computation process.

[1] 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.

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

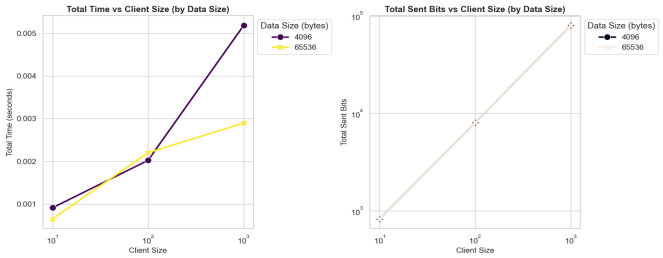

Fig. 3. Experimental results obtained using the different parameter settings

2) **Scenario factor: Required calculation type & function.** Accurately determine the number of unique artists across all CMOs to prevent overestimation or underestimation of average earnings due to duplicated data entries.
3) **Scenario factor: Resource-constrained.** The deployment design should be capable of handling computations involving numerous CMOs, some of which may have limited computational resources.
4) **Scenario factor: Robust, even in case of client failure** Ensure that the computation process remains robust and continues to function correctly even if some clients fail.
5) **Scenario factor: Regulatory and adversarial context.** Balance the need for trust among CMOs with the potential for conflicts in a contract-regulated business environment.
6) **Scenario factor: No collusion** Prevent collusion among CMOs or servers that could lead to unauthorized access to sensitive data.
7) **Scenario factor: Performant.** Optimize the computation process for operations such as summation and averaging, which are central to revenue analysis.
8) **Scenario factor: Required calculation type & function.** Required calculation function: $\mu = \frac{S}{N} = \frac{\sum_{i=1}^{k} \sum_{a \in A_i} e_a^i}{\left| \bigcup_{i=1}^{k} A_i \right|}$ [3][4][5]. Involved computation types: sum (addition), mean (division), and set.

VI. Discussion

A. Validation: usability of framework

The carry-out of S1 "Average revenue of venues" serves as a case study to refine our approach and inform the design of a more comprehensive and rigorous framework. The successful result we have had in this proof-of-concept testing shows the feasibility and usability about finding an appropriate PPC technology for a specific scenario through our proposed design procedure and decison-making framework. 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 solution design procedure development.

Practical constraints as another dimention. Through practical construction and tryout (S1: "Average revenue of venues") of the immature mapping framework, we argue that some practical constraints (eg., cost, trust level, resource-constrained) cannot be neglected. It is an important factor in decision-making of SMPC deployment model types. Also, these constraints usually affect the trade-off between more robust security guarantees and limited computation/communication resources.

Priority weighing between factors/features. Priority as a metric can be investigated and identified by some principles in the mapping framework. For example, in specific scenario analysis, the priority of performance requirements will depend on whether there exists a low-latency need (e.g., real-time analytics). One example principle can be: For resource-limited parties (e.g., small venues), favor lightweight protocols (Prio+) over wide-featured but computationally heavy ones.

Technique dataset. Our experiences in music DBE scenarios demonstrate that protocol mapping and selection require careful consideration of multiple key technique features. As a result, we plan to develop a structured dataset as an appendix for our final mapping framework. This dataset would rigorously classify methods by key technical features: threat security model, computing model, deployment model, support functions, and even possible existing benchmark testing results. It would enable efficient mapping to DBE requirements while supporting reproducible research.

Need for broader types of RQTs.. The current list of 12 RQTs (grouped under six security patterns) 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.

B. Possible contributions

This research preview highlights critical challenges in addressing complex security requirements within digital business ecosystems (DBEs). We suggest that a unified and comprehensive framework can be concluded and generalized to show the entire mapping and linking process. The framework will be a reference guidance for easy access and application of PPC (SMPC) technologies for business companies in non-secure computing fields to make DBE more powerful for private data value investigation.

To improve our framework, the scenario-specific needs involving security requirements, functional requirements and other constraints will be coherently considered and then be linked to key components in PPC (SMPC) technology. Trade-offs and preferences between different types of these key

[3] μ: The average earning is the sum of total earnings divided by the total number of unique artists.

[4] S: The total earnings of each artist is the sum of their earnings in all associated CMOs. Assuming there are k CMOs, the set of artists in the ith CMO is A_i, and the earning of the artist a in the ith CMO is e_a^i.

[5] N: The total number of unique artists is the union size of all CMO artist sets.

components in multiple technical protocols will be conducted during the mapping and decision-making. Certain criteria and principles will be derived, generalized and then iterated from case validation in the research design cycle we presented in Fig. 1 as the main part of the framework we want to propose. The practicality and effectiveness of our proposed framework and the framework design approach in Fig. 2 will both be validated through proof-of-concept testing on scenarios in music DBE using several selected PPC protocols.

C. Potential risks & Limitations

The potential risks and limitations of this research are as follows:

1) Deployment complexity remains a significant barrier and hard to estimate the effort needed, even for the proof-of-concept testing, as existing PPC protocols are mostly implemented as command-line prototypes/applications or provided as programming libraries. Therefore, implementation difficulty and possible time cost should also be used as one of the measurement indicators of the proposed framework.
2) Scenarios that have multiple privacy computation requirements at the same time have not been explored enough. The real-world scenario complexity may lead to possible protocol switching or some safety-related considerations trade-off. Comprehensive reasoning and mapping mechanisms need to be realized in some way in the proposed framework and standardized interfaces and cross-protocol compatibility may also need to be included in the real-project implementation.
3) Our study design primarily focuses on the specific use case within music DBE, and further validation is needed across different domains to generalize our findings.

ACKNOWLEDGMENT

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] R. Wieringa and J. Gordijn, *Digital Business Ecosystems*. Soest, The Netherlands: TVE Press, 2023. [Online]. Available: https://www.thevalueengineers.nl/digital-business-ecosystems-book?page=digital-business-ecosystems-book

[2] J. Gordijn and R. Wieringa, *E3value User Guide - Designing Your Ecosystem in a Digital World*, 1st ed. The Value Engineers, 2021.

[3] Y. Lindell, "Secure multiparty computation for privacy preserving data mining," in *Encyclopedia of Data Warehousing and Mining*. IGI global, 2005, pp. 1005–1009.

[4] A. O. Almagrabi and A. K. Bashir, "A classification-based privacy-preserving decision-making for secure data sharing in internet of things assisted applications," *Digital Communications and Networks*, vol. 8, no. 4, pp. 436–445, 2022.

[5] J. Feng, L. T. Yang, N. J. Gati, X. Xie, and B. S. Gavuna, "Privacy-preserving computation in cyber-physical-social systems: A survey of the state-of-the-art and perspectives," *Information Sciences*, vol. 527, pp. 341–355, 2020.

[6] A. Boulemtafes, A. Derhab, and Y. Challal, "A review of privacy-preserving techniques for deep learning," *Neurocomputing*, vol. 384, pp. 21–45, 2020.

[7] C. Zhao, S. Zhao, M. Zhao, Z. Chen, C.-Z. Gao, H. Li, and Y.-a. Tan, "Secure multi-party computation: theory, practice and applications," *Information Sciences*, vol. 476, pp. 357–372, 2019.

[8] D. Evans, V. Kolesnikov, M. Rosulek *et al.*, "A pragmatic introduction to secure multi-party computation," *Foundations and Trends® in Privacy and Security*, vol. 2, no. 2-3, pp. 70–246, 2018.

[9] P. Salini and S. Kanmani, "Survey and analysis on security requirements engineering," *Computers Electrical Engineering*, vol. 38, no. 6, pp. 1785–1797, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045790612001644

[10] A. van Lamsweerde, "Goal-oriented requirements engineering: a guided tour," in *Proceedings Fifth IEEE International Symposium on Requirements Engineering*, 2001, pp. 249–262.

[11] V. M. B. Werneck, A. d. P. A. Oliveira, and J. C. S. do Prado Leite, "Comparing gore frameworks: i-star and kaos." in *WER*. Citeseer, 2009.

[12] D. Mellado, H. Mouratidis, and E. Fernández-Medina, "Secure tropos framework for software product lines requirements engineering," *Computer Standards Interfaces*, vol. 36, no. 4, pp. 711–722, 2014, security in Information Systems: Advances and new Challenges. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0920548913001803

[13] J. P. McDermott and C. Fox, "Using abuse case models for security requirements analysis," *Proceedings 15th Annual Computer Security Applications Conference (ACSAC'99)*, pp. 55–64, 1999. [Online]. Available: https://api.semanticscholar.org/CorpusID:206580824

[14] N. Mayer, A. Rifaut, and E. Dubois, "Towards a risk-based security requirements engineering framework," in *Proceedings of REFSQ*, vol. 5, 2005. [Online]. Available: https://api.semanticscholar.org/CorpusID:8837909

[15] H. Mouratidis and P. Giorgini, "Secure tropos: A security-oriented extension of the tropos methodology," *International Journal of Software Engineering and Knowledge Engineering*, vol. 17, 04 2007.

[16] N. Mead and T. Stehney, "Security quality requirements engineering (square) methodology," *ACM SIGSOFT Software Engineering Notes*, vol. 30, pp. 1–7, 07 2005.

[17] M. T. J. Ansari, D. Pandey, and M. Alenezi, "Store: Security threat oriented requirements engineering methodology," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 2, pp. 191–203, 2022.

[18] I. Maskani, J. Boutahar, and S. E. G. El Houssaïni, "Analysis of security requirements engineering: towards a comprehensive approach," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 11, 2016.

[19] P. K. Senyo, K. Liu, and J. Effah, "Digital business ecosystem: Literature review and a framework for future research," *International journal of information management*, vol. 47, pp. 52–64, 2019.

[20] J. Gordijn and R. Wieringa, "The business model of digital ecosystems: Why and how you should do it," in *Advances in Enterprise Engineering XVI*, ser. Lecture Notes in Business Information Processing, C. G. M. J. S. Guerreiro, Ed. Germany: Springer-Verlag, 2023.

[21] S. Suuronen, J. Ukko, R. Eskola, R. S. Semken, and H. Rantanen, "A systematic literature review for digital business ecosystems in the manufacturing industry: Prerequisites, challenges, and benefits," *CIRP Journal of Manufacturing Science and Technology*, vol. 37, pp. 414–426, 2022.

[22] A. C.-C. Yao, "How to generate and exchange secrets," in *27th annual symposium on foundations of computer science (Sfcs 1986)*. IEEE, 1986, pp. 162–167.

[23] S. Micali, O. Goldreich, and A. Wigderson, "How to play any mental game," in *Proceedings of the Nineteenth ACM Symp. on Theory of Computing, STOC*. ACM New York, 1987, pp. 218–229.

[24] J. Guo, Q. Wang, X. Xu, T. Wang, and J. Lin, "Secure multiparty computation and application in machine learning," *Journal of Computer Research and Development*, vol. 58, pp. 2163–2186, 2021.

[25] Y. Lindell, "Secure multiparty computation," *Communications of the ACM*, vol. 64, no. 1, pp. 86–96, 2020.

[26] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A survey on homomorphic encryption schemes: Theory and implementation," *ACM Computing Surveys (Csur)*, vol. 51, no. 4, pp. 1–35, 2018.

[27] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the forty-first annual ACM symposium on Theory of computing*, 2009, pp. 169–178.

[28] W. Jin, Y. Yao, S. Han, J. Gu, C. Joe-Wong, S. Ravi, S. Avestimehr, and C. He, "Fedml-he: An efficient homomorphic-encryption-based privacy-preserving federated learning system," *arXiv preprint arXiv:2303.10837*, 2023.

[29] C. Dwork, "Differential privacy," in *International colloquium on automata, languages, and programming*. Springer, 2006, pp. 1–12.

[30] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS Q.*, vol. 28, no. 1, p. 75–105, Mar. 2004.

[31] R. Wieringa and A. Moralı, "Technical action research as a validation method in information systems design science," in *International Conference on Design Science Research in Information Systems*. Springer, 2012, pp. 220–238.

[32] S. A. White and D. Miers, *BPMN modeling and reference guide: understanding and using BPMN*. Future Strategies Inc., 2008.

[33] K. Lenkenhoff, U. Wilkens, M. Zheng, T. Süße, B. Kuhlenkötter, and X. Ming, "Key challenges of digital business ecosystem development and how to cope with them," *Procedia Cirp*, vol. 73, pp. 167–172, 2018.

[34] H. Susanto, L. F. Yie, D. Setiana, Y. Asih, A. Yoganingrum, S. Riyanto, and F. A. Saputra, "Digital ecosystem security issues for organizations and governments: Digital ethics and privacy," in *Web 2.0 and cloud technologies for implementing connected government*. IGI Global, 2021, pp. 204–228.

[35] S. Bian, W. Jiang, and T. Sato, "Privacy-preserving medical image segmentation via hybrid trusted execution environment," in *2021 58th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2021, pp. 1347–1350.

[36] Q. Zhang, C. Xin, and H. Wu, "Privacy-preserving deep learning based on multiparty secure computation: A survey," *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10 412–10 429, 2021.

[37] R. W.-q. L. G.-p. W. Z.-x. HAN Wei-Li, SONG Lu-shan, "Secure multi-party learning: From secure computation to secure learning," *CHINESE JOURNAL OF COMPUTERS*, vol. 46, no. 7, pp. 1494–1512, 2023.

[38] "Ieee recommended practice for secure multi-party computation," *IEEE Std 2842-2021*, pp. 1–30, 2021.

[39] A. McLaren, Y. Wang, C. van de Velde, E. Green, R. Wieringa, and J. Gordijn, "Value-based security requirements in a highly decentralized digital ecosystem: the music360 case," in *PoEM-Companion 2024 PoEM 2024 Forum, M4S, FACETE, AEM, Tools and Demos*, ser. CEUR Workshop Proceedings, S. Hacks and B. Roelens, Eds. CEUR-WS, 2024, pp. 1–15, 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.

[40] S. Addanki, K. Garbe, E. Jaffe, R. Ostrovsky, and A. Polychroniadou, "Prio+: Privacy preserving aggregate statistics via boolean shares," in *International Conference on Security and Cryptography for Networks*. Springer, 2022, pp. 516–539.