MapXchange: Designing a Confidentiality-Preserving Platform for Exchanging Technology Parameter Maps

Jan Pennekamp
jan.pk@comsys.rwth-aachen.de
RWTH Aachen University
Aachen, Germany

Joseph Leisten
leisten@comsys.rwth-aachen.de
RWTH Aachen University
Aachen, Germany

Paul Weiler
p.weiler@wzl.rwth-aachen.de
RWTH Aachen University
Aachen, Germany

Markus Dahlmanns
dahlmanns@comsys.rwth-aachen.de
RWTH Aachen University
Aachen, Germany

Marcel Fey
m.fey@wzl.rwth-aachen.de
RWTH Aachen University
Aachen, Germany

Christian Brecher
c.brecher@wzl.rwth-aachen.de
RWTH Aachen University
Aachen, Germany

Sandra Geisler
geisler@cs.rwth-aachen.de
RWTH Aachen University
Aachen, Germany

Klaus Wehrle
wehrle@comsys.rwth-aachen.de
RWTH Aachen University
Aachen, Germany

Abstract

Technology parameter maps summarize experiences with specific parameters in production processes, e.g., milling, and significantly help in designing new or improving existing production processes. Businesses could greatly benefit from globally exchanging such existing knowledge across organizations to optimize their processes. Unfortunately, confidentiality concerns and the lack of appropriate designs in existing data space frameworks—both in academia and industry—greatly impair respective actions in practice. To address this research gap, we propose MapXchange, our homomorphic encryption-based approach to combine technology parameters from different organizations into technology parameter maps while accounting for the confidentiality needs of involved businesses. Central to our design is that it allows for local modifications (updates) of these maps directly at the exchange platform. Moreover, data consumers can query them, without involving data providers, to eventually improve their setups. By evaluating a real-world use case in the domain of milling, we further underline MapXchange's performance, security, and utility for businesses.

CCS Concepts

• **Security and privacy → Privacy-preserving protocols**; • **Applied computing →** *Engineering*; • **Information systems**;

Keywords

secure industrial collaboration; homomorphic encryption; data sharing; exchange platform; process planning; Internet of Production

ACM Reference Format:
Jan Pennekamp, Joseph Leisten, Paul Weiler, Markus Dahlmanns, Marcel Fey, Christian Brecher, Sandra Geisler, and Klaus Wehrle. 2025. MapXchange: Designing a Confidentiality-Preserving Platform for Exchanging Technology Parameter Maps. In *The 40th ACM/SIGAPP Symposium on Applied Computing (SAC '25), March 31-April 4, 2025, Catania, Italy.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3672608.3707734

1 Introduction

Just like other areas, the Industrial Internet of Things (IIoT) greatly benefits from information sharing [19, 51]. Already today these benefits lead to the establishment of various data ecosystems in industry as exemplified by commercial platforms, e.g., Open-es, MindSphere, and Skywise [32]. These data ecosystems range from direct, bilateral data sharing between only two parties to data exchange between many stakeholders over the infrastructure of a data space [21]. However, in business-driven environments like the IIoT, where maintaining a competitive advantage is essential, confidentiality concerns often impede information sharing [27]. In fact, 80 % of industrial data is neither exploited nor shared [18]. While providing numerous benefits, data spaces may not always provide sufficient guarantees to protect sensitive data [31], preventing large-scale adoption. Hence, insufficient consideration of confidentiality hinders the benefits of widespread information today.

In the IIoT, the scope of shared data also has a significant impact on what businesses consider acceptable for disclosing. The scope can range from direct, bilateral information flows (along supply chains) to exchanging highly-specialized data among competitors (across supply chains) [42]. Here, domain knowledge of and experience with materials, machines, or processes greatly influences how valuable shared information is and what kind of details and know-how are derivable from seemingly abstract data [40]. While industry [2, 13, 14, 32, 46] and academia [6, 15, 39, 50] proposed several platforms and protocols to mitigate the situation, they are not universally applicable (cf. Section 2.5). Instead, developing domain- and use case-specific approaches for the IIoT that introduce certain reliable security guarantees is a common practice nowadays.

SAC '25, March 31-April 4, 2025, Catania, Italy

ACM ISBN 979-8-4007-0629-5/25/03
https://doi.org/10.1145/3672608.3707734

Accordingly, in this work, we focus on the novel use case of exploiting technology parameters across organizational boundaries. Technology parameters in milling document the engagement conditions of a milling tool and provide insights into the productivity of the milling process. The selection and optimization of process parameters rely on implicit expert knowledge [9]. By aggregating optimal technology parameters into technology parameter maps (cf. Figure 1) and subsequently sharing them, this implicit knowledge is made available (globally). This sharing results in resource-efficient production design and shorter ramp-up phases (when integrated into process planning). Given the lack of global exploitation, so far, this knowledge is seldomly captured (compiled). Specifically, today's data-sharing approaches and data ecosystems do not yet convincingly support said exchange for multiple reasons: They either rely on organizational security [31], only protect data providers *or* data consumers [10], or do not support updates [15].

To address this lack, in this paper, we propose a confidentiality-preserving exchange platform called MapXchange, which enables sharing technology parameter maps in the IIoT while accounting for (i) the required functionality when processing and exploiting technology parameter maps, (ii) their sensitivity, and (iii) the expected scale of data sharing in industrial settings. Central to our design is Homomorphic Encryption (HE), which hides sensitive details from the exchange platform while still allowing for operations on the encrypted data. This way, we establish a practical, confidentiality-preserving data space for technology parameter maps. Utilizing real-world data that covers milling processes, we further demonstrate the performance, security, and utility of MapXchange.

Conceptually, our research is not limited to milling. We see potential of exchanging maps for a variety of manufacturing processes: Injection molding [12], electrochemical machining (ECM) [28], and electrical discharge machining (EDM) [24], among others.

Contributions. Our main contributions are as follows.

- We enable exchanging technology parameter maps in industry by preserving the confidentiality of parameter maps *and* queries, promising a broader utilization of these maps.
- Our integrated data-validation mechanism allows for attested data quality, more specifically the genuineness of updates, without sacrificing the expected confidentiality.
- We demonstrate that MapXchange scales to industry-sized applications while providing technical security guarantees.

Open Science. We open-source our artifacts and a prototype of MapXchange [41] to ensure reproducibility and reusability.

2 Scenario and Research Gap

In this section, we first introduce the use case of exchanging technology parameter maps along with its benefits (Sections 2.1 and 2.2) as well as challenges that currently hinder its real-world use (Section 2.3). Based on this description, we then formulate corresponding design goals (Section 2.4) and discuss related work (Section 2.5).

2.1 Use Case: Technology Parameter Maps

Technology parameter maps allow for aggregating the (optimal) technology parameter for a specific manufacturing process and may further track additional information, such as reported usage statistics. For the use case of milling, a technology parameter map documents the (optimal) engagement condition of a milling tool for a specific combination of machine tool and material. The depth of cut a_p, the width of cut a_e, and the feed per tooth f_z then characterize said engagement conditions, which are defined by the selected technology parameters of a milling process. In sum, these three factors determine the material removal rate per cutting edge, which serves as an indicator for the productivity of a milling process [17].

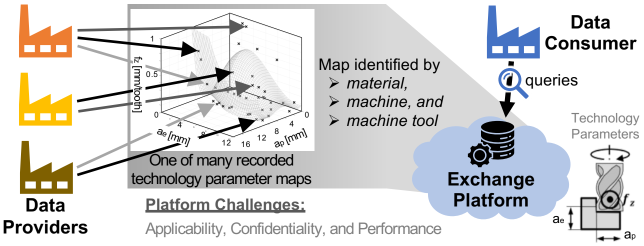

Figure 1: Data providers offload their insights for compensation. Data consumers query maps to improve their setups.

Beyond productivity, a high material removal rate also indicates shorter process times, i.e., reduced energy consumption per part [16]. However, a corresponding disadvantage is the increase in tool wear, which reduces both, part quality and the tool's life [30, 58]. This situation highlights the trade-off between process productivity and part quality. Due to the complex wear mechanisms of a cutting edge, calculating the optimal engagement conditions for a milling tool is infeasible at this point [59]. Instead, selecting technology parameters relies on expert knowledge of a process planner (i.e., implicit knowledge), which the machine operator adjusts adaptively to account for environmental influences to achieve the required part quality [9]. Due to a lack of comparability, this approach cannot ensure the selection of optimal technology parameters, resulting in different feed rates per tooth with the same cutting depth and width, leading to inefficient cutting processes. In the context of milling processes, "optimal" refers to achieving a high material removal volume per cutting edge under stable process conditions.

By using a dexel-based material removal simulation [29] in parallel to the real process, the engagement conditions of a milling tool can be calculated based on machine internal sensor signals. Areas of constant engagement conditions are detected using a change point detection algorithm [20] to extract support points for a technology parameter map from the resulting time series signal. A support point i is defined by the tuple $(a_{p,i}, a_{e,i}, f_{z,i})$. Collecting multiple supporting points (incl. their reported use) results in a technology parameter map, which provides information on the optimal engagement conditions of a tool for a specific (milling) process (cf. Figure 1). Collaborative efforts are highly beneficial for its quality and thus value. Due to the temporal decoupling through the extraction of support points, technology parameter maps do not reveal any insights about underlying (machining) processes, which is a necessary prerequisite for an interorganizational exchange of data.

2.2 Key Benefits of an Exchange Platform

Most importantly, technology parameter maps contain implicit knowledge about processes derived from the intuition and experience of a process planner or machine operator, which cannot be economically captured through analogy experiments or comprehensively modeled [59]. Facilitating an exchange of technology parameter maps, in turn, enables a domain-specific, interorganizational transfer of expert knowledge, even within competitive environments [48]. Incorporating a platform (cf. Figure 1) as an intermediary eases the preservation and aggregation of expert knowledge. In production technology, implicit expert knowledge is a decisive factor in the design and optimization of processes [38], so the need to exchange expert knowledge is not limited to the use case of milling but can be applied to other areas (cf. Section 1).

Exchanging technology parameter maps offers data providers the opportunity to globally monetize their acquired expert knowledge. Moreover, by retrieving and utilizing technology parameter maps, data consumers benefit from the potential of an efficient process design and shortened ramp-up phases of a process for a given combination of machine, machine tool, and material. Then again, technology parameters further enable the selection of the most suitable tool under given machining conditions, supporting a demand-based utilization of process resources. Finally, they can provide suppliers with insight into user behavior, serving as the basis for the development of demand-oriented tools. Thus, interorganizational data sharing enables not only resource-efficient process planning but also facilitates agile process adaptation in dynamically changing environments, strengthening corporate resilience.

2.3 Challenges and Threat Model

Despite the indisputable benefits (cf. Section 2.2), corresponding exchange platforms are still missing. Apart from the need for scaling to applications in the IIoT, we attribute this situation to three factors.

Confidentiality Concerns. First, businesses are extremely cautious with sensitive data because they fear a loss of control [27, 49]. Given that technology parameter maps inherently contain domain-specific expert knowledge, which ultimately constitutes a competitive advantage, preserving this confidentiality is essential. In this context, businesses have precise expectations: They do not want their data to end up in a public repository or to be freely and unrestrictedly accessible to other businesses (or even competitors), i.e., they expect an approach that convincingly addresses their needs.

Aggregation Support. Second, platforms have to aggregate (combine and update) information from several data providers while accounting for the aforementioned data sensitivity. Seemingly naïve approaches that operate on plaintext data may offer this support at the expense of failing to preserve the required confidentiality. Specifically, minimizing side-channel leaks, for example, revealing the updating data provider, constitute a significant challenge in this regard. In addition to minimizing the amount of (sensitive) information the platform discloses to (unauthorized) parties, they also have to obfuscate the relationships of cooperating, i.e., exchanging, entities to protect trade secrets. Reliably providing this aggregation support while preserving confidentiality is far from trivial.

Deployment Model. Third, assuming that technology parameters are retrievable *at all times* from the data providers is not realistic, because data-providing businesses want to (a) minimize their overhead, i.e., interaction with third parties, and still (b) maximize their monetary compensation. This desire is also the reason why we consider bilateral approaches, which omit a central platform, as impractical. Repeatedly retrieving data from all data providers *for each query* simply does not scale. Relying on a third-party platform rather than round-based protocols limits the communication overhead, improves the overall scalability of the exchange platform, and thus constitutes a promising direction. However, "simply" pursuing a basic data ecosystem that offers organizational security measures, i.e., collecting all available information, persisting it in a central register, and configuring modules for authentication and authorization, is not in compliance with the aforementioned confidentiality expectation, as also highlighted in related work [31].

To better understand the challenges associated with designing a *secure and confidentiality-preserving* exchange platform, we now look at the capabilities potential attackers have in this setting. Given that the utility of exchanging technology parameter maps increases with the number and diversity of participants, we cannot assume established trust relationships between data providers, data consumers, or their intersections. In our research, we thus consider malicious-but-cautious adversaries [47] as the primary threat model: Involved businesses have an incentive to cheat (for their individual gain) while also depending on their reputation and being bound to specific legislation. In particular, the exchange platform, data providers, and data consumers may attempt to gain as much (sensitive) information without leaving a trace of their malicious actions.

2.4 Postulated Research Goals

We now derive three design goals based on the information we presented in Section 2 to postulate a concrete set of desired properties.

G1: Applicability. First, supporting monetary compensation for data providers introduces a significant incentive to participate, which might be missing otherwise. Therefore, sufficient information on queries and returned results should be collected and handled for billing purposes. Second, the exchange platform should respond to queries with accurate results. This aspect is particularly important when designing protocols that conceal sensitive data, e.g., through noise or when approximating it (with limited precision). Third, to ensure valuable, up-to-date data from various data providers at all times, technology parameter maps have to remain updatable. Hence, any design must be able to flexibly join (aggregate) data from different sources. The platform thus also has to feature a component that reliably identifies the true optimum after receiving new data. Fourth, data consumers may be interested in different responses. As a result, designs should support multiple query types: regular—index-based ones—and reverse queries, which allow for comparing multiple technology parameter maps with each other.

G2: Confidentiality. Due to the sensitivity of both, data and queries in this business setting, designs have to account for the confidentiality needs of both, data providers *and* data consumers. The former expect that offloaded data is kept private, except when queried, to maintain their competitive advantage. Moreover, third parties may not learn anything about a data-providing business. Likewise, the latter desire that submitted queries (and access patterns) are not traceable for third parties, including data providers, to hide which information they are after. Lastly, the required trust in the data-sharing platform should be kept minimal to (i) avoid a single point of failure and (ii) convince businesses to participate.

G3: Performance. Since our research focuses on globally sharing information in the IIoT, a suitable exchange platform and the overhead introduced by preserving the confidentiality needs to scale to deployments in industry, i.e., it has to support large numbers of data providers, data consumers, and queries, as well as corresponding technology parameter maps; ideally while using commodity hardware. That being said, the offloading of data or queries must conclude in reasonable time: Given that these actions are not everyday tasks (for example, commissioning a production line may take several weeks [39]), certain processing (hours to days) is acceptable. However, if the introduced overhead is too excessive, businesses might be discouraged from participating in the exchange at all.

2.5 Related Work: Common Shortcomings

Next, we give an overview of related work, which we also summarize in Table 1. Despite being aware of the business' confidentiality expectations, large-scale initiatives [7, 34, 36] *still* fail to provide concrete, general-purpose realizations [25, 56]. To the best of our knowledge, they largely rely on organizational security and thus cannot satisfy the strong confidentiality needs horizontal collaboration and coopetition introduce [42]. Other commercial offers either require trust in a single party [46] or only host a lookup service (without data points) [2] and are thus not applicable to our setting.

Technology Overview. For the sake of a more detailed analysis, we thus limit ourselves to approaches that (i) are readily deployable and (ii) promise reasonable performance for industrial settings (**G3**). Accordingly, we further exclude approaches that exclusively apply oblivious transfers [44] or secure multiparty computation [57]. While the former adds significant computational overhead, greatly exceeding acceptable runtimes and storage needs, the latter usually requires data providers to repeatedly participate in queries, contradicting **G1**. In Section 6.1, we later discuss the potential of designs with building blocks different from HE in greater detail.

For the remaining approaches, we notice two primary issues:

Applicability. First, confidentiality-preserving approaches [6, 11, 15, 39] cannot support the required data aggregation and updating functionality, which is strictly required when joining information from multiple data providers (cf. **G1**). Particularly, BPE [39] and PDBQ [6] may further be challenged by their somewhat limited scalability. Regardless of these critical limitations, these approaches highlight the diversity in underlying building blocks (e.g., Bloom filters, private set intersections, or zero-knowledge proofs) when designing privacy-preserving protocols for information retrieval.

Confidentiality. Second, (older) approaches [5, 10, 50], which generally promise the required applicability features (**G1**), lack the required level of confidentiality (**G2**) promising approaches require (cf. Table 1). For example, PIR [10] does not consider the confidentiality needs of data providers. Likewise, PKSE [5] does not limit the data consumer's access to the offloaded data. Lastly, SSE [50] requires an established trust relationship between data consumers and data providers, which is unrealistic for our scenario and for data sharing between organizations in the IIoT in general. Thus, they are not applicable to our setting (cf. Section 2.3) either.

Research Gap. A viable approach that (i) enables globally exchanging and handling technology parameters, (ii) reliably accounts for the sensitivity of processed data, and (iii) scales to industry-sized use cases and applications in the IIoT is still missing. Our approach MapXchange (Section 4) supports the required functionality (**G1**). However, by design, it slightly sacrifices **G2** by (i) tracking access patterns for billing purposes and (ii) trading off data correctness (i.e., the utility of exchanged data) and confidentiality (cf. Section 4.4).

Table 1: Classifying related work in light of our design goals.

Approach	G1: Applicability	G2: Confidentiality	G3: Performance
BPE [39]	✗	✓	(✓)
PDBQ [6]	✗	✓	(✓)
PIR [10]	✓	✗	✓
PKSE [5]	✓	✗	✓
PPSSI [15]	✗	✓	✓
RKS [11]	✗	✓	✓
SSE [50]	✓	✗	✓
MapXchange	✓	(✓)	✓

Prior work is really diverse and excels in their respective settings. However, practical and performant designs, which support updates to offloaded information while considering the confidentiality needs of data providers and *consumers, are still missing. Research thus need to consider their applicability in real-world business settings to offer practical approaches for exchanging technology parameter maps.*

3 Preliminaries

To close the outlined research gap, we rely on a well-established building block, whose functionality we present in the following.

Homomorphic Encryption (HE) enables calculations on encrypted data without the need to decrypt the ciphertexts, thus maintaining data confidentiality [1]. As a result, HE is frequently applied in (untrusted) cloud environments or to enable the secure outsourcing of computations [33, 43, 55, 60]. With respect to the outlined research gap, HE promises to enable modifications of technology parameter maps without the exchange platform learning anything about the sensitive business data it is processing. As a result, businesses have to invest less trust in the exchange platform. Over time, several variants (cryptosystems) of HE emerged. They feature distinct implications on usability and performance. For example, Fully Homomorphic Encryption (FHE) [22, 52] supports a larger set of operations. However, it introduces computational overhead, additional storage needs, and decreased accuracy [1]. In contrast to FHE, **Partially Homomorphic Encryption (PHE)** [23, 37, 45] is limited in its set of operations but demands fewer resources for operation [1]. PHE cryptosystems are either additively or multiplicatively homomorphic, i.e., they support either *only* adding or *only* multiplying encrypted numbers, respectively. As detailed in the following section, we apply homomorphic encryption to values offered for exchanges in two cases: Addition into an encrypted aggregate sum and obfuscation by adding a random offset. For these purposes, additively homomorphic encryption is sufficient.

4 Design: MapXchange

In this section, we introduce MapXchange, our approach for preserving confidentiality while globally exchanging technology parameter maps. In Section 4.1, we introduce this concept for a data ecosystem before presenting the involved entities in more detail in Section 4.2. As we elaborate on in Section 4.3, MapXchange features two types of queries for improved utility. Lastly, in Section 4.4, we discuss how MapXchange ensures genuineness of updates, despite being able to only process encrypted information at the exchange platform.

4.1 Design Overview and Processing Sequence

MapXchange distinguishes three actions during operation to realize data provision, validation, and retrieval (i.e., queries), respectively. The core of this endeavor is the conceptually-centralized exchange platform (Figure 2). MapXchange distributes the competencies to account for the sensitivity of handled data and processed queries (**G2**), and thus, the exchange platform consists of two components, a *key server* to handle relevant key material and a *map server*, which records, updates, and returns technology parameter maps without having access to the key material. We thus require non-collusion between these components, as we further discuss in Section 5.4. On a technical level, we rely on PHE (cf. Section 3) to enable the map server to locally, i.e., at the server, alter recorded maps while preserving confidentiality (**G1**). Next, we present the three actions.

Data Provision. By designing a conceptually-centralized platform, we account for the data providers’ desire to offload unique data points only once (cf. Section 2.3). The data provision works as follows. First, ①, the data provider retrieves the PHE key material for the specific technology parameter map she intends to provide from the key server. As part of ②, the data provider homomorphically encrypts her data, which is subsequently ③ send to the map server for further processing and eventual integration/aggregation into the parameter map. MapXchange utilizes the tuple (machine, machine tool, material) to index technology parameter maps. Specifically, it relies on existing standards, such as ISO 13399 [26], which captures the most relevant properties of cutting tools. Similar “ontologies” are available across the entire industrial landscape.

Data Validation. After provision, data must be validated to ensure its genuineness before integrating it into an existing map. The map server cannot assess this genuineness because all sensitive information is encrypted to account for its confidentiality. MapXchange addresses this correctness challenge with offloaded data-validation operations (labeled as Ⓥ in Figure 2), which the map server triggers and also involves the key server. It is independent of the real-world production process, i.e., no knowledge about or access to the machine, machine tool, or material is needed. As a result, depending on the chosen realization, data consumers, data providers, a (trusted) third party, or any combination of these entities may be selected to validate newly-provided data. After receiving a unanimous verdict, the map server integrates said (encrypted) data into existing maps. In Section 4.4, we provide more information on the proposed data-validation mechanism in MapXchange.

When replacing PHE with FHE, the data validation could take place directly on the map server without the need to offload it. However, we argue that the FHE-associated ciphertext overheads (cf. Section 3) do not scale to practical deployments in the IIoT.

Data Queries. While the previous actions facilitate the basis for having technology parameter maps available, any platform is only valuable if it also supports queries. In particular, MapXchange supports two types of queries for its data consumers to use: (i) Regular queries allow them to optimize their setup for a given technology, and (ii) reverse queries enable the selection of a “best-suited” technology by enabling comparisons of multiple technology parameter maps. For more details on their differences, we refer to Section 4.3. Their abstract operation is as follows. After retrieving the respective key material from the key server (Ⓐ) and the (encrypted) map from the map server (Ⓑ), the data consumer decrypts the PHE ciphertexts (Ⓒ) to access and exploit the queried data. Queries make use of same indexing/standards as the data provision.

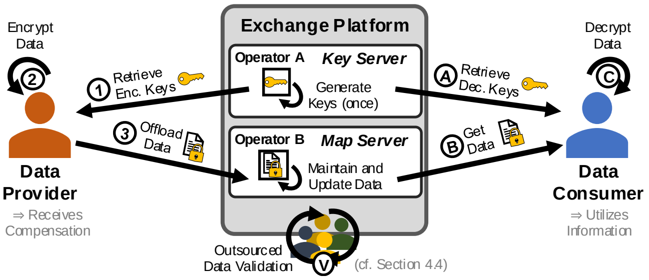

Figure 2: MapXchange features three logical steps: Data provision (①–③; left side), data validation (Ⓥ; bottom center), and data retrieval (Ⓐ–Ⓒ; right side). The two-server exchange platform separates the key material from the encrypted data in the technology parameter maps (i.e., PHE ciphertexts).

To account for **G2**, MapXchange depends on the homomorphic properties of the transferred PHE ciphertexts in two cases. First, technology parameter maps also track the usage per data point businesses report from their production (cf. Section 2.1). Depending on the use case, other details such as motor current or energy consumption could be tracked by the map server as well. The map server maintains these usage statistics by summing up multiple PHE ciphertexts. Second, during data validation and for reverse queries, the map server obfuscates the offloaded data to preserve its confidentiality by adding a random offset to the (encrypted) data points in question. After each successful, outsourced data validation, the map server uses the encrypted, validated data points to update the technology parameter map directly at the server.

Lastly, data providers and their offloaded data are associated with a pseudonym, which enables the map server to track the number of accesses even though the sensitive information remains confidential. This way, MapXchange also implements a compensation module, satisfying the respective aspect in **G1**. Due to its simplicity, we omit the exact billing mechanism in the remainder of this paper.

4.2 Entities and Responsibilities

As shown in Figure 2, MapXchange depends on data providers, data consumers, and the exchange platform itself. Given the business setting, users (i.e., data providers and consumers) have to authenticate themselves toward the platform. An anonymous participation is neither supported nor desired in our setting (cf. Section 2.3).

While data providers and data consumers are logically independent, businesses interacting with MapXchange can take on multiple roles. In the following, we describe the respective entities in detail.

Data Provider(s). Data providers are essential for the exchange platform to gain traction and attract data consumers. While idealistic motives may drive some providers, monetization of their data will convince the majority. The respective compensation is only realistic if the exchange platform offers sufficient data security and protects the stored data accordingly. Therefore, MapXchange builds on a strict separation of key material and encrypted data. Moreover, to reduce the load for data providers, by design, they only have to interact with the platform once when offloading their data.

Data Consumer(s). In contrast, data consumers have a desire to retrieve relevant data from the exchange platform while also keeping their intentions private. Consequently, they also benefit from MapXchange's approach of encrypting sensitive information and separating key material and data. While their interactions with the exchange platform are identical, we still have to logically distinguish "applying clients" and "tool suppliers". The former are interested in improving their own setup by using information handled by MapXchange. The latter have a specific interest in receiving insights into the best-performing parameters, e.g., to further improve their products. To maximize the utility of queried data, they are also interested in obtaining usage data for the respective technology parameters. MapXchange tracks this information by summing up submitted usage data from multiple data providers for the same data point over time (cf. Section 4.1). Data consumers can opt for either submitting regular or reverse queries (cf. Section 4.3).

Key Server. Within the exchange platform, the key server is responsible for generating and handing out key material. The key material is distinct for each technology parameter map. As such, its computational and storage burden is manageable. To ensure confidentiality, it may not collude with any other entity (cf. Section 5.4).

Map Server. Following the distribution of competencies, the map server maintains the (encrypted) technology parameter maps. After receiving new data from any data provider, the map server updates the usage data (homomorphic addition) and schedules the submission for the outsourced data validation (cf. Figure 3). This task includes obfuscating the data before sharing it for validation with a third party. After successful validation, the map server incorporates said data into the existing map. In addition to handling the data provision, it also processes data consumer queries (cf. Section 4.3). Again, MapXchange cannot tolerate any collusion with another entity since access to the key material would allow the map server to look at corresponding data points (cf. Section 5.4).

4.3 Supported Query Types

Data consumers (either "applying clients" or "tool suppliers") can submit queries to the map server to retrieve information from the exchange platform. The map server tracks which data it is returning to later compensate data-providing businesses. As we have outlined in Section 4.1 and Figure 2, the message sequences are identical for both query types, but their content and the intended result differ.

4.3.1 Regular Query. When submitting a regular query, a data consumer is interested in improving its local setup for a specific technology, e.g., when commissioning a new production line. By submitting uniquely-identifying information on the technology parameter map in question, by specifying the tuple (machine, machine tool, material), the data consumer indicates which data he wants to query. The map server accesses the map in question and selects the corresponding (encrypted) data points along with the usage data and returns them to the data consumer. After retrieving the corresponding key material from the key server and decrypting the data, he can then proceed with optimizing its setup (cf. Section 6.2).

4.3.2 Reverse Query. The second supported query type is broader in nature (and thus induces higher costs for the querying data consumer). Specifically, reverse queries allow for answering more fundamental questions by comparing multiple technology parameter maps with each other (hence, reverse). For example, a data consumer might be looking for the best-performing tool for a given material and machine combination, i.e., the machine tool is a wildcard in the query tuple (machine, *, material). Relatedly, tool suppliers are interested in multiple maps and their usage data to (better) understand how tools are being applied in practice. For confidentiality preservation, MapXchange homomorphically adds fixed offsets to the returned data points. This way, data consumers can directly compare the proportions of the retrieved but obfuscated maps with restricted access to all sensitive (and valuable) data points.

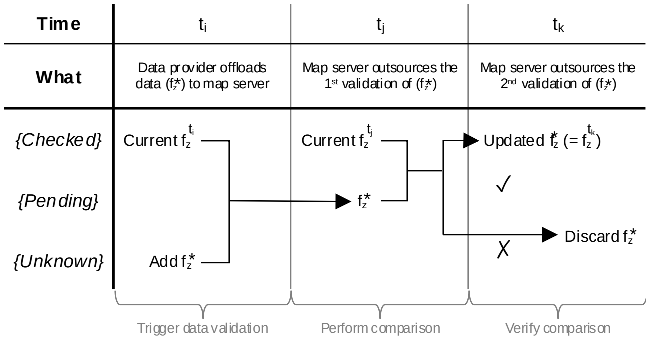
Time
t_i
t_j
t_k
What
Data provider offloads data (f_z^*) to map server
Map server outsources the 1st validation of (f_z^*)
Map server outsources the 2nd validation of (f_z^*)
{Checked}
{Pending}
{Unknown}
Current $f_z^{t_i}$
Add f_z^*
Current $f_z^{t_j}$
f_z^*
Updated f_z^* (= $f_z^{t_k}$)
✓
Discard f_z^*
✗
Trigger data validation
Perform comparison
Verify comparison

Figure 3: After receiving new data, the map server initiates the data validation, which consists of two independent checks. Before outsourcing the check, the map server obfuscates the relevant data to preserve its confidentiality.

Depending on the exchanged technology parameter map, queries may also support disclosing an optimality criterion for the platform to take into account. For instance, a data consumer may be interested in optimizing process stability, productivity, or component quality [9]. Overall, MapXchange supports diverse and feature-rich queries that allow for data provider compensation without (publicly) revealing which data has been exchanged or queried.

4.4 Data-Validation Mechanism

MapXchange's data-validation mechanism is integral to ensuring that provided data updates are genuine, i.e., that this new data should be integrated into a map. Without outsourcing this validation, the platform cannot determine the need for an update because the map server only has access to ciphertexts, and the key server does not have access to the data. We exploit the fact that businesses are interested in having "optimal" data available because they also benefit from it when querying the platform. Thus, they have an incentive to participate in the validation. Since we obfuscate all data before sharing it with third parties (they can only learn the proportions of data points in the maps, e.g., different feed rates f_z, but not the sensitive values themselves), MapXchange does not have specific (trust) requirements regarding validating entities (we consider malicious-but-cautious [47] entities; cf. Section 2.3).

In particular, the map server maintains three sets: (i) *{Checked}* contains the queryable, fully-validated data, (ii) *{Pending}* includes data that has been validated once, (iii) *{Unknown}* is the list of newly-provided data that has not yet been processed. While a FIFO processing of the data points is reasonable, MapXchange does not depend on a specific order for their validation. We only have to ensure that every submitted data point is being validated eventually.

Figure 3 exemplifies the data-validation sequence for a technology parameter map that records forces. Basically, the validation consists of two independent (subsequent) checks. If the "Unknown"-check has been completed, it moves to the "Pending" set. If this check is concluded as well, the verdict is unanimous, and the provided data warrants an adjustment, the map server updates the information in the "Checked" set. If the information is identical, it only updates the recorded usage data. Otherwise, the information is discarded. To ensure genuineness of checks, the map server has to take care that data providers do not validate their own data/updates.

MapXchange preserves confidentiality by utilizing homomorphic encryption as well as separating key material and ciphertexts. To ensure applicability, it (i) allows data providers to offload their data and (ii) supports data consumers to submit regular and reserve queries.

5 Evaluation of MapXchange

We now evaluate MapXchange to assess the goals **G1**, **G2**, and **G3**.

5.1 Implementation and Experimental Setup

To prepare for our evaluation, we now detail how we realized MapXchange and which evaluation environment we prepared.

Implementation. We prototypically realized MapXchange [41] in Python 3. We rely on SQLite and SQLAlchemy to store and access the PHE ciphertexts. For communication, we set up a web framework based on Flask, which provides data providers and data consumers with TLS 1.3-enabled APIs required for operation. Users have to pass HTTP basic authentication, and the servers further log all interactions for access tracking and billing. As PHE cryptosystem, we utilize Paillier [37] (CSIRO Data61's Python library) and configure it with a key length of 2048 bit to achieve 112 bit of security. We assume that the utilized building blocks are secure.

Experimental Setup. All entities run on a single server (Intel Xeon E5-2630 and 32 GB RAM) and communicate using the loopback interface. We report the arithmetic mean of 30 runs per experiment and present 99 % confidence intervals. We measured the reported data transmission volumes with tcpdump.

Use Case Data. To also cover a real-world use case, we extracted 30 optimal feed rate values from 1064 timestamped measurements, which were recorded as part of three milling processes. They employed identical material, machine, and tool configurations. Together with domain experts, we identified the optimal parameters with the approach described in Section 2.1. In our evaluation (Section 5.2.3), data consumers submit queries to retrieve them.

5.2 Performance Evaluation

Given that MapXchange directly operates on ciphertexts, the underlying data is irrelevant when assessing its performance. Since our real-world use case data is limited in size, we initially take it as a foundation for creating a large-scale synthetic dataset. This way, we are also able to holistically assess MapXchange's scalability. Eventually, in Section 5.2.3, we report on our real-world setting.

We structure our discussion into data provisioning and querying.

5.2.1 *Data Provision.*

Data providers can freely decide when and how many data points they want to offload to the exchange platform. Thus, we first discuss the performance of this action before also looking at the—subsequent but independent—data validation.

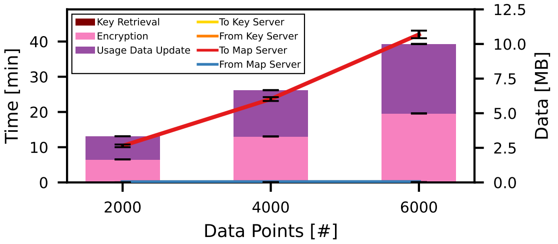

Figure 4: The time to offload data increases linearly with the number of data points. The usage data update, i.e., a PHE addition, only takes place if usage data has been captured before. The data validation is independent of these runtimes.

Providing Maps. As expected, in Figure 4, we observe a linear increase in runtime at the data provider with an increasing number of provided data points (and usage data), which is driven by the PHE encryption process. The ciphertexts also drive the data transmission to the map server, but the Paillier cryptosystem introduces comparably little storage overheads. The remaining processing and communication are negligible. Consequently, MapXchange's data provision scales well even when providing thousands of data points.

By design, i.e., to support the offloading of data, the map server must store the ciphertexts for all indexed maps. Even in the worst case of having to process data in all three sets (*{Checked}*, *{Pending}*, and *{Unknown}*) for 60 000 entries in a single map, less than 250 MB of disk storage is needed. Thus, when simultaneously handling all combinations of 10 materials, 20 machines, and 100 machine tools (i.e., overall 20 000 maps), the map server "only" requires 5 TB, which is manageable. In contrast, the key server's needs are negligible as each map only consumes 512 B, indicating good scalability.

Tracking data consumers' access to technology parameter maps for monetary compensation reveals further negligible storage needs.

Validating Maps. The proposed data-validation mechanism (cf. Section 4.4) barely puts any burden on the validators. Per validation, they only retrieve a private key (512 B) and two ciphertexts (512 B each) from the exchange platform. After decrypting the ciphertexts and comparing the plaintexts, the validators return a short response. Thus, in theory, thousands of data points are validated within minutes, covering data communication, decryption, and comparison. As a result, we consider the proposed mechanism to be practical.

After successful data validation, the map server homomorphically updates (aggregates) the corresponding technology parameter map. This operation is constant per data point and thus also scales linearly with the number of data points (2000: 9.68 ± 0.30 min; 4000: 19.36 ± 1.94 min; and 6000: 28.96 ± 0.26 min). Consequently, the post-validation runtime at the map server is acceptable as well.

5.2.2 *Queries.*

Compared to the data provision, querying technology parameter maps using MapXchange introduces little burden to data consumers and the exchange platform. Again, the runtime and communication costs are driven by the Paillier ciphertexts.

Regular Query. As we illustrate in Figure 5, the runtime for retrieving (querying) data points increases linearly with their number. Key and map servers barely have any computational burden as they simply return information based on the provided index. Thus, even when processing thousands of queries, the queries conclude within minutes, promising scalability to massive industrial settings.

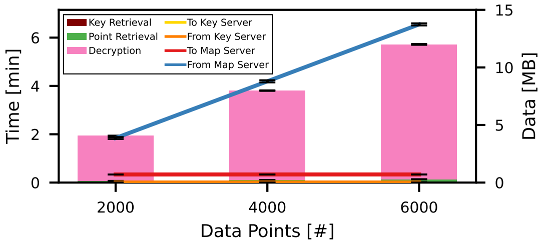

Figure 5: Regular queries exhibit excellent scalability.

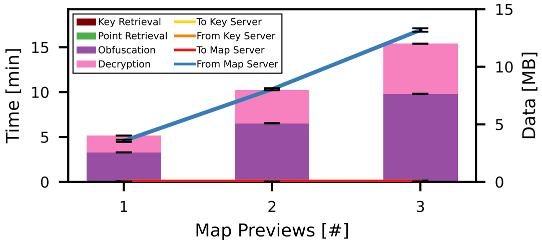

Figure 6: Retrieving maps, with 2000 data points each and obfuscated feed rates, scales linearly in the number of maps.

Reverse Query. Likewise, reverse queries scale linearly with the number of retrieved technology parameter maps (Figure 6). Given that MapXchange obfuscates these maps before returning them (to only share proportions), additional computational load burdens the map server. Hence, from a business perspective, the map server may charge data consumers for reverse queries accordingly. Overall, we believe that the measured performance is equally acceptable.

5.2.3 Real-World Evaluation. After evaluating MapXchange on a large scale with synthetic data, we now provide insights into expectable real-world deployments (cf. Section 5.1). Specifically, for the provision of all data points of a single milling process, we measure a total runtime of only 9.52 ± 0.49 s. Data transmissions with key and map servers take 0.06 MB and 0.23 MB, respectively.

Looking at the query runtime of the entire map, we measure 3.80 ± 0.07 s, which includes the decryption by the data consumer. Since we only had a single map available, we refer to our synthetic evaluation (cf. Section 5.2.2) for insights into reverse queries.

Based on the reported numbers, we attest MapXchange practical performance and good scalability, i.e., it is ready for real-world use.

5.3 Accuracy Assessment

Since operations directly on homomorphic ciphertexts might introduce inaccuracies (cf. Section 3), we also look at the accuracy when performing additions using Paillier. It expresses integers in the range of $[-\text{max\_int}, +\text{max\_int}]$, with $\text{max\_int} = \left\lfloor \frac{n}{3} \right\rfloor - 1$. Given the sole support for integers, we multiply all feed rates and usage data with a fixed constant to remove any floating point numbers. For every n that satisfies the configured key length of 2048 bit (cf. Section 5.1), we can thus accurately express the feed rates and usage data. While performing additions during regular operation, we do not observe any deviations or overflows; all results remain within the range. As such, MapXchange's accuracy is not limited by the use of PHE but rather by the (initial) measurement of parameters.

*Based on this analysis, we conclude that the achieved accuracy of MapXchange conforms with the derived applicability needs (**G1**).*

5.4 Security Discussion

To conclude our evaluation, we now discuss how MapXchange addresses the derived confidentiality goals (**G2**) while simultaneously accounting for the setting (**G1**; applicability). Our design addresses the presented threat model (cf. Section 2.3), which also highlights the incentives of the participating businesses. For example, on a general note, all of them depend on their reputation and are bound to a jurisdiction. Even more, specifically data providers are interested in receiving compensation when using MapXchange. In light of these properties, we have the following considerations regarding the operators of the exchange platform (i.e., key and map server).

Operators. Given that MapXchange's security guarantees build on the separation of key material and ciphertexts, its operators must be trusted to not collude with each other or with the data consumers. Looking at the manufacturing industry, well-known organizations like the *Association of German Engineers (VDI)* [53] and the *Mechanical Engineering Industry Association (VDMA)* [54] could reasonably act as semi-trusted operators. These organizations are usually funded through membership fees and are thus a better fit to serve as operators than random, potentially unreliable third parties. Similar deployment considerations have been raised before [39].

Preserving Confidentiality. At all times, the exchange platform only handles encrypted information, i.e., it is without plaintext access. Given the separation of key material and ciphertexts, compromising a single entity does not lead to a disclosure of sensitive information either. As part of the data validation (cf. Section 4.4), sensitive information is shared with potentially untrusted entities by design. However, due to the applied obfuscation, they may only learn the proportions of two data points without gaining any insights into (sensitive) data. Additionally, solely authenticated data consumers can query for key material and ciphertext. Since they have to compensate data providers for retrieved data, we consider the likelihood of misuse to be low. To still mitigate this slim chance, the platform could optionally enforce some kind of rate limiting.

The Exchange Platform. Both, map and key servers, can (*and should for compensation purposes*; **G1**) track the other parties' access patterns (third parties cannot track queries or access patterns). Thus, in principle, they could create histograms to study the frequency and durability of data points. However, they can neither link this information to plaintext data nor derive whether specific technology parameters or entire maps have really been used in practice (businesses decide for themselves). After discussing with domain experts, we conclude that being aware of maps that are available and which ones are being accessed is not an issue because the existence of corresponding records is not surprising. Hence, only the (encrypted) data recorded within such maps is sensitive.

Threat of Collusion. Overall, the risks associated with collusion are limited. Data providers and data consumers colluding does not reveal new (or unintended) insights. Likewise, a data consumer could also share a ciphertext with the key server or the key server could share a key with the data consumer. However, in both cases, they cannot extract additional information. In contrast, if a data consumer relays a decryption key to the map server, the map server can track the corresponding information (even after updates). Thus, MapXchange cannot tolerate such a collusion. Lastly, as we outlined above, key and map server may not collude. Regardless, even though

MapXchange is conceptually centralized, it allows for scaling out by distributing the ID space, i.e., sets of technology parameter maps, across instances. This way, constraining how much information operators can access is possible (even if collusion occurs).

Side-Channel Information. Since communication with the exchange platform is TLS-protected, interacting with the exchange platform only reveals that communication takes place without leaking details on the content. Consequently, we argue that third parties cannot learn anything meaningful by analyzing such side channels.

MapXchange accounts for and convincingly addresses the security needs of businesses when exchanging technology parameter maps.

6 Discussion: Utility of MapXchange

Looking beyond MapXchange's technical features, we now focus on the bigger picture to stress its added value for the IIoT.

6.1 Drawbacks of Alternative Designs

We identified three conceptual alternatives to realize this exchange platform: First, one could simply offer a look-up service, which allows data consumers to individually contact data providers. However, this approach is infeasible when trying to join data from different data providers into technology parameter maps—✗ **G1**. Second, when relying on secure multiparty computations to jointly maintain the maps, the data providers have to remain online when sharing any data. Moreover, such a design introduces significant performance overhead, especially when increasing the number of involved entities, greatly impairing the scalability—✗ **G3**. Third, maintaining the data in a central data space is not feasible given that the data space either handles the data of multiple providers (see above) or requires plaintext access to sensitive data—✗ **G2**.

Consequently, we argue that our conceptually-centralized *and* confidentiality-preserving approach in MapXchange is inevitable.

6.2 Real-World Impact on Production

MapXchange facilitates the interorganizational transfer of technology parameter maps, thereby enabling the sharing of implicit, domain-specific knowledge even in coopetitive production environments while accounting for the businesses' confidentiality needs. Integrating technology parameters into the planning of processes has the potential to augment the individual expert knowledge of a process planner with the collective knowledge of different businesses, supporting the selection of optimal technology parameters. Specifically, optimal feed rates for a machine, machine tool, and material combination under given engagement conditions are queryable from the exchange platform, i.e., constituting a regular query. At an earlier stage of the process design, technology parameter maps also allow for selecting a suitable (milling) tool for a given component contour through a reverse query, taking into account the machine tool and material. Both queries support the shortening of running-in processes and enable a resource-saving process design to meet the demand for sustainable production. Fortunately, this concept is not limited to the exchange of parameters within milling. Globally establishing such platforms has the potential to capture, aggregate, and exchange domain-specific knowledge across organizations to fuel innovation and to comply with the idea of networked production, as exemplified through recent trends [8, 32].

6.3 Lessons Learned for Data Ecosystems

We see security (guarantees) as a crucial prerequisite when exchanging sensitive and business-critical data between stakeholders in a data ecosystem [27]. While current solutions in data ecosystems rely on organizational security, we show with MapXchange that a technical security solution is applicable, leveraging the sharing, validation, and completion of confidential knowledge. MapXchange can complement the set of services in an IIoT data ecosystem, combining privacy-preserving joint computation as well as efficient and secure querying. We strive for compatibility and exchange with large initiatives, such as IDS [34, 35] or Gaia-X [7], as they offer holistic, full-fledged architectures for data exchange between stakeholders but still lack implementations of privacy-preserving data services—only concepts extending data space connectors have been introduced so far [3]. By design, we avoid that MapXchange presents a single point of failure by separating map and key server and data validation. Given its conceptually-centralized nature, map and key servers can be organized as distributed systems as well. Irrespectively, their responsibilities could also be shared among data ecosystem participants or fulfilled by a trusted third party. While MapXchange presents a practical solution to exchange technology parameter maps, where additive HE is sufficient for completion and validation, the generalization to more complex data structures and purposes requires further substantial research efforts.

6.4 Future Work and Outlook

To complement the previous discussions concerning the utility of MapXchange, we now discuss (potential) future research activities.

Security Improvements. Our design of MapXchange satisfies the confidentiality needs in light of the considered threat model (cf. Section 2.3), as we also outline in Section 5.4. When targeting non-business settings with stronger attackers, we identify two additional measures to improve MapXchange's security. First, the exchange server could periodically replace the key material. Just like the regular data validation (cf. Section 4.4), the map server could outsource the re-encryption to any third party, again requiring an obfuscating of the recorded information. Alternatively, MapXchange could utilize a homomorphic encryption scheme, which supports proxy re-encryption [4], eliminating the need for an outsourced re-encryption. Second, an adapted version of MapXchange could feature oblivious key retrieval, similar to the protocol proposed by BPE [39], effectively concealing the data providers' and data consumers' access patterns. When utilizing oblivious transfers [44], the key server would not be able to observe meaningful patterns.

Holistic View. We primarily researched the confidentiality-preserving exchange of technology parameter maps from a technical point of view. Moving on, research should increasingly study the economic perspective. For example, data consumers could—in theory—simply pay to get access to all recorded data. We leave corresponding game-theoretic questions for social sciences and business experts. On a technical level, implementing rate limiting would be a straightforward countermeasure (cf. Section 5.4).

Universality. When assessing MapXchange's suitability for exchanging information beyond technology parameter maps, we see that our considered data features an optimum, which is also relevant when updating and aggregating information at the exchange platform. Likewise, the viability of our aggregation strategy (cf. Section 4.4) depends on an unambiguous ordering of data points. Hence, future work could investigate whether other use cases, which do not feature unambiguous notions of "optimality", can be adapted for exchanging data using MapXchange. Despite our specific focus on milling, we expect that MapXchange may also be compatible with other use cases (cf. Section 1), pending minor changes.

7 Conclusion

In this paper, we have proposed MapXchange, our approach to exchanging technology parameter maps in the IIoT while simultaneously addressing the confidentiality needs of participating businesses. The insufficient consideration of their needs is the primary reason why they are hesitant to share (sensitive) data, especially across organizational boundaries. We address this issue and thereby complement existing data ecosystems with a secure, homomorphic encryption-based design. Specifically, our work allows for modifications (updates) directly at the exchange platform without the need to locally decrypt sensitive information. Given that MapXchange further supports monetization, offloading data, and two types of queries, it promises utility for practical deployments in production. Our evaluation, which also covers a real-world use case, stresses its adequate performance and scalability for industrial settings.

Acknowledgments

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC-2023 Internet of Production – 390621612.

References

[1] Abbas Acar et al. 2018. A Survey on Homomorphic Encryption Schemes: Theory and Implementation. *ACM Comput. Surv.* 51, 4.

[2] Advaneo GmbH. 2020. Data Marketplace: sharing platform for data analytics, data mining, IoT, ML. https://www.advaneo-datamarketplace.de/en/.

[3] Mehdi Akbari Gurabi et al. 2024. Towards Privacy-Preserving Machine Learning in Sovereign Data Spaces: Opportunities and Challenges. In *Privacy and Identity*.

[4] Reda Bellafqira et al. 2017. Proxy Re-Encryption Based on Homomorphic Encryption. In *ACSAC*.

[5] Dan Boneh et al. 2004. Public Key Encryption with Keyword Search. In *EUROCRYPT*.

[6] Dan Boneh et al. 2013. Private Database Queries Using Somewhat Homomorphic Encryption. In *ACNS*.

[7] Arnaud Braud et al. 2021. The Road to European Digital Sovereignty with Gaia-X and IDSA. *IEEE Netw.* 35, 2.

[8] Philipp Brauner et al. 2022. A Computer Science Perspective on Digital Transformation in Production. *ACM Trans. Internet Things* 3, 2.

[9] Christian Brecher et al. 2023. Sustainability in Production Lines. In *AWK*.

[10] Benny Chor et al. 1995. Private Information Retrieval. In *IEEE FOCS*.

[11] Markus Dahlmanns et al. 2019. Privacy-Preserving Remote Knowledge System. In *IEEE ICNP*.

[12] Xuan-Phuong Dang. 2014. General frameworks for optimization of plastic injection molding process parameters. *Simul. Model. Pract. Theory.* 41.

[13] Data4Industry-X. 2024. Data4Industry-X. https://www.data4industry-x.com/.

[14] Dawex Systems. 2015. Data Exchange Platform. https://www.dawex.com/en/.

[15] Emiliano De Cristofaro et al. 2010. Privacy-preserving Sharing of Sensitive Information. Cryptology ePrint Archive 2010/471.

[16] Berend Denkena et al. 2020. Energy efficient machine tools. *CIRP Annals* 69, 2.

[17] Berend Denkena et al. 2011. *Spanen: Grundlagen.* Springer.

[18] European Commission. 2022. Data Act: Commission proposes measures for a fair and innovative data economy. Press Release IP/22/1113.

[19] Bahar Farahani et al. 2023. Smart and collaborative industrial IoT: A federated learning and data space approach. *Digit. Commun. Netw.* 9, 2.

[20] Piotr Fryzlewicz. 2014. Wild binary segmentation for multiple change-point detection. *Ann. Stat.* 42, 6.

[21] Sandra Geisler et al. 2022. Knowledge-Driven Data Ecosystems Toward Data Transparency. *J. Data Inf. Qual.* 14, 1.

[22] Craig Gentry. 2009. Fully Homomorphic Encryption Using Ideal Lattices. In *ACM STOC*.

[23] Shafi Goldwasser et al. 1984. Probabilistic encryption. *J. Comput. Syst. Sci.* 28, 2.

[24] Bhiksha Gugulothu. 2020. Optimization of process parameters on EDM of titanium alloy. *Mater. Today: Proc.* 27, Part 1.

[25] International Data Spaces Association. 2022. 4.3.10 Privacy Perspective. IDS RAM 4.

[26] International Organization for Standardization. 2006. Cutting tool data representation and exchange. ISO 13399.

[27] Ilka Jussen et al. 2024. Issues in inter-organizational data sharing: Findings from practice and research challenges. *Data Knowl. Eng.* 150.

[28] Nimo Singh Khundrakpam et al. 2020. Optimizing the process parameters of ECM using Taguchi method. *Mater. Today: Proc.* 26, Part 2.

[29] Michael Königs et al. 2018. Process-parallel virtual quality evaluation for metal cutting in series production. *Procedia Manuf.* 26.

[30] Mathew Kuttolamadom et al. 2015. Correlation of the Volumetric Tool Wear Rate of Carbide Milling Inserts With the Material Removal Rate of Ti–6Al–4V. *J. Manuf. Sci. Eng.* 137, 2.

[31] Johannes Lohmöller et al. 2024. The Unresolved Need for Dependable Guarantees on Security, Sovereignty, and Trust in Data Ecosystems. *Data Knowl. Eng.* 151.

[32] Ilaria Mancuso et al. 2024. Value creation in data-centric B2B platforms: A model based on multiple case studies. *Ind. Mark. Manag.* 119.

[33] Michael Naehrig et al. 2011. Can homomorphic encryption be practical?. In *ACM CCSW*.

[34] Boris Otto et al. 2016. *Industrial Data Space: Digital Souvereignity over Data.* White Paper. Fraunhofer.

[35] Boris Otto et al. 2019. International Data Spaces: Reference architecture for the digitization of industries. Springer, Chapter 8.

[36] Boris Otto et al. 2019. Designing a multi-sided data platform: findings from the International Data Spaces case. *Electron. Mark.* 29, 4.

[37] Pascal Paillier. 1999. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In *EUROCRYPT*.

[38] Dorothea Pantförder et al. 2017. Making Implicit Knowledge Explicit – Acquisition of Plant Staff's Mental Models as a Basis for Developing a Decision Support System. In *HCI*.

[39] Jan Pennekamp et al. 2020. Privacy-Preserving Production Process Parameter Exchange. In *ACSAC*.

[40] Jan Pennekamp et al. 2019. Dataflow Challenges in an *Internet* of Production: A Security & Privacy Perspective. In *ACM CPS-SPC*.

[41] Jan Pennekamp et al. 2024. MapXchange. https://github.com/COMSYS/MapXchange.

[42] Jan Pennekamp et al. 2024. An Interdisciplinary Survey on Information Flows in Supply Chains. *ACM Comput. Surv.* 56, 2.

[43] Raluca Ada Popa et al. 2011. CryptDB: Protecting Confidentiality with Encrypted Query Processing. In *ACM SOSP*.

[44] Michael O. Rabin. 2005. How To Exchange Secrets with Oblivious Transfer. Cryptology ePrint Archive 2005/187.

[45] Ronald Rivest et al. 1978. A Method for Obtaining Digital Signatures and Public-key Cryptosystems. *Commun. ACM* 21, 2.

[46] Rosswag GmbH. 2022. AddiMap. https://addimap.com/.

[47] Mark D. Ryan. 2014. Enhanced Certificate Transparency and End-to-end Encrypted Mail. In *NDSS*.

[48] Hanna Shvindina. 2019. Coopetition as an Emerging Trend in Research: Perspectives for Safety & Security. *Saf.* 5, 3.

[49] Emiliano Sisinni et al. 2018. Industrial Internet of Things: Challenges, Opportunities, and Directions. *IEEE Trans. Ind. Informat.* 14, 11.

[50] Dawn Xiaoding Song et al. 2000. Practical Techniques For Searches On Encrypted Data. In *IEEE SP*.

[51] Thomas Usländer et al. 2022. Industrial Data Spaces. In *Designing Data Spaces: The Ecosystem Approach to Competitive Advantage*. Springer.

[52] Marten Van Dijk et al. 2010. Fully Homomorphic Encryption over the Integers. In *EUROCRYPT*.

[53] VDI Verein Deutscher Ingenieure e.V. 2020. VDI – The Association of German Engineers. https://www.vdi.de/en/home.

[54] VDMA e. V. (Mechanical Engineering Industry Association). 2015. The VDMA – VDMA. https://www.vdma.org/en/.

[55] Alexander Viand et al. 2021. SoK: Fully Homomorphic Encryption Compilers. In *IEEE SP*.

[56] Linda Weigl et al. 2023. Mediating the tension between data sharing and privacy: The case of DMA and GDPR. In *ECIS*.

[57] Andrew C. Yao. 1982. Protocols For Secure Computations. In *SFCS*.

[58] Guoqing Zhang et al. 2016. Relationships of tool wear characteristics to cutting mechanics, chip formation, and surface quality in ultra-precision fly cutting. *Int. J. Adv. Manuf. Technol.* 83.

[59] Yang Zhou et al. 2022. Tool wear mechanism, monitoring and remaining useful life (RUL) technology based on big data: a review. *SN Appl. Sci.* 4, 8.

[60] Jan Henrik Ziegeldorf et al. 2017. BLOOM: BLoom filter based Oblivious Outsourced Matchings. *BMC Medical Genom.* 10 (Suppl 2).