Offering Two-Way Privacy for Evolved Purchase Inquiries

JAN PENNEKAMP, RWTH Aachen University, Germany
MARKUS DAHLMANNS, RWTH Aachen University, Germany
FREDERIK FUHRMANN, RWTH Aachen University, Germany
TIMO HEUTMANN, Fraunhofer IPT, Germany
ALEXANDER KREPPEIN, Fraunhofer IPT, Germany
DENNIS GRUNERT, Fraunhofer IPT, Germany
CHRISTOPH LANGE, Fraunhofer FIT, Germany and RWTH Aachen University, Germany
ROBERT H. SCHMITT, RWTH Aachen University, Germany and Fraunhofer IPT, Germany
KLAUS WEHRLE, RWTH Aachen University, Germany

Dynamic and flexible business relationships are expected to become more important in the future to accommodate specialized change requests or small-batch production. Today, buyers and sellers must disclose sensitive information on products upfront before the actual manufacturing. However, without a trust relation, this situation is precarious for the involved companies as they fear for their competitiveness. Related work overlooks this issue so far: Existing approaches only protect the information of a single party only, hindering dynamic and on-demand business relationships. To account for the corresponding research gap of inadequately privacy-protected information and to deal with companies without an established trust relation, we pursue the direction of innovative privacy-preserving purchase inquiries that seamlessly integrate into today's established supplier management and procurement processes. Utilizing well-established building blocks from private computing, such as private set intersection and homomorphic encryption, we propose two designs with slightly different privacy and performance implications to securely realize purchase inquiries over the Internet. In particular, we allow buyers to consider more potential sellers without sharing sensitive information and relieve sellers of the burden of repeatedly preparing elaborate yet discarded offers. We demonstrate our approaches' scalability using two real-world use cases from the domain of production technology. Overall, we present deployable designs that offer two-way privacy for purchase inquiries and, in turn, fill a gap that currently hinders establishing dynamic and flexible business relationships. In the future, we expect significantly increasing research activity in this overlooked area to address the needs of an evolving production landscape.

CCS Concepts: • **Security and privacy** → **Privacy-preserving protocols**; *Domain-specific security and privacy architectures*; • **Applied computing** → Engineering;

Authors' addresses: Jan Pennekamp, pennekamp@comsys.rwth-aachen.de, Communication and Distributed Systems, RWTH Aachen University, Aachen, Germany; Markus Dahlmanns, dahlmanns@comsys.rwth-aachen.de, Communication and Distributed Systems, RWTH Aachen University, Aachen, Germany; Frederik Fuhrmann, fuhrmann@comsys.rwth-aachen.de, Communication and Distributed Systems, RWTH Aachen University, Aachen, Germany; Timo Heutmann, timo.heutmann@ipt.fraunhofer.de, Production Quality, Fraunhofer IPT, Aachen, Germany; Alexander Kreppein, alexander.kreppein@ipt.fraunhofer.de, Production Quality, Fraunhofer IPT, Aachen, Germany; Dennis Grunert, dennis.grunert@ipt.fraunhofer.de, Production Quality, Fraunhofer IPT, Aachen, Germany; Christoph Lange, christoph.lange-bever@fit.fraunhofer.de, Data Science and Artificial Intelligence, Fraunhofer FIT, Sankt Augustin, Germany, Information Systems and RWTH Aachen University, Aachen, Germany; Robert H. Schmitt, r.schmitt@wzl.rwth-aachen.de, Machine Tools and Production Engineering, RWTH Aachen University, Aachen, Germany, Production Metrology and Quality Management and Fraunhofer IPT, Aachen, Germany; Klaus Wehrle, wehrle@comsys.rwth-aachen.de, Communication and Distributed Systems, RWTH Aachen University, Aachen, Germany.

1533-5399/2023/11-ART53 $15.00
https://doi.org/10.1145/3599968

Additional Key Words and Phrases: bootstrapping procurement; secure industrial collaboration; private set intersection; homomorphic encryption; Internet of Production

ACM Reference Format:
Jan Pennekamp, Markus Dahlmanns, Frederik Fuhrmann, Timo Heutmann, Alexander Kreppein, Dennis Grunert, Christoph Lange, Robert H. Schmitt, and Klaus Wehrle. 2023. Offering Two-Way Privacy for Evolved Purchase Inquiries. *ACM Trans. Internet Technol.* 23, 4, Article 53 (November 2023), 30 pages. https://doi.org/10.1145/3599968

1 INTRODUCTION

Digitalization has already had a significant impact on (existing) supply chains [6, 31, 58], especially on their management. Mainly, by relying on additional information, the decision-making concerning the planning and adjustment of flows of goods improved throughout the complete supply chain [48–51]. Novel concepts, such as the Internet of Production [15, 69], suggest to further extend these advances by inter and intra supply chain information exchange between stakeholders, including competitors [70]. Thus, shaping the underlying communication architecture and utilized application layer protocols is crucial to enable innovation and to ensure a secure evolution of manufacturing and production. Due to this increased utility of information and advantages of data sharing [29, 70], these concepts even envision dynamic relationships that allow companies to further improve their production [46]. The expected benefits cover improved product quality, reduced costs, enhanced sustainability, and establishing relationships with the most suitable suppliers. In this context, relevant supplier evaluation criteria include, e.g., timeliness, quality, sustainability, or price [19, 73].

The future of manufacturing is thus primarily driven by specific customer requests or small-batch production. A foundation for the manufacturing of (custom) products is the availability of relevant parts and components. Given the diversity in small-batch production, companies are likely to flexibly source parts from different suppliers—many of which they do not have an established trust relationship with. Thus, these envisioned dynamic relationships require companies to repeatedly discover suitable business partners [46]. However, currently, appropriate solutions are still missing as today's approaches fail to account for the privacy needs in such competitive environments [67]. We are not aware of any work that focus on the companies' privacy during the initial steps procurement. Most work covers subsequent steps of procurement, such as auctions [13, 54, 56], electronic markets [41], or private e-tendering [63], where sensitive information had to be (partially) shared already. When looking at the applicability of existing approaches from other domains, such as advertising [32, 34], we notice that they only focus on privacy needs of a single party, i.e., in our setting, they fail to protect the information of both buyers and sellers. Hence, one party has to reveal its intentions upfront even if no business relationship is established eventually. Consequently, they are unsuitable for an application as part of the envisioned privacy-preserving procurement.

We identify the following privacy needs. On the one hand, buyers do not want to reveal any information on products they intend to acquire, especially not to previously unknown sellers. On the other hand, sellers do not want to reveal their full product catalog in advance. To the best of our knowledge, we are the first to explicitly focus on this research gap. Accordingly, in light of the requirements of future manufacturing and small-batch production, companies are in need of a privacy-preserving approach to securely establish buyer-seller relationships between previously unaffiliated (and possibly mutually distrustful) companies in an effort to account for the risk-free establishment of dynamic and flexible relationships in an evolved production landscape.

In this work, motivated by two real-world use cases in the domain of machine tools, we first investigate the privacy needs when establishing new relationships. Building upon well-established building blocks in the domain of privacy-preserving computing, e.g., private set intersection [22] or homomorphic encryption [2], we propose two designs that technically ensure two-way privacy

for purchase inquiries with slightly different privacy and performance implications (to offer use case-specific flexibility). Given that solutions must not only respect buyer and seller privacy, but must also scale to industry needs, we extensively evaluate our real-world use cases and demonstrate our work's feasibility. With our work, which is oblivious of concrete use cases and domains, we make an important step to turn the vision of an Internet of Production [15] into reality: We provide a crucial building block that enables companies to conveniently establish dynamic business relations even with untrusted parties by conducting purchase inquiries privacy-preservingly and reliably.

Contributions. Our main contributions can be summarized as follows.

- We are the first to summarize the privacy issues originating from dynamic buyer-seller relationships, and based on this work, we derive a universally-valid set of five design goals.
- Utilizing private computing, we propose two approaches to realize the innovative idea of privacy-preserving purchase inquiries. We focus on real-world settings and show that our work has no invasive (or negative) impact on today's well-established procurement processes.
- We open-source our implementations and use case data to the public [68].
- With our practical, real-world feasible approaches and the corresponding discussions, we make an important step toward a secure, sustainable, and efficient production landscape by enabling a risk-free establishment of dynamic buyer-seller relationships.

Paper Organization. In Section 2, we introduce procurement processes, purchase inquiries, and our real-world machine tool use cases in more detail. Subsequently, in Section 3, we derive universally-valid design goals for privacy-preserving purchase inquiries. Then, we introduce relevant building blocks that are commonly known from private computing (Section 4). In Section 5, we detail our designs and present their implementations in Section 6. We demonstrate the corresponding performance, security, and applicability, including our real-world use case, in Section 7. Then, in Section 8, we discuss related work, and we conclude our paper in Section 9.

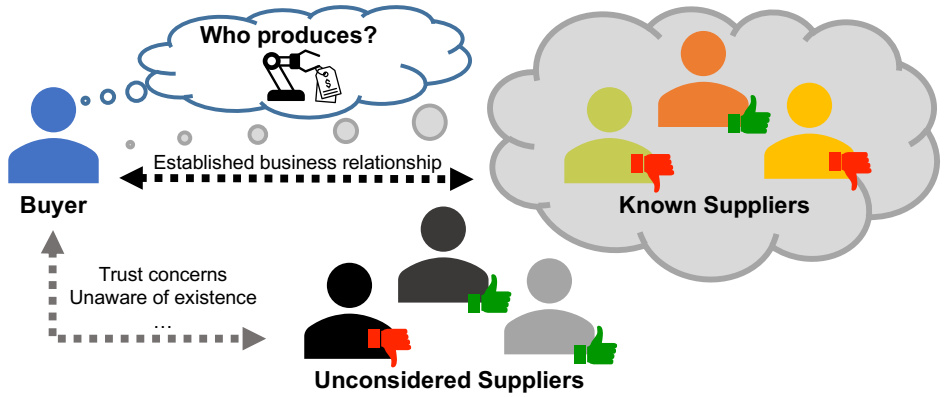

Fig. 1. Buyers tend to re-use their existing business networks when looking for suitable suppliers. Due to privacy concerns (the need to share information upfront), other, potentially superior suppliers are excluded.

2 SCENARIO

In this section, we motivate the need for improved privacy in purchase inquiries by enabling dynamic business relationships that are currently hindered due to privacy concerns. We further illustrate the associated benefits of using real-world machine tool use cases.

2.1 Purchase Inquiries

We first introduce our view of purchase inquiries, before highlighting today's purchase behavior.

As part of the procurement process [57, 88] in business-to-business markets, interested companies, so-called *buyers*, contact potential suppliers (*sellers*) and inquire about specific resources, parts, or products [19]. While we refer to this step as a *purchase inquiry*, it is also known as request for quotation (RFQ) [19, 25]. On a higher level, the primary goal of this step is to find a supplier who can satisfy the requested order. If several suppliers are able to satisfy the order, additional criteria, e.g., the price, can be used to eventually select a single seller, effectively avoiding price discrimination by sellers [84]. Most notably, a purchase inquiry is only a single step in the process of identifying, managing, and integrating suppliers. For an in-depth introduction of today's well-established procurement processes in business-to-business markets, we refer to a summary in Appendix A.

In Figure 1, we illustrate a typical purchase inquiry situation: A buyer is interested in a specific product, e.g., a robotic arm, and contacts all relevant suppliers in the field that she has worked with in the past. To this end, she provides them with detailed specifications about the robotic arm, such as rotation angle, lifting capacity, and supported communication protocols. After their replies, which indicate whether they are able to deliver the requested product, the buyer might be able to select a suitable supplier. However, due to the focus on her existing network of suppliers, a large number of other, potentially superior suppliers is excluded from this inquiry step. Related work confirms that finding suitable suppliers is a major issue [33]. To mitigate this situation, centralized platforms promise to simplify this matching [33]. However, they learn all details, which constitutes a significant breach of privacy. Even without such platforms, buyers have privacy concerns, i.e., they want to avoid sharing sensitive information with companies without a previous relationship [3, 5, 55]. Furthermore, they might not even be aware of their existence or supported capabilities. Thereby, when making such decisions, the buyer currently has a limited view of all available options.

Likewise, potential sellers are also dissatisfied with sharing sensitive information upfront. As their counterpart, they have to openly share details on their capabilities, their delivery times, and price expectations even if no trust relationship has been established. This knowledge could potentially benefit their competitors to outmatch their offers. Therefore, privacy in purchase inquiries is a two-way street [5] and should be treated accordingly. Otherwise, the risk-free establishment of relationships with the goal of selecting the most suitable supplier(s) will not succeed.

Zeng et al. [94] outline the different types of (sensitive) information in detail and hence confirm the need for corresponding approaches. To overcome the issue of neglecting many suitable business partners, privacy preservation during procurement is thus a crucial aspect. A corresponding solution should mitigate this issue, address any concerns, and improve today's established "buyer-seller" matchmaking. Apart from an improved matching, further benefits could follow from the handling of specialized products [27] and the ability to swiftly react toward customer change requests [70].

We discovered the discussed privacy issues and needs in collaboration with industry practitioners from Fraunhofer IPT. The relevance and innovativeness of this issue has been confirmed (in writing) in (confidential) off-the-record conversations by various industrial actors, ranging from registered research associations to registered research communities focusing on quality improvements in industry.

2.2 Discovering (New) Machine Tool Suppliers

To date, the procurement process in companies is a critical, privacy-invasive task. Although a contractual relationship has not yet been established, the companies must exchange sensitive information. Consider, for example, an automotive manufacturer (OEM) who wants to launch a new model variant on the market. Thus, the OEM requires various machine tools for specific production tasks. Car manufacturers rely on an on-time delivery of such machine tools, and on privacy when developing new models. This privacy already applies when ordering machine tools, as the tools are individualized for the model and often contain critical data regarding technological advances [92].

To allow a matching of the buyer's requirements and specifications of the machine tool with the capabilities of a potential seller, both parties must exchange confidential information that both parties prefer not to exchange. For example, on the one hand, the buyer does not want to disclose anything that might reveal aspects of her future model(s) or technological advances that are incorporated in the machine tool. On the other hand, the supplier does not want to reveal her entire product catalog and capabilities. This situation leads to a dilemma: the more accurate the information, the better the offer. However, at the same time, the more accurate the information, the more sensitive details have to be disclosed. In practice, this dilemma has two consequences: Either the buyer does not contact the potential supplier at all, or the customer contacts only a few suppliers whom the supplier trusts based on past experiences. This pre-selection severely restricts the market and excludes a number of potentially more suitable suppliers, impacting their business.

Trade between companies is key to a functioning market economy. However, today's concerns during procurement severely limit this process. Two-way privacy during purchase inquiries would significantly benefit both buyers and (potential) suppliers in practice.

Nowadays, companies are pursuing a wide range of procurement strategies. Unfortunately, all strategies are somehow problematic for companies, especially when dealing with goods that are worth protecting (and not just "everyday parts"). In the case of machine tool suppliers, machine tools are complex goods that require a variety of highly accurate machining, and thus increase the added value. This situation demands the protection of related purchase inquiries. Consequently, some companies rely on close partnerships with only a few trusted suppliers (single sourcing) [93], leading to a strong increase in dependency [40]. Likewise, such behavior severely limits the potential relationship in terms of quality, functionality, innovation, and costs [40]. Other companies diversify widely, at the risk of "leaking" (disclosing) information. Today, a wide range of strategies also exists in-between: Many companies use indirect methods, such as signing a non-disclosure agreement (NDA) early on [88]. However, such practices do not protect the data directly, i.e., detecting breaches or attributing them, especially when offers are obtained from many companies, is very challenging.

To summarize, the process for requesting offers is characterized by the distrust of the involved parties. To receive a useful offer, both companies have to exchange sensitive data, which they prefer not to disclose, especially when dealing with untrusted parties. Therefore, (technical) means are needed to sufficiently address their confidentiality needs. Ultimately, such approaches would enable fairer and broader competition industry-wide, exceeding far beyond the directly involved buyers and sellers.

3 DESIGN GOALS FOR IMPROVED PRIVACY

Based on our considered scenario, we now derive a set of five distinct, universal design goals, which must be addressed by any approach that improves the privacy of purchase inquiries. These goals summarize the needs of the individual parties (**G1** and **G2**) as well as universal conceptual requirements (**G3**, **G4**, and **G5**).

G1: Buyer Privacy. Companies are interested in keeping sensitive information on their business practices and orders private [5, 55], especially in light of untrusted third parties. Consequently, they are only willing to reveal this information to their suppliers, i.e., in our scenario, the deliberately selected seller. They want to avoid sharing anything upfront with other parties, especially if no business relationship is established after all. This information is not limited to the requested product or its specification but also includes other sensitive criteria, such as their price expectations or preferred delivery schedules. Likewise, buyers cannot tolerate any linking of their queries. Overall, buyers are concerned with leaking valuable data, fearing a loss of their competitiveness.

G2: Seller Privacy. Even though buyers are more likely to share information upfront (cf. **G1**), sellers also have an interest in privacy. Today, their privacy is at stake in two ways [5]. First, sellers reply to purchase inquiry requests with specific offers. Apart from the effort invested in this possibly

unrewarding task (if no sale is closed), made offers reveal a variety of sensitive details. For example, they contain the company's production capabilities, its declared price, and potentially additional insights into available production resources or schedules. Thus, competitors might be able to derive the sellers' profit margins or other valuable details, eventually providing them with means to undercut offers. Second, currently, sellers might resort to publicly announcing their catalog (cmp. consumer mail-ordering businesses) to attract business, i.e., their privacy is similarly affected.

G3: Protocol Resistance. The realization of two-way privacy for purchase inquiries further demands secure approaches, i.e., colluding parties should not be able to extract any additional information, especially about third parties. Similarly, the result of any privacy-preserving purchase inquiry must be sound, i.e., no manipulation must be possible at any time. This requirement primarily concerns the comparison of price expectations, as otherwise, sellers could pretend to provide goods at every price to extract the requesting buyer's price limit. Furthermore, falsely advertised products by a seller would be immediately noticeable to the buyer as the subsequent negotiation (cf. Appendix A) would fail right away (the "matched" seller cannot provide said products). Regardless, such unsound results would also diminish the value of and the trust in the protocol, i.e., sensible approaches should prevent such attacks to ensure technology acceptance.

G4: Applicability. Given our focus on real-world settings, any approach must satisfy the constraints of our representative use cases. Namely, this goal covers both performance and scalability, where scalability refers to the number of products that are globally comparable and the number of contactable sellers in a specific period. Solutions not fulfilling these requirements might only be able to improve the companies' privacy while failing to significantly improve the status quo.

Since purchases are started well in advance (usually providing several weeks of buffer), mainly due to the manual effort that is needed and to account for delivery delays, having an (automated) protocol run conclude within a single day constitutes a reasonable upper bound. If conducting a privacy-preserving purchase inquiry is not feasible, the envisioned dynamic business relationships remain impractical as they would incur significant overhead. Thus, to profit from all benefits, e.g., the ability to react to customer change requests, improved product quality, and lower costs, any proposed approach must scale to real-world needs. Thus, this goal is key for any solution's success and its technology acceptance (in industry).

G5: Ease of Use. To stress parts of the previous goal (**G4**), we explicitly model the ease of use as a distinct goal. In particular, we demand the independence of the involved parties, i.e., a purchase inquiry should not be bound to a fixed set of potential sellers. Instead, buyers should be able to contact sellers on demand and as needed, e.g., if no satisfying match has been made or the subsequent negotiations cannot be concluded. On a related note, to mitigate any concerns stemming from the chosen setup and to ease real-world deployment, solutions should avoid using a centralized service. For privacy reasons, companies are reserved to use a centralized service to manage their purchase inquiries [18]. All in all, this addition aligns neatly with **G3**, which mandates preventing all sorts of information leaks, because it effectively limits the number of involved stakeholders in a single protocol run to a minimum, i.e., by excluding uninvolved parties. Moreover, a direct bilateral protocol between a buyer and a single seller most likely reduces the load on the sellers (cf. **G4**) as they are only contacted if needed. Consequently, we argue that approaches to improve the privacy in purchase inquiries should avoid a trusted third party (to limit the threat of data leaks) and should not make use of multi-party computation (to improve the flexibility and to avoid round-based runs with fixed sets of potential sellers), i.e., we call for flexible, bilateral approaches.

Non-Goals. In this work, we primarily focus on approaches that allow buyers and sellers to extend their established network of business relationships in business-to-business markets without fearing any leaks of sensitive information. In particular, we do not want to replace today's procurement processes, contract negotiations, or approaches for bidding on products or prices

in any way. Consequently, fuzzy queries are uncalled-for as buyers know the product properties that they inquire about. Instead, we intend to augment these established approaches with an intermediate step to establish new relationships without revealing sensitive information upfront.

These derived design goals are crucial when designing a reliable and secure solution that attempts to improve the privacy in purchase inquiries for all involved parties. We argue that they are universally valid and independent of the domain in which the procurement process takes place.

4 CRYPTOGRAPHIC PRELIMINARIES

To establish a common background of our utilized building blocks, we briefly introduce their specifics in this section. For our designs (cf. Section 5), we either rely on a private set intersection (PSI) and order-revealing encryption (ORE), or homomorphic encryption (HE).

Private Set Intersection (PSI) is a cryptographic primitive that allows two parties to calculate the intersection of two confidential sets without revealing the included elements to the other party [22]. Depending on the implementation, only one or both parties learn the content or the size of the intersection [21]. PSIs utilize different concepts: For improved security, many efficient designs utilize oblivious transfers [45, 74, 77] that introduce processing and networking overhead. For flexibility, other PSI variants rely on RSA and Bloom filters [44] without impairing the security.

Order-Revealing Encryption (ORE) [11] extends the concept of order-preserving encryption (OPE) [4], and provides a tool that allows for efficient range queries and sorting on encrypted data [16, 43]. ORE addresses some of the limitations of OPE, e.g., to mitigate inference attacks and prevent information leakage of the "encrypted" plaintext, and accordingly intends to only reveal the order of the encrypted plaintexts [11]. Thus, the server only learns the ordering of the ciphertexts.

Homomorphic Encryption (HE) allows for calculations on encrypted data without requiring access to the underlying raw data, thus maintaining data confidentiality [2]. Thus, HE is commonly used for private computing or secure outsourcing [53, 75, 86, 95]. Concerning our scenario, HE promises to perform computations on sensitive company data without revealing any details of the inputs, i.e., companies do not have to trust a third party. Different variants of HE feature distinct implications on usability and performance, including Fully Homomorphic Encryption (FHE) [28, 83] and Partially Homomorphic Encryption (PHE) [30, 62, 78]. While FHE allows a larger set of operations, it introduces computational overhead, additional storage needs, and decreased accuracy [2]. PHE is limited in the allowed operations, but implies fewer required resources [2].

5 SECURE BILATERAL PURCHASE INQUIRIES

We now propose our two designs to realize two-way privacy in purchase inquiries. Thereby, we address our outlined research gap and intend to move toward an evolved production landscape. Concerning the required security level in our scenario, we focus on a setting with semi-honest (participating) companies and service providers as they live off their reputation and must follow the law. Our work is not limited to specific deployments given that it is oblivious of the handled products. We only require a deterministic modeling of production properties to ensure their unambiguity.

5.1 Notation for Purchase Inquiries

To establish a common foundation for our design, we now introduce a semi-formal definition of purchase inquiries and the properties of each involved party. While our definition focuses on a single Buyer B who considers n potential Sellers $S_1, \ldots, S_n$, our proposed, bilateral protocols only handle a single seller S_i at a time. We can handle multiple independent buyers by parallelizing protocol runs conveniently. Thus, corresponding designs scale to any real-world setting at hand.

Informally, on the one hand, the buyer has a query q that contains different, parameterized products as well as a maximum price for each of these products. On the other hand, each seller maintains a product catalog c with producible items and corresponding minimum prices.

Product Modeling. Every relevant product P is representable by a unique identifier. To this end, for discretization, we define a global (domain-specific) modeling function $f : \mathbb{P} \longrightarrow \mathbb{N}, \forall P \in \mathbb{P} : f(P) \in [0, \dots N]$, where $\mathbb{P}$ matches all relevant products, $N \geq |\mathbb{P}|$ is a fixed integer and globally defined together with f for a specific domain. In a query, buyers and sellers must use the same f.

Product-Price Mapping(s). First, we define a set $X = f(P) \times \ddot{\mathbb{M}}$, where $\ddot{\mathbb{M}}$ refers to a price, e.g., in USD. A tuple in X semantically corresponds to $(id, price)$. We define a Buyer B's query q as $P_q^B \subset X$, and for every Seller S_i, we define a set containing all producible items in her product catalog c as $P_c^{S_i} \subset X$. As we require specific price expectations per product, we further note that $\nexists\,(id, m_1), (id, m_2) \in P_q^B \wedge \nexists\,(id, m_1), (id, m_2) \in P_c^{S_i}$ where $m_1 \neq m_2$. Buyers and sellers independently populate their sets P_q^B and $P_c^{S_i}$: For each product P with id, max_{id}^B defines the maximum *price* a Buyer B is willing to pay for it, and $min_{id}^{S_i}$ indicates the minimum *price* expected by Seller S_i. For eventual sales negotiations, the respective *min* and *max* values are kept private.

Buyer. A Buyer B's query q is expressed through the set P_q^B, where to-be-queried products are stored in direct connection with their envisioned maximum prices. We further define a function g_B that indicates whether Buyer B is interested ($1 \equiv true$) in a specific product, i.e., $g_B : \mathbb{N} \longrightarrow \{0, 1\}, \forall id \in f(\mathbb{P}) : g_B(id) = 1 \Leftrightarrow (id, m) \in P_q^B$, else $g_B(id) = 0$. For each query q, we further compute a specific price threshold $\perp_B = \sum_{id \in f(\mathbb{P})} g_B(id) \cdot max_{id}^B$ to fix the maximum costs.

Seller. In addition to the product catalog c and the conceived minimum prices that are expressed through $P_c^{S_i}$, we define a globally-defined price threshold $\top \notin \ddot{\mathbb{M}}$. We rely on $\top$ as a price placeholder for every product id that is not listed in the seller's catalog c, i.e., $(id, \top) \in X \Leftrightarrow \nexists (id, m) \in P_c^{S_i}$.

Purchase Inquiry. We express a purchase inquiry PI between Buyer B and Seller S_i as $PI_i(P_q^B, P_c^{S_i})$. The PI's result is either: (i) A more expressive (fine-granular), yet more revealing PI, Buyer B details a result for each queried product in her query q: $\forall\,(id, price) \in P_q^B : (id, price, \{0, 1\})$, or (ii) she only learns a single result 1 or 0 (indicating a match or no match, respectively), stripping all details.

5.2 Design Overview

Our designs to realize *privacy-preserving* purchase inquiries consist of four phases, as we illustrate in Figure 2. Initially, as part of a global setup, a global modeling function $f(P)$ must be defined, which is later used to deterministically map products to *id*s. Next, each buyer-seller pair (B, S_i) bilaterally executes all subsequent protocol steps, i.e., to consider multiple potential sellers, the buyer reruns the protocol multiple times. In the first protocol step, buyer and seller can individually pre-process certain steps, such as preparing the query q or indexing the catalog c. The exact opportunity for this pre-processing depends on the specific design, i.e., PPI or HPI, in use.

Afterward, we *bilaterally* conduct the privacy-preserving comparison phase to obliviously identify matches (in terms of product(s) and price range(s)) between the buyer's query and the seller's catalog. For our straightforward, PSI-based purchase inquiry design (PPI), we utilize a two-phased approach: an independent matching and a subsequent price comparison for each matched product. In the second phase, we have to outsource the ORE-based price comparison to a third party to preserve privacy as ORE supports symmetric-key cryptography only. Thus, PPI partially violates the ease of use goal (**G5**). In contrast, our HE-based approach (HPI) directly returns a single result for all queried products, following the computation. The buyer repeats this phase for each potential seller. Eventually, she knows which sellers are, in principle, able to satisfy her requested order within the expected price range. These two aspects are the most important decision factors [91].

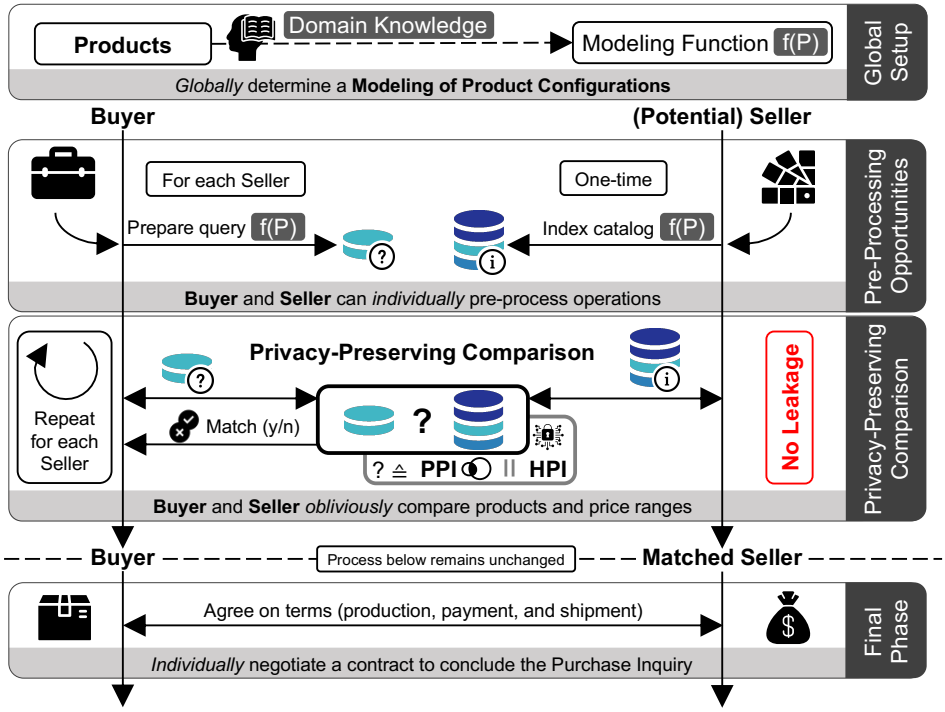

Fig. 2. Our newly proposed designs (PPI and HPI) ensure a privacy-preserving comparison of a buyer's query and a seller's catalog. After a successful match for a query (in terms of product(s) and price range(s)), the traditional procurement process (cf. Appendix A) can continue.

Due to its more performant PSI-based building block, PPI outperforms HPI (cf. Section 7.2). However, its superior flexibility and performance come at the expense of slightly weaker privacy guarantees due to its straightforward design, as we detail in Section 7.5. We further propose a cloud-tailored design variant of HPI, called cHPI, that obliviously offloads expensive HE computations in Section 7.3. For simplicity, we initially limit our explanations to PPI and HPI in the following.

Finally, the buyer can contact any number of matched sellers to continue with the final negotiations, i.e., to agree on a price, a delivery schedule, and other relevant aspects. This final phase, after our privacy-preserving comparison, remains unchanged as to established procurement processes. Thereby, both buyers and sellers keep the same flexibility as they are accustomed to today.

Our designs generally ensure buyer and seller privacy (**G1** & **G2**) by utilizing building blocks from private computing. Moreover, thanks to our modular phases, we can gradually adjust and tune specific parts of each protocol if needed. As a protocol run only concerns the buyer and a specific seller, we also account for protocol resistance (**G3**), unlinkability of buyer queries (**G1**), and the desired ease of use (**G5**). Furthermore, with our non-invasive impact on today's procurement, our privacy-preserving purchase inquiries integrate neatly into established processes: We allow buyers to consider a larger set of potential sellers and relieve sellers from the need to draft offers early on while removing the need to disclose any sensitive information upfront for all parties. The exact contract negotiations (with optional soft criteria) are part of subsequent procurement steps.

Overall, our designs provide two-way privacy during the procurement process. Thereby, buyers can reliably increase the number of considered sellers as they do not have to fear any disadvantages from sharing sensitive information upfront (with unsuitable or untrusted sellers).

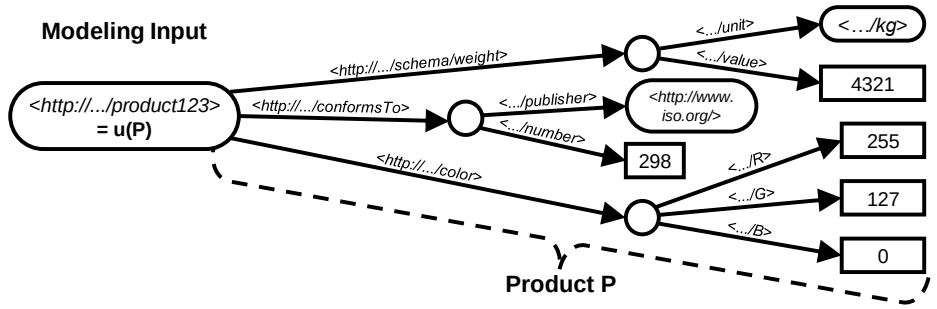

Fig. 3. Our modeling utilizes RDF. We exemplarily illustrate a product description using an RDF graph.

5.3 Specifics of the Protocol Phases

Now, we discuss the phases of our proposed protocols in more detail. We follow the logical flow as presented in Figure 2 and include protocol-specific sequence charts.

5.3.1 Modeling of Product Configurations. In the first phase, we obtain a unique product identifier from a parameterized product description. Subsequent phases only make use of these discrete product identifiers. Rather than an algorithm that computes them from a given description, we present a data governance process that leads to machine-comprehensible product descriptions with unique identifiers – an approach widely used, e.g., in e-commerce in context with the schema.org product description schema (cf. Section 6). This global and deterministic approach also conveniently enables additions, refinements, updates, and product deprecations. The input into this process captures all relevant product information, except the price. The output is a graph-shaped representation in a uniform terminology, where the graph's root node carries the product's identifier.

To ensure global access to the market for all procuring companies, product identifiers and descriptions, as well as purchase inquiries, need to be FAIR [89], i.e., findable, accessible, interoperable, and reusable. While "I" and "R" directly apply in our scenario, "F" and "A" are only desirable in some use cases, e.g., a seller maintaining a public catalog. Technically, FAIR data is often implemented along with the principles for 5-star open data [1, 35] and linked data [9, 39], with an open license being an optional concept: URIs are used as globally unique identifiers of things (products, standards, etc.), and data is linked to other data to provide context, e.g., "what standards does product P conform with", or "how is the 'weight' of a product defined". 5-star data implementations usually adopt the graph-based Resource Description Framework (RDF [90]) data model, which natively uses URIs for instances, e.g., products, as well as on the schema level, e.g., to define a property *weight*, as we exemplarily illustrate for a product in Figure 3.

Schema terms are usually agreed upon by a larger community in the domain (often moderated by a standardization body). These definitions are typically formalized using logical axioms or rules to enable automated data processing. Then, such a schema is called *ontology*. Thus, a product P is modeled as an RDF resource, i.e., a node in the graph that has a description in terms of outgoing edges pointing to further nodes. Given a URI $u(P)$ of P, which is unique and discrete, $f(u(P))$ (henceforth abbreviated as $f(P)$) maps it to a positive integer, as required by our proposed protocols. As we lack a universal, canonical definition of how to generate URIs for resources with RDF, in the future, we additionally have to agree on globally-defined data governance (or implementations).

5.3.2 Pre-Processing Opportunities. After the global definition of the modeling function $f(P)$ to discretize products P, we now present our purchase inquiry protocols. Given that the pre-processing opportunities vary between the two approaches, we now discuss them individually.

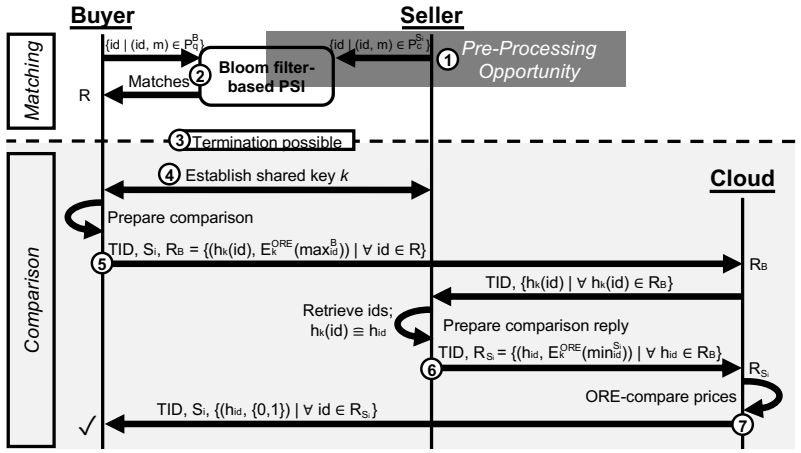

Fig. 4. Sequence chart detailing the consecutive two steps of PPI. The seller can initially pre-compute the Bloom filter for the PSI matching. For respective matches from the PSI, the cloud (third party) obliviously conducts an ORE-based price comparison upon the buyer's request.

PPI: A Two-Phased Approach. We illustrate this design in Figure 4. ① For the initial comparison of PPI (the matching step), Seller S_i can pre-generate the Bloom filter (containing all offered products $P_c^{S_i}$) that is used for the PSI. Using $f(P)$ and RSA blinds [44], the potential seller inserts all computed product *id*s in the Bloom filter. Buyers later receiving the Bloom filter cannot brute-force the inserted products, as all inserted product *id*s are signed using the seller's private key. While this step is the most computationally expensive task, it still is reasonable as the seller only has to perform it once. If desired, Seller S_i can create a new Bloom filter by re-generating the RSA key to protect against buyers colluding to derive her producible items, i.e., the respective *id*s, from $P_c^{S_i}$ that match to her product catalog c. Otherwise, multiple buyers could later merge their queries, as the seller RSA-signs the queried *id*s according to the RSA-PSI protocol [44] using the same private key. This pre-processing, including the RSA blinding, exploits the unmodified RSA-PSI protocol by Kiss et al. [44] for efficiency. We simply outsource this step from the originally proposed protocol sequence.

To prepare a specific query, Buyer B can also pre-process the (quick and inexpensive) derivation of product *id*s in P_q^B using $f(P)$ for the subsequent comparison.

HPI: An HE-Based Protocol. Figure 5 visualizes a sequence chart of this design. In HPI, ① Buyer B must homomorphically encrypt the result of $g_B(id)$ for every product id to pre-process a query, creating C_B for subsequent use in HPI. Additionally, she can pre-compute the query-specific price threshold $\perp_B$ and PHE-encrypt it. To prevent correlation attacks among contacted sellers, she can prepare individual PHE ciphertexts for each Seller S_i. Alternatively, she can re-use them. ② Each potential seller S_i can already blind her prices from $P_c^{S_i}$ to compute M_{S_i} for subsequent use in HPI. However, this step is not computationally expensive as no encryption or signing is required. After this pre-processing phase, using the modeling function $f(P)$, the sets P_q^B and $P_c^{S_i}$ have been individually prepared for the purchase inquiry $PI_i(P_q^B, P_c^{S_i})$ by Buyer B and Seller S_i.

5.3.3 Privacy-Preserving Comparison. Subsequently to the parties computing their query and product catalogs, the buyer can initiate the inquiry with each seller, i.e., she can trigger respective protocol runs for as many sellers as needed, e.g., until a suitable seller has been found. Depending on their (privacy) needs, parties can either choose PPI or HPI, while the protocols cannot be mixed.

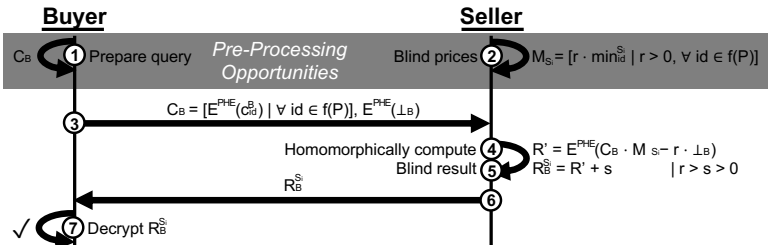

Fig. 5. In HPI, the buyer can pre-encrypt the query using PHE. Subsequently, the seller homomorphically computes a blinded result and returns it to the buyer for decryption, indicating the inquiry's outcome.

PPI: A Two-Phased Approach. In Figure 4, we illustrate the two individual steps of PPI, i.e., matching and comparison. ② The matching step continues the Bloom filter-based PSI protocol by Kiss et al. [44]. First, Buyer B blinds her P_q^B using the public RSA key of Seller S_i and sends the blinded query to the seller. Then, the seller signs these entries using her private RSA key before returning the results to the buyer. Lastly, in the matching step, the buyer can remove the respective blinds and check for containment in the Bloom filter, which concludes the matching. For each product, the buyer learns whether the seller is able to produce it or not. Now, ③ the buyer can gracefully terminate PPI, e.g., if the result indicates no or partial matches of the requested products.

When continuing with the purchase inquiry, Buyer B can subsequently trigger the comparison step of PPI. Initially, ④ Buyer B and Seller S_i establish a shared key k that is used to order-revealingly encrypt their price expectations. Using a cryptographic, keyed hash function h_k, they further obfuscate the matched product *id*s. In line with our ease of use goal (**G5**), we consider involving a third party as disadvantageous. Still, in PPI, using a random transaction ID TID, the buyer must task an independent third party in the cloud to compare the prices under ORE. To this end, the cloud explicitly queries the (encrypted) prices from Seller S_i. First, ⑤ the buyer transmits R_B, a set containing the obfuscated *id*s and her ORE-encrypted price expectations, to the cloud, which requests the seller's price expectations. Second, ⑥ the seller shares R_{S_i}, which likewise contains the obfuscated *id*s and her ORE-encrypted price expectations, with the cloud to allow for an ORE-based price comparison. Finally, ⑦ Buyer B receives a result for each queried/matched product from the cloud, revealing whether the price expectations overlap as well.

HPI: An HE-Based Protocol. We detail the steps of HPI in Figure 5. First, ③ Buyer B transmits a PHE-encrypted vector C_B, containing encrypted zeros and ones from $g_B(id)$ and her encrypted price threshold ($\perp_B$), to Seller S_i. Subsequently, ④ the seller can homomorphically compute the result of the purchase inquiry using her blinded prices (M_{S_i}), i.e., she multiplies C_B with her product prices (as scalar product). Afterward, she blinds $\perp_B$ and subtracts it from the scalar product to obtain R'. ⑤ She blinds the intermediate result (R') before ⑥ returning the blinded result $R_B^{S_i}$ to the buyer. Finally, ⑦ the buyer can decrypt the PHE ciphertext ($R_B^{S_i}$).

If the result is smaller than 0, the seller can produce the queried items and offers them in the desired price range. Otherwise, the contacted seller is not a match. In contrast to the default behavior of PPI, in HPI, the buyer gains no knowledge about individual matches in her query P_q^B.

After this phase, the purchase inquiry PI_i is concluded, and the buyer is aware of the result(s), i.e., whether her query (including the price thresholds) fits the sellers' catalog and price expectations. In both designs, the privacy-preserving comparison is oblivious, i.e., no sensitive information is leaked or exchanged. In contrast to PPI, where the buyer is aware of the matches for each product, in HPI, the buyer only learns whether a seller can produce all requested products within the specified price range. We detailedly discuss the slightly weaker security guarantees of PPI in Section 7.5.

5.3.4 Concluding the Purchase Inquiry. In the final phase, the buyer can contact one or multiple matched sellers to individually continue with the procurement process. In particular, they have to negotiate contracts, i.e., agree on specific terms. The buyer is also able to include additional aspects, such as sustainability or reputation, into the decision-making when contacting matched sellers.

The introduction of our privacy-preserving purchase inquiry does not affect other steps (cf. Appendix A). Thus, we consider this specific phase to be out of scope for our work and refer to related work. The conducted (privacy-preserving) comparison is an indicator that, in theory, an agreement can be reached. Fortunately, buyers and sellers can still freely negotiate prices and terms.

6 REAL-WORLD REALIZATION

In the following, we present our Python-based implementations, which we use for our evaluation. Overall, the seller and cloud components provide RESTful APIs through Flask [79] web servers with a secure version of TLS. Furthermore, to manage all tasks, e.g., incoming queries, we utilize Celery [82]. To account for numerous requests in parallel, we support a separation of frontend (API) and backend (workers). Thus, companies can scale their infrastructure as needed, e.g., by relying on cloud computing to offload computationally expensive tasks or to improve network speeds.

Modeling of the Product Configuration. For our evaluation in this paper, we rely on a simple discretization approach as modeling (cf. Section 7.4.1). For real-world use, we would have to define a way of generating ("minting") URIs to identify the things described [81] (cf. Section 5.3.1). In this regard, market participants are free to agree on either descriptive URIs (https://trusted-marketplace.com/machine-tools/dmu-50-gen3) or non-descriptive URIs (https://vdma.org/id/23597656-0e29-4a04-9f6f-0a8d856c3769), as long as it has been agreed upon how to retrieve information about a thing, given its URI. For example, following pure linked data best practices does not require directory services but requires URIs to be HTTP URLs, from which RDF metadata is downloadable. A trusted party, such as an industrial association, can maintain these globally used URIs on behalf of the products' providers within an independent Internet domain to also ensure the privacy of retrieving companies. The RDF data should use agreed-upon Schemas. Thus, we argue to prefer re-using existing schemas, e.g., ECLASS for product classes and product properties [23], for which an old, unofficial ontology exists [37], and an official one is under development, or many advanced units of measurement ontologies to choose from [42]. Despite being less widespread, unofficial ontologies also exist for describing standards [7]. When ontologies for use with f are missing, we recommended adapting existing ones, e.g., specializing the general, domain-independent GoodRelations ontology for product descriptions in e-commerce (now part of the search engine standard schema.org [36]), or generalizing ontologies for specific machine tool applications, e.g., ExtruOnt [76].

PPI: A Two-Phased Approach. We rely on a Bloom filter-based PSI [44] for the matching step as this variant promises to be very performant (resulting in fewer round trips and the ability to pre-process the product catalog). We utilize a Python library [8], which is based on PyCryptodome [24], and added (de)serialization support. For the comparison, we use a Python library [64], which implements the ORE scheme by Chenette et al. [16].

HPI: An HE-Based Protocol. Our second design builds on PHE. We rely on the Paillier cryptosystem [62]. More specifically, we use CSIRO Data61's Python library [20] for our implementation.

Our designs build on well-known building blocks by linking their operations to preserve the participants' privacy. To create secure prototypes, we rely on popular libraries implementing them.

7 EVALUATION

As we detailed in Section 5, our proposed designs already fulfill the conceptual goals of buyer privacy (**G1**), seller privacy (**G2**), and ease of use (**G5**). To verify that our designs are indeed applicable in real-world settings (**G4**), we further conducted a performance evaluation. First, in

Section 7.1, we present our experimental setup. Afterward, in Section 7.2, we report on the general performance of our designs to study the compliance with **G4**. Subsequently, in Section 7.3, we introduce a third, specifically cloud-tailored design variant of HPI, called cHPI, which allows sellers to offload resource-consuming tasks to a dedicated public cloud (third party)—slightly weakening **G5** (use of a third party) for improved performance. Moving on, in Section 7.4, we evaluate two real-world use cases in the domain of machine tools (cf. Section 2.2). To assess the protocol resistance (**G3**), we extensively discuss the security of our approaches in Section 7.5 while highlighting any influences on **G1** and **G2**, before summarizing our conducted evaluation and its implications in Section 7.6.

7.1 Experimental Setup

We utilized a single server (Intel Xeon E5-2630 and 32 GB RAM) to host all involved parties. We ran Docker containers [52], and they communicated via the loopback interface. We report on the arithmetic mean of 30 runs, calculate 99 % confidence intervals, and present all measurements in logarithmic scales. When evaluating PPI, we configured an RSA key size of 2048 bit for the PSI, and the defined key size of the hash function in ORE was 20 bit. For the Paillier cryptosystem, in HPI and cHPI, we relied on a key size of 3072 bit to encrypt the PHE ciphertexts.

7.2 Performance Measurements

We measured the computation time, maximum RAM usage, and the size of outgoing transmissions per party. As the number of products in the product catalog significantly influences the performance, we consider different, real-world-derived values between 5000 and 1 Mio. to evaluate our implementations. Contrary, the query size only has negligible influence on the runtime during the PSI-based matching as the buyer has to request an encryption for each product (in PPI) or no influence at all as every product *id* has to be encrypted anyway (HPI). Given that $P_q^B \ll P_c^{S_i}$ and real-world queries cover only a few products, we fix the query size at 10 for our evaluation.

7.2.1 PPI: A Two-Phased Approach. As we present in Figure 6, the runtime of PPI increases linearly with the product catalog's size due to the signed values used in the Bloom filter-based PSI (cf. Section 5.3.2). Notably, it is dominated by the seller's pre-processing, i.e., the runtime is dominated by a phase only needed before the first protocol run and can be omitted by subsequent runs with different buyers. The remaining runtime splits between the PSI during the matching step (negligible: 500 ms with a product catalog of 1 Mio. entries and 10 products in the query) and the ORE-based price comparison computed in the cloud, which sequentially involves buyer and seller. In the latter step, the most workload is on the seller (36 s with 1 Mio. total products) as she has to hash all products with the shared key k to identify the cloud-requested prices. In contrast, as part of the price comparison, the cloud is only affected by the number of overall matches. For example, with 10 matches, we measure 75 ms of workload at the cloud.

The memory needs of PPI (Figure 7) are low. While the cloud's use is independent of the catalog's size (max. 35 MB), the memory consumption at the buyer correlates with the size of the Bloom filter (4 MB with 1 Mio. products). The seller's memory usage during the matching is only 0.5 MB and increases linearly with the catalog's size during the price comparison (262 MB with 1 Mio. products). Most memory is needed for the pre-processing (at most 1.9 GB for 1 Mio. products).

Similar observations also hold for the transmission sizes (Figure 8). For a query with 10 products (and in our setting 10 matches), buyer and cloud transmit 7.1 kB and 1.2 kB, respectively. The size of the transmitted data by the seller increases linearly but remains maintainable (only 4 MB with 1 Mio. products in the product catalog).

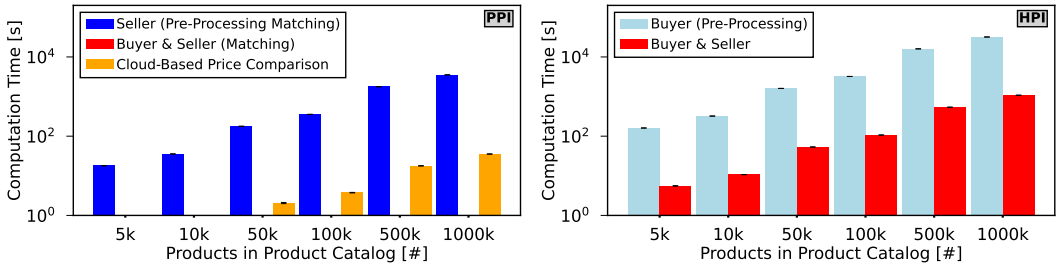

Fig. 6. The runtime scales linearly with the size of the product catalog. PPI (left) is faster than HPI (right) by one order of magnitude. While PPI allows for seller pre-processing, HPI enables buyer pre-processing.

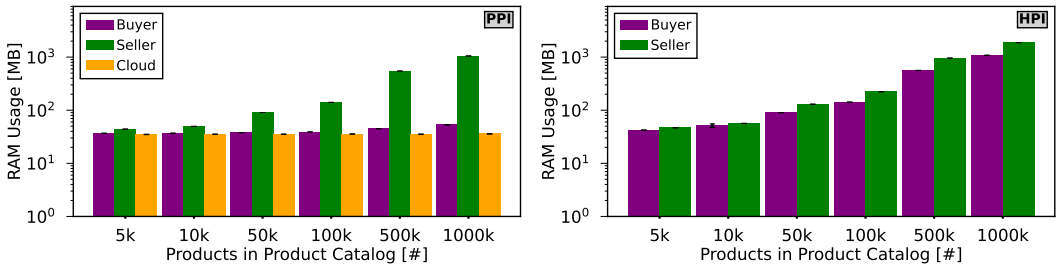

Fig. 7. In PPI (left), the seller fully keeps the Bloom filter in memory. Thus, her usage grows with the product catalog. HPI (right) demonstrates comparable, linearly increasing RAM usage for buyers and sellers.

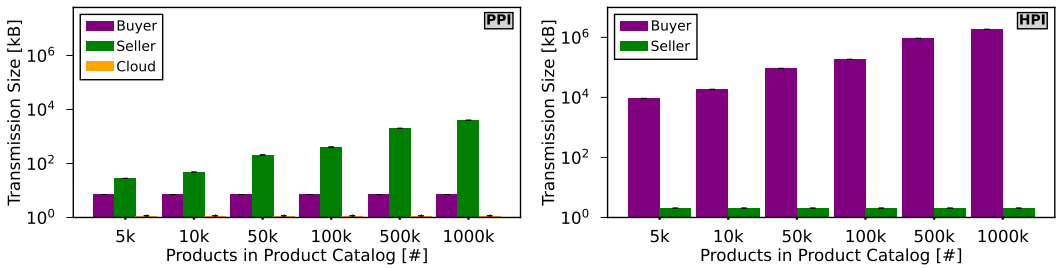

Fig. 8. In PPI (left), the seller has to transfer the Bloom filter for the PSI-based matching. Buyer and cloud correlate with the query size and the number of matches, respectively. In HPI (right), the buyer has to upload the encrypted catalog and only receives a single result from the seller. Thus, we notice a significant asymmetry.

In general, with a total runtime of 59 min (including the one-time pre-processing), moderate RAM usage, and limited network transmissions even for catalogs with 1 Mio. products, PPI is very suitable for real-world use. Sellers using pre-processed catalogs or buyers aborting the protocol whenever the matching step was unsatisfactory further reduce the overall resource needs of PPI.

7.2.2 HPI: An HE-Based Protocol. The runtime of HPI (Figure 6) increases linearly with the number of products. While it is dominated by the buyer's pre-processing (8.8 h for 1 Mio. products), the remaining privacy-preserving comparison, involving both parties, is nearly 30 times faster ($\approx$ 18 min for 1 Mio. products). Hence, HPI primarily burdens the buyer. However, she can re-use every pre-processed query by sending it to all considered sellers and thereby significantly reduce her load.

In HPI, the maximal memory usage (Figure 7) correlates linearly with the number of products as the buyer and seller load all ciphertexts into memory for the pre-processing and computation, respectively. The measured maximum lies at maintainable 1.1 GB for 1 Mio. products on the buyer's and 1.9 GB on the seller's side. Still, to reduce the memory usage resulting in a constant maximum, we could easily adjust our implementation to process the products (computation) in smaller batches.

Concerning the network transmissions (Figure 8), the buyer has to upload all PHE ciphertexts. Thus, the amount scales linearly with the number of products (1.85 GB for 1 Mio. products). In contrast, the seller only returns a single result, i.e., we constantly measure only 2.1 kB.

While the necessary resources are steep in HPI, its needs are still maintainable for real-world deployments. Today's non-automated procurements usually take several days up to weeks. Thus, introducing a processing of 10 h for catalogs with 1 Mio. products adds no significant overhead. A RAM consumption of at most 1.9 GB and network transmissions of 1.85 GB do not overload today's infrastructures and the resources of companies either.

7.2.3 Comparing PPI and HPI. The resource usage and distribution over all participants is crucial for real-world use. Hence, we now look into our protocols in more detail.

Computation Time. Given that the total runtime scales with the product catalog's size for both designs, all parties can estimate their computation time properly before starting the execution. While both protocol variants are well feasible for all involved entities with a reasonable product catalog size, we notice that PPI is one order of magnitude faster than HPI. This difference in runtime bases on the PHE-induced overhead in HPI, which enables very strong privacy in comparison to the comparably inexpensive PSI-based, Bloom filter-enabled matching in PPI. Furthermore, both protocols feature pre-computable parts, which can be re-used over several runs. The quicker PPI protocol allows the seller to pre-process their offers, which constitutes a one-time setup that is independent of the number of buyer-triggered purchase inquiries. In contrast, in HPI, the buyer has the most workload when preparing her (reusable) query. As identifying the "best-fitting" seller by contacting several sellers is a common application, this re-use is highly beneficial in practice.

RAM Usage. Our designs do not require excessive amounts of memory at any time of the operation and are runnable on commodity hardware. Still, the memory usage and its distribution over the participants differ. In PPI, the memory usage is driven by the Bloom filter as part of the PSI-based matching step. Consequently, it barely requires memory at the buyer (4 MB for 1 Mio. products) or the cloud (only around 200 B per matched product), but mostly burdens the seller (with a feasible RAM usage of 1 GB during the pre-processing of a product catalog 1 Mio. products). In comparison, HPI distributes the memory usage over both parties, seller and buyer (at most 1.9 GB for 1 Mio. products; with batching support).

Outgoing Transmissions. Again, the data transmission of both protocols is well feasible with today's infrastructures. While, in PPI, the seller has to transmit most of the data (only at most 4 MB, even for 1 Mio. products due to the efficient underlying Bloom filter), HPI features a larger transmission from the buyer to the seller. For example, with $|X|$=5000 products, we measure around 9 MB of data transmissions. This number increases to 1.85 GB for $|X|$=1 Mio. Thus, network limitations affect HPI's runtime to a larger extent (in comparison to PPI).

Overall, we notice that HPI requires slightly more resources due to the PHE-induced overhead in comparison to PPI. Later, in Section 7.5, we discuss the privacy benefits of HPI, which warrant the overhead for settings with strict guarantees. Regardless, we conclude that the performance is suitable for real-world deployments as our protocols can run on commodity hardware in a reasonable time, even when working with large product catalogs. To relieve the seller of computational load in HPI, we propose a cloud-tailored variant, called cHPI, in the next section.

7.3 cHPI: Offloading Parts of HPI to the Cloud

To relieve the seller in HPI from some computational load, we can offload parts of the (costly) computation to an untrusted cloud, which is inspired by work that outsources vector multiplications [17]. While such a design seems to contradict **G5**, due to the properties of PHE, the cloud

learns nothing about the data it operates on nor the final result. Thus, we do not require any trust in this third party. We refer to this protocol variant as cHPI.

In principle, cHPI only differs from HPI as follows. The buyer sends her encrypted vector directly to the cloud, and the seller shares her blinded prices with the cloud. For security, these inputs are blinded with a random vector that buyer and seller agreed upon. The cloud then computes the (costly) scalar product before returning the result to the seller (to allow for blinding as in HPI).

We refer to Appendix B.1 for more details on the protocol and a detailed sequence chart. In Appendix B.2, we further present a performance evaluation of cHPI that shows how this design relieves the seller from computational load without prolonging the overall runtime.

7.4 A Real-World Setting for Purchase Inquiries

To further verify the real-world applicability of our designs, we source authentic products from the domain of machine tools and conduct an evaluation that is representative for real-world use cases. In the following, we first introduce our considered datasets before discussing the evaluation results.

7.4.1 Queries in the Domain of Machine Tools. For our real-world evaluation, we consider two distinct queries on machine tools with an application in injection molding: Our queries exemplarily describe properties (of products) and offers from sellers of (i) clamping units where the material during the process is injected, i.e., the units determine the shape of produced workpieces (referenced to as *tool use case*), and (ii) machines to produce such clamping units (*machine tool use case*).

First, we require a unique product modeling that allows both potential sellers and interested buyers to have an unambiguous understanding of the product. Typically, a domain-specific set of parameters is used to define the modeling function. While a detailed discussion of all parameters is out of scope for this paper, we exemplarily present a subset used in the *tool use case* (and provide further details on these parameters in Appendix C): (i) The *size factor* describes the maximum mounting length, which is crucial as machines are frequently limited in their workspace. (ii) The *shape complexity factor*, the *aspect ratio*, and the *filigree factor* identify the shape of the workpiece and, therefore, also of the mold. These aspects often introduce constraints on the tooling of supplier's machinery and thus reduce the number of suitable suppliers. (iii) In addition, *tolerance* and *material factor* label the requirements concerning the surface quality of the product in question.

To combine the selected parameters into a modeling function that also unites value ranges of specific parameters (keeping the product catalog small), we define a binning scheme for each relevant parameter. We appropriately configure the respective parameter ranges and the bins' granularity for the respective use case. For example, when querying for a tool, for the important size factor, we work with 5 bins, each covering a distinct set of tool configurations. In contrast, other parameters are only binary, e.g., indicating whether they feature dielectric operation (cf. Appendix C). Finally, we source this binning scheme to discretize the products into product *id*s.

With this approach, we end up with 38 880 possible configurations (based on 10 parameters (each with 2 to 5 bins)) in our *tool use case* and 944 784 unique product configurations in our *machine tool use case*. In the latter case, we express the different products of the catalog through 14 parameters.

7.4.2 Real-World Performance Measurements. With these use cases, we are able to underline the real-world applicability of our approaches. The products in the catalogs (38 880 and 944 784 modeled products) match the values that we considered in our previous performance evaluation (cf. Section 7.2). Our measurements are in line with our previous results because our designs are oblivious to the exact numbers they are comparing. Hence, they are universally applicable.

Tool Use Case. For the first use case, we report a total runtime of 2.3 min ± 0.7 s for PPI and 21.6 min ± 0.2 min for HPI, respectively. Again, these numbers constitute a difference by one order of magnitude. In real-world settings with constrained network links, the speedup of PPI will be

even higher due to the smaller total transmission size. Regardless, this use case underlines that the designs allow buyers to quickly query sellers for available tools.

Machine Tool Use Case. Our second use case features a significantly larger product catalog. The overall runtime exhibits this aspect as well. PPI takes 57 min ± 15.5 s to conclude its run, while HPI finishes after 525 min ± 3 min. Given that the purchase of a new, complex machine tool (choosing from more than 940k product variants) is not an everyday query, which is usually planned well in advance, the real-world performance of our designs is feasible for industrial settings. Thus, we argue that we could easily support even larger scenarios without limiting the value of our newly proposed privacy-preserving purchase inquiries.

Conducting purchase inquiries with other companies is a critical task for many manufacturing companies as they cannot produce every required part or production resource themselves. Thus, they are forced to trade with other companies. The presented performance of real-world use cases underlines that identifying a suitable supplier from the multitude of possible suppliers on a global market is possible, even privacy-preservingly. In subsequent steps of the procurement process, companies can then also take other important factors, such as delivery times, into account. Overall, across all kinds of use cases and applications, such an approach protects sensitive information from unsuitable suppliers, i.e., effectively avoiding competitive disadvantages for inquiring companies.

Our real-world measurements underline that privacy-preserving purchase inquiries are indeed suitable for realistic and large-scale deployment. Thus, buyers can easily consider a larger set of suppliers. Thereby, we enable them to consider additional factors, e.g., sustainability, more broadly.

7.5 Security Discussion

In the following, we discuss the security of our designs that promise two-way privacy for purchase inquiries. Primarily, we consider buyer and seller privacy (**G1** & **G2**). Moreover, we look at the protocol resistance (**G3**) of our designs to ensure protocol acceptance in real-world deployments.

Attacker Model. In our setting (cf. Section 2), all participating entities, i.e., buyers and sellers, are well-known (registered) companies who depend on their reputation to secure and attract business, respectively. Thus, due to the authenticated communication, all participating companies are identifiable. We detail that they do not have any incentive to input incorrect information into our protocols. Furthermore, we envision that a trusted industry association, such as the VDMA [85], operates—funded by membership fees—the third party as a public service when using PPI (or cHPI). Finally, all companies and the third-party operator are bound by legislation and to specific jurisdictions. Thus, we focus on malicious-but-cautious attackers [80] and do not permit cartel collusion between companies [38, 60]. Our main threat is a company who tries to extract as much information as possible, jointly with the cloud, while behaving according to the specification.

Our work bases on the security of the established secure communication channels, the used (technical) building blocks, and properly chosen key lengths (adequately secure mode of operation).

Bilateral Protocols. All designs concern a single buyer-seller pair only: For security, we utilize secure two-party building blocks from private computing (PSI or HE) in our protocols. Given that all protocol runs (i) are independent of each other and (ii) do not source third-party data, only the involved parties can potentially attack the protocols with the goal of extracting information. Uninvolved parties, especially other (uninvolved) buyers and sellers, cannot gain any information.

Information Leakage. Theoretic information leaks for the involved, protocol-compliant behaving parties are limited by design: First, a buyer could repeatedly send queries to a single seller to brute-force her prices or re-construct her offered product catalog. Given that no centralized third party is handling the purchase inquiries, sellers can freely decide whether they want to apply some kind of rate limiting for any buyer to enforce their privacy needs (**G2**). However, we assume that,

in practice, no such action is needed as the workload for buyers renders frequent requests unlikely (cf. Section 7.2). Second, due to the weaker privacy guarantees in PPI (two-phased design), a buyer could fear that her requested products are leaked (partially) during the comparison phase. However, we only work with granular capabilities, and thus, no detailed product information is revealed at any time. Next, we individually discuss the respective implications of each design.

PPI: A Two-Phased Approach. We generally ensure buyer and seller privacy (**G1** & **G2**) in both phases of PPI. First, the matching step relies on PSI and only returns the matches to the buyer, i.e., the seller cannot learn anything from this phase (accounting for **G1**). In theory, the buyer could request every possible product to derive the seller's catalog. However, as the seller is involved in the preparation of the buyer's query (cf. Section 5.3.3), the seller can simply rate-limit the buyer if she seems to be requesting too many products. For the same reason, buyers cannot brute-force products listed in the Bloom filters that index the seller's catalog. Thus, apart from sharing intermediate results on matches with the buyer, we also address seller privacy (**G2**) in the matching step.

In the comparison step of PPI, we rely on a third party (cloud). The used ORE ciphertexts are never shared with any other party except for the cloud, which only operates on these ciphertexts. Thus, as intended (also a design goal of ORE), it can only learn the ordering of any two ciphertexts without any knowledge of the compared products. As buyers and sellers agree on a shared key k, the cloud cannot analyze comparisons over time, limiting the cloud's insights to a single query, i.e., the cloud cannot perform frequency analysis to, e.g., uncover frequently requested products. However, PPI can "leak" the matches (prior to the price comparison) to the seller as the cloud only queries the prices for requested matches by default. At the expense of moderate performance overhead, e.g., a runtime of 100 s to encrypt 1 Mio. prices, we could require the seller to share ORE ciphertexts for all prices by default, effectively preventing said intermediate information leak. Thus, in a semi-honest setting, we can tune the fulfillment of **G1** & **G2** during the comparison.

HPI: An HE-Based Protocol. This protocol is secure and ensures privacy by design [26]: The buyer encrypts its data homomorphically (**G1**), and the contacted seller never provides any information to others. Thus, no data, except for the final result, is shared as all computations operate on PHE ciphertexts. The seller can, at no point, decrypt the buyer's data nor the result. To ensure confidentiality (**G2**), the seller blinds the result before returning it to the buyer. Eventually, the buyer learns a single result only. We discuss the security properties of cHPI in Appendix B.3.

Entity Misbehavior. Buyer and seller could still behave according to the protocol while working with manipulated queries or product catalogs, respectively. First, in theory, the buyer could send bogus queries. In this case, the protocols return matches on products that the buyer is not interested in, i.e., the information has no added value to him. In addition to his own resources, such misbehavior would, however, also consume some of the seller's computational resources. Second, the seller could compile a fake catalog to increase the probability of matches and, thereby, the likelihood of subsequent negotiations. By inserting additional products, the seller might know what products the buyer is interested in (if being contacted). However, as he cannot produce them, no sale will be made (cf. **G3**), and his reputation will suffer. Higher prices would even lower the probability, and getting the buyer to negotiate by specifying lower prices either leads to lower prices during the sale, i.e., no misbehavior, or no sale at all (cf. **G3**). Thus, entity misbehavior leads to no benefits.

Collusion Attacks. Possible collusion between multiple parties could potentially increase the illegitimate information gain for involved parties. Thus, we now discuss the threat of multiple colluding parties. Due to the lack of a third party, such attacks are not possible in HPI.

Buyer & Seller. Such collusions contradict our setting (cf. Section 2) as these parties can exchange all sensitive information anyway. Besides, they cannot gain any additional (third-party) data as our protocols are bilateral by design. Thus, such collusions are irrelevant to assess the security.

Buyer & Third Party. In PPI, these parties can reveal the prices $min_{id}^{S_i}$ of (matched) products during the comparison step due to the buyer's knowledge of the shared key k, i.e., he knows Seller S_i's lowest selling price. Thus, the seller's privacy (**G2**) depends on a carefully selected third party.

Seller & Third Party. In PPI, such a collusion can reveal the prices max_{id}^{B} of matched products during the comparison step due to the seller's knowledge of k, i.e., he is aware of Buyer B's highest purchase price. Thus, a carefully selected trusted third party is recommended to also ensure **G1**.

To conclude, due to the shared key k in PPI, we cannot tolerate any collusion with the third party as it would allow for leaks of the minimum, respectively maximum prices, slightly violating **G1** & **G2**. Thus, strict privacy needs mandate the use of HPI (or cHPI). Regardless, we consider PPI to be resistant in terms of **G3** as such collusions do not influence the outcome of PPI's computation.

Multiple Buyers. Such a collusion contradicts our defined attacker model as syndicated procurements [59] are only permitted through a wholesaler, which in turn equals a single buyer regarding our protocol use, i.e., our protocols are not affected by any (buyer) collusion in this case.

Multiple Sellers. Likewise, such a collusion contradicts our defined attacker model as cartels [60] are forbidden by law. Consequently, they cannot use our proposed protocols in practice to, e.g., globally drive up product prices. Hence, such (seller) collusion is out of scope for our discussion.

Our security discussion underlines the privacy benefits of HPI. However, even with the intuitive and more performant PPI, most information is kept private by design as part of our purchase inquiries.

7.6 The Potential of Private Purchase Inquiries

Our evaluation underlines that our designs are suitable for our considered scenario (cf. Section 2). With our HPI protocol, we are further able to address all design goals (cf. Section 3) to enable privacy-preserving purchase inquiries as part of the procurement process in the industry.

While our protocols generally ensure buyer and seller privacy (**G1** & **G2**; PPI with minor deductions) by design (cf. Section 5.2), we demonstrate their real-world applicability (**G4**) using our machine tool use cases (cf. Section 7.4). These runs conclude within an appropriate time and reasonably consume resources to be suitable for real-world use. During our security discussion (cf. Section 7.5), we further looked at our protocols' robustness (**G3**): Overall, we present secure protocols as their security mainly builds on established building blocks with attested security.

When considering both performance and security, we notice the trade-off between no information leakage in HPI at the expense of some computational overhead when compared to PPI. More specifically, in settings where an ORE-based price comparison, conducted by a cloud service, is acceptable, companies can benefit from the superior performance of PPI. Its performance is superior to HPI because the protocol only involves the seller's catalog $P_c^{S_i}$ and not every product *id*. Thus, in practice, fewer product *id*s are part of the pre-processing and the subsequent comparison. Apart from fewer computational resources, PPI also supports indicating matches for each queried product, i.e., queries are answered with more granularity. However, adjusting the cloud's protocol to only return a single result, as in HPI (cf. Section 5.1), is a simple, non-invasive adjustment.

Our approaches further scale well with the number of potential sellers that should be considered as the protocols runs are independent of each other, and buyers can easily trigger as many purchase inquiries as computationally supported. Thus, due to this flexibility, our proposed designs also fulfill the targeted goal of ease of use (**G5**). Hence, we consider them suitable for real-world deployment.

Our evaluation shows that our designs for bilateral purchase inquiries are well-suitable for real-world settings regarding their performance and security. Thus, according to their needs, companies can select a design with technical guarantees that provides two-way privacy during procurement.

8 RELATED WORK

To the best of our knowledge, no approach to significantly improve the privacy during traditional procurement processes is widely in use today. Instead, companies protect themselves by limiting their considered set of potential sellers in practice [5] or requiring third parties to sign NDAs at an early stage of the negotiations [88], which is a time-consuming and tedious task (cf. Section 2.2).

Similar scenarios such as auctions [13, 54, 56], electronic markets [41], or private e-tendering [63] exist but fail to provide companies with the required flexibility in selecting their suppliers, e.g., based on other relevant evaluation criteria (cf. Section 1). Basically, they all target settings in which the purchase is concluded (agreeing on a fixed price). Thereby, they exceed our scenario, where only a pre-selection (matching) is needed to allow companies to still negotiate final prices. Furthermore, these works usually assume that one entity reveals what they are buying or selling, i.e., commonly, sellers have to reveal their product catalogs or products. Thus, they all contradict our goal of not sharing any information upfront as they only consider the privacy of one party.

In the area of advertising, related work [32, 34] only focuses on the privacy of ad receivers (cf. **G1**) as the second party, i.e., the ad provider, does not demand specific privacy guarantees. Thus, these approaches violate the need for seller privacy (**G2**). Regardless, they intuitively highlight similar privacy issues in other settings. To conclude, all of these advances fail to protect the privacy of all involved entities as they commonly focus on a single party only. Hence, they are not suitable.

Moving toward business eco-systems, initiatives such as the federated secure data infrastructure Gaia-X [14] and International Data Spaces (IDS) [61] aim at standardizing (industrial) data sharing. However, they are broader in scope, impose significant technical and organizational requirements for participating in an eco-system [47], and so far have not specifically addressed purchase inquiries. On a related note, privacy-preserving building blocks are frequently used to realize matching in various domains, such as production [66, 71], data management [75], or genome processing [95]. Other work protects the privacy of all parties in industrial comparisons [72]. However, their data sharing settings do not translate directly to our considered scenario, where multiple dimensions must be compared (product(s) and price range(s)). Consequently, prior to our proposed designs, suitable approaches for privacy-preserving purchase inquiries were still missing.

Our designs augment existing work in the area of real-world deployable matching. Thereby, they address the outlined gap (cf. Section 2) and efficiently enable two-way privacy during procurement.

9 CONCLUSION & FUTURE WORK

In this paper, we have raised and carefully compiled privacy concerns about procurement processes. These concerns especially impact the future of manufacturing, which introduces dynamic and flexible business relationships to effectively deal with change requests and small-batch production. Thus, we identify the need for privacy-preserving purchase inquiries to address future challenges within manufacturing and production, i.e., a way to discover and match fitting sellers without sharing any information upfront. Due to the lack of applicable related work, we developed two designs, PPI and HPI (as well as cHPI), to innovatively realize secure and scalable purchase inquiries. By relying on established building blocks from private computing, we address this research gap. Thereby, we augment existing procurement practices in a non-invasive way by allowing companies to establish new business relationships privacy-preservingly. Thus, companies can bypass our privacy-preserving procurement step whenever privacy is not necessary, or none of the contacted sellers match the request, e.g., to get suppliers to innovate their offered products. Since our design are oblivious of the handled products and domains, they are universally applicable. In our evaluation, we study two representative use cases from the domain of machine tools. This evaluation and our security discussion underline the real-world impact of our designs for deployments in industry.

Thereby, we non-invasively augment today's procurement processes, and pave the way for privacy-preserving procurement that turns the vision of dynamic business relationships into reality.

We expect that the research efforts in this direction will significantly increase to properly address this overlooked area. In our view, future work should mainly deal with challenges associated with real-world deployments and potential reservations against the acceptance of our novel privacy-preserving purchase inquiries as an immediate next step. Thus, we call for economic studies measuring the (monetary) impact of our privacy-preserving purchase inquiries in the industry. Moreover, we look forward to real-world acceptance studies, also in light of our designs' ease of use. Once the evolution of this crucial aspect of production has started, we expect further refined solutions to improve the maturity of privacy-preserving purchase inquiries overall. With a closer focus on our specific designs, we are particularly interested in future developments of privacy-preserving building blocks that are used over the Internet to facilitate the secure exchange of sensitive information. In particular, evolved building blocks in this area could allow us to enable (external) verifiability in our proposed approaches. Currently, to the best of our knowledge, no real-world deployable solutions exist that protect all confidential inputs while also allowing (external) parties to verify the conducted computation. Finally, we intend to look into potentials to improve PPI by replacing our ORE-based price comparison with another secure approach, e.g., secure multi-party computation or homomorphic encryption, to remove the third party in PPI.

Our presented designs improve the status quo in today's manual, privacy-invasive procurement by offering two-way privacy for the involved companies, a feature that has been missing so far.

ACKNOWLEDGMENTS

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC-2023 Internet of Production – 390621612. Within this work, we followed an abstract research methodology [65] to structure and organize our collaboration.

REFERENCES

[1] 5stardata.info. 2012. 5-star Open Data. https://5stardata.info/.
[2] Abbas Acar, Hidayet Aksu et al. 2018. A Survey on Homomorphic Encryption Schemes: Theory and Implementation. *Comput. Surveys* 51, 4, 1–35.
[3] Evgeniy A. Ageshin. 2001. E-procurement at work: A case study. *Production and Inventory Management Journal* 42, 1, 48–53.
[4] Rakesh Agrawal, Jerry Kiernan et al. 2004. Order Preserving Encryption for Numeric Data. In *2004 ACM SIGMOD International Conference on Management of Data (SIGMOD '04)*. ACM, 563–574.
[5] Rebecca Angeles and Ravi Nath. 2007. Business-to-business e-procurement: success factors and challenges to implementation. *Supply Chain Management* 12, 2, 104–115.
[6] Lennart Bader, Jan Pennekamp et al. 2021. Blockchain-Based Privacy Preservation for Supply Chains Supporting Lightweight Multi-Hop Information Accountability. *Information Processing & Management* 58, 3.
[7] Sebastian R. Bader, Irlan Grangel-Gonzalez et al. 2020. A Knowledge Graph for Industry 4.0. In *17th International Conference on The Semantic Web (ESWC '20)*. Springer, 465–480.
[8] Ayoub Benaissa. 2020. PyPSI. https://github.com/OpenMined/PyPSI.
[9] Tim Berners-Lee. 2010 (2006). Linked Data - Design Issues. https://www.w3.org/DesignIssues/LinkedData.html.
[10] Alex Bilsing. 2007. *Kennzahlengestützte Bewertung der technologischen Leistungsfähigkeit der Fertigung im Werkzeug- und Formenbau.* Ph. D. Dissertation. RWTH Aachen University.
[11] Dan Boneh, Kevin Lewi et al. 2015. Semantically Secure Order-Revealing Encryption: Multi-input Functional Encryption Without Obfuscation. In *34th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT '15)*. Springer, 563–594.
[12] Wolfgang Boos, Christoph Maximilian Bernd Kelzenberg et al. 2018. *Erfolgreich Lieferanten Managen im Werkzeugbau.* Technical Report. WBA Aachener Werkzeugbau Akademie GmbH.
[13] Felix Brandt. 2006. How to obtain full privacy in auctions. *International Journal of Information Security* 5, 4, 201–216.
[14] Arnaud Braud, Gaël Fromentoux et al. 2021. The Road to European Digital Sovereignty with Gaia-X and IDSA. *IEEE Network* 35, 2, 4–5.

[15] Philipp Brauner, Manuela Dalibor et al. 2022. A Computer Science Perspective on Digital Transformation in Production. *ACM Transactions on Internet of Things* 3, 2.
[16] Nathan Chenette, Kevin Lewi et al. 2016. Practical Order-Revealing Encryption with Limited Leakage. In *Revised Selected Papers of the 23rd International Conference on Fast Software Encryption (FSE '16)*. Springer, 474–493.
[17] Qiong Cheng and Chong-Zhi Gao. 2017. A cloud aided privacy-preserving profile matching scheme in mobile social networks. In *2017 IEEE International Conference on Embedded and Ubiquitous Computing (EUC '17)*. IEEE, 195–198.
[18] Richard Chow, Philippe Golle et al. 2009. Controlling Data in the Cloud: Outsourcing Computation without Outsourcing Control. In *2009 ACM Workshop on Cloud Computing Security (CCSW '09)*. ACM, 85–90.
[19] Leon Yang Chu, Ying Rong et al. 2020. The Strategic Benefit of Request for Proposal/Quotation. *Operations Research* 70, 3, 1410–1427.
[20] CSIRO's Data61. 2014. python-paillier. https://github.com/data61/python-paillier.
[21] Paolo D'Arco, María Isabel González Vasco et al. 2012. Size-Hiding in Private Set Intersection: Existential Results and Constructions. In *5th International Conference on Cryptology in Africa (AFRICACRYPT '12)*. Springer, 378–394.
[22] Emiliano De Cristofaro and Gene Tsudik. 2010. Practical Private Set Intersection Protocols With Linear Complexity. In *14th International Conference on Financial Cryptography and Data Security (FC '10)*, Vol. 6052. Springer, 143–159.
[23] ECLASS e.V. 2007. ECLASS – Standard for Master Data and Semantics for Digitalization. https://www.eclass.eu/.
[24] Helder Eijs. 2014. PyCryptodome. https://www.pycryptodome.org/.
[25] Fredrik Elgh. 2012. Decision support in the quotation process of engineered-to-order products. *Advanced Engineering Informatics* 26, 1, 66–79.
[26] Caroline Fontaine and Fabien Galand. 2007. A Survey of Homomorphic Encryption for Nonspecialists. *EURASIP Journal on Information Security* 2007.
[27] Michael Funke and Ralf Ruhwedel. 2001. Product Variety and Economic Growth: Empirical Evidence for the OECD Countries. *IMF Staff Papers* 48, 2, 225–242.
[28] Craig Gentry. 2009. Fully Homomorphic Encryption Using Ideal Lattices. In *41st Annual ACM Symposium on Theory of Computing (STOC '09)*. ACM, 169–178.
[29] Lars Gleim, Jan Pennekamp et al. 2020. FactDAG: Formalizing Data Interoperability in an Internet of Production. *IEEE Internet of Things Journal* 7, 4, 3243–3253.
[30] Shafi Goldwasser and Silvio Micali. 1984. Probabilistic encryption. *J. Comput. System Sci.* 28, 2, 270–299.
[31] Peter Gonczol, Panagiota Katsikouli et al. 2020. Blockchain Implementations and Use Cases for Supply Chains-A Survey. *IEEE Access* 8, 11856–11871.
[32] Saikat Guha, Bin Cheng et al. 2011. Privad: Practical privacy in online advertising. In *8th USENIX Symposium on Networked Systems Design and Implementation (NSDI '11)*. USENIX Association, 169–182.
[33] Sung Ho Ha and Sang Chan Park. 2001. Matching Buyers and Suppliers: An Intelligent Dynamic-Exchange Model. *IEEE Intelligent Systems* 16, 4, 28–40.
[34] Hamed Haddadi, Pan Hui et al. 2011. Targeted Advertising on the Handset: Privacy and Security Challenges. In *Pervasive Advertising*. Springer, 119–137.
[35] Ali Hasnain and Dietrich Rebholz-Schuhmann. 2018. Assessing FAIR Data Principles Against the 5-Star Open Data Principles. In *ESWC 2018 Satellite Events on the Semantic Web (ESWC '18)*. Springer, 469–477.
[36] Martin Hepp. 2015. The Web of Data for E-Commerce: Schema.org and GoodRelations for Researchers and Practitioners. In *15th International Conference on Web Engineering (ICWE '15)*. Springer, 723–727.
[37] Martin Hepp and Andreas Radinger. 2010. eClassOWL – The Web Ontology for Products and Services. http://www.heppnetz.de/projects/eclassowl/.
[38] Kai Hüschelrath and Heike Schweitzer. 2014. *Public and Private Enforcement of Competition Law in Europe*. Springer.
[39] Bernadette Hyland, Ghislain Atemezing et al. 2013. Linked Data Glossary. W3C Working Group Note.
[40] Roman Inderst. 2008. Single sourcing versus multiple sourcing. *The RAND Journal of Economics* 39, 1, 199–213.
[41] Joakim Kalvenes and Amit Basu. 2006. Design of Robust Business-to-Business Electronic Marketplaces with Guaranteed Privacy. *Management Science* 52, 11, 1721–1736.
[42] Jan Martin Keil and Sirko Schindler. 2019. Comparison and evaluation of ontologies for units of measurement. *Semantic Web* 10, 1, 33–51.
[43] Florian Kerschbaum and Anselme Tueno. 2019. An Efficiently Searchable Encrypted Data Structure for Range Queries. In *24th European Symposium on Research in Computer Security (ESORICS '19)*. Springer, 344–364.
[44] Ágnes Kiss, Jian Liu et al. 2017. Private Set Intersection for Unequal Set Sizes with Mobile Applications. *Proceedings on Privacy Enhancing Technologies Symposium (PETS '17)* 2017, 4, 177–197.
[45] Vladimir Kolesnikov, Ranjit Kumaresan et al. 2016. Efficient Batched Oblivious PRF with Applications to Private Set Intersection. In *2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16)*. ACM, 818–829.
[46] Maria Linnartz, Ursula Motz et al. 2021. Increasing Resilience in Procurement in the Context of the Internet of Production. *Journal of Production Systems and Logistics* 1, 2021.

[47] Johannes Lohmöller, Jan Pennekamp et al. 2022. On the Need for Strong Sovereignty in Data Ecosystems. In *Proceedings of the 1st International Workshop on Data Ecosystems (DEco '22)*. CEUR Workshop Proceedings.

[48] Sidra Malik, Volkan Dedeoglu et al. 2022. PrivChain: Provenance and Privacy Preservation in Blockchain enabled Supply Chains. In *Proceedings of the 2022 IEEE International Conference on Blockchain (Blockchain '22)*. IEEE, 157–166.

[49] Sidra Malik, Volkan Dedeoglu et al. 2019. TrustChain: Trust Management in Blockchain and IoT supported Supply Chains. In *2019 IEEE International Conference on Blockchain (Blockchain '19)*. IEEE, 184–193.

[50] Sidra Malik, Naman Gupta et al. 2021. TradeChain: Decoupling Traceability and Identity in Blockchain enabled Supply Chains. In *Proceedings of the 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom '21)*. IEEE.

[51] Sidra Malik, Salil S. Kanhere et al. 2018. ProductChain: Scalable Blockchain Framework to Support Provenance in Supply Chains. In *2018 IEEE 17th International Symposium on Network Computing and Applications (NCA '18)*. IEEE.

[52] Dirk Merkel. 2014. Docker: Lightweight Linux Containers for Consistent Development and Deployment. *Linux Journal* 2014, 239.

[53] Michael Naehrig, Kristin Lauter et al. 2011. Can Homomorphic Encryption Be Practical?. In *3rd ACM Workshop on Cloud Computing Security Workshop (CCSW '11)*. ACM, 113–124.

[54] Moni Naor, Benny Pinkas et al. 1999. Privacy Preserving Auctions and Mechanism Design. In *1st ACM Conference on Electronic Commerce (EC '99)*. ACM, 129–139.

[55] Rajesh Narang and Tanmay Narang. 2017. Preserving Confidentiality and Privacy of Sensitive Data in e-Procurement System. *International Journal of Cyber-Security and Digital Forensics* 6, 4, 186–197.

[56] Khanh Quoc Nguyen and Jacques Traoré. 2000. An Online Public Auction Protocol Protecting Bidder Privacy. In *5th Australasian Conference on Information Security and Privacy (ACISP '00)*. Springer, 427–442.

[57] Robert A. Novack and Stephen W. Simco. 1991. The Industrial Procurement Process: A Supply Chain Perspective. *Journal of Business Logistics* 12, 1, 145–168.

[58] Gilbert N. Nyaga, Judith M. Whipple et al. 2010. Examining supply chain relationships: do buyer and supplier perspectives on collaborative relationships differ? *Journal of Operations Management* 28, 2, 101–114.

[59] Organisation for Economic Co-operation and Development. 2013 (accessed April 20, 2023). Fighting bid rigging in public procurement. https://www.oecd.org/competition/cartels/fightingbidriggininpublicprocurement.htm.

[60] Organisation for Economic Co-operation and Development. 2013 (accessed March 6, 2022). Cartels and anti-competitive agreements. https://www.oecd.org/competition/cartels/.

[61] Boris Otto, Sören Auer et al. 2016. *Industrial Data Space: Digital Souvereignity over Data.* White Paper. Fraunhofer.

[62] Pascal Paillier. 1999. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In *International Conference on the Theory and Application of Cryptographic Techniques (EUROCRYPT '99)*, Vol. 1592. Springer, 223–238.

[63] Vijayakrishnan Pasupathinathan, Josef Pieprzyk et al. 2008. A Fair E-Tendering Protocol. In *5th International Conference on Security and Cryptography (SECRYPT '08)*. SCITEPRESS, 294–299.

[64] Constantinos Patsakis. 2017. OrderRevealingEncryption. https://github.com/kpatsakis/OrderRevealingEncryption.

[65] Jan Pennekamp, Erik Buchholz et al. 2021. Collaboration is not Evil: A Systematic Look at Security Research for Industrial Use. In *Workshop on Learning from Authoritative Security Experiment Results (LASER '20)*. ACSA.

[66] Jan Pennekamp, Erik Buchholz et al. 2020. Privacy-Preserving Production Process Parameter Exchange. In *36th Annual Computer Security Applications Conference (ACSAC '20)*. ACM, 510–525.

[67] Jan Pennekamp, Frederik Fuhrmann et al. 2021. *Confidential Computing-Induced Privacy Benefits for the Bootstrapping of New Business Relationships.* Technical Report RWTH-2021-09499. RWTH Aachen University. Blitz Talk at the 2021 Cloud Computing Security Workshop (CCSW '21).

[68] Jan Pennekamp, Frederik Fuhrmann et al. 2023. Offering Two-Way Privacy for Evolved Purchase Inquiries. https://github.com/COMSYS/purchase-inquiries.

[69] Jan Pennekamp, René Glebke et al. 2019. Towards an Infrastructure Enabling the Internet of Production. In *2019 IEEE International Conference on Industrial Cyber Physical Systems (ICPS '19)*. IEEE, 31–37.

[70] Jan Pennekamp, Martin Henze et al. 2019. Dataflow Challenges in an *Internet* of Production: A Security & Privacy Perspective. In *ACM Workshop on Cyber-Physical Systems Security & Privacy (CPS-SPC '19)*. ACM, 27–38.

[71] Jan Pennekamp, Martin Henze et al. 2021. Unlocking Secure Industrial Collaborations through Privacy-Preserving Computation. *ERCIM News* 126, 24–25.

[72] Jan Pennekamp, Johannes Lohmöller et al. 2023. Designing Secure and Privacy-Preserving Information Systems for Industry Benchmarking. In *Proceedings of the 35th International Conference on Advanced Information Systems Engineering (CAiSE '23)*, Vol. 13901. Springer, 489–505.

[73] Jan Pennekamp, Roman Matzutt et al. 2021. The Road to Accountable and Dependable Manufacturing. *Automation* 2, 3, 202–219.

[74] Benny Pinkas, Thomas Schneider et al. 2014. Faster Private Set Intersection Based on OT Extension. In *23rd USENIX Conference on Security Symposium (SEC '14)*. USENIX Association, 797–812.

[75] Raluca Ada Popa, Catherine M.S. Redfield et al. 2011. CryptDB: Protecting Confidentiality with Encrypted Query Processing. In *23rd ACM Symposium on Operating Systems Principles (SOSP '11)*. ACM, 85–100.
[76] Víctor Julio Ramírez-Durán, Idoia Berges et al. 2020. ExtruOnt: An ontology for describing a type of manufacturing machine for Industry 4.0 systems. *Semantic Web* 11, 6, 887–909.
[77] Peter Rindal and Mike Rosulek. 2017. Malicious-Secure Private Set Intersection via Dual Execution. In *2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17)*. ACM, 1229–1242.
[78] Ronald Rivest, Adi Shamir et al. 1978. A Method for Obtaining Digital Signatures and Public-key Cryptosystems. *Commun. ACM* 21, 2, 120–126.
[79] Armin Ronacher. 2010. Flask. https://palletsprojects.com/p/flask/.
[80] Mark D Ryan. 2014. Enhanced Certificate Transparency and End-to-end Encrypted Mail. In *21st Annual Network and Distributed System Security Symposium (NDSS '14)*. Internet Society.
[81] Leo Sauermann and Richard Cyganiak. 2008. Cool URIs for the Semantic Web. W3C Interest Group Note.
[82] Ask Solem. 2009. Celery: Distributed Task Queue. http://www.celeryproject.org/.
[83] Marten Van Dijk, Craig Gentry et al. 2010. Fully Homomorphic Encryption over the Integers. In *Proceedings of the 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT '10)*, Vol. 6110. Springer, 24–43.
[84] Hal R. Varian. 1989. Chapter 10 Price discrimination. In *Handbook of Industrial Organization*. Vol. 1. Elsevier, 597–654.
[85] VDMA e.V. (Mechanical Engineering Industry Association). 2015. The VDMA – VDMA. https://www.vdma.org/en/.
[86] Alexander Viand, Patrick Jattke et al. 2021. SoK: Fully Homomorphic Encryption Compilers. In *2021 IEEE Symposium on Security and Privacy (SP '21)*. IEEE, 1092–1108.
[87] Gordon Walker and David Weber. 1987. Supplier Competition, Uncertainty, and Make-or-Buy Decisions. *Academy of Management Journal* 30, 3, 589–596.
[88] Ulrich Weigel and Marco Ruecker. 2017. *The Strategic Procurement Practice Guide*. Springer.
[89] Mark D. Wilkinson, Michel Dumontier et al. 2016. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data* 3.
[90] David Wood, Markus Lanthaler et al. 2014. RDF 1.1 Concepts and Abstract Syntax. W3C Rec..
[91] Yu Xia, Bintong Chen et al. 2008. Market-Based Supply Chain Coordination by Matching Suppliers' Cost Structures with Buyers' Order Profiles. *Management Science* 54, 11, 1861–1875.
[92] Xun Xu. 2017. Machine Tool 4.0 for the new era of manufacturing. *The International Journal of Advanced Manufacturing Technology* 92, 5, 1893–1900.
[93] Jiho Yoon, Srinivas Talluri et al. 2020. Procurement decisions and information sharing under multi-tier disruption risk in a supply chain. *International Journal of Production Research* 58, 5, 1362–1383.
[94] Yong Zeng, Lingyu Wang et al. 2012. Secure collaboration in global design and supply chain environment: Problem analysis and literature review. *Computers in Industry* 63, 6, 545–556.
[95] Jan Henrik Ziegeldorf, Jan Pennekamp et al. 2017. BLOOM: BLoom filter based Oblivious Outsourced Matchings. *BMC Medical Genomics* 10 (Suppl 2).

A AN OVERVIEW ON PROCUREMENT PROCESSES

In the following, we provide more background on today's procurement processes in industry that are the foundation for the subsequent manufacturing of products. Furthermore, we highlight the process changes that follow from our newly-proposed privacy-preserving purchase inquiry.

Today, procurement processes in industry typically cover the phases supplier identification and supplier management [12], as we detail in Figure 9. While the main goal of the supplier identification is the initial selection of supplier(s), the subsequent supplier management contains steps that directly concern concretely selected supplier(s). Overall, the two general phases consist of altogether seven individual steps, which we now discuss in more detail.

First, during the *requirements elicitation*, the buyer compiles the product's characteristics to make their "make-or-buy" decision [87], i.e., either the product should be manufactured locally, or procured elsewhere. To this end, the buyer usually crafts a (detailed) specification sheet, which lists all relevant product properties. Based on these requirements, during the *supplier identification*, the initial identification of all potential suppliers commences, i.e., to create a set of potential sellers. Subsequently, in the *supplier pre-assessment*, the buyer conducts a preliminary review of the identified suppliers to evaluate them. In these critical steps, the buyer and potential suppliers

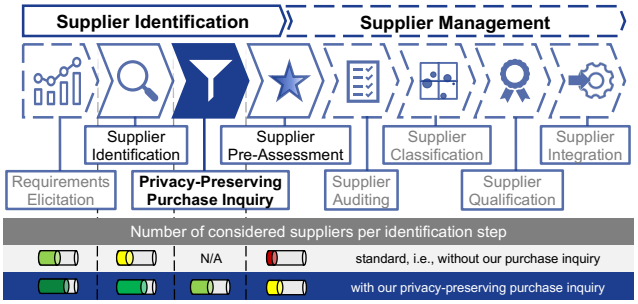

Fig. 9. We augment existing, established procurement processes [12] with a non-invasive, additional step in the supplier identification. Thereby, we intend to increase the set of considered suppliers in practice.

(sellers) exchange (sensitive) information. Afterward, based on these details, the supplier(s) usually prepare their offers. Eventually, they are reviewed by the inquiring buyer.

After this step, one (or multiple) supplier(s) are selected based on the available offers and other relevant information, e.g., delivery schedules. Thus, this step concludes the supplier identification phase, where the trade is finalized. Afterward, the procurement process continues with the supplier management phase, which focuses on the overall relationship between buyer and selected seller, i.e., it is more general than a specific purchase decision.

In this phase, the selected supplier is subject to a more detailed examination to ensure the delivery of the product with regard to the negotiated quality and delivery date, which is often realized in the form of a *supplier auditing* and a subsequent *supplier classification*. The classification enables the supplier to be rated more precisely for subsequent procurements, i.e., classify suppliers based on the strategic relevance for the buyer. Then, as part of the *supplier qualification*, the buyer gathers and formalizes all aspects of their (future) business relationship, also covering a variety of economic aspects. Finally, during the *supplier integration*, the buyer integrates the supplier in their product development phase, thus creating an economic dependency by directly incorporating suppliers into local (decision) processes. Thus, after concluding the supplier management, the buyer has (i) a deepened business relationship with the seller and (ii) additional (background) information on the seller. This information can prove in subsequent procurements as additional soft criteria, e.g., schedule flexibility, sustainability, or timeliness, is recorded at first hand by the buyer. Such details are especially relevant when having multiple suppliers that provide nearly identical offers.

As we detail in Section 2, when considering an exchange of information, the most critical steps during the identification of new suppliers are at an early phase of the procurement process, mainly because it also involves completely unknown, likely untrusted suppliers. Thus, buyers cannot forecast what these potential sellers will do with such (sensitive) information, i.e., they fear for their competitive advantage. Likewise, suppliers are also not interested in disclosing all of their capabilities, as they also fear damaging effects. Consequently, both buyers and sellers avoid sharing information upfront, if possible.

To account for this dilemma (cf. Section 2.2), we propose the introduction of a new, eighth step—***privacy-preserving purchase inquiries*** (cf. Figure 9)—that directly integrates into existing procurement processes. Due to its non-invasive nature, it only slightly affects the traditional steps (*supplier identification* and *supplier pre-assessment*) as parts of these steps are now covered by our newly added step. All other steps of the procurement process (gray labels in Figure 9) remain completely oblivious to this change, i.e., companies can, for the most part, stick to their best practices. For technical details on the new step, we refer to our presented design (cf. Section 5).

Our new privacy-preserving step is highly beneficial for use in real-world deployments, because it allows buyers to consider significantly more sellers at the early phases of the procurement process

(supplier identification phase). Using technical means, we ensure that no sensitive information is leaked to other parties, and especially unsuitable sellers. Given the larger number of initially considered sellers (visualized at the bottom of Figure 9), buyers might be able to source their sellers from a large(r) set of fitting suppliers. With today's practices, these suppliers might not have been considered at all (cf. Section 2.1).

B cHPI: OUR CLOUD-TAILORED DESIGN VARIANT

In this section, we present cHPI, a cloud-tailored design variant of HPI, in more detail. First, in Section B.1, we give an overview of this variant. Afterward, in Section B.2, we report on our evaluation and compare the performance of cHPI to HPI. This discussion also covers our real-world use cases (cf. Section 7.4) and the corresponding performance measurements. Finally, in Section B.3, we augment our security discussion from Section 7.5 to also capture the security properties of cHPI.

B.1 Protocol Overview of cHPI

We now introduce our cloud-tailored variant of HPI. cHPI follows the same phases as HPI, while allowing the seller to offload the homomorphic computation of HPI to the cloud, i.e., by proposing cHPI, we relieve the seller of computational workload (reducing the overhead of privacy-preserving purchase inquiries for sellers). While ① the pre-processing of cHPI is identical to HPI (as presented in Section 5.3.2), ② Buyer B and Seller S_i afterward agree on a random vector N to blind their pre-processed values (C_B and M_{S_i}). Otherwise, the cloud would know the seller's prices as they are not encrypted in HPI due to the local computation (no threat of information leakage).

As we illustrate in Figure 10, the subsequent protocol steps of cHPI are slightly different as well. Buyer B must share her encrypted inputs with both ③ the cloud (C_B, unblinded vector) to eventually allow for a removal of the blinds from the cloud-computed result by the seller and ④ the seller ($\langle N \cdot C_B \rangle$, blinded scalar product). ⑤ Using the seller's blinded prices ($\langle N \cdot M_{S_i} \rangle$), the cloud can homomorphically compute the (computationally-expensive) scalar product, i.e., R' (as in HPI). Then, ⑥ the cloud returns this result (R') to the seller, who ⑦ removes the random vector N using the blinded scalar product $\langle N \cdot C_B \rangle$. Finally, as the only local computation step, the seller subtracts the blinded $\perp_B$ to compute the intermediate result R". ⑧ She further blinds the result with s to obtain $R_B^{S_i}$, and ⑨ returns this single result to the buyer, who ⑩ can decrypt the result using her private PHE key. In cHPI, the result's semantics are identical to the one in HPI.

Overall, cHPI reduces the seller's computational load at the expense of greatly increased network traffic for the seller, following the offloading to the cloud. Thus, depending on the available computing and networking resources, cHPI can be a suitable alternative.

B.2 Measuring the Performance of cHPI

We repeated our evaluation from Section 7.2 to study the performance of cHPI. Additionally, we also measured our real-world use cases (cf. Section 7.4) using cHPI.

B.2.1 Performance Measurements. We repeated our measurements for cHPI and compare them to HPI to underline the (positive) performance implications of this cloud-tailored design variant.

First, in Figure 11 (left), we detail the overall runtime of cHPI. We notice that the offloading takes most of the runtime (excluding the buyer's initial pre-processing) in cHPI. Notably, in comparison to HPI (Figure 11 (right)), the seller's workload is reduced by nearly one order of magnitude due to the offloaded scalar product computation relieving many of her resources. Thus, the offloading to the cloud significantly relieves the seller as most workload is now offloaded to the cloud. In contrast, the buyer's pre-processing remains identical as she still has to encrypt the same number of PHE ciphertexts when using cHPI.

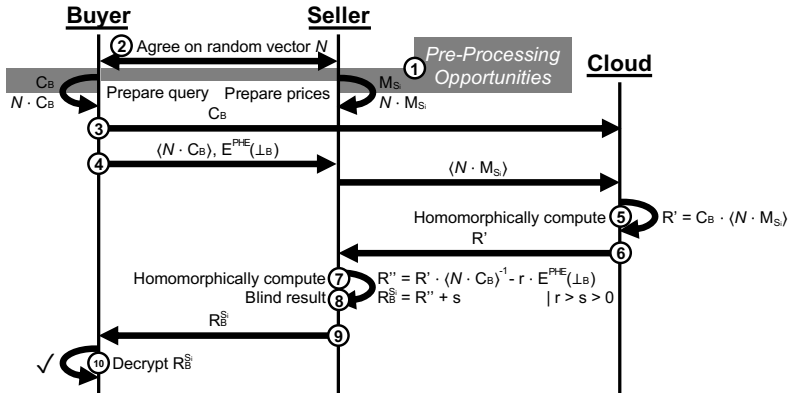

Fig. 10. In comparison to HPI, the seller offloads the expensive homomorphic computation (scalar product) to a cloud in cHPI for performance reasons. Thus, cHPI reduces her local workload and memory needs.

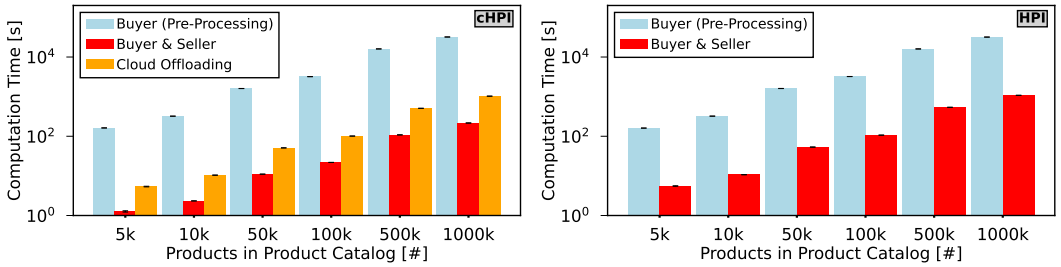

Fig. 11. cHPI (left) reduces the seller's workload by one order of magnitude in comparison to the purely local HPI protocol (right). Thereby, it is a adequate candidate in settings with resource-constrained sellers.

Concerning the RAM usage, the seller is also relieved in cHPI as she no longer has to keep all PHE ciphertexts in memory, as we detail in Figure 12. Instead, the cloud inherits this task, effectively reducing the seller's needs when comparing HPI to cHPI. As for HPI (cf. Section 7.2.2), we could also adjust our cHPI implementation to support batch processing. Thus, we could end up with an upper bound for the cloud if needed.

Finally, as expressed before (cf. Section B.1), we measure significant differences in network usage when comparing HPI to cHPI. As we illustrate in Figure 13, the seller has the burden of offloading her blinded prices to the cloud when using cHPI, resulting in a transmission overhead when compared to HPI. Thus, in cHPI, the seller's network use correlates with the number of products. This observation also holds for the buyer in both HPI and cHPI. The cloud's network use is constant as the result is always a single PHE ciphertext, which is irrespective of the number of products that were used as initial input. This behavior matches the seller's network usage in HPI.

B.2.2 Real-World Setting. To underpin the feasibility of cHPI, we also analyze the performance of our real-world datasets, with the performance being independent of the exact inputs (cf. Section 7.2).

Tool Use Case. In comparison to the reported runtime of HPI (21.6 min±0.2 min, cf. Section 7.4.2), we measure an overall runtime of 21.6 min ± 5.2 s when using cHPI for our first real-world use case. Hence, offloading the computation of small real-world datasets does not influence the overall performance significantly with unconstrained network links.

Machine Tool Use Case. With larger product catalogs, the offloading and additional blinding in cHPI slightly prolong the overall runtime of the design variant in comparison to HPI. In particular, for cHPI, we measure a runtime of 530 min ± 4 min for our second real-world use case. With the same input, the runtime of HPI is slightly faster, measured as 525 min ± 3 min (cf. Section 7.4.2).

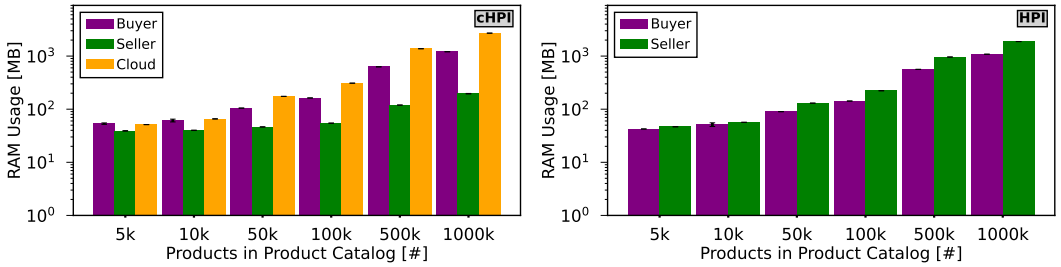

Fig. 12. cHPI (left) significantly reduces the seller's local RAM usage when compared to HPI (right) by utilizing the cloud. Therefore, the virtually unlimited resources in the cloud easily allow for parallel executions.

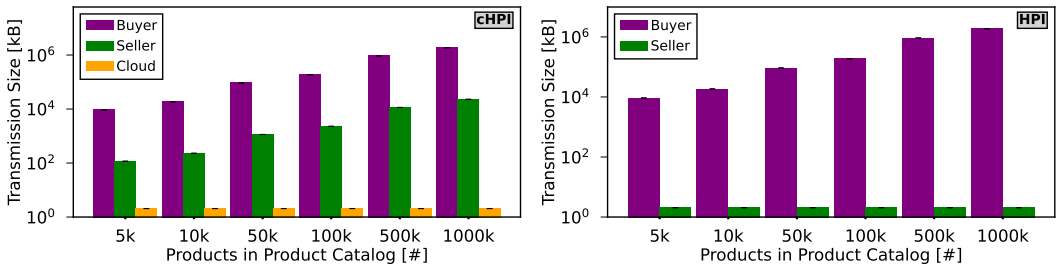

Fig. 13. In contrast to HPI (right), the seller offloads the PHE-based computation to the cloud when using cHPI (left). Thus, companies need to consider the trade-off between computational and network load.

Our results corroborate that cHPI (just like our initial designs HPI and PPI) is feasible with real-world data. While the performance of our designs is oblivious to the exact inputs, real-world settings do not influence the performance in comparison to our prior evaluation.

B.2.3 Conclusion. While the runtime of cHPI is identical to HPI, in practice, the presented results would likely differ as most companies only have access to constraint network links. The offloading in cHPI significantly increases the transmission size of the seller when compared to HPI. Regardless, depending on the scenario, the significantly reduced computational and memory needs at the seller will outweigh this disadvantage. Given that the overall runtime is still suitable for real-world use, even with constrained network links, e.g., a product catalog with 1 Mio. products incurs a seller upload of only 23 MB, cHPI is a promising design variant for real-world deployments.

B.3 Discussing the Security of cHPI

For the security of cHPI, the same aspects, including the attacker model, as for PPI and HPI hold (cf. Section 7.5). Additionally, we consider the third party (cloud) in cHPI to be untrusted, i.e., it must not have access to any data in the clear or observe any insightful patterns. In the remainder of this section, we discuss the security-related differences of cHPI in comparison to HPI (cf. Section 7.5).

Information Leakage. Like HPI, cHPI is secure and privacy-preserving by design: It still bases on PHE, i.e., only the buyer can decrypt the ciphertexts. All remaining parties (seller and cloud) directly operate on ciphertexts, i.e., they only compute data homomorphically.

To ensure privacy while offloading unencrypted inputs to the cloud, the seller must blind her price expectations to still achieve seller privacy (**G2**). To this end, buyer and seller jointly agree on a random vector N. Consequently, the cloud cannot conduct any correlation attacks as the individual prices are obfuscated in a different way for each protocol run and query. The buyer's pre-processing capabilities are not limited in any way by this step as the cloud still receives the unblinded PHE ciphertexts C_B. Using the scalar product $\langle N \cdot C_B \rangle$, the seller can eventually remove

the random vector N, i.e., she locally unblinds the cloud-computed result. As for HPI, the seller also blinds the result in cHPI to ensure confidentiality (cf. Section 7.5) before returning the result to the buyer. Thus, we still establish seller privacy (**G2**) when deploying cHPI.

Entity Misbehavior. Our previous security discussion (cf. Section 7.5) is independent of the used protocol and thus also holds when considering the security of cHPI.

Collusion Attacks. In cHPI, we have to discuss collusion attacks due to the third party.

Buyer & Third Party. If a buyer is colluding with the third party, they can jointly extract all price expectations of the seller, e.g., to only offer the seller's lowest selling prices during the subsequent negotiations: The third party has access to the blinded prices, and the buyer knows the random vector N, which can be used to remove the applied blinds. Hence, cHPI cannot tolerate such a collusion to ensure seller privacy (**G2**). However, when relying on a trusted operator, as for PPI (cf. Section 7.5), we can effectively reduce the probability of such an attack.

Seller & Third Party. A collusion of the seller and the third party has no negative consequences for the buyer, because both parties operate with PHE ciphertexts anyway, and without the encryption key, they cannot decrypt the ciphertexts. Thus, such a collusion has no effect on the buyer's privacy (**G1**). If the seller trusts the third party, we could even refrain from blinding the price expectation using N, reducing the overhead of the cloud offloading in cHPI.

Our design variant cHPI enables privacy-preserving purchase inquiries (as with HPI) while simultaneously reducing the required resources for participating sellers. Thus, it serves as an alternative when (local) computing resources are scarce. As the cloud in cHPI is oblivious of the processed data, it fits nicely with our design goals and can be considered when the disadvantages (here mainly, networking overhead and operating costs) from an offloading of computations to the cloud are not decisive.

C MODELING MACHINE TOOLS IN PRACTICE

To give some more insights into our real-world use cases, which we presented in Section 7.4.1, we discuss the relevant properties of tools (Table 1a) and machine tools (Table 1b). These properties [10] were compiled by the Laboratory for Machine Tools and Production Engineering (WZL) of RWTH Aachen University in conjunction with leading companies in the domain of machine tools. Notably, researchers use these properties frequently for other evaluations, i.e., our datasets are well-known in the domain of machine tools. Thus, they constitute a realistic foundation for the representation of products, i.e., the underlying properties are well suitable when modeling products in said domain.

Table 1. Overview on the modeling parameters that are relevant as part of our real-world use cases.

(a) In our real-world **tool use case**, we consider 7 properties that are expressed by a total of 10 crucial parameters. The defined number of bins is parameter-specific.

Property	Parameter	# Bins
Size factor	maximum mounting length	5
Shape compl. factor	number of diff. contour elements	3
	percentage of 3D-surfaces	2
Aspect ratio	largest cavity depth	4
	minimal horizontal radius	3
Filigree factor	minimal feature size	3
Surface factor	roughness	3
	percentage of active surface	2
Tolerance factor	IT class	2
Material factor	machinability	3

(b) For our **machine tool use case**, we consider 3 general aspects. We define them with 14 parameters in total, i.e., we can express complex products.

Property	Parameter	# Bins
Productivity	palletizing system	2
	workpiece changer	3
	maximum removal rate	3
Flexibility	installation space	3
	maximum weight	3
	maximum cutting height	3
	number of machine axes	3
	conicity angle	3
Quality	machine age	3
	climate control	2
	temperature compensation	2
	dielectric	2
	positional accuracy	3
	minimum wire diameter	3