Designing Secure and Privacy-Preserving Information Systems for Industry Benchmarking

✉ Jan Pennekamp[1], Johannes Lohmöller[1], Eduard Vlad[1], Joscha Loos[1], Niklas Rodemann[2], Patrick Sapel[3], Ina Berenice Fink[1], Seth Schmitz[2], Christian Hopmann[3], Matthias Jarke[4,6], Günther Schuh[2], Klaus Wehrle[1], and Martin Henze[5,7]

[1] Communication and Distributed Systems, [2] Laboratory for Machine Tools and Production Engineering, [3] Institute for Plastics Processing, [4] Information Systems, and [5] Security and Privacy in Industrial Cooperation. [1–5] *are affiliated to RWTH Aachen University*, Aachen, Germany. [6] Fraunhofer FIT, Germany, and [7] Fraunhofer FKIE, Germany.

Abstract. Benchmarking is an essential tool for industrial organizations to identify potentials that allows them to improve their competitive position through operational and strategic means. However, the handling of sensitive information, in terms of (i) internal company data and (ii) the underlying algorithm to compute the benchmark, demands strict (technical) confidentiality guarantees—an aspect that existing approaches fail to address adequately. Still, advances in private computing provide us with building blocks to reliably secure even complex computations and their inputs, as present in industry benchmarks. In this paper, we thus compare two promising and fundamentally different concepts (hardware- and software-based) to realize privacy-preserving benchmarks. Thereby, we provide detailed insights into the concept-specific benefits. Our evaluation of two real-world use cases from different industries underlines that realizing and deploying secure information systems for industry benchmarking is possible with today's building blocks from private computing.

Keywords: real-world computing · trusted execution environments · homomorphic encryption · key performance indicators · benchmarking.

1 Introduction

Benchmarking is either a one-time or continuous process of identifying best practices to improve the performance using indicators [17]. More precisely, *industry benchmarking* is an essential tool for businesses to identify potentials by comparing themselves internally, with partners, or competitors through specific key performance indicators (KPIs). It provides companies with insights into the effectiveness of their processes (qualitatively and quantitatively). Further, it allows them to identify processes that are worthwhile to improve to close the gap between the market leader (best in class) and their own position, e.g., by avoiding

a waste of resources [17,30]. To this end, we can distinguish between two common types of industry benchmarks, i.e., internal (involving departments of a single company) and external benchmarking (comparing multiple companies) [17].

From an information systems' perspective, benchmarks build on distributed information systems (ISs) with the goal of improving the overall performance (expressed through use case-tailored KPIs), within certain peer groups [14].

Overall, we identify three crucial dimensions when designing such systems: (1) *benchmarking frequency* (one-time vs. continuous benchmarking), (2) *openness of data* (i.e., open vs. closed data), and (3) *openness of the algorithm* (i.e., open vs. closed benchmarking algorithms). Traditionally, benchmarks utilized labor-intensive, manual interviews [17, 23] to collect the required data to compute the defined KPIs by following a fixed algorithm, i.e., we can classify them as one-time benchmarks that source closed data and a closed algorithm. Data-driven approaches increasingly evolve toward continuous benchmarking systems, which might additionally rely on public (open) data and algorithms (e.g., governmental applications). As such benchmarks (open data and open algorithms) do not require elaborate security mechanisms, they are comparably easy to realize.

However, benchmarks in industry require strong security as they operate on sensitive (closed) company data using valuable, use case-tailored, and complex (closed) algorithms. Without sufficient security guarantees, companies fear a loss of control over their sensitive data, and in turn, of their competitive advantages. Thus, privacy-preserving benchmarks increase the number of companies willing to participate, which significantly affects the utility of benchmarks and the revenue of the analyst (the developer of the underlying algorithm). Moreover, such systems also help to ensure the confidentiality of the algorithm. Thereby, they protect the analyst's intellectual property, counteracting potential losses of subsequent compensations through unauthorized and unpaid reuse of algorithms.

Given these confidentiality needs, realizing privacy-preserving industry benchmarks is challenging. Fortunately, developments in the area of private computing provide us with two diametric concepts for designing such privacy-preserving systems. On the one hand, *Trusted Execution Environment*s (TEEs) [27] provide hardware-based guarantees through dedicated computing enclaves for the private computation on sensitive data. On the other hand, *Fully Homomorphic Encryption* (FHE) [2] schemes are a software-based approach that enables privacy-preserving computations on encrypted data without revealing any details of the computation or its inputs. The question is which direction is best-suited.

In this paper, we answer this question by studying the suitability and applicability of two diametrical designs to benchmark organizations in industry, i.e., a hardware-based TEE and a software-based FHE approach, with the latter basing on our prior work [23]. Using real-world use cases in the domains of injection molding and global production networks, we evaluate our designs qualitatively and quantitatively. These use cases originate from the interdisciplinary research cluster "Internet of Production" [7, 22], which connects researchers from various domains and more than 30 institutes. Thus, we holistically discuss the foundations for and implications of such privacy-preserving information systems.

Contributions. Our primary contributions in this paper are as follows.

- We study two diametrical designs[1] that secure ISs for industry benchmarking with modern concepts from private computing, i.e., using TEEs and FHE.
- We discuss the performance, limitations, and security guarantees of each design to give an intuition of the real-world implications, i.e., we provide a holistic overview for practical deployments of such information systems.

Organization. In Sec. 2, we first introduce industry benchmarking, two real-world use cases, and relevant related work. Subsequently, in Sec. 3, we present the two diametrical private computing concepts, which we build upon when describing our two benchmarking designs (Sec. 4). In Sec. 5, we evaluate these information systems also in light of our use cases before concluding in Sec. 6

2 Company Benchmarking in Industry

The need for privacy-preserving benchmarks motivates us to study existing concepts that promise to secure corresponding information systems. In Sec. 2.1, we first provide essential aspects of industry benchmarking, its actors, benefits, and privacy requirements. Subsequently, in Sec. 2.2, we detail two real-world use cases, which also serve as a basis for our evaluation (Sec. 5), before discussing past efforts to realize secure and privacy-preserving industry benchmarks in Sec. 2.3.

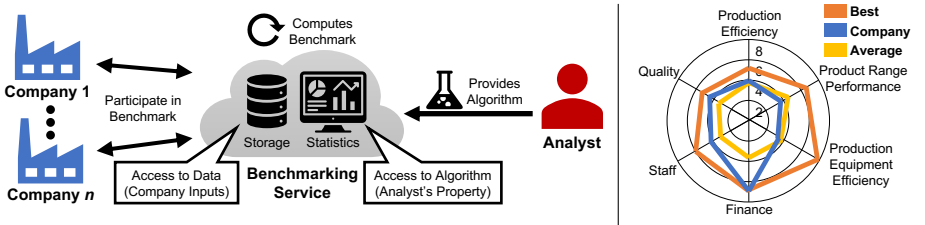
2.1 Company Benchmarking 101

Industry benchmarking usually focuses on practices such as the company's operations and the management of a company or a department [19]. The main objectives are to evaluate the company's current market position to identify the gap between the company and a recognized leader, as well as to adapt local processes to close this identified gap. For example, Xerox, a manufacturer of photocopiers and document management systems, improved its annual productivity gains from 3 %–5 % to 10 % [31] after comparing its processes with L.L. Bean, a retailer of outdoor sporting goods, and addressing the benchmark's findings.

Benchmarks operate on key performance indicators (KPIs), allowing for quantitative comparisons of products, services, or implemented practices [11, 14]. Thus, we can also understand KPIs as digital shadows [5, 12, 18], i.e., an abstraction that represents the companies' performance. Nowadays, the sets of relevant KPIs frequently change. For instance, they increasingly cover sustainability, which also allows for comparisons of environmental and social aspects [6].

Fig. 1(a) illustrates the process of benchmarking, including the main actors: an analyst, the benchmarking service, and participating companies. First, the analyst develops suitable algorithms to compute meaningful KPIs, which are usually kept private due to their value and intellectual property [10]. For example, the business models of credit scoring agencies, such as Experian or Schufa, largely depend on the confidentiality of the algorithm. The benchmarking service

[1] We open-sourced them at `https://github.com/COMSYS/industry-benchmarking`



(a) Benchmarking relies on data and the algorithm (b) Visualization of KPIs

Fig. 1: Using the algorithm and the companies' inputs, the benchmarking service computes all target KPIs, which allows companies to judge their performance.

collects the corresponding inputs from participants and computes the KPIs to compare them as part of the benchmark. Eventually, the participants receive the general results and their own KPIs. Companies can then investigate their performance in comparison to the average and "best in class", as shown in Fig. 1(b).

Previous work [4, 14, 29] frequently outlined the need for confidential KPIs. The prevalence of closed benchmarking algorithms stresses the need to also protect the computation of the compared KPIs as it represents the analyst's competitive advantage who invests significant effort to derive meaningful KPIs [23]. We, therefore, identify two crucial privacy requirements that need to be considered: (1) the provided company inputs *and* computed company-specific KPIs, which we define analogous to personal privacy as *company privacy*, and (2) *algorithm confidentiality*. Mitigating data leaks is a significant challenge. Accordingly, company data should not even be accessible to the benchmarking service. However, neither should the algorithm be public nor (partly) accessible to the companies.

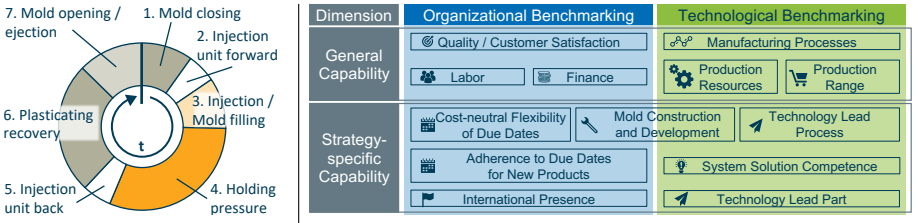
As a further complication, the KPI computation can be very complex in real-world benchmarks, i.e., a single KPI can be based on several formulas with dependencies, diverse operations, and hundreds of inputs [23]. Overall, a single benchmark may consist of up to 200 KPIs [14], thus, demanding computational resources for its operation. Next, we take a look at two real-world use cases.

2.2 Real-World Use Cases of Company Benchmarking

In this paper, we consider two real-world use cases, which we introduce next.

Benchmarking Companies in Injection Molding (IM). Our first use case is injection molding as primary shaping: It is widely applicable in different industries and domains and allows for the processing of complex part geometries without subsequent rework. The raw plastic material is plasticized by heat and friction and then injected into the mold, which is the negative of the plastic part to be produced. After a pre-defined cooling time, the final part can be ejected from the mold [13]. Given the multitude of steps within a single cycle and their sensitivity, injection molding is a highly complex process, as shown in Fig. 2(a).

This use case bases on a real-world benchmark from 2014 that still utilized a centralized, paper-based approach [23] (cf. Sec. 2.3). Back then, companies in the injection molding industry compared their performance in organizational and



(a) Process steps of an injection molding cycle

(b) The benchmark evaluates organizational and technological dimensions through specifically derived KPIs.

Fig. 2: Our real-world benchmarking use case in the domain of injection molding captures the complexity of the production process through various KPIs.

technological aspects to advance or consolidate their positions (cf. Fig. 2(b)). The underlying algorithm and hence most of the resulting KPIs are highly specialized for this domain. As we detail in Tab. 1, the complexity of this example is high, with computations in up to 49 sequential operations, over 600 inputs, and more than 2700 operations, i.e., the analyst's effort is significant. In addition to elementary arithmetic, the KPI computation also sources exponentiation (x^y), roots ($\sqrt[y]{x}$), as well as absolute ($|x|$) and extrema values (min/max). In return, participants received detailed results due to the large number of KPIs (Fig. 2(b) highlights some of them). As such, this use case is a representative real-world example, and it is along the lines of the number of expected KPIs (cf. Sec. 2.3).

This benchmark must be privacy-preserving as it builds on private inputs and computes sensitive KPIs. Here, prominent examples are data comprising costs of labor or manufacturing processes. For an in-depth presentation of the setup and the benchmark itself within this use case, we refer to previous work [23].

Measuring the Efficiency of Global Production Networks (PN). Our second use case [24] benchmarks the performance of production sites in globalized production networks to exemplarily study another setting and a different type of algorithm. Distributing production sites and supply chains can yield significant advantages as the geographic, regulatory, and technological conditions of each location can be exploited best [32]. However, a competitive advantage is only given if the beneficial performance is being attested regularly, e.g., through benchmarks where companies compare their inventory, efficiency, and equipment over different days, products, and production sites. The data necessary to generate the KPIs for comparing companies' production networks requires highly sensitive company data. Thus, it must be treated confidentially, as it would allow others to draw conclusions about the corporate strategy and relationships [16].

Table 1: Overview of our two real-world use cases and their algorithm complexity.

Dataset	Inputs	KPIs	Depth (Max.)	Depth (Avg.)	Formulas	Operations
IM	674	48	49	12	627	2704
PN	35 (n-dim)	14	12	6	14	100

In this context, benchmarking the performance of individual production sites is particularly interesting to compare the efficiency of companies or even locations within a single company. For example, a KPI can express the unit costs of a product at a specific location for this purpose. By breaking down the unit product costs, companies can then identify the main drivers, such as the degree of automation, the wage level, or even the characteristics of the machine park. In this use case, the product portfolio complexity has been identified as a major driver of unit costs, which can be traced back to the need to interrupt production sequences with setup processes, resulting in reduced machine utilization.

In comparison to IM, the underlying algorithm of PN features three interesting differences for the design of an IS: (1) arrays as input values with variable length, which might implicitly reveal sensitive company details, (2) component-wise operations on arrays, and (3) summation (Σ) or extrema over arrays. Hence, despite its small size, with 14 KPIs and 100 operations (cf. Tab. 1), it is of great relevance for our work due to the complex operations contained within.

With these real-world use cases, we intend to provide a holistic view on the features any suitable design must adhere to while studying their implications. In the following, we look at past efforts in realizing secure benchmarking systems.

2.3 Related Work in Securing Company Benchmarks

Traditional benchmarking services utilize a centralized design that digitizes the paper-based responses of participants before computing the KPIs and comparing them [23]. Apart from their labor-intensive realization, such benchmarking services conceptually serve as a trusted third party as they have access to all sensitive inputs. Such centralized designs protect the algorithm but fail to account for the sensitive company inputs (*company privacy*). In contrast, local computations by the participants, who only return the computed KPIs, protect sensitive inputs but fail to account for the required *algorithm confidentiality*. In a general direction, advances in privacy-preserving data processing emerge in research [1]. However, they frequently build upon disclosing the algorithms as well. Ongoing developments in the area of private computing promise to privacy-preservingly secure industry benchmarks while reliably mitigating this critical drawback.

Software-based Approaches with Private Computing. In related work, we discover several software-based designs utilizing secure multi-party computation or homomorphic encryption. The former approaches usually have two major drawbacks: (1) they are commonly round-based, i.e., all participants need to participate simultaneously [4, 14], and (2) the scalability is, at best, quadratic [4,15] in the number of participants. Initial homomorphic encryption-based approaches [28, 29] come with a limited set of supported operations that challenge the computation of complex operations directly on encrypted data. These approaches have in common that they do not consider *algorithm confidentiality* [23], i.e., they only protect the comparison of KPIs but fail to account for the sensitivity of the KPI computation (the analyst's intellectual property).

In 2020, improved FHE schemes allowed us to propose a Privacy-preserving Company Benchmarking (PCB) [23], which considers *both* privacy requirements.

Using FHE-encrypted inputs, PCB locally computes simple operations on encrypted data and locally compares encrypted KPIs, achieving *company privacy*. While PCB offloads complex computations to the participants, it allows for algorithm obfuscation to probabilistically ensure *algorithm confidentiality*. In this work, we refer to an increment of PCB with fewer needs of offloading as SW-PIB.

Hardware-based Approaches. On the other side of the spectrum of private computing, we have hardware-based concepts, such as TEEs, which only emerged after the *majority* of software-based benchmarking approaches had already been proposed. The range of applications that utilize TEEs to securely execute programs is immense, with them also moving toward mobile devices, such as smartphones, these days. However, we did not discover any approach that utilizes TEEs to secure industry benchmarks. Thus, in this work and based on their secure computing enclaves, we study the opportunities hardware-based designs offer for the secure realization of benchmarking information systems.

3 Preliminaries: Recent Advances in Private Computing

Given the huge potential of private computing for benchmarking information systems, we now introduce the ideas and variants of the corresponding diametrical extremes, i.e., hardware- and software-based concepts, in more detail. Afterward, we study two designs for privacy-preserving industry benchmarking that base on these fundamentally different concepts to evaluate their practical feasibility.

Trusted Execution Environments (TEEs) [27] shield and protect confidential data and code during the processing via *hardware* mechanisms that are implemented in modern CPUs. The core idea is that confidential data and code remain inaccessible for the untrusted part of the system. To this end, TEEs allow attesting the hard- and software for validity such that users then share their data securely with a trusted party. For benchmarking, we can utilize these properties to protect the inputs and the analyst's algorithm. Using remote attestation, participants can verify the configuration of the system they are sending data to, as well as details on the software running in the TEE. Intel SGX is a popular and widely available implementation of this concept, which is also present in commodity hardware and today's cloud environments, such as Microsoft Azure [25].

Fully Homomorphic Encryption (FHE) [2] is a *software-based* approach that relies on the homomorphic property of special cryptosystems that allow for operations on the ciphertext to also be reflected in the plaintext, i.e., FHE enables computations directly on encrypted data even on untrusted hardware. Particularly, the widespread adoption of cloud computing contributes to the increasing application of FHE. FHE schemes can have varying cryptographic foundations that differ in terms of supported operations and encrypted data types (e.g., Booleans, integers, or approximated reals), each with individual overhead and constraints [2]. After a certain number of operations on a single ciphertext, ciphertexts need to be refreshed: either interactively using the owner's key pair or through (local) bootstrapping. In practice, the ideal design choice depends on the specific confidentiality needs and availability of computing resources.

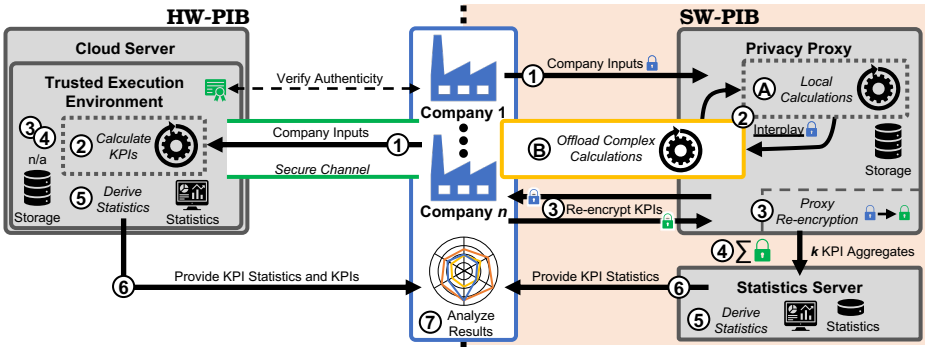


Fig. 3: Our hardware security-based approach **HW-PIB** (left) can be realized with a single server. Contrary, our software security-based approach **SW-PIB** (right) uses two non-colluding servers to ensure the confidentiality of inputs and the valuable algorithm. Eventually, companies analyze the benchmarking results.

4 Privacy-Preserving Company Benchmarking Designs

We propose two reference designs (**H**ard**w**are- or **S**oft**w**are-based) for **P**rivacy-preserving **I**ndustry **B**enchmarking (PIB), i.e., HW-PIB and SW-PIB, to study their suitability for real-world information systems through qualitative and quantitative analysis. Apart from the frequently addressed company privacy, our designs also consider algorithm confidentiality (cf. Sec. 2.1). In Sec. 4.1, we provide a high-level overview to express the general processing steps. Subsequently, we discuss crucial details of HW-PIB and SW-PIB in Sec. 4.2 and 4.3, respectively.

4.1 Design Overview

The main difference between our designs lies in the underlying private computing concept (hardware vs. software-based). While TEEs can retain inputs and computed KPIs of each company within the protected enclave in HW-PIB, SW-PIB's privacy proxy only operates on encrypted data, and the statistics server only has access to aggregates. Designing and evolving the actual benchmarking algorithms is entirely independent of our designs, which focus on securing the *operation* of benchmarking algorithms. Thus, the development of benchmarking algorithms remains unchanged. Conceptually, the logical steps to compute a benchmark are identical in our designs, and the overall steps are largely comparable. However, the individual realizations differ significantly. Thus, we provide a high-level description at this point. We visualize both designs in Fig. 3.

In ①, the participating companies share their inputs with the benchmarking service. In HW-PIB, the companies send their sensitive data through a secure (TLS) channel directly into the TEE. In contrast, SW-PIB requires the participants to homomorphically encrypt their inputs with their own public keys. Subsequently, in ②, using the analyst's algorithms, the KPIs are computed. While HW-PIB operates directly on plaintext data within the TEE, SW-PIB

deals with ciphertexts: Thus, in SW-PIB, depending on the operation, the computation is either Ⓐ performed (locally) on the privacy proxy if supported by the FHE scheme or Ⓑ it is offloaded to the participant. We refer to offloading as the process where the participant receives the operation and ciphertext(s) from the privacy proxy to (1) decrypt the input ciphertext(s), (2) compute the operation on the decrypted plaintexts, (3) homomorphically encrypt the result, (4) and return it to the privacy proxy. Thereby, we circumvent the restricted set of FHE-supported computations on ciphertexts and provide analysts with the flexibility to include arbitrary operations in the benchmarking algorithms.

Steps ③ and ④ are only relevant for SW-PIB as HW-PIB directly operates on plaintext data within the protected enclave, i.e., no additional security measures are needed. First (③), the ciphertexts must be re-encrypted with the statistics server's key. Depending on the underlying FHE scheme, we can either utilize proxy re-encryption directly on the proxy or we have to offload the re-encryption to the company. Second (④), the privacy proxy aggregates the KPIs of k participants [23] that are all encrypted with the statistics server's key and forwards these aggregates to the statistics server, which can decrypt them.

⑤–⑦ are again identical for HW-PIB and SW-PIB. The benchmarking service derives the KPI statistics (⑤) and shares them with the companies (⑥). Finally, in ⑦, companies analyze their results to derive management decisions.

Next, we look at the designs' specifics and our prototypical implementations.

4.2 HW-PIB: Shielding the Computations

HW-PIB, our hardware-based design, utilizes TEEs to process the companies' sensitive inputs while preserving confidentiality. The design builds on the isolation property of TEEs together with memory encryption and storage sealing to restrict the access to sensitive information to software within the enclave.

Setup. Since the enclave has access to company inputs as plaintext data, the setup first needs to establish trust between the running enclave, the analyst, and participating companies. This trust includes (a) the correct and benign functionality of code running inside the enclave and (b) that the enclave actually runs the intended software on a trustworthy platform. We resolve (a) by open-sourcing the enclave code, such that any interested entity can verify its functionality, and (b) via remote attestation by a trusted certificate authority. Upon successful attestation, it issues and signs an enclave-specific certificate. This certificate serves as an enclave identifier and proves successful attestation to all entities who connect via a secure channel. The analyst and the companies then provision the enclave with their configuration and data (①), respectively.

KPI Computation. Due to the use of a trusted enclave, the TEE may have access to all data in plaintext. Hence, HW-PIB locally supports arbitrary complex operations (②) and does not require any offloading. The TEE's memory encryption ensures that the input and all intermediate computation results remain confidential, i.e., they are only accessible by/within the enclave itself.

Aggregation. Due to HW-PIB's computations on plaintexts, it does not require any preparatory aggregation steps (③–④). Instead, HW-PIB directly

calculates the KPI statistics (⑤). Together with their individual KPIs, in ⑥, the general statistics are sent to the companies via TLS. Afterward, the enclave may terminate to ensure that any data and the KPIs are no longer accessible.

Remarks. As a hardware-based design, HW-PIB depends on a TEE-enabled cloud server, which various vendors offer. In this work, we utilize Intel SGX.

4.3 SW-PIB: Realizing Oblivious Computations

Now, we focus on the specifics of SW-PIB and the implications of utilizing FHE.

Setup. During the setup, the analyst (cf. Fig. 1) configures the privacy proxy (by sharing the algorithm and configuring k). Moreover, the statistics server generates an FHE key pair that is used to compute the aggregates in ④. Finally, each participant must generate an FHE key pair as well (used in ①–③).

KPI Computation. To ensure algorithm confidentiality, the privacy proxy tries to compute as many operations on ciphertexts as supported locally (Ⓐ). The support for complex operations (i.e., beyond $+$,$-$, and $\cdot$) depends on the utilized FHE scheme. Accordingly, unsupported operations need to be offloaded to the client (Ⓑ). Here, the analyst may configure obfuscation strategies (cf. [23]). This continuous interplay (②) concludes once all KPIs have been computed.

Aggregation. The realization of ③ depends on the support of proxy re-encryption in the utilized FHE scheme: Either the KPI re-encryption (to encrypt with the statistics server's key) is offloaded to the participant (who simultaneously learns its own KPIs), or the re-encryption is performed locally at the proxy (while the encrypted KPIs are shared to the participant for decryption). Once the KPIs of k companies have been aggregated (④), these aggregates are then sent to the statistics server, which combines them with existing statistics in ⑤. Eventually, in ⑥, the general statistics are retrievable for all participants.

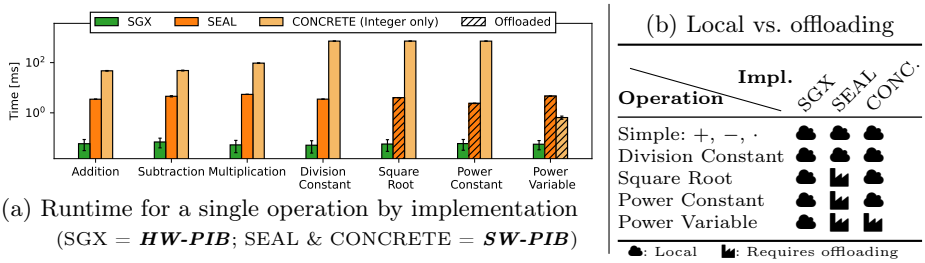
Remarks. In SW-PIB, we have no requirements on the required hardware, as data is protected through a software-based (FHE) approach. However, this design comes with limitations of the locally supported FHE operations. Furthermore, separating privacy proxy and statistics server is crucial to prevent the decryption of (unaggregated) ciphertexts that contain sensitive company inputs or KPIs.

5 Evaluating Secure Industry Benchmarking Systems

To evaluate the feasibility of our discussed designs, we study their performance (Sec. 5.1) using synthetic measurements and two real-world benchmarking algorithms. We further discuss their respective security guarantees (Sec. 5.2) and compare them (Sec. 5.3). With this overview, we provide insights into concept-specific benefits and their applicability for benchmarking information systems.

5.1 Performance and Overhead Evaluation

We conduct our evaluations using our open-sourced implementations of the designs. In particular, we focus on the performance of the KPI computation as it



(a) Runtime for a single operation by implementation (SGX = ***HW-PIB***; SEAL & CONCRETE = ***SW-PIB***)

Fig. 4: The locally supported (complex) operations differ across implementations.

covers the majority of relevant operations. We do not report any numbers on the aggregation phase (③–⑤) due to its low computational footprint.

Implementation. For HW-PIB, we utilize Scone [3], running on Intel SGX. For SW-PIB, we (i) re-implemented and extended PCB [23], which builds on Microsoft SEAL [20], with array computations, as required by our PN use case, and (ii) built a proof of concept that employs CONCRETE [8]. Since Microsoft SEAL does not (yet) implement proxy re-encryption, we resort to offloading in Step ③. However, this limitation of SEAL is not a conceptual issue in SW-PIB.

Experimental Setup. Our implementations run on a commodity computer with moderate resources (Intel i7-7700 with 16 GB RAM and a regular SSD). All entities communicate over the loopback interface. We conduct 50 runs for each measurement, compute the mean, and calculate 99 % confidence intervals. We rely on 128 bit-level security. In SEAL, we configure polynomial moduli of 16 384 (7 levels) and 8192 (4 levels) for IM and PN, respectively (cf. Sec. 2.2). To ensure consistency across the reported numbers and avoid bias in our results, we followed the same evaluation methodology for all conducted experiments.

Performance. To assess the performance, we have to look at the setup and run times. We observe that the setup times are negligible ((17.893 ± 0.015) s for HW-PIB and (3.424 ± 1.164) s for SW-PIB). Looking at the run times, we first investigate the performance of single operations. As we illustrate in Fig. 4(a), these synthetic measurements show that HW-PIB is one order of magnitude faster than our SEAL- and CONCRETE-based implementations of SW-PIB, and it does not require offloading (Fig. 4(b)). The performance of the CONCRETE-based version will likely deteriorate once additional datatypes, such as floats or larger integers, are supported. Still, we want to emphasize the potential of programmable bootstraps, i.e., this FHE scheme supports additional complex operations without the need for offloading. While the overhead of computing FHE ciphertexts at the privacy proxy is already significant, the need to offload operations further slows down SW-PIB; especially for constrained network links.

Moving to our real-world examples, we notice that both designs are practical for real-world deployments. For the larger IM example (cf. Tab. 1), HW-PIB and SW-PIB finish after (0.115 ± 0.019) s and (634.008 ± 0.538) s, respectively. Thus, from a suitability perspective, analysts could even offer significantly larger yet confidential benchmarks. In contrast, our PN example is an order of magnitude faster, with (0.080 ± 0.001) s and (34.409 ± 0.044) s. Overall, the runtimes

for single operations, as we have illustrated in Fig. 4(a), amplify in real-world benchmarks. Hence, the performance of SW-PIB remains inferior to HW-PIB.

Accuracy. While HW-PIB is exact by design, our SEAL-based implementation of SW-PIB uses approximate arithmetic, i.e., when processing floats, we encounter precision losses. As we perform computations on approximated numbers, the precision loss amplifies, especially for long chains of operations. When using SW-PIB and suffering from insufficient accuracy, the benchmarking algorithm can be tweaked to better fit the precision of the utilized FHE scheme. For example, numbers can be scaled to account for precision losses of approximate ciphertext representations. Nonetheless, SW-PIB is feasible as we only observe minor deviations. Overall, we measure $(4.0 \pm 0.3)\,\%$ for IM and $<0.01\,\%$ for PN.

Ciphertext Overhead. HW-PIB does not introduce noteworthy storage and network overhead by design. Thus, we now focus on SW-PIB: Relying on FHE introduces storage and network overhead due to comparably large ciphertext sizes. For IM, we measure a size of at most 1.842 MB for a single ciphertext, i.e., even the up- and download of thousands of ciphertexts (when sharing inputs or during offloading) is feasible over bandwidth-constrained network links. Hence, ciphertext overheads do not prohibit real-world applications of SW-PIB.

Moving on, we discuss our designs' security before comparing them in detail.

5.2 Security Discussion

From a security perspective, we expect malicious-but-cautious entities [26], i.e., they want to extract as much information as possible without leaving any traces of the extraction. This assumption is especially reasonable in scenarios with businesses that are bound to specific legislation. Consequently, we exclude collusion attacks that involve multiple entities. Next, we look at our two designs in detail.

HW-PIB. The security builds upon hardware-based security. Consequently, the hardware vendor must be trusted, i.e., it serves as the root of trust. Using remote attestation (a key feature of TEEs), we can establish a trust chain to the enclave and the code running within it. Hence, participants only have to verify this chain and the running code using certificates and cryptographic signatures. If the security has been correctly attested, all information and computations are shielded within the TEE. Thus, in this case, HW-PIB is secure by design. However, the multitude of (past) vulnerabilities in TEEs [9] could negatively impact the trust in this technology. Consequently, we also consider SW-PIB.

SW-PIB. This design protects the company inputs and intermediate results using FHE. Its security builds on the privacy proxy and statistics server not colluding. Then, the privacy proxy never has access to any information in plaintext as it lacks the corresponding decryption keys. Moreover, the statistics server only receives aggregates of k participants, i.e., it cannot deduce any details about specific companies if k is reasonably large (i.e., $k > 3$) [23]. Hence, sensitive company data is protected (encrypted) at all times. In both designs, the algorithm is never shared with the participants. However, in SW-PIB, we need to offload specific complex operations (cf. Sec. 4.3). Thus, fractions of the algorithm along with their intermediate results need to be shared with the participants,

Table 2: Comparison of hardware- and software-secured benchmarking designs.

Criteria \ Design	HW-PIB		SW-PIB			
	SGX		SEAL		CONCRETE	
Setting	IM	PN	IM	PN	IM	PN
Performance	★★★		★★☆		★★☆	
▶ Setup	Remote attestation		Exchange of key material			
▶ Run Time [s]	0.11 ± 0.02	0.08	634.0 ± 0.5	34.4	Unknown	
Accuracy Loss [%]	Exact		4.0 ± 0.3	0.0	Unknown	
Ciphertext Overhead	★★★		★☆☆		★★☆	
▶ Offloading [#]	None		↓1487 ↑745	↓53 ↑28	↓84 ↑42	↓0 ↑0
▶ Networking [≤×MB]	No overhead		1.842	1.053	Unknown	
Ease of Use	★★★		★★½		Unknown	
Security	★★★		★★☆		★★★	
▶ Assumptions	Trusted hardware		Secure FHE scheme			
▶ Trust in Participants	Not required		Non-collusion required			
▶ Privacy Issues	None		Minor (offl.)		Barely any (offl.)	
▶ Own KPIs	After/with agg.		Before agg.		After/with agg.	

slightly violating the intended algorithm confidentiality. To mitigate these implications of offloading, the benchmarking service can utilize different obfuscation strategies, such as dummy requests, blinding, and request randomization [23]. Consequently, SW-PIB ensures the privacy needs of real-world benchmarks.

5.3 HW-PIB vs. SW-PIB: Selecting the Fitting Design

We compare both diametrical security concepts when realizing benchmarking information systems in Tab. 2 to give a concise overview and to allow for well-founded deployment decisions. Now, we briefly summarize the specific properties.

Performance. The benchmarking setup is a one-time task and thus negligible with times below 18 s. Given that benchmarking is not an everyday task, the run time for each company is more than suitable for real-world applications, even with significantly larger benchmarks. The real-world use cases further underline this claim (IM: (634.008 ± 0.538) s and PN: (34.409 ± 0.044) s). The TEE- and FHE-induced overheads are reasonable in light of the confidentiality benefits.

Accuracy. HW-PIB features exact computations by design, and the loss of precision for SW-PIB is tolerable as (i) the deviations affect all companies and (ii) benchmarks primarily concern the relative positioning [19]. Moreover, the evaluated real-world algorithms were not tailored to the use with FHE. Given that the inaccuracies follow from small numbers [23], the analyst could easily scale the inputs and formulas to mitigate such deviations to a large extent.

Ciphertext Overhead. In addition to the noticeable ciphertext overhead in SW-PIB, we further have to rely on offloading to compute a subset of complex computations locally at the companies. Recent advances, such as CONCRETE [8], even promise to reduce the required offloading. Regardless of such advances, companies receive more ciphertexts than they send, which fits to the

imbalance of Internet connections. Even with ciphertext sizes of 1.842 MB, for IM, the upload of 1.391 GB per company is feasible in constrained networks.

Ease of Use. We consider our designs to be practical for real-world use as companies can easily participate through common web browsers. While our implementation of HW-PIB natively features a web-based client, we have shown in previous work [23] that our SEAL-based implementation supports WebAssembly-based web clients as well. Concerning reoccurring operational costs, HW-PIB only requires a server with TEE support, which is commercially available at major (cloud) vendors. In contrast, FHE-based SW-PIB does not introduce specific hardware requirements, but its operations are computationally more expensive.

In real-world deployments, analysts could operate the cloud server and privacy proxy, respectively, and fund them through participation fees. If needed, our designs support scaling out the cloud server and privacy proxy, respectively, e.g., to support a tremendous number of participants. Aside from that, industry associations could fund the statistics server in SW-PIB using their membership fees to prevent collusion attacks [23]. Generally, HW-PIB is cheaper to operate with fewer overheads *if TEEs are trusted*, compared to FHE-based SW-PIB.

Security. HW-PIB requires specific hardware for its operation and is secure and privacy-preserving if the trusted hardware is realized as intended. Given that companies establish a secure tunnel into the enclave, HW-PIB reliably protects the algorithm, all inputs, and the computed KPIs. In contrast, SW-PIB does not depend on specifically trusted hardware but on secure and properly configured FHE schemes. We further require non-collusion between privacy proxy and statistics server to ensure company privacy. Obfuscation strategies can help to prevent offloading-induced information leaks (companies have access to the operator and intermediate data). As indicated before, modern FHE schemes promise to further reduce the required offloading. While our SEAL-based SW-PIB uses offloading for the KPI re-encryption (③), which enables companies to abort the protocol (then, they only have access to their KPIs), implementations that support proxy re-encryption provide the same security properties as HW-PIB.

Takeaways. Nowadays, concepts from private computing are readily available to secure information systems in real-world deployments. Thus, when designing secure information systems for industry benchmarks, the key question is which conceptual technology should serve as the root of trust, i.e., trusted hardware or a secure FHE scheme, mainly because the remaining properties do not prohibit practical realizations, as we briefly summarize in the following.

Looking at the performance, both designs fulfill the needs of real-world benchmarks, with HW-PIB computationally outperforming SW-PIB. While HW-PIB's accurate computations promise quick and precise results, SW-PIB is easier to deploy as it is designed for untrusted hardware (despite requiring two entities, i.e., privacy proxy and statistics server). Thus, industry should indeed be able to offer secure and privacy-preserving benchmarks in practice. The exact realization (design) then likely depends on the availability of a TEE and the willingness to build on its associated security assumptions (e.g., trusting the underlying security concept, the vendors, and remote attestation). Otherwise, FHE-based

implementations promise secure and practical real-world deployments. For now, we recommend our revised SEAL-based version, but in the future, CONCRETE-based implementations with fewer offloading needs could outperform it.

6 Conclusion

In industry, companies frequently rely on industry benchmarks to identify potentials that allow them to improve their competitive position through strategic and operational adjustments. Given its benefits, industry benchmarking is a valuable tool. However, benchmarks depend on valuable information: First, the underlying complex formulas to compute meaningful key performance indicators (KPIs) constitute the analyst's intellectual property and must be kept private. Likewise, the KPI computation sources sensitive company inputs. The inputs, as well as the KPIs, must thus remain confidential. Consequently, to increase the number of participants, industry benchmarks must be operated privacy-preservingly.

Confidentiality requirements have hindered the wide use of corresponding information systems (ISs) so far. Given the latest advances in private computing, we compared two fundamentally different concepts (hardware- and software-based security) to realize privacy-preserving ISs that are capable of offering real-world industry benchmarks while ensuring both algorithm confidentiality and company privacy. Our corresponding designs each offer concept-specific benefits: While the performance of HW-PIB and its accurate computations promise quick and precise results, SW-PIB is easier to deploy and does not depend on specific hardware or its associated security guarantees. Our evaluation of two real-world industrial use cases (IM & PN) demonstrates that secure benchmarking deployments are practical with today's concepts from private computing. In the future, we look forward to the rapid evolution of private computing and its implications on information systems beyond our application in industry benchmarking.

Acknowledgments. Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC-2023 Internet of Production – 390621612. We thank Jan-Gustav Michnia for his initial exploration of the FHE library CONCRETE [8]. We followed an abstract research methodology [21] to structure and organize our research collaborations.

References

1. van der Aalst, W.M.P.: Federated Process Mining: Exploiting Event Data Across Organizational Boundaries. In: IEEE SMDS (2021)
2. Acar, A., Aksu, H., et al.: A Survey on Homomorphic Encryption Schemes: Theory and Implementation. ACM Comput. Surv. **51**(4) (2018)
3. Arnautov, S., Trach, B., et al.: SCONE: Secure Linux Containers with Intel SGX. In: USENIX OSDI (2016)
4. Becher, K., Beck, M., Strufe, T.: An Enhanced Approach to Cloud-based Privacy-preserving Benchmarking. In: NetSys (2019)

5. Bibow, P., Dalibor, M., et al.: Model-Driven Development of a Digital Twin for Injection Molding. In: CAiSE (2020)
6. Boos, W.: Production Turnaround — Turning Data into Sustainability. Tech. rep., RWTH Aachen University (2021), white Paper
7. Brauner, P., Dalibor, M., et al.: A Computer Science Perspective on Digital Transformation in Production. ACM Trans. Internet Things **3**(2) (2022)
8. Chillotti, I., Joye, M., et al.: CONCRETE: Concrete Operates oN Ciphertexts Rapidly by Extending TfhE. In: WAHC (2020)
9. Fei, S., Yan, Z., et al.: Security Vulnerabilities of SGX and Countermeasures: A Survey. ACM Comput. Surv. **54**(6) (2021)
10. Gunasekaran, A., Putnik, G.D., et al.: An expert diagnosis system for the benchmarking of SMEs' performance. Benchmarking **13**(1–2) (2006)
11. Herrmann, D., Scheuer, F., et al.: A Privacy-Preserving Platform for User-Centric Quantitative Benchmarking. In: TrustBus (2009)
12. Jarke, M.: Data Sovereignty and the Internet of Production. In: CAiSE (2020)
13. Kamal, M.R., Isayev, A.I., Liu, S.J.: Injection Molding: Technology and Fundamentals. Hanser (2009)
14. Kerschbaum, F.: Practical Privacy-Preserving Benchmarking. In: IFIP SEC (2008)
15. Kerschbaum, F.: Secure and Sustainable Benchmarking in Clouds. Bus. Inf. Syst. Eng. **3**(3) (2011)
16. Kerschbaum, F., Oertel, N., Weiss Ferreira Chaves, L.: Privacy-Preserving Computation of Benchmarks on Item-Level Data Using RFID. In: ACM WiSec (2010)
17. Kozak, M.: Destination Benchmarking: Concepts, Practices and Operations. CABI (2004)
18. Liebenberg, M., Jarke, M.: Information Systems Engineering with DigitalShadows: Concept and Case Studies. In: CAiSE (2020)
19. Marti, J.M.V., d. R. Cabrita, M.: Entrepreneurial Excellence in the Knowledge Economy: Intellectual Capital Benchmarking Systems. Palgrave Macmillan (2012)
20. Microsoft, Inc.: Microsoft SEAL. https://github.com/Microsoft/SEAL (2018)
21. Pennekamp, J., Buchholz, E., et al.: Collaboration is not Evil: A Systematic Look at Security Research for Industrial Use. In: LASER (2021)
22. Pennekamp, J., Glebke, R., et al.: Towards an Infrastructure Enabling the Internet of Production. In: IEEE ICPS (2019)
23. Pennekamp, J., Sapel, P., et al.: Revisiting the Privacy Needs of Real-World Applicable Company Benchmarking. In: WAHC (2020)
24. Rittstieg, M.: Einflussfaktoren der Leistungsfähigkeit von Produktionsstandorten in globalen Produktionsnetzwerken. Ph.D. thesis (2018)
25. Russinovich, M.: Azure confidential computing. `https://azure.microsoft.com/en-us/blog/azure-confidential-computing/` (2018 (accessed March 20, 2023))
26. Ryan, M.D.: Enhanced Certificate Transparency and End-to-end Encrypted Mail. In: NDSS (2014)
27. Sabt, M., Achemlal, M., Bouabdallah, A.: Trusted Execution Environment: What It is, and What It is Not. In: IEEE TrustCom (2015)
28. Sahin, C., Kuczenski, B., et al.: Privacy-Preserving Certification of Sustainability Metrics. In: ACM CODASPY (2018)
29. Sobati-Moghadam, S., Fayoumi, A.: Private Collaborative Business Benchmarking in the Cloud. In: SAI (2018)
30. Teicholz, E.: Facility Design and Management Handbook. McGraw-Hill (2001)
31. Tucker, F.G., Zivan, S.M., Camp, R.C.: How to Measure Yourself Against the Best. Harv. Bus. Rev. **65**(1) (1987)
32. Verhaelen, B., Mayer, F., et al.: A comprehensive KPI network for the performance measurement and management in global production networks. Prod. Eng. (2021)