Journal of
Cybersecurity
and Privacy

Article

Potential of Homomorphic Encryption for Cloud Computing Use Cases in Manufacturing

Raphael Kiesel [1,*], Marvin Lakatsch [1], Alexander Mann [2], Karl Lossie [2], Felix Sohnius [1] and Robert H. Schmitt [1,2]

[1] Laboratory for Machine Tools and Production Engineering (WZL), RWTH Aachen University, Campus-Boulevard 30, 52074 Aachen, Germany
[2] Fraunhofer Institute for Production Technology IPT, Steinbachstraße 17, 52074 Aachen, Germany
* Correspondence: mail@raphael-kiesel.de; Tel.: +49-176-32641066

Abstract: Homomorphic encryption enables secure cloud computing over the complete data lifecycle. As so-called in-use encryption methodology, it allows using encrypted data for, e.g., data analysis—in contrast to classic encryption methods. In-use encryption enables new ways of value creation and an extensive use of cloud computing for manufacturing companies. However, homomorphic encryption is not widely implemented in practice yet. This is mainly since homomorphic encryption has higher computation times and is limited regarding its calculation operations. Nevertheless, for some use cases, the security requirements are a lot stricter than, e.g., timeliness requirements. Thus, homomorphic encryption might be beneficial. This paper, therefore, analyzes the potential of homomorphic encryption for cloud computing in manufacturing. First, the potential and limitations for both classic and homomorphic encryption are presented on the basis of a literature review. Second, to validate the limitations, simulations are executed, comparing the computation time and data transfer of classic and homomorphic encryption. The results show that homomorphic encryption is a tradeoff of security, time, and cost, which highly depends on the use case. Therefore, third, manufacturing use cases are identified; the two use cases of predictive maintenance and contract manufacturing are presented in detail, demonstrating how homomorphic encryption can be beneficial.

Keywords: homomorphic encryption; cloud computing; manufacturing

check for updates

Citation: Kiesel, R.; Lakatsch, M.; Mann, A.; Lossie, K.; Sohnius, F.; Schmitt, R.H. Potential of Homomorphic Encryption for Cloud Computing Use Cases in Manufacturing. *J. Cybersecur. Priv.* **2023**, *3*, 44–60. https://doi.org/10.3390/jcp3010004

Academic Editor: Savio Sciancalepore

Received: 18 December 2022
Revised: 28 January 2023
Accepted: 29 January 2023
Published: 6 February 2023

1. Introduction

Against the background of constantly increasing demands on modern IT infrastructures, especially in terms of their storage capacity and computing power when processing data, the application of cloud computing in manufacturing is growing rapidly. In particular, small and medium-sized enterprises are no longer dependent on having large data storage facilities or on carrying out complex data analyses on internal systems [1–4]. In addition to the cost reduction by adapting storage capacity and computing performance based on company's needs, cloud computing has several advantages. Due to permanent, location-independent availability of relevant data, decision-making times are shorter [5,6]. Cloud computing allows cross-site access to systems through central synchronization and analysis of data [7]. Furthermore, off-site cloud backups protect relevant data in the event of a local disaster [8]. These advantages result in a predicted globally annual growth rate of cloud computing in manufacturing of 16.1% from 2020 to 2030 [9]. Global revenue from public cloud offerings is estimated to exceed 350 billion USD in 2022 [10]. Despite this immense potential, cloud computing is currently mostly used to only store data. The main reason for this limited use of cloud computing lies in data security. As shown by the cloud security alliance, security threats are the major reasons why companies are not yet using cloud computing for purposes other than pure storage [11,12].

Security challenges include the risk of data misuse by the cloud provider, cloud infrastructure security, and control over third-party providers [1,11]. However, the challenge level of data security depends on the state of data during cloud computing [1,13]. The literature distinguishes among three different states of data: data in storage, data in transit, and data in use [13–15]. Data in storage and data in transit have been protected with classical methods of cryptography for years. However, classical encryption methods cannot be applied for data in use, as they do not allow executing calculations with cipher text. Thus, data are decrypted in the cloud to be used for analyses and, therefore, revealed to the cloud provider. This consequently results in the abovementioned security risks [13,16]. To solve these risks, in-use encryption technologies are the focus of current research. They allow calculations with encrypted data. The in-use encryption technology "homomorphic encryption" promises a high potential to solve security risks within cloud computing [14,15,17,18]. It requires, however, more computing efforts than classic encryption technologies and is limited regarding certain calculation operations [19].

Considering both the advantages and the current challenges of homomorphic encryption, the goal of this paper is to evaluate the potential of homomorphic encryption for cloud computing use cases in manufacturing. Therefore, Section 2 presents the fundamentals within this field of research. Section 3 presents encryption for cloud computing, analyzing the limitations of classic encryption, as well as the potential and challenges of homomorphic encryption. Section 4 presents manufacturing use cases for cloud computing in manufacturing and analyzes their improvement when deploying homomorphic encryption. Section 5 summarizes the findings and gives an outlook for further research requirements. Sections 3.3 and 4 represent the main contribution and novelty of this paper, as the paper is the first to simulate, in detail, homomorphic encryption's applicability for manufacturing, along with manufacturing use cases.

2. Fundamentals

This section defines the fundamental terms relevant to evaluate the potential of homomorphic encryption for cloud computing use cases in manufacturing.

2.1. Cloud Computing

The National Institute of Standards and Technology (NIST) defines cloud computing as "*[. . .] a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources [. . .] that can be rapidly provisioned and released with minimal management effort or service provider interaction*" [20]. According to the NIST, cloud computing, thus, refers to access at any time to a pool of computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction [20].

Cloud computing can be categorized according to the cloud deployment model, of which NIST defines four: private cloud (1), public cloud (2), community cloud (3), and hybrid cloud (4). Deployment models basically differ regarding cloud ownership and cloud accessibility, which in turn influences the cloud's functionalities and the cloud cost [13,20]. A private cloud (1) is owned by one organization. Access to the private cloud is restricted to this organization's employees. A private cloud allows the highest flexibility regarding cloud functionalities but is the most expensive deployment model. In contrast, a public cloud (2) is accessible by the broad public. It is owned and provided by a cloud provider. In general, it is the cheapest deployment option, but also limits the cloud's functionalities. A community cloud (3) is accessible for a defined group of organizations that share the same interests or goals. It is either owned by one of the community members or an external cloud provider. If two or more of the deployment models mentioned are combined, this is referred to as hybrid cloud (4) [3,13,20,21]. For further analysis, the tradeoff among functionality, cost, and accessibility is important.

Independent of the deployment model, three different states of data exist during cloud computing being relevant for encryption [13–15]:

- *In-storage data*: data that are currently stored in the cloud,
- *In-transit data*: data that are currently sent from the cloud to the user's storage,

- *In-use data*: data that are currently being used, e.g., for analyses.

As not every encryption method is applicable for each state, this differentiation is important for encryption, as further described in Section 2.2.1.

2.2. Encryption

2.2.1. General Encryption Process

Encrypting information, messages, or data has been established for decades. The oldest known encryption technologies were already used in Sparta or Rome for military communication [22]. Encryption aims to prevent unauthorized persons from accessing information, messages, or data. To achieve this goal, almost all encryption technologies are designed to provide maximum computational security. This means that the computational effort to decrypt data is not in proportion to the benefit of its reveal [23,24].

Regardless of the encryption technology, the general encryption process is identical and can be described based on the quintuple (P, E, K, C, and D) of cryptosystems, as shown in Figure 1. The information or data to be secured is given as plaintext (P). To turn this plaintext into an illegible form, encryption (E) is executed. Encryption is a mathematical transformation of the plaintext by combining an invertible encryption function with the corresponding key (K). This turns the readable plaintext into ciphertext (C), which is illegible for everyone but the intended recipient. To then make the ciphertext readable again, decryption (D) is executed, using the inverted encryption function in combination with the key to display plain text again [23–26].

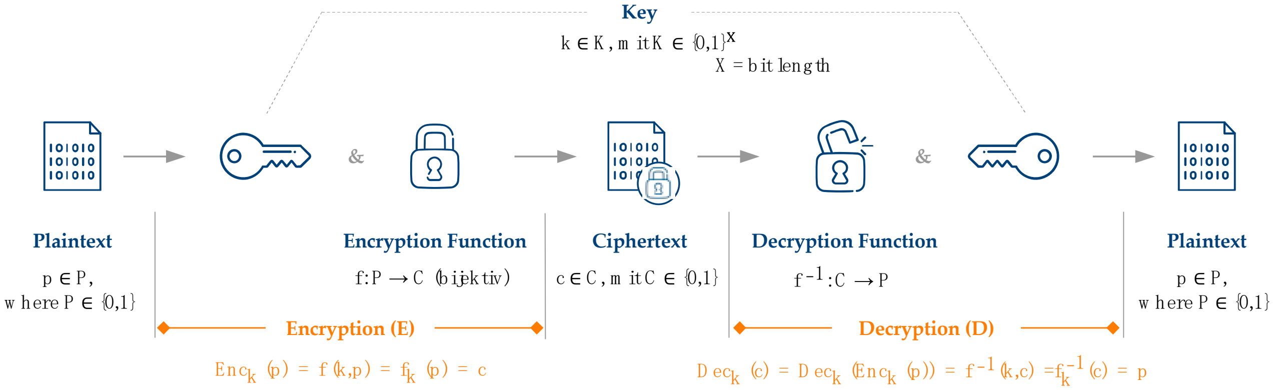

Figure 1. Basic encryption process based on the quintuple of cryptosystems.

In most cases, the encryption function is based on known algorithms [24]. However, the algorithms and functions of classical encryption technologies can only be applied to data in storage and data in transit (see Section 2.1) [27]. Calculations and analyses on data encrypted with classical encryption functions would either result in a wrong result or might even lead to errors due to the illegible ciphertext, which might not even be readable by the algorithms. Thus, calculations are not possible on the ciphertext, but only on plaintext. Data would need to be decrypted to be used for calculations and, thus, be readable by the person or company that is analyzing the data [27].

However, so-called in-use encryption technologies allow calculations on encrypted data. The most promising technology is homomorphic encryption, which is presented in Section 2.2.2 [18,28].

2.2.2. Homomorphic Encryption

Homomorphic encryption is a further development of existing cryptographic approaches. Its central feature is the usability of encrypted data, meaning that homomorphic encryption allows calculations and analyses on encrypted data [19,29]. The term homomorphic derives from the Greek words “homos” and “morphe”, which together mean “same shape” [28]. In the field of algebra, the term homomorphism is used for “a structurepreserving mapping between two algebraic structures" [30]. In terms of cryptography, this means that the decisive properties of an unencrypted dataset are passed on to the encrypted data set during encryption [29]. Figure 2 describes homomorphism of an encryption according to the introduced quintuple of cryptosystems. Analogously to the mathematical homomorphism, encryption (Equation (1)) and decryption (Equation (2)) are defined as follows:

$$e_k\left(p \lozenge p^{'}\right) = e_k(p) \circ e_k\left(p^{'}\right) = c \circ c^{'}, \tag{1}$$

$$d_k(c \circ c') = d_k(c) \lozenge d_k(c') = p \lozenge p^{'}, \tag{2}$$

where $\lozenge \in \{\oplus, \otimes\}\ \forall\ p, p^{'} \in P$, e_k is an encryption function, d_k is a decryption function, $p,\ p^{'}$ is plaintext, $c,\ c^{'}$ is cypher text, $\oplus$ denotes addition, and $\otimes$ denotes multiplication.

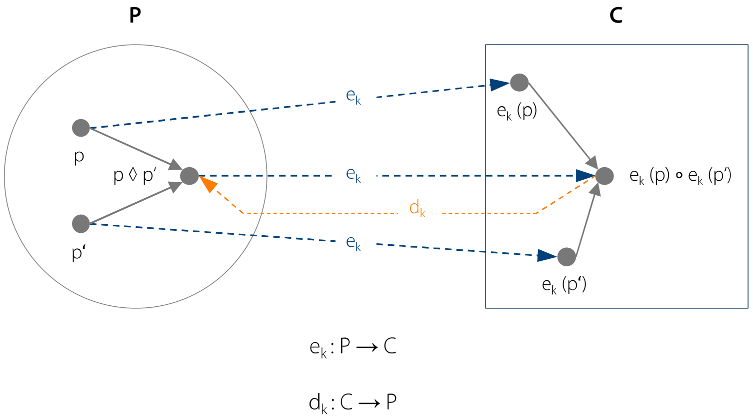

Figure 2. Homomorphism of an encryption function.

The possible operations of homomorphic encryption methods are limited to addition and multiplication in known methods. Nevertheless, homomorphic encryption methods are able to perform arbitrary operations, as proven by Gentry [31]. He demonstrated that circuit functions that are based on only additions and multiplications enable modeling arbitrary operations [31].

The possible operations and the number of operations define the three types into which homomorphic encryption is usually categorized: partially homomorphic encryption, somewhat homomorphic encryption, and fully homomorphic encryption [28,32]. The computation time increases with the complexity of the methods, with fully homomorphic encryption requiring the greatest computational effort compared to the other methods [18]. The difference between somewhat and fully homomorphic encryption, thus, lies in the error correction of fully HE, so-called bootstrapping. If calculations of encrypted data are executed with two operators, small errors occur. With each calculation, these errors increase. Using fully homomorphic encryption algorithms, bootstrapping is applied in case the defined error value crosses the threshold, which reduces the error value. This allows an unlimited number of calculations but requires higher calculation efforts. Somewhat HE does not apply bootstrapping and, thus, requires less calculation effort. However, for this reason, the number of calculations is limited until the error threshold is reached; otherwise, the calculations error would become too large [33,34].

Independent of the HE category, all methods work with consistently encrypted data, and the data security of all methods can be classified as high. Table 1 describes and characterizes these three categories.

Table 1. Characterization of homomorphic encryption categories.

HE Category	Partially HE	Somewhat HE	Fully HE
Possible operations	Addition OR multiplication	Addition AND multiplication	Addition AND multiplication
Operations amount	Unlimited	Limited	Unlimited
Calculation effort	+	++	+++
Security	+++	+++	+++

+ Low. ++ Medium. +++ High.

3. Encryption for Cloud Computing in Manufacturing

To ensure a secure cloud computing, encryption of data is key. Thus, after introducing the fundamentals of cloud computing and encryption in Section 2, Section 3 analyzes the potential and limitations of different encryption technologies for cloud computing. First, Section 3.1 presents the limitations of classic encryption for cloud computing. Second, Section 3.2 presents the potential and challenges of using homomorphic encryption during cloud computing. To further analyze and validate both potential and challenges, Section 3.3 experimentally compares classic and homomorphic encryption.

3.1. Limitations of Classic Encryption for Cloud Computing

To ensure comparability with other technologies, a scenario for the use of cloud computing is introduced. In this example, a company is assumed to use a cloud service for the storage and processing of data collected in manufacturing. These data are transferred from the on-premise company IT to the cloud in the form of a database and stored there. The following use cases should be possible: backing up encrypted manufacturing data as a database in the cloud; performing analyses using the data of the stored database.

The cloud computing scenarios for the two manufacturing use cases are shown in Figure 3. The potentials (green) and limitations (orange) are highlighted.

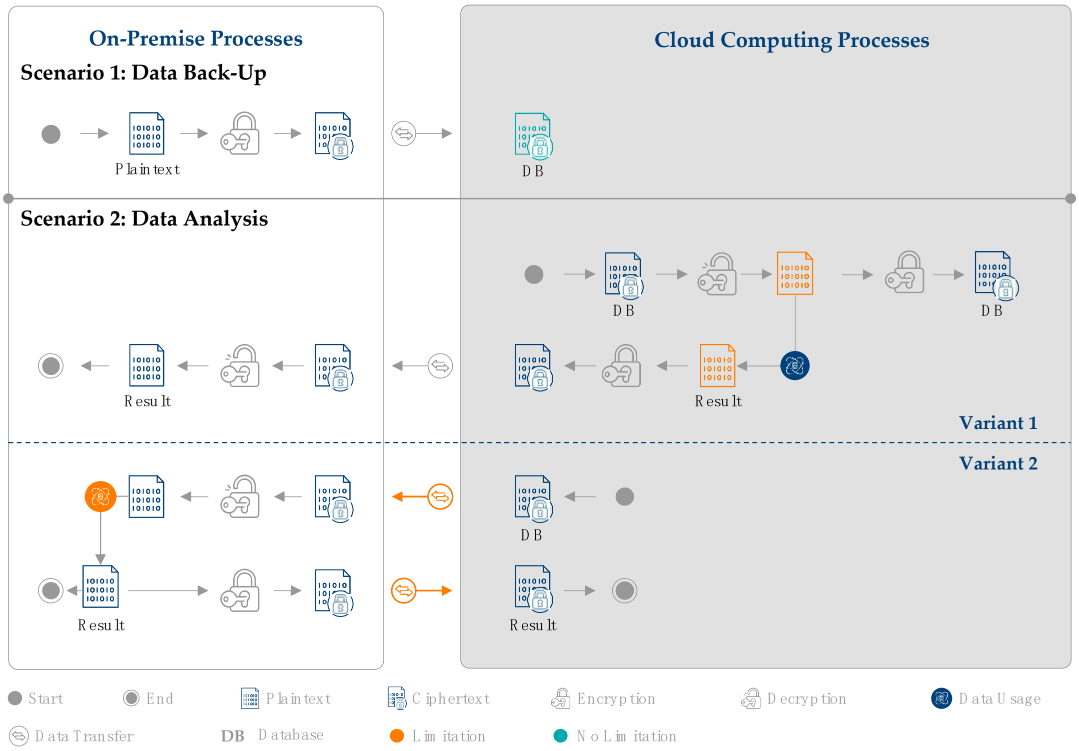

Figure 3. Limitations of classic encryption during cloud computing.

In scenario 1 (data backup), the data are encrypted at the company and transferred to the cloud. It is noticeable that no limitations occur in the pure storage of data using classic encryption methods. The database is encrypted during data transfer and data storage in the cloud. Thus, the data are secure over the entire data lifecycle. This means that classic encryption is fully suitable for data backup. For scenario 2 (data analysis), two possible variants exist. In variant 1 of scenario 2, the analysis is performed in the cloud. For the analysis to be possible, the encrypted database must first be decrypted in the cloud as it is not possible to analyze classical encrypted data (see Section 2.2.1). The result of the analysis is then encrypted in the cloud again and transferred back to the company. The company finally decrypts the result of the analysis for on-premise use. As the data analysis is carried out in the cloud, the company relies on the flexible computing resources of the cloud, which is a benefit. However, since the database is decrypted in the cloud at this point, data security is not guaranteed over the entire data lifecycle. In variant 2 of scenario 2, the analysis is carried out on premise. Here, the backed up encrypted database is first downloaded on the company's on-premise IT. Here, the database is decrypted to perform the analysis.

To have an updated backup, the result is then encrypted again and stored in the cloud. In this variant, data security is guaranteed throughout the whole lifecycle as the decrypted data are only on premise and, thus, only visible to the company. However, as the analysis is not carried out in the cloud, the computing resources of the cloud are not utilized. Furthermore, since the database must be downloaded from the cloud before each analysis, the amount of transferred data is constantly increasing. This analysis shows how the central limitation of classic encryption, i.e., not being able to use in-use data, affects the security of cloud computing: either the company relies on cloud resources with the risk of revealing the data, or the data are secure while cloud resources are not used. Pure data backup, however, has no limitations with classic encryption methods.

3.2. Potentials and Challenges of Homomorphic Encryption for Cloud Computing

3.2.1. Potentials of Homomorphic Encryption

Homomorphic encryption is an evolution of existing cryptographic approaches that has as its central feature the usability of encrypted data. As data backup in the cloud (scenario 1, Section 2.2.2) is realizable without limitations and does not include data usage, it is not further considered for homomorphic encryption. However, the potential of homomorphic encryption for data analysis in the cloud (scenario 2) is considered. As shown in Figure 4, the database is first encrypted at the company (on premise) and stored in the cloud. Once the company wants to perform an analysis, this encrypted database can now be used without any decryption in the cloud. The complete analysis is executed in the cloud, and the result is also encrypted. This encrypted result is then sent back to the on-premise IT, where it is decrypted by the company, to make the result visible. Compared to classic encryption, homomorphic encryption tackles both major limitations (see Section 3.1), as demonstrated by the color code in Figure 4. First, homomorphic encryption enables data security over the whole data life cycle. Second, as this continuous data security allows analyzing data in the cloud with any risks, homomorphic encryption allows revealing the full potential of cloud computing and flexibly using the cloud resources.

3.2.2. Current Challenges of Homomorphic Encryption

Despite the potential that homomorphic encryption offers for cloud computing, there are still several challenges regarding the implementation of homomorphic encryption in several branches in practice, as well as in manufacturing companies.

Homomorphic encryption, especially fully homomorphic encryption, currently has high energy requirements due to high computational efforts. These computational efforts result in longer computing times, thus lowering efficiency, which is why homomorphic encryption is often not yet suitable for practical use [35]. In addition to the efficiency problem, the possible uses of homomorphic encryption are also limited by the characteristics of the application in question. Highly complex algorithms or the processing of disproportionately large datasets is currently not feasible with homomorphic encryption methods [19]. Furthermore, no "one-fit-all" solution for homomorphic encryption exists yet. There are several domain-specific homomorphic encryption algorithms existing (e.g., for medical data); however, finding a suitable homomorphic encryption technology requires time and expertise and very much depends on the use case [33].

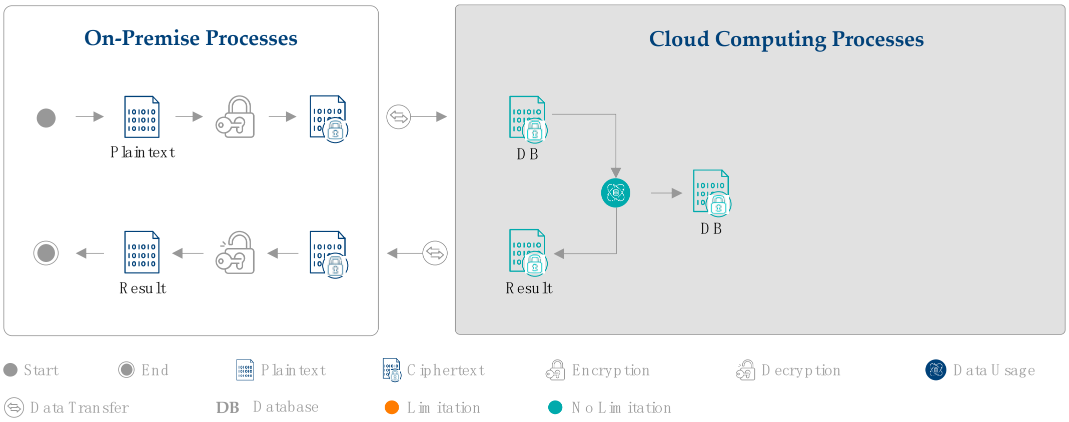

Figure 4. Potential of HE for cloud computing in production.

Nevertheless, these challenges are qualitative and must always be considered regarding the use cases and the use case requirements. Therefore, in Section 3.3, an experimental comparison in conducted to more deeply analyze the challenges and potential of homomorphic encryption for manufacturing.

3.3. *Experimental Comparison of Classic and Homomorphic Encryption*

The challenges of homomorphic encryption considered in Section 3.2 have shown that the computing time is significantly longer than for classical encryption methods. However, the chapter also highlighted that the challenges of homomorphic encryption majorly depend on the use case. To investigate the actual differences between homomorphic encryption methods and classical encryption for different operations (i.e., use cases) for cloud computing, an experimental validation was performed. In this section, the parameters and assumptions of the performed validation are explained (Section 3.3.1) and its results are presented (Section 3.3.2). As this paper is the first to conduct such a detailed analysis, this section is one of the main contributions of this paper. The used dataset and the GitLab repository are provided in the data availability statement at the end of this paper.

3.3.1. Experimental Setup

To pursue the described aim of comparing homomorphic and classic encryption for different scenarios, three different analysis scenarios are defined that could be applicable in manufacturing. These scenarios differ regarding their calculation operation as follows, as summarized in Table 2:

- *Addition* (e.g., daily consumption of cooling lubricant on one machine),
- *Multiplication* (e.g., electricity costs of the manufacturing line in a certain interval),
- *Average* (e.g., average temperature in a cold store over a certain time interval).

Depending on the need for partially (only one operator, with unlimited calculations), somewhat (both operators, with limited calculations), or fully homomorphic encryption (both operators, with unlimited calculations), different homomorphic encryption methods were applied, as summarized in Table 2. The Paillier method was used for partially homomorphic encryption, implemented via a Python library. This is a probabilistic asymmetric method whose security is based on the mathematical n-th residual problem. As is usual for asymmetric methods, the cryptosystem consists of an algorithm for key generation and the encryption and decryption algorithm. The Paillier system has an additive homomorphic property and is used, for example, in electronic voting systems [18,36,37]. The Brakerski–Fan–Vercauteren (BFV) method was used for somewhat homomorphic encryption. BFV is a homomorphic encryption scheme that uses the ring learning with error (RLWE) problem and encrypts data in ring polynomial form. The encryption and decryption in this scheme involve high-degree polynomial multiplication [38]. The Cheon–Kim–Kim–Song (CKKS) method was used for fully homomorphic encryption [39]. CKKS is also based on the RLWE assumption and does not require expensive bootstrapping operations. Plaintext messages are encoded and encrypted into vectors, and all computations are vectorized. Both BFV and CKKS were implemented over the Pyfhel Library.

Table 2. Experimental setup to validate the potential and challenges of homomorphic encryption.

Calculation Operation	Addition	Multiplication	Average
Manufacturing analysis scenario	Calculation of daily consumption of cooling lubricant on one machine	Calculation of electricity costs of the manufacturing line in a certain interval	Calculation of average temperature in a cold store over a certain time interval
Compared classic encryption technology	Python Fernet Library/Fernet		
HE category (library/algorithms)	PHE (Python/Paillier)	SWHE/FHE (Pyfhe/BFV/CKKS)	SWHE/FHE (Pyfhe/CKKS)
Hardware/operating system	AMD Ryzen 7 3700X 8 Kern 3600 MHz/Linux Debian 64 bit		

HE: homomorphic encryption, PHE: partially homomorphic encryption, SWHE: somewhat homomorphic encryption; FHE: fully homomorphic encryption.

As classic encryption has no limitation regarding calculation operations, the compared classic encryption technology was the same for each scenario (Symmetric Fernet Procedure, Python Fernet Library).

All calculations were performed on Linux Debian 64 bit and an AMD Ryzen 7 3700X 8 core 3600 MHz processor. For each scenario, sample datasets with 6048 values were generated and used for the calculations. The used dataset and the GitLab repository are provided in the data availability statement at the end of this paper.

For the experimental comparison, our main condition was that the data are secure over the whole data lifecycle. As explained in Section 3.2.1, this is given for homomorphic encryption. For classic encryption, however, this is only given if the data are analyzed on premise. Thus, during our experiment, we assumed that the database was downloaded for the calculations to not reveal data in the cloud. This refers to the comparison scenario of Scenario 2 Variant 2, as shown in Figure 3.

Since all calculation were executed on the same hardware, cloud usage was simulated by setting a latency of 20 ms for each data transfer that would have taken place between the company's on-premise IT and the cloud (this only affects the comparison of data transfer, see Section 3.3.2 and Figure 5).

3.3.2. Experimental Results

To compare homomorphic with classic encryption, two major categories are considered: computation time and the amount of transferred data. The computation time is, on the one hand, directly related to the energy usage and, thus, cost. On the other hand, it is an important factor in the manufacturing industry in order to cope with the latency requirements of applications [40]. The computation time was measured by recording the computation time of the used hardware. The data transfer is important, as the cost of data transfer is constantly rising, and data transfer might be a significant cost factor in the future [5]. The data transfer was measured by recording the number of kilobytes.

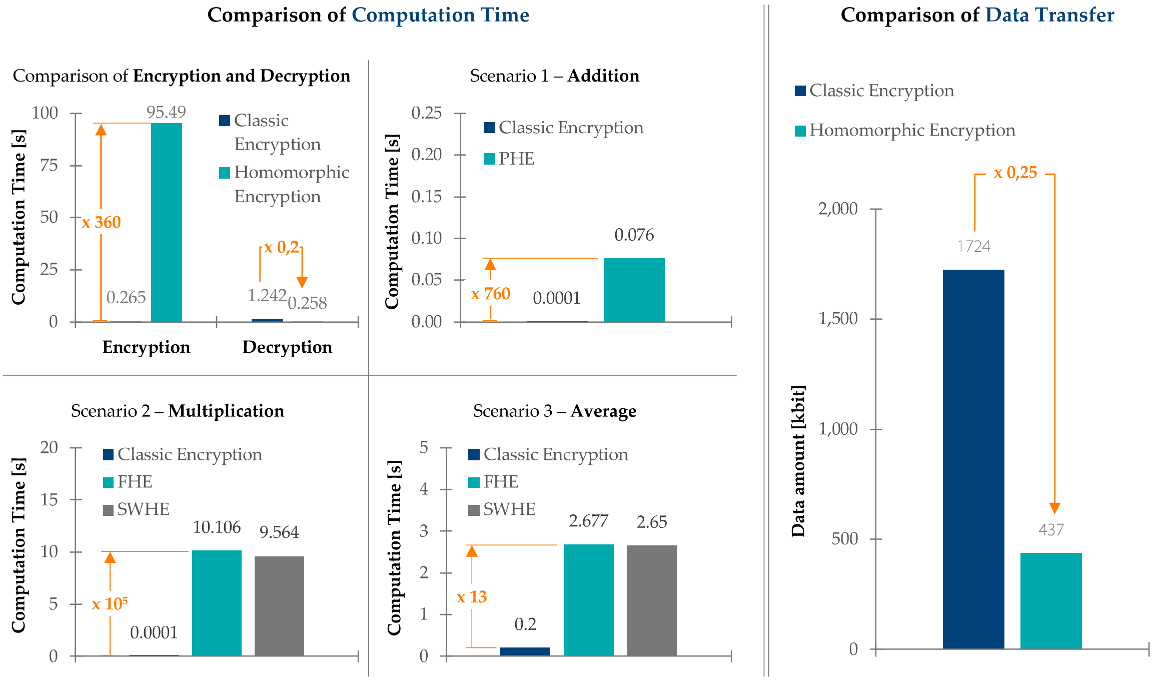

Figure 5. Results of experimental comparison.

The computation time was analyzed for encryption and decryption in three scenarios. For encryption and decryption, the computation time for the entire dataset was compared between classic encryption and homomorphic encryption. It was found that encryption using homomorphic encryption took 360 times longer than with classic encryption. The total decryption time of homomorphic encryption, on the other hand, only took 20% of classic encryption. These results can be attributed to the fact that only the results had to be decrypted when using homomorphic encryption; using classic encryption, decryption was necessary for each calculation, leading to a higher total description time. Nevertheless, despite a slightly faster decryption time, the total encryption and decryption time of homomorphic encryption was significantly higher. Moreover, the absolute value (95.49 s homomorphic encryption vs. 0.265 s classic encryption) can be an issue in manufacturing practice.

The first cloud computing scenario considered an addition. Since this calculation only covers one operator, a partially homomorphic encryption (PHE) method was considered for the comparison with classical encryption. As shown in Figure 5, the calculation times between classical and homomorphic encryption differed by a factor of 760. However, for both classic and homomorphic encryption, the absolute values were below 0.1 s. In practice, this difference might be neglectable.

The second cloud computing scenario considered a multiplication. All homomorphic encryption categories would be suitable; however, as the available PHE libraries are not suitable for multiplications, we only considered somewhat homomorphic encryption (SWHE) and fully homomorphic encryption method (FHE). Accordingly, the number of operations was limited to 16; otherwise, the calculation error of the SWHE would have been too large (see Section 2.2.2). The comparison shows that SWHE required slightly less computing time. As shown in Figure 5, both SWHE and FHE required approximately 10^5 times more computing time than classic encryption. Again, the applicability of absolute values of 10 s for SWHE and FHE must be considered for the specific application.

The third cloud computing scenario considered an average calculation, whereby both addition and multiplication operations are necessary. Therefore, only SWHE and FHE were applicable, again only with a limited number of operations, due to the increasing error of SWHE. Again, the classic encryption required less computation time (factor 13) than SWHE and FHE, which were again similar. As for the other three scenarios, the suitability of the absolute values (2.6 s) must be considered for the use case.

For the comparison of the transferred data amount, the overall data transfer for all three scenarios was analyzed. This shows that the total data amount of homomorphic encryption was only 25% of the classic encryption. This was the consequence of down- and uploading data for each calculation using classic encryption. Again, this up- and download was simulated on the same hardware, which has no effect on the data amount that would be down- and uploaded to the cloud. Therefore, the results are transferrable to a real cloud scenario.

In summary, the validation results generally confirm the knowledge of the literature presented in Section 3.3.2. Nevertheless, it is important to consider whether the absolute values of PHE, SWHE, and FHE are applicable for the practice use case.

3.4. Conclusions

Section 3 analyzed the limitations and potentials of both classic and homomorphic encryption methodologies for cloud computing in manufacturing. While homomorphic encryption guarantees data security over the whole data life cycle and a resulting exploitation of the advantages of cloud computing, homomorphic encryption requires higher computation time and, thus, energy. This was proven via a literature review, as well as in an experimental validation. However, the latter also showed that the absolute values of operations with homomorphic encryption are in a reasonable, absolute time span that might be applicable in practice use cases. Therefore, the next section considers possible use cases to analyze if homomorphic encryption can be implemented in manufacturing.

4. Use Cases for Homomorphic Encryption for Cloud Computing in Manufacturing

In this section, the integration of homomorphic encryption for certain use cases in manufacturing is analyzed. First, Section 4.1 describes how use cases were identified on the basis of the findings of Section 3. Second, Sections 4.2 and 4.3 describe the integration of homomorphic encryption for two use cases (predictive maintenance for SME and contract manufacturing for chipsets). Third, Section 4.4 summarizes the findings. As this paper is the first to conduct a systematic analysis of use cases for homomorphic encryption in manufacturing, it is a main contribution and novelty of this paper.

4.1. Identification of Use Cases

To identify use cases, the described findings of homomorphic encryption in manufacturing were combined; thus, potential benefits of cloud computing with the benefits and limitations of homomorphic encryption are considered.

On the one hand, to identify, cluster, and analyze use cases, the creativity technique used was the morphological box, as it combines creativity, logic, and systematics in one approach [41]. The morphological box consists of four characteristics:

- *Cloud benefits:* the reasons for using cloud computing as described in Section 2.1.
- *Cloud type:* the type of cloud as described in Section 2.1.
- *Homomorphic encryption (HE) category:* the applied homomorphic encryption category as described in Section 2.2.2.
- *Company size:* the company size and associated characteristics.

On the other hand, the restrictions that were concluded in Section 3.4 were used as boundary conditions and restrictions for the use case:

- *Security:* the use case must have high data security requirements, especially in the usage phase. Otherwise, classic encryption methods could be used.
- *Time:* the time restrictions of the use case must not be too low, which means that latencies in the minute range must be acceptable.
- *Cost:* the cost for homomorphic encryption compared to classic encryption are yet higher, despite savings due to lower data transfer. Thus, the implementation of homomorphic encryption should increase the revenue to cope with the increased cost.

Figure 6 summarizes the above. Thus, to identify the use case, first, the categories of the morphological box were systematically combined (e.g., availability, public cloud, SWHE, and SME). Second, it was checked whether these use cases fulfilled the boundary conditions. For example, a use case with low latency requirements was excluded from the further analysis. In the end, a total of 13 promising use cases was identified, two of which are presented below.

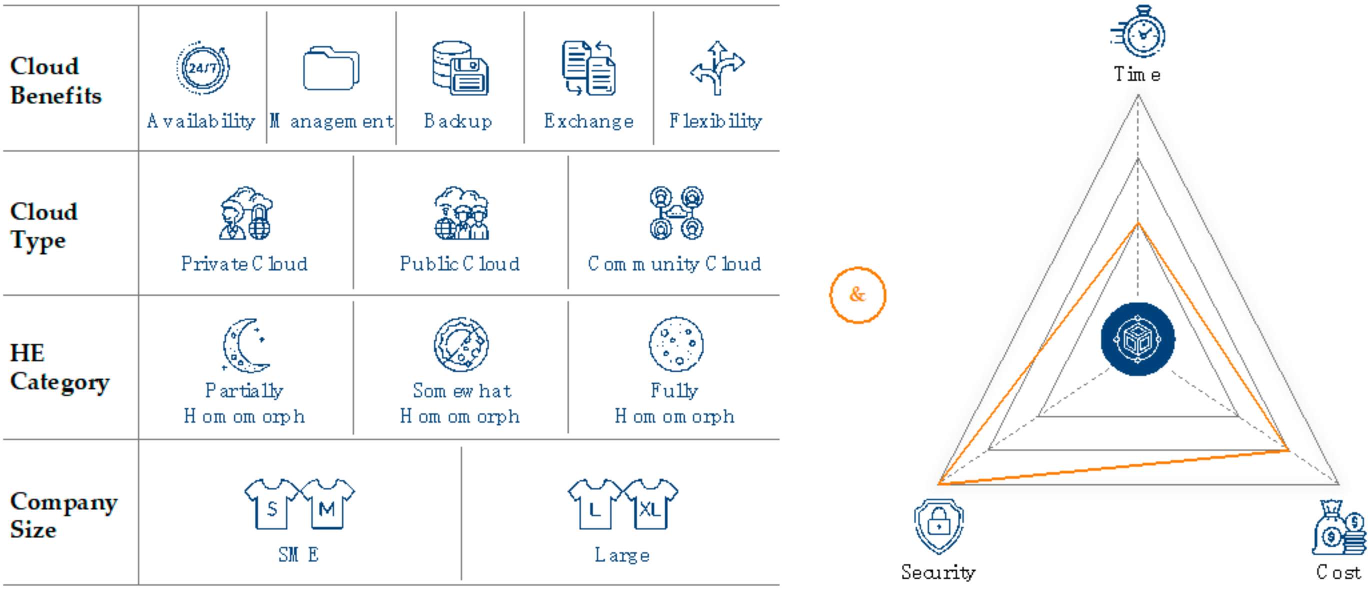

Figure 6. Identification of use cases.

4.2. Predictive Maintenance for SME

The first use case describes the secure use of predictive maintenance by means of cloud computing and homomorphic encryption in an SME. In the use case considered, the SME is a plastic fiber manufacturer where predictive maintenance is to be introduced for an extrusion process. The manufacturing process of the plastic fiber is a competitive advantage for the SME in this use case, which is why all process data should remain secret.

Predictive maintenance is a form of maintenance strategy in manufacturing companies that ensures the availability of machines and systems, guarantees high product quality, and improves overall equipment effectiveness [42,43]. However, the use of predictive maintenance is not widespread among SMEs, as high costs and computing power are incurred for the development of corresponding maintenance systems. Recording data, conducting test runs, software development, and subsequent validation for the development of such systems take financial and time resources that an SME cannot afford [44]. In addition, SMEs usually do not have the expert knowledge to model such a system or the computing resources needed to implement it [45].

Therefore, the outsourcing of predictive maintenance operations is considered. In this way, the lack of computing resources of the SME can be compensated. By using predictive maintenance as a cloud service, the costly and time-consuming development of an in-house predictive maintenance system is eliminated. This also avoids the need to build up expert knowledge within the company, as the cloud provider offers a ready-to-use predictive maintenance solution.

Due to the size of the company (SME) and the general goal of cost savings, the public cloud is chosen as the cloud type. Cloud computing, is therefore, primarily used for data management and because of its flexibility. Fully homomorphic encryption is applied as the data model requires both addition and multiplication operations (only operations possible with FHE, see Section 2.2.2) and has no defined number of calculations. The selection of these characteristics is shown in the morphological box of the predictive maintenance use case in Figure 7.

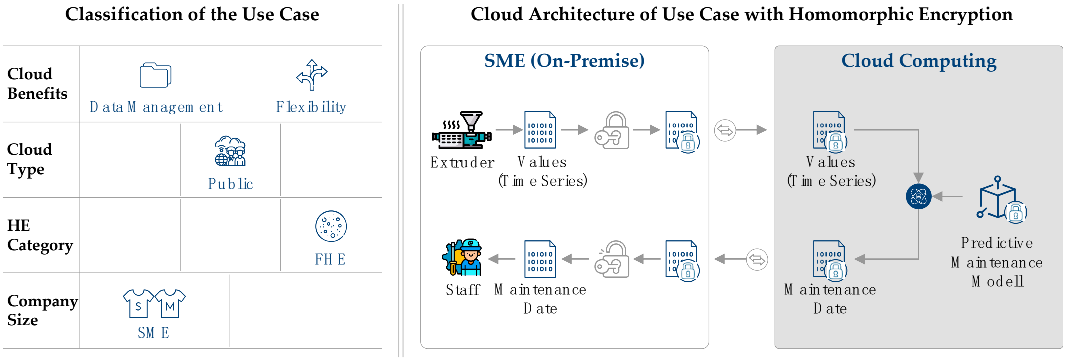

Figure 7. Classification and cloud architecture for predictive maintenance of an extrusion process.

The case also fulfills the requirements security, time, and cost. First, the data are highly sensitive for the company's business model, are analyzed by a third party, and must be secure of the whole data life cycle. Second, the planned predictive maintenance applications do not require real-time information; instead, maintenance requirements are queried at specific time intervals. Third, since the development of a predictive maintenance system can be avoided with the help of the cloud and, since a relatively inexpensive public cloud is selected, the cost requirement is also met.

The cloud architecture of the predictive maintenance use case is also shown in Figure 7. It shows the basic data and information flows between the SME and the cloud, the recording and processing of the information, and the encryption and decryption of the data. First, measured values are recorded on the extruder during the extrusion process, and a time series of condition data is generated in this way. Different status data such as process temperature, raw material flow, and coil speed are recorded, which are necessary as input for the predictive maintenance model. Since, as defined at the beginning, all data must remain secret, the measured values are homomorphically encrypted by the SME. The encrypted data are then transferred to the cloud, which also contains the cloud provider's predictive maintenance model. The model is then calculated in the cloud using the SME's input data. Since both the data and the model are homomorphically encrypted, the calculation can take place on the encrypted data; hence, in-use data are used in an encrypted manner, and end-to-end data security is ensured. The result of the model execution is a report with specific information about the machine's maintenance time, according to the model assumptions and the SME's input data. This report is encrypted in the cloud due to the use of homomorphic encryption and is transmitted to the SME. At the SME, the report is decrypted on premise, and the report is made available to maintenance in manufacturing.

For the use case, an FHE process is envisaged, as this offers the most comprehensive functionality and is, therefore, best suited for the use of a complex predictive maintenance model. In contrast to the other homomorphic encryption methods, the FHE method allows an unlimited number of additions and multiplications, with which all mathematical operations can be mapped. These functionalities form the basis for the development of a homomorphically encrypted predictive maintenance model.

The use of homomorphic encryption has different advantages in the considered use cases. For the SME, the focus is on the security of the data, as this is a process that is crucial for competition. Homomorphic encryption ensures the security of the data as it uses encrypted data over the entire data life cycle. As part of this, the use of homomorphic encryption also allows the complete outsourcing of model computation to the cloud, as the encrypted in-use data can be used in combination with the model.

In summary, the use case illustrates that costs can be saved by strategically selecting a public cloud and service offerings. Combined with the use of homomorphic encryption, resource-intensive processes can be delegated to the cloud, while ensuring the security of the data throughout its lifecycle. There is no need for a tradeoff between using cloud resources and ensuring data security, as homomorphic encryption enables both.

4.3. Contract Manufacturing for Chipsets

The second use case enables anonymized cost determination for contract manufacturers in chip manufacturing using homomorphic encryption via a cloud platform. Due to the rising costs in the manufacturing of microchips, many companies in the semiconductor industry are abandoning their own chip manufacturing [46]. So-called fabless companies design and develop microchip technology, and then outsource the entire manufacturing to contract manufacturers, the so-called chip foundries. This is advantageous for chip developers, as setting up their own manufacturing involves significantly higher costs than just developing microchips [47]. The use case focuses on the process of evaluating the manufacturing and cost feasibility of producing a new chip design.

For the chip manufacturers under consideration, the developed microchip technology is a unique feature, which must, therefore, be specially protected. Due to the existing dangers, such as theft of the technology, to which the chip design is exposed when manufacturing is outsourced, the chip developer focuses on the greatest possible data security [46]. While determining the feasibility of manufacturing, the chip developer's data should, therefore, remain encrypted throughout. However, for efficiency reasons, a digital and centralized platform for exchange between chip developers and contract manufacturers is to be used. This platform should both standardize the quotation process and be used for data exchange after conclusion of the contract.

In the considered use case, the cloud initially has the function of a platform for the exchange between chip developers and the contract manufacturer. Similar cloud manufacturing platforms already exist and represent an Internet-based variant of contract manufacturing [48]. Due to the extensive resources offered by a cloud, the entire quotation calculation can be transferred to the cloud. At the same time, the accessibility of the cloud via the Internet allows easy access for exchange at any time. In the use case, a group of several chip developers is considered, who do not have their own manufacturing facilities and use a contract manufacturer that takes over the manufacturing of microchips for these developers. Since all products are semiconductors, the manufacturing requirements are similar for all chip developers. Due to the similar requirements for the cloud, which is used for the calculation of the costing model, among other things, a community cloud represents an efficient and cost-effective solution. Figure 8 shows the architecture and the data flow between the involved parties.

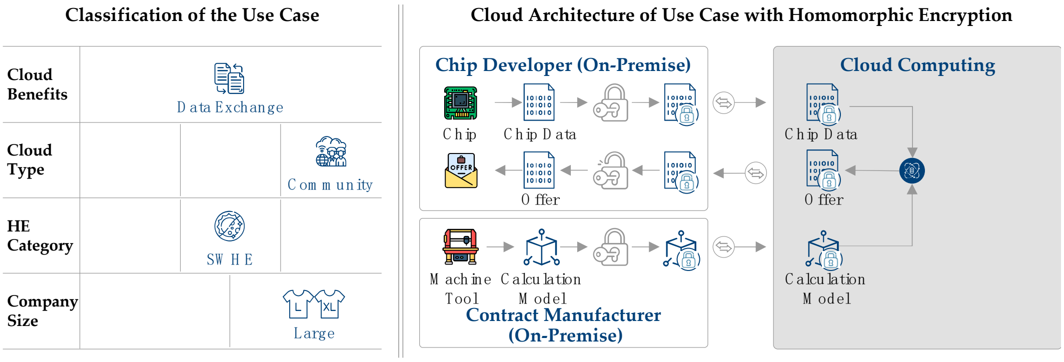

Figure 8. Classification and cloud architecture for contract manufacturing of chipsets.

In summary, the main task of the cloud is to provide a platform for data exchange between chip developers and the contract manufacturer. Due to the composition of the companies using the cloud platform, a community cloud is used. Encryption takes place using SWHE, and the use case is aimed at large enterprises. The selection of characteristic values is shown in the morphological box of the chip manufacturer use case Figure 8.

Again, the three requirements security, time, and cost are fulfilled. First, as described above, data security is of high importance, as data might otherwise be visible for competitors. Second, offers must not be created in real time. Thus, the additional calculation time is not issue. Third, as the whole offer calculation is automized and a mid-prize cloud in form of a community cloud is used, costs are saved.

As the architecture in Figure 8 shows, the process starts with the creation and provision of the costing model for quoting by the contract manufacturer. The model is defined on the basis of the manufacturing parameters for a given chip type and considers the contract manufacturer's costs and intended margins. The model is homomorphically encoded and transmitted to the cloud platform. If the chip designer has developed a new microchip and wants to request a quote from the contract manufacturer, it also homomorphically encrypts the design data and transmits it to the cloud.

In the cloud, the data are then evaluated using the model, and a quotation is generated. The offer is initially also encrypted and is only decrypted by the chip developer after the transfer. In this way, both the chip developer's technology and the contract manufacturer's model remain secret before a contract is concluded between the two parties. This process takes place via the community cloud every time one of the companies has developed a new chip design and wants to initiate a new manufacturing order. The contract manufacturer can add new costing models as needed, expanding the ability to quote for additional chip types. An SWHE method is envisioned for the use case since the calculations necessary to determine quotation terms and manufacturing durations are known in advance. SWHE procedures only allow a limited number of different operations. The operations necessary for the calculation of the offer can be fixed in advance in such a way that an SWHE model can be used. Compared to FHE, this saves time and computing effort.

Homomorphic encryption allows the bidding calculation to be performed without any prior legal assurance, as sensitive technology is not present in unencrypted form in the cloud. After the necessary data are entered by the contract manufacturer, the model calculates the offer including costs and delivery time, which represent the only information presented to the chip developer. For the chip developer, there is no risk of competition-critical chip design information being made public during the bidding calculation. Because of this, the additional work involved in bidding can be significantly reduced, and the duration of the bidding process can be reduced. In addition, the process also keeps the exact costing bases of the contract manufacturer secret since the model is also encrypted.

The chip developer use case shows that a group of companies with similar needs can achieve efficiency benefits by using a community cloud. For example, a template is first generated by the contract manufacturer that provides information about all relevant data that must be available for quotation costing. The chip developers then must compile their data on the basis of the template, encode them homomorphically, and transmit them to the cloud. In the cloud, the same calculation process now takes place automatically for every request. In the process, competition-critical information remains secret from the other chip developers and the contract manufacturer, while ensuring fair bidding within the same constraints for all chip developers. The use of homomorphic encryption forms the basis for secure bidding and ensures that no competition-critical data can be made public. Without homomorphic encryption, it would not be possible to calculate a bid without lengthy legal safeguards in advance.

4.4. *Summary of Use Case Analysis*

In summary, the use of homomorphic encryption in combination with cloud computing is suitable for different use cases from the manufacturing environment. However, the use case analysis showed that homomorphic encryption would not improve every possible use case. When using the morphological box to develop the use cases, it is noticeable that not all possible combinations of characteristic values were used. First, regarding cloud benefits, the cloud is not used for data storage in the use cases since pure data storage makes the use of in-use encryption methods obsolete. The type of cloud used is limited to the public and the community cloud in the use cases. Second, regarding cloud type, the use of a private cloud is not applicable, as a private cloud is usually used by a single company, which configures it and can specify the security of the infrastructure in advance. Access to the cloud can be restricted as required, and the cloud can also be hosted locally if there are special security requirements.

There is no data exchange or shared data use with other companies, as is possible with the other cloud types. For this reason, the use of homomorphic encryption is not advantageous compared to classic encryption, since the in-use data utilization is not exploited. Third, regarding homomorphic encryption category, either FHE or SWHE methods are used, but not PHE methods. In principle, the use of a PHE method is possible; however, the application possibilities are limited, since only additions or only multiplications are possible. In the use cases considered, models of varying complexity are applied, but they all use both additions and multiplications. In addition, cloud computing is typically used to move the computation of complex models to a cloud. FHE and SWHE methods are better suited for such models.

5. Conclusions and Outlook

This paper evaluated the potential of homomorphic encryption for cloud computing use cases in manufacturing. After presenting the fundamentals in Section 2, Section 3 presented encryption for cloud computing, analyzing the limitations of classic encryption, as well as the potential and challenges of homomorphic encryption. This showed that homomorphic encryption has a big advantage regarding security over the whole data lifecycle, but has the consequence of higher computing times, thus leading to higher computing energy and costs. However, the section also showed that the potential and challenges of homomorphic encryption highly depend on the use case. Therefore, Section 4 identified use cases for homomorphic encryption for cloud computing in manufacturing and analyzed how homomorphic encryption improves this use case. This showed that, in general, several beneficial manufacturing use cases exist. However, mainly FHE and SWHE methods are applicable in manufacturing. Furthermore, it is probably only beneficial if using public or community clouds, as private clouds generally fulfill security requirements per se; thus, using homomorphic encryption does not have any further benefits.

Sections 3.3 and 4 represented the main contribution and novelty of this paper, as this paper is the first to simulate homomorphic encryption applicability for manufacturing and present in detail manufacturing use cases on the level of data transfer.

Despite the shown potential of homomorphic encryption for manufacturing, there is further research necessary to increase the use of homomorphic encryption in manufacturing. For increased use in manufacturing, the development of efficient manufacturing specific FHE methods is necessary, which offer correspondingly extensive functionality, but entail less computing time and reduced costs in the application. For this purpose, the specific requirements of manufacturing applications could first be determined, which can be sensibly outsourced to a cloud. This includes, e.g., the definition of error threshold of the bootstrapping for manufacturing, as well as necessary operations other than addition and multiplication. Furthermore, a tradeoff between the use of cloud computing and data security should take place for the consideration of the topic. A mathematical modeling could be envisaged, which weighs the additional effort caused by homomorphic encryption against the advantages of cloud computing. In this way, the decision on the use of homomorphic encryption could be systematized and, thus, facilitated. Interesting developments also exist around business models and current implementation of homomorphic encryption. For example, IBM offers so-called homomorphic encryption services. This is a service offering that includes the option of using a fully homomorphic encryption process [49]. Research in this area could be further expanded in the future and continued by other companies. On the other hand, many public libraries already exist, such as SEAL or HElib, which can be used to implement homomorphic encryption [50]. At this point, another property of homomorphic encryption should be mentioned that might become relevant in the context of cryptographic security in the future. FHE theoretically provides complete security with respect to the threats posed by the advancement and improvement of quantum computers. The development of viable homomorphic encryption systems may, therefore, offer important advantages for the future of cryptography [29].

Author Contributions: Conceptualization, R.K., M.L. and K.L.; methodology, R.K. and M.L.; software, A.M.; validation, A.M., R.K. and M.L.; writing—original draft preparation, R.K. and M.L.; writing—review and editing, K.L., F.S. and R.H.S.; visualization, R.K.; supervision, R.H.S. All authors read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The simulation data can be downloaded at https://git.rwth-aachen.de/m1ndbr34ker/he_experiments (accessed on 17 December 2022).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Herzwurm, G.; Henzel, R. Cloud-Computing–gekommen um zu bleiben. In *Handbuch Digitale Wirtschaft*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 877–909.
2. Hentschel, R.; Leyh, C. Cloud Computing: Status quo, aktuelle Entwicklungen und Herausforderungen. In *Cloud Computing*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 3–20.
3. Langmann, R.; Stiller, M. Industrial Cloud—Status und Ausblick. In *Industrie 4.0*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 29–47.
4. Repschläger, J.; Pannicke, D.; Zarnekow, R. Cloud Computing: Definitionen, Geschäftsmodelle und Entwicklungspotenziale. *HMD Praxis Wirtsch.* **2010**, *47*, 6–15. [CrossRef]
5. Armbrust, M.; Fox, A.; Griffith, R.; Joseph, A.D.; Katz, R.; Konwinski, A.; Lee, G.; Patterson, D.; Rabkin, A.; Stoica, I. A View of Cloud Computing. *Commun. ACM* **2010**, *53*, 50–58. [CrossRef]
6. Möhring, M.; Keller, B.; Schmidt, R. Cloud-Computing. In *CRM in der Public Cloud*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 11–19.
7. Fallenbeck, N.; Eckert, C. IT-Sicherheit und Cloud Computing. In *Handbuch Industrie 4.0 Bd. 4*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 137–171.
8. Rao, B.T. A study on data storage security issues in cloud computing. *Procedia Comput. Sci.* **2016**, *92*, 128–135.
9. Market Research Future. Cloud Manufacturing Market (ID: MRFR/ICT/4546-CR); New York, NY, USA. 2021. Available online: https://www.marketresearchfuture.com/reports/cloud-manufacturing-market-6004 (accessed on 17 December 2022).
10. Gartner. Gartner Forecasts Worldwide Public Cloud Revenue to Grow 17% in 2020. 2019. Available online: https://www.gartner.com/en/newsroom/press-releases/2019-11-13-gartner-forecasts-worldwide-public-cloud-revenue-to-grow-17-percent-in-2020 (accessed on 17 December 2022).
11. Cloud Security Alliance. Top Threats to Cloud Computing—Pandemic Eleven. 2022. Available online: https://cloudsecurityalliance.org/artifacts/top-threats-to-cloud-computing-pandemic-eleven/ (accessed on 17 December 2022).
12. Cloud Security Alliance. Top Threats to Cloud Computing—The Egregious 11. 2020. Available online: https://cloudsecurityalliance.org/artifacts/top-threats-to-cloud-computing-egregious-eleven (accessed on 17 December 2022).
13. Marinescu, D.C. *Cloud Computing: Theory and Practice*; Morgan Kaufmann: Cambridge, MA, USA, 2022.
14. Archer, D.; Chen, L.; Cheon, J.H.; Gilad-Bachrach, R.; Hallman, R.A.; Huang, Z.; Jiang, X.; Kumaresan, R.; Malin, B.A.; Sofia, H. Applications of homomorphic encryption. In *Crypto Standardization Workshop, Microsoft Research*; Microsoft: Redmond, WA, USA, 2017.
15. Iezzi, M. Practical privacy-preserving data science with homomorphic encryption: An overview. In Proceedings of the 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 10–13 December 2020; pp. 3979–3988.
16. Curran, K.; Carlin, S.; Adams, M. Cloud computing security. *J. Netw. Eng.* **2011**, *37*, 4069–4072.
17. Das, D. Secure Cloud Computing Algorithm Using Homomorphic Encryption and Multi-Party Computation. In Proceedings of the 2018 International Conference on Information Networking (ICOIN), Chiang Mai, Thailand, 10–12 January 2018; pp. 391–396.
18. Mohan, M.; Devi, M.K.; Prakash, V.J. Homomorphic Encryption—State of the Art. In Proceedings of the 2017 International Conference on Intelligent Computing and Control (I2C2), Coimbatore, India, 23–24 June 2017; pp. 1–6.
19. Armknecht, F.; Boyd, C.; Carr, C.; Gjøsteen, K.; Jäschke, A.; Reuter, C.A.; Strand, M. A Guide to Fully Homomorphic Encryption. *Cryptol. ePrint Arch.* 2015. Available online: https://eprint.iacr.org/2015/1192.pdf (accessed on 17 December 2022).

20. Mell, P.; Grance, T. *The NIST Definition of Cloud Computing (Special Publication 800-14)*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2011.
21. Münzl, G.; Przywara, B.; Reti, M.; Schäfer, J.; Sondermann, K.; Weber, M.; Wilker, A.; Barot, P.; Becker, B.; Bühr, O. Cloud Computing—Evolution in der Technik, Revolution im Business; Berlin, Germany. 2009. Available online: https://www.bitkom.org/sites/default/files/file/import/090921-BITKOM-Leitfaden-CloudComputing-Web.pdf (accessed on 17 December 2022).
22. Spitz, S.; Pramateftakis, M.; Swoboda, J. *Kryptographie und IT-Sicherheit*; Springer: Berlin/Heidelberg, Germany, 2011.
23. Ertel, W.; Löhmann, E. *Angewandte Kryptographie*; Carl Hanser Verlag GmbH Co KG: Munich, Germany, 2019.
24. Pohlmann, N. Kryptografie. In *Cyber-Sicherheit*; Springer Fachmedien Wiesbaden GmbH: Wiesbaden, Germany, 2022.
25. Schwenk, J. *Sicherheit und Kryptographie im Internet: Theorie und Praxis*; Springer: Berlin/Heidelberg, Germany, 2020; Volume 5.
26. Wendzel, S. *IT-Sicherheit für TCP/IP-und IoT-Netzwerke*; Springer: Berlin/Heidelberg, Germany, 2021; Volume 2.
27. Küsters, R.; Wilke, T. *Moderne Kryptographie*; Springer: Berlin/Heidelberg, Germany, 2011.
28. Ogburn, M.; Turner, C.; Dahal, P. Homomorphic Encryption. *Procedia Comput. Sci.* **2013**, *20*, 502–509. [CrossRef]
29. Schulze, M. *Homomorphe Verschlüsselung und Europas Cloud: Ein Baustein für Europas Digitale Souveränität*; Stiftung Wissenschaft und Politik, Deutsches Institut für Internationale Politik und Sicherheit: Berlin, Germany, 2021.
30. Modler, F.; Kreh, M. *Tutorium Analysis 1 und Lineare Algebra 1*; Springer: Berlin/Heidelberg, Germany, 2011.
31. Gentry, C. Computing arbitrary functions of encrypted data. *Commun. ACM* **2010**, *53*, 97–105. [CrossRef]
32. Acar, A.; Aksu, H.; Uluagac, A.S.; Conti, M. A survey on homomorphic encryption schemes: Theory and implementation. *ACM Comput. Surv. (Csur)* **2018**, *51*, 1–35. [CrossRef]
33. Müller-Quade, J.; Huber, M.; Nilges, T. Daten verschlüsselt speichern und verarbeiten in der Cloud. *Datenschutz Datensicherheit-DuD* **2015**, *39*, 531–535. [CrossRef]
34. Gentry, C. *A Fully Homomorphic Encryption Scheme*; Stanford University: Stanford, CA, USA, 2009.
35. Bouti, A. *Homomorphe Verschlüsselung und Cloud-Computing*; Fernuniversität Hagen: Hagen, Germany, 2020.
36. Alkharji, M.; Liu, H.; Washington, C. Homomorphic Encryption Algorithms and Schemes for Secure Computations in the Cloud. In Proceedings of the 2016 International Conference on Secure Computing and Technology, Virginia International University, Fairfax, VA, USA, 4–5 November 2016; p. 19.
37. Yi, X.; Paulet, R.; Bertino, E. *Homomorphic Encryption and Applications*; Springer: Berlin/Heidelberg, Germany, 2014.
38. Sutisna, N.; Jonatan, G.; Syafalni, I.; Mulyawan, R.; Adiono, T. Polynomial Multiplication Systolic Array for Homomorphic Encryption in Secure Network Communications. In Proceedings of the 2020 IEEE International Conference on Communication, Networks and Satellite (Comnetsat), Batam, Indonesia, 17–18 December 2020; pp. 390–394.
39. Paul, J.; Annamalai, M.S.M.S.; Ming, W.; Badawi, A.A.; Veeravalli, B.; Aung, K.M.M. Privacy-Preserving Collective Learning with Homomorphic Encryption. *IEEE Access* **2021**, *9*, 132084–132096. [CrossRef]
40. Kiesel, R. *Techno-Economic Evaluation of 5G Technology for Latency-Critical Applications in Production*; Apprimus Verlag: Aachen, Germany, 2022; Volume 1.
41. Kaufmann, T. *Strategiewerkzeuge aus der Praxis*; Springer: Berlin/Heidelberg, Germany, 2021.
42. Wöstmann, R.; Strauß, P.; Deuse, J. Predictive Maintenance in der Produktion. *Anwend. Einführungsvoraussetzungen Erschließung Ungenutzter Potentiale Werkstattstech. Online* **2017**, *107*, 524–529.
43. Zhai, S.; Reinhart, G. Predictive Maintenance als Wegbereiter für die instandhaltungsgerechte Produktionssteuerung. *Z. Wirtsch. Fabr.* **2018**, *113*, 298–301. [CrossRef]
44. Lee, J.; Kao, H.-A.; Yang, S. Service innovation and smart analytics for industry 4.0 and big data environment. *Procedia Cirp* **2014**, *16*, 3–8. [CrossRef]
45. Kang, H.E.D.; Kim, D.; Kim, S.; Kim, D.D.; Cheon, J.H.; Anthony, B.W. Homomorphic Encryption as a Secure PHM Outsourcing Solution for Small and Medium Manufacturing Enterprise. *J. Manuf. Syst.* **2021**, *61*, 856–865.
46. Knechtel, J.; Patnaik, S.; Sinanoglu, O. Protect Your Chip Design Intellectual Property: An Overview. In Proceedings of the International Conference on Omni-Layer Intelligent Systems, Crete, Greece, 5–7 May 2019; pp. 211–216.
47. Hung, H.-C.; Chiu, Y.-C.; Wu, M.-C. Analysis of competition between IDM and fabless–foundry business models in the semiconductor industry. *IEEE Trans. Semicond. Manuf.* **2017**, *30*, 254–260. [CrossRef]
48. Ellwein, C.; Riedel, O.; Meyer, O.; Schel, D. Rent'n'produce: A secure cloud manufacturing platform for small and medium enterprises. In Proceedings of the 2018 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC), Stuttgart, Germany, 17–20 June 2018; pp. 1–6.
49. IBM. Innovative Factory Processes with Cloud and AI. Available online: https://www.ibm.com/industries/industrial/resources/business-transformation-interactive/scenes/smart-factory/select/details/production-optimization/ (accessed on 16 October 2022).
50. Sathya, S.S.; Vepakomma, P.; Raskar, R.; Ramachandra, R.; Bhattacharya, S. A review of homomorphic encryption libraries for secure computation. *arXiv* **2018**, arXiv:1812.02428.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.