Computers and Chemical Engineering 176 (2023) 108295

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/cace

Cryptographic protocol for privacy-preserving integration of HAZOPs in modular process plants☆,☆☆

Zarina Chokparova [a,d,*], Kilian Becher [b], Anselm Klose [c,e], Thorsten Strufe [b,f], Leon Urbas [c,d]

[a] Boysen-TU Dresden-Research Training Group, TU Dresden, Chemnitzer Str. 48b, Dresden, 01187, Germany
[b] Chair of Privacy and Data Security, TU Dresden, Nöthnitzer Str. 46, Dresden, 01187, Germany
[c] Chair of Process Control Systems, TU Dresden, Helmholtzstraße 10, Dresden, 01069, Germany
[d] Process Systems Engineering Group, TU Dresden, Helmholtzstraße 10, Dresden, 01069, Germany
[e] Process-to-Order Lab, TU Dresden, Helmholtzstraße 10, Dresden, 01069, Germany
[f] Chair of IT Security, Karlsruhe Institute of Technology, Am Fasanengarten 5, Karlsruhe, 76131, Germany

ARTICLE INFO

Keywords:
Modular HAZOP
Confidentiality
Privacy-preserving computation
Homomorphic encryption

ABSTRACT

Information which is contained in Hazard & Operability (HAZOP) studies is highly sensitive since it can reveal the vulnerabilities of a system and potential ways in which to bypass safeguards. Through the design of systems involving collaboration along a value chain, at some point this information is shared between several parties. In this paper, we propose a methodology for the secure exchange of safety information whilst preserving sensitive information for the application of modular Hazard & Operability (HAZOP) studies. We use homomorphic encryption in a workflow for the sharing of information between plant owners and operators as well as module vendors. We apply encryption to the risks from different modular HAZOPs (mHAZOPs), and combine and compare them without disclosing the risk level. Our contribution is a privacy-preserving protocol for mHAZOP comparison during the integration of modular process and equipment. We provide an exemplary implementation of the protocol and demonstrate the protocol's privacy and correctness.

1. Introduction

In digitalized industries, sharing information with the right partners at the right time is essential for efficient processes. Industrial partners exchange information to determine what is needed for their own individual value creation. The value for the parties is sometimes the information itself, which is why this information is confidential.

To reduce the risk of unwanted information disclosure, the collaboration between the parties is limited to single aspects, such as the specifications of certain equipment. Alternatively, the information can be shared in a privacy-preserving manner that prevents the revealing of sensitive data. With this approach, neither of the industrial players is disclosing confidential information, but the effort needed to collaborate could be increased.

Especially for modular process plants, the exchange of information is essential. The approach for the design of a modular plant is to choose suitable modules, so-called process equipment assemblies (PEAs). PEAs are intended to fulfill desired process steps (Schindel et al., 2021). Here, in addition to the technical specifications of the equipment, the safety requirements of the process should be satisfied, e.g., risks of the process must be matched to the safeguards of the equipment (Klose et al., 2019; Pelzer et al., 2021a). This can be done by comparing the different modular Hazard & Operability (HAZOP) information of the process and equipment (Klose et al., 2019, 2021). This information is essential for the suitability of process and equipment. It is also confidential since it contains the operational risks and, therefore, the vulnerabilities of a process or plant. HAZOP representations comprise sensitive information, such as causalities between process parameters (Klose et al., 2021). Leaking this kind of information can reveal intellectual property in the form of control sequences and system operation and represents a threat to the company's competitive advantage.

This paper addresses an issue of HAZOP information confidentiality during the integration of modular equipment within a manufacturing process, as well as proposes a methodology for privacy-preserving computation of compatibility between corresponding HAZOP systems. As special case for modular equipment, the safety analysis in form of an HAZOP can be done separately by Module Vendors

☆ The authors would like to thank the Boysen-TU Dresden Research Training Group for the financial support that has made this publication possible.
☆☆ The authors thank the Process-to-Order-Lab for discussing the concepts as applied to existing demonstrators.
* Corresponding author at: Boysen-TU Dresden-Research Training Group, TU Dresden, Chemnitzer Str. 48b, Dresden, 01187, Germany.
E-mail addresses: zarina.chokparova@tu-dresden.de (Z. Chokparova), kilian.becher@tu-dresden.de (K. Becher), anselm.klose@tu-dresden.de (A. Klose), thorsten.strufe@kit.edu (T. Strufe), leon.urbas@tu-dresden.de (L. Urbas).

https://doi.org/10.1016/j.compchemeng.2023.108295
Received 22 December 2022; Received in revised form 6 February 2023; Accepted 15 May 2023
Available online 19 May 2023

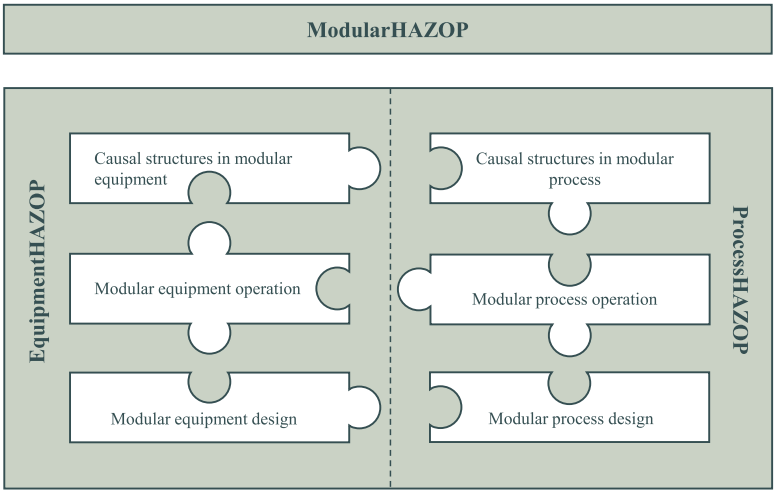


Fig. 1. Integration between EquipmentHAZOP and ProcessHAZOP.

and Owner/Operators, which makes the exchange of these information relevant in the first place.

1.1. HAZOP and mHAZOP

The HAZOP analysis is a common approach for safety assessment in chemical process industries (Kletz, 2018). This analysis is conducted by a group of experts in design, operation, and maintenance of a chemical plant (Vaidhyanathan and Venkatasubramanian, 1996) or by using computer-aided design tools (Cui et al., 2010). The experts systematically review piping and instrumentation diagrams (P&IDs) of the process to diagnose any deviations from design intent or normal operation, and identify their causes and subsequent hazards.

An mHAZOP study is the application of the HAZOP methodology for modular plants. Its difference from the original methodology is that separate analyses for equipment and process are conducted, which are combined for a complete evaluation of the modular system (Klose et al., 2019). With that approach, the flexibility of modular plants can be followed. The EquipmentHAZOP addresses risk and hazard information within the modular plant or its components, whereas the ProcessHAZOP refers to the HAZOP structures within the process that is to be integrated into the modular plant. Integration of modular EquipmentHAZOP and ProcessHAZOP (Fig. 1) requires the internal causal relations and process flows of operation and design intents to fit. Information contained in mHAZOP parts can be owned by different entities (VDI 2776-3, 2022). Within this work, the party which provides the EquipmentHAZOP will be referred to as the Module Vendor (MV) and the one providing the ProcessHAZOP is referred to as the Owner/Operator (O/O). To operate a chemical process, the O/O needs equipment (modular plant, PEA, or component) which can be purchased or outsourced from the MV. Due to confidentiality concerns, neither party is willing to disclose their mHAZOP information to each other before an official agreement is made, e.g., a sales contract. Such an agreement can only be made if the suitability of the equipment for the process can be shown.

We resolve this apparent contradiction through a secure-computation protocol based on homomorphic encryption that enables the privacy-preserving evaluation of equipment suitability. Homomorphic encryption (HE), first proposed by Rivest et al. (1978b), is a special form of encryption that enables computations on encrypted data without intermediate decryption. Given that, our protocol ensures that no party learns another party's confidential data.

1.2. Problem statement and goals

This study addresses the confidentiality requirement of safety-related operation information and presents a methodology for a comparison of risk levels during the adaptation of process equipment provided by a Module Vendor within the process of an Owner/Operator. Specifically, the equipment risks have to be aligned with or be higher than process risks in order to tolerate them during operation. Since the equipment and process can be designed by separate entities, both can have their secrets associated with the operation and control of the *HAZOPElements*. Therefore, the parties need to share this information in a privacy-preserving manner.

As a result, in this paper, a protocol for the confidential comparison of information pieces of mHAZOP *HAZOPElements* of Equipment and Process is proposed. The confidentiality protocol uses a partially homomorphic encryption scheme with additive homomorphism for comparison of process parameters and risk values contained in mHAZOPs to subsequently determine whether the safety integrity level (SIL) values of the systems are suitable for integration and operation.

This work investigates integration between the safety models in modular process plant and aims to make the following contributions:

- Investigation of the significance of sensitive information contained in mHAZOPs that can be revealed during the integration of process and process equipment;
- Proposal of a procedure for checking the safety-based compatibility between process and equipment mHAZOPs in a modular production concept;
- Protocol with application of homomorphic cryptosystem to enable the privacy-preserving comparison of process and equipment mHAZOP risks;
- Demonstration of the result of privacy-preserving comparison with a simulation and a numerical example.

The paper is structured as follows: Section 2 provides background information on modular safety and mHAZOP model structures, as well as an outline of confidentiality and privacy-preserving comparison techniques; Section 3 describes the procedure for the combination of *Equipment* and *Process HAZOPs*, lists confidentiality requirements, and describes employed cryptographic methods; Section 4 presents the protocol with the use of a homomorphic cryptosystem and relevant assumptions; Section 5 illustrates a comprehensive example of the compared information and outcomes from a simulation; and Section 6 concludes the paper with a summary and emphasizes the value of the presented work for process industries.

2. Preliminaries

Modular process plants (Roy, 2017) is a concept of flexible and adaptable production systems that can be set up at various locations. The modularization (ProcessNet, 2016) allows for the distributed operation of processes and manufacturing of products through the combined efforts of multiple industrial parties. The operation of these plants requires safety considerations and confidentiality preservation of parties' trade secrets. The following sections give an overview of these trade secrets, the resulting confidentiality requirements, and approaches to preserving confidentiality during the processing of such data.

2.1. Modules and safety

In modular chemical process systems and small-scale production, safety concerns prevail due to the risk and hazards of process operation conditions and exposure to harmful chemicals (Kockmann et al., 2017). In order to prevent accidents in process plants and improve process performance, the HAZOP approach was developed (Kletz, 2001). Process safety considers important topics and parameters, such as hazard characterization, pressure and temperature control, thermal stability, reaction control, variations of concentration, flow rate control, storage and transportation, waste disposal, etc.

Numerous process and chemical industrial players have adopted the concept of modular production (Baldea et al., 2017). The continuous process is divided into standardized modules that comprise operational units and contain engineering information about its design and life cycle. The modules can be flexibly arranged and combined into a fully operational process plant. Similar to conventional plants, modular plants are required to comply with safety regulations, including the analysis of hazards and risks and strategies to prevent accidents (Pelzer et al., 2021a). This is performed through the implementation of risk assessment techniques such as HAZOP. Since each module contains a standardized risk assessment analysis integrated into the system, the overall safety evaluation of the plant can be shortened (Kockmann et al., 2017; Klose et al., 2019).

Modular safety includes intra- and intermodular risk evaluation (VDI 2776-3, 2022). While intramodular safety involves process equipment assembly (PEA) risks, intermodular safety investigates minimizing risks which result from a combination of intramodular safety functions (Pelzer et al., 2021b). HAZOP information is highly relevant for use in modular process plants and can be integrated into the modular plant hierarchy consisting of PEAs, which contain functional equipment assemblies (FEAs) (Klose et al., 2019).

2.2. Modular HAZOP structure

For the application of mHAZOP as an approach to solving flexibility for the safety analysis, Klose et al. (2019) proposed an information model which structures HAZOP studies in *HAZOPNodes* for the analyzed sub-systems or process steps, *HAZOPCases* for the individual scenarios, and *HAZOPElements* for causes, deviations and consequences as well as safeguards. The resulting information model is shown in Fig. 2 and a resulting exemplary table is shown in Table 1. In the table, all needed information is documented whereas one *HAZOPCase* presents one row. The resulting risk is derived e.g., with the risk-graph as later shown in Fig. 3.

The approach proposed by Klose et al. (2019) states that causes, consequences, and deviations can all be described in the same way as *HAZOPElements* (hE). As a consequence, this opens the possibility for *HAZOPElements* to take different roles (causes, deviations or consequences) for different *HAZOPCases* (hC) (Klose et al., 2019, 2021). With that, a network of *HAZOPElements* is created which describe safety relevant situations. The overall risk of the identified scenarios must not exceed the acceptable risks, which is why safeguards are implemented in the equipment. The safeguards provide a specific safety capability of the equipment which can be used to safely contain the process.

During the comparison of the safety requirements from the process with the safety capabilities of the equipment, the O/O needs to check if the identified process risks can be mitigated by the safeguards of the module. For that, the mHAZOPs of both parties need to be combined by finding identical *HAZOPElements* in the shared *HAZOPNode* and the risk levels of the combined Elements need to be compared. Two *HAZOPElements* are identical if the guideword and parameter of the *HAZOPElements* are identical and they belong to the same *HAZOPNode* (Klose et al., 2021). Assuming that the built equipment can be operated in a safe way, suitability is guaranteed if the risk capability of the equipment is higher than the risk requirement from the process.

2.3. Integration of mHAZOPs

Structuring HAZOP information within the parts of a modular plant and providing a methodology for adaptation and information exchange between partners is an essential step in the integration of equipment and process. This is necessary to ensure that possible risks and their effects on the system are identified, the relevant safeguards are specified, and the impacts of changes in process operation on the overall modular plant performance are considered.

Before modules are integrated with each other, the design information and potential risk areas need to be assessed. Modules have flexibility in the range of substances that can be processed as well as in the corresponding operational conditions for chemical reaction or separation (Kockmann et al., 2017; Fleischer et al., 2015). For example, one of the characteristics of the module can be a temperature class, which describes the limitations of the equipment regarding possible substances that can be operated without a threat of ignition or structural failure. As a result, safety considerations of the modular component and the intended operation process need to be in compliance. Considering the intermodular safety, the HAZOP study includes safety related interactions of combined modules and external impact factors on the whole modular process chain.

mHAZOP information contains the causal relations within the processes and equipment (Klose et al., 2021). This represents how some parameters are related to and affected by other parameters, what are the probabilities of hazardous events, as well as their severity. This information facilitates the identification of the posed risks and derivation of appropriate safety measures. For example, increasing the flow rate of the reactants in a plug flow reactor decreases the residence time. This directly leads to a drop in the conversion value of the reactor. Therefore, the goal would be to apply measures for flow rate control in order to achieve a higher conversion of reactants. As a counteraction, the residence time can be prolonged by changing the length of a tubular reactor and, therefore, allowing for higher conversion rate, or by increasing the temperature in the reactor. This case demonstrates the causal chain between dependent events and parameters. Disclosure of mHAZOP knowledge regarding the fundamental causal relationships can be a threat to the integrity of process operation. Sharing the potential risks of a process increases the potential of malicious attacks similar to Stuxnet (Baezner and Robin, 2017).

During the HAZOP study, Process Flow Diagrams and P&IDs are systematically reviewed by multidisciplinary teams (Dunjó et al., 2010). The complex systems are partitioned into sections, or nodes, where safety and risks can be analyzed. Traditionally, target function parameters are defined in a process node. The causes of deviations in a process are represented by a combination of guideword and parameter. Thus, the causal relations in HAZOPs and process risks are sensitive information, as they describe and identify the process and operation.

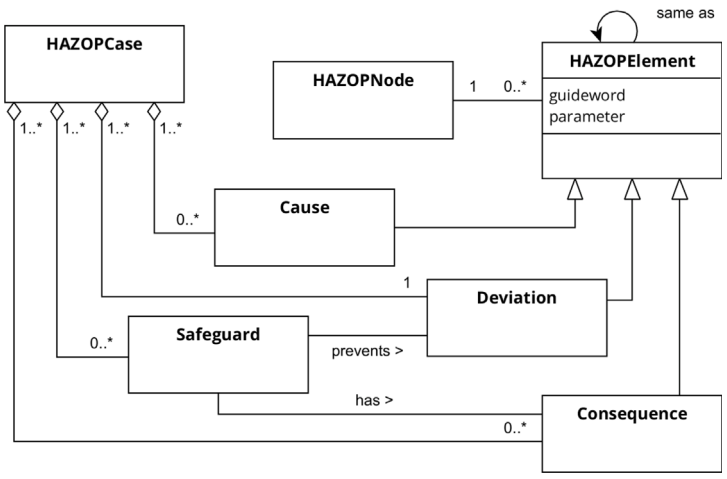
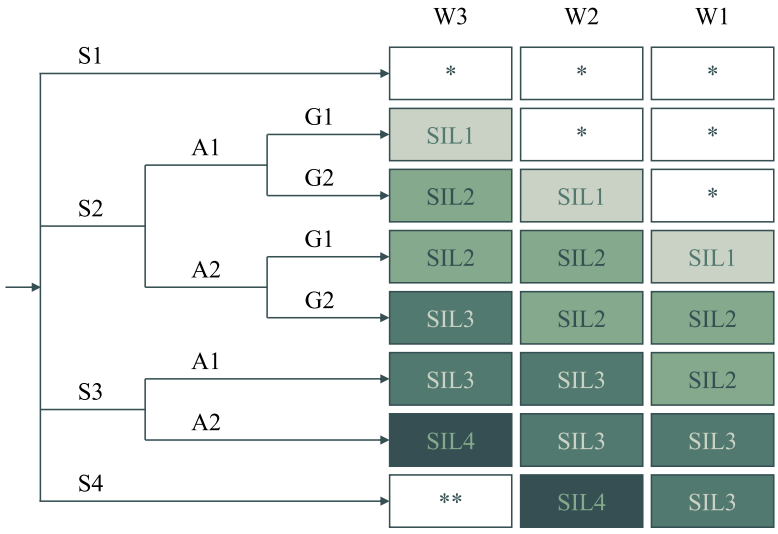


Fig. 2. mHAZOP Information model with *HAZOPNode*, *HAZOPCases*, and *HAZOPElements* from Klose et al. (2019).



* safety instrumented function is not needed or basic process control system protection layer
** safety instrumented function is deficient

Fig. 3. SIL classification for the risk according to the parameters of VDI/VDE 2180 (2019).

Table 1
Exemplary equipment-mHAZOP for a reactor.

HAZOP	Deviation				Cause				Consequence			
case	*HAZOP Node*	Guideword	Parameter	S, A, G, W	*HAZOP Node*	Guideword	Parameter	S, A, G, W	*HAZOP Node*	Guideword	Parameter	S, A, G, W
1	F03-Vessel	More	Pressure	S3, A1, -, W2	F03-Inlet	More	Flow	S2, A2, G1, W3	F03-Outlet	More	Pressure	S3, A1, G1, W1
2	F03-Vessel	More	Level	S2, A2, G2, W3	F03-Inlet	More	Flow	S2, A1, G2, W2	F03-Outlet	More	Flow	S2, A2, G2, W1
3	F03-Vessel	Less	Pressure	S2, A2, G1, W1	F03-Outlet	More	Flow	S2, A2, G1, W3	F03-Inlet	More	Flow	S1, A2, G1, W1
4	F03-Vessel	More	Temperature	S3, A1, -, W1	F03-Inlet	More	Temperature	S2, A1, G2, W2	F03-Outlet	More	Temperature	S2, A2, G1, W1

2.4. Confidentiality

In the context of modern process industries, manufacturing businesses own and produce confidential information including intellectual property and trade secrets (Corallo et al., 2020). Confidentiality concerns and protection of knowledge assets of organizations, such as operational data, resources, experience, product knowledge, strategies, etc. are important research topics (Ahmad et al., 2014). The exposure of such sensitive information could weaken a company's competitive advantage and harm their position on the market. Moreover, it could impose financial penalties due to violating commercial agreements.

As discussed in D'Amours et al. (1999), firms collaborate with each other over virtual manufacturing networks, which allows them to control and configure the scheduling and operations within the network. Similarly, by outsourcing production and resources, industries need to supply information (Bardhan et al., 2006). Information sharing is an essential criterion of an effective networked manufacturing and improves collaboration between parties (Kenyon et al., 2016). Another study (Gaonkar and Viswanadham, 2001) has shown the cost and logistical benefits of information sharing by conducting numerical experiments. Due to multistage manufacturing processes and their complexity, equipment producers outsource their assets to other

manufacturing specialists. However, information shared during such interactions can contain sensitive data and values related to their processes that parties must not reveal to each other.

As parts of modular plant operation, the sub-systems, such as various industrial equipment or smart devices, are connected with one another over the communication chain. Through the communication of parts and systems originating from distinct providers, information is exchanged. Thus, parties could utilize received information to derive additional knowledge about the processes or operation. For example, a simple alteration of increasing or decreasing parameter values, including pressure or temperature, could reveal additional information about the process or ingredients. Time series data generated by process units or specific control and operational commands and their sequences could also disclose information about processes. With knowledge about the process, it would be feasible to derive possibilities of how to manipulate the process. This could be used for malicious purposes, e.g., by increasing the temperature of a reaction to create a runaway reaction. Such actions could affect product quality or, in the worst case, even the safety of the plant. The HAZOP usually contains information about critical process steps, making them an easy entry point for system manipulation.

To ensure confidentiality of sensitive data, secure-computation protocols can be employed. Such protocols utilize advanced cryptographic mechanisms and cryptographic primitives for the computation of joint functions in a privacy-preserving form (Micali and Rogaway, 1992). One useful tool for privacy-preserving computations are homomorphic cryptosystems. A review of homomorphic encryption and its applications is provided by Alloghani et al. (2019). Example applications include genome analysis for medical research (Kantarcioglu et al., 2008), cross-company benchmarks (Becher et al., 2019), and supply-chain verification (Becher et al., 2020). A protocol that utilizes homomorphic encryption for privacy-preserving transfer of process operation information between the value provider and the secret owner is presented in Chokparova and Urbas (2021-09). As a result, the use of a secure-computation protocol with homomorphic encryption aims to preserve competitive advantages of the industrial players.

Therefore, in order to deal with the confidentiality issue of HAZOP information throughout the integration of process and equipment, we propose a protocol utilizing a homomorphic cryptosystem.

2.5. Privacy-preserving comparisons

Through secure computation, parties are able to compute a common output function on confidential inputs without exposing them. Privacy-preserving comparison is a common case in secure computation and is often illustrated by means of the Millionaires' problem (or GT, or "greater than"), where two millionaires want to determine who is wealthier without letting each other know how much they own. After the formulation of the problem by Yao (1982), multiple solutions to tackle secure comparison and perform equality tests have been developed. There are numerous application areas of secure comparisons for sorting data (Baldimtsi and Ohrimenko, 2015), solving optimization tasks (Atallah and Li, 2005), and other operations in different areas, including signal processing (Rane and Boufounos, 2013), statistical analysis (Bogdanov et al., 2018), auctions (Damgård et al., 2007), etc.

Commonly, secure comparison protocols are constructed by implementing garbled circuits (Yao, 1982), oblivious transfer (Rabin, 1981), homomorphic encryption (Rivest et al., 1978a), other privacy-preserving techniques, and their combinations. In order to solve the problem of secure comparison, the protocols approach it in a general way by evaluating the "greater than" circuit, or in a problem specific way by evaluating special properties of the case (Lin and Tzeng, 2005).

In the case of garbled circuits, the procedure for computation is comprised of garbling the circuit that represents the desired function by one party and evaluation of the circuit gate outputs by another party (Beaver et al., 1990; Yao, 1982). The first party G_1 constructs a Boolean circuit and assigns two random strings, called labels, to each wire of the circuit, one label for Boolean 0 and one for 1. For each gate, G_1 then replaces the entries in the corresponding truth table with the respective label, encrypts the output labels with the two corresponding input labels as keys, and randomly permutes the rows of the truth table. G_1 provides these *garbled* truth tables and the labels that represent the confidential input of G_1 to the second party, G_2. Then, G_2 employs oblivious transfer (Cramer, 1999; Rabin, 1981) to receive the labels that represent G_2's confidential input and evaluates the circuit gate by gate without learning G_1's input. The main disadvantage of this secure comparison approach is its large computational and communication complexity, which eventually leads to long running times and poor scalability (Ioannidis and Grama, 2003).

In this study, a secure comparison protocol is applied to compare the risk values of *HAZOPElements* to ensure the process's and equipment's safety parameters are suitable for each other and the integration in a modular process operation is viable.

3. Concept and methodology

Traditional *HAZOPElements* contain information about the location or a part that is studied during process operation. For example, this can represent a vessel, reactor, feed, pump, separator, etc. Therefore, a *HAZOPElement* refers to a node of a plant. The next consideration in a HAZOP workflow is a parameter that deviates during the operation. This can be pressure, temperature, flow rate, level, composition, pH, viscosity, etc. The next information piece is the guideword, which describes a deviation from a parameter. Guidewords include "no"/"not", "more"/"less", "part of", "as well as", etc.

According to the (VDI/VDE 2180, 2019) standard, the risk quantification is performed by considering four affecting variables, namely S (severity of consequence of the hazardous event), A (frequency of presence in the hazardous zone multiplied by the exposure time), G (possibility of avoiding the consequences of the hazardous event), and W (probability of the hazardous event). The corresponding safety integrity level (SIL) (VDI/VDE 2180, 2019) can be determined based on the combination of mentioned risk variables and are structured in Fig. 3. More generally, SIL defines the extent of necessary risk reduction using the control tools for process safety.

To integrate the process and the equipment from an mHAZOP compatibility perspective, the SIL values of the *HAZOPElements* need to be compared. It is required that, for every process *HAZOPElement*, there is a corresponding element on the equipment/module side. If the SILs of the equipment are equivalent to the SILs of the process or greater than them, then the integration is feasible. If this is not the case, then integration is not possible.

$$\forall hE_P \exists! hE_E | [g.hE_P = g.hE_E] \wedge [p.hE_P = p.hE_E] \quad (1)$$

In Eq. (1), hE represents the *HAZOPElements* for the equipment ($_E$) and the process ($_P$). The *HAZOPElements* are a match if the guideword ($g.hE$) and the parameter ($p.hE$) of the compared *HAZOPElements* are the same.

3.1. Confidentiality requirements

We require our protocol to ensure confidentiality of the following pieces of information.

- mHAZOP Elements supplied by the Owner/Operator must not be revealed during the comparison protocol and vice versa;
- Both parties must not know which particular guideword and parameter match occurred;
- The SIL difference between process and equipment must not be revealed.

3.2. Homomorphic encryption

An asymmetric cryptosystem is a tuple $CS = (G, E, D)$ that consists of three polynomial-time algorithms. Given a security parameter κ, the probabilistic key-generation algorithm $G(\cdot)$ outputs a pair (pk, sk) of a (public) encryption key pk and a (secret) decryption key sk. The (probabilistic) encryption algorithm $E(\cdot)$ takes as input a plaintext $m \in \mathcal{M}$ and pk and outputs the ciphertext $c = E_{pk}(m) \in C$, where $\mathcal{M}$ and C denote the plaintext and ciphertext space, respectively. Given a ciphertext c and sk, the decryption algorithm $D(\cdot)$ outputs the plaintext $m = D_{sk}(c) = D_{sk}(E_{pk}(m))$.

Homomorphic encryption (HE) schemes allow computations on ciphertexts with predictable effect on the underlying plaintext. Assume ciphertexts $E_{pk}(m_1)$ and $E_{pk}(m_2)$ are encrypted under the same key pk. A cryptosystem CS is homomorphic if it offers an operation $\circ$ on C that maps to a homomorphic operation $\bullet$ on $\mathcal{M}$, such that $E_{pk}(m_1) \circ E_{pk}(m_2)$ yields an encryption of $m_1 \bullet m_2$. This can be formalized as:

$$D_{sk}(E_{pk}(m_1) \circ E_{pk}(m_2)) = m_1 \bullet m_2.$$

Common homomorphic operations are addition and multiplication.

$$D_{sk}(E_{pk}(m_1) \oplus E_{pk}(m_2)) = m_1 + m_2 \tag{2}$$

$$D_{sk}(E_{pk}(m_1) \odot E_{pk}(m_2)) = m_1 \cdot m_2 \tag{3}$$

Partially homomorphic encryption (PHE) schemes, such as Paillier's (Paillier, 1999) and RSA (Rivest et al., 1978b), enable either addition or multiplication of the underlying plaintexts. Fully homomorphic encryption (FHE) schemes provide both addition and multiplication and allow the privacy-preserving evaluation of arbitrary arithmetic functions (Gentry, 2009). A review of existing homomorphic encryption schemes was provided by Acar et al. (2018).

3.3. Paillier's cryptosystem

Paillier's encryption scheme is a probabilistic asymmetric encryption scheme that was proposed by Paillier (1999). Its security relies on the decisional composite residuosity assumption, which states that for a composite number a and an integer b, it is hard to decide if there exists an integer c such that $b \equiv c^a \pmod{a^2}$. We follow the notation of Acar et al. (2018) and formalize key generation, encryption, and decryption with Paillier's cryptosystem as follows.

Key generation Choose two large primes p, q such that $gcd(pq, (p-1)(q-1)) = 1$ and set $n = pq$ as well as $\lambda = lcm(p-1, q-1)$. Choose $g \in \mathbb{Z}^*_{n^2}$ uniformly at random by checking whether $gcd(n, L(g^{\lambda} \bmod n^2)) = 1$ for a function L such that $L(u) = (u-1)/n$ for every $u \in \mathbb{Z}^*_{n^2}$. The key pair is defined as $(pk = (n, g), sk = (p, q))$. In practice, n should be at least 2048 bits long to ensure a sufficient level of security.

Encryption For encrypting a plaintext message $m \in \mathbb{Z}_n$, first select $r \in \mathbb{Z}^*_n$ uniformly at random. Then compute the ciphertext c as

$$c = E_{pk}(m) = g^m \cdot r^n \bmod n^2. \tag{4}$$

The plaintext and ciphertext spaces are $\mathcal{M} = \mathbb{Z}_n$ and $C \in \mathbb{Z}^*_{n^2}$, respectively. Therefore, plaintext messages that are real numbers that need to be scaled and rounded prior to encryption.

Decryption Decryption of a ciphertext $c \in \mathbb{Z}^*_{n^2}$ to obtain the plaintext m with the help of the secret key sk works as follows.

$$m = D_{sk}(c) = \frac{L(c^{\lambda} \bmod n^2)}{L(g^{\lambda} \bmod n^2)} \bmod n \tag{5}$$

Homomorphic properties Paillier's cryptosystems has the following additive homomorphic property:

$$\begin{aligned} &D_{sk}\left(E_{pk}(m_1) \oplus E_{pk}(m_2)\right) \\ &= D_{sk}\left(E_{pk}(m_1) \cdot E_{pk}(m_2) \bmod n^2\right) \\ &= m_1 + m_2 \bmod n. \end{aligned} \tag{6}$$

That is, multiplication of ciphertexts maps to addition of the underlying plaintexts. Homomorphic addition can also be performed as a ciphertext–plaintext operations as follows:

$$\begin{aligned} &D_{sk}\left(E_{pk}(m_1) \oplus m_2\right) \\ &= D_{sk}\left(E_{pk}(m_1) \cdot g^{m_2} \bmod n^2\right) \\ &= m_1 + m_2 \bmod n. \end{aligned} \tag{7}$$

Based on homomorphic addition, Paillier's cryptosystem also enables homomorphic multiplication by plaintexts. A ciphertext raised to the power of a plaintext yields an encryption of the product of the two plaintexts:

$$\begin{aligned} &D_{sk}\left(E_{pk}(m_1) \odot m_2\right) \\ &= D_{sk}\left(E_{pk}(m_1)^{m_2} \bmod n^2\right) \\ &= m_1 \cdot m_2 \bmod n. \end{aligned} \tag{8}$$

4. Protocol for integration of mHAZOPs

An effective combination of mHAZOPs involves comparison of risk parameters and a subsequent selection of the higher risk value. However, both process risk and module risk values are regarded as sensitive information. Therefore, the Module Vendor and the Owner/Operator need to compare their corresponding mHAZOPs and select the one with the higher risks without revealing individual parameter values. This essentially reduces to an instance of the Millionaires' problem. There are numbers of approaches with proven security that solve similar problems. Solutions to this two-party secure computation task can be classified based on the choice of the cryptographic primitive used in the secure computation protocol.

One of the most important tools applicable in solving the Millionaire's problem is homomorphic encryption. For example, in Blake and Kolesnikov (2004), Paillier's additively homomorphic cryptosystem (Paillier, 1999) was applied. Despite the fact that numerous protocols for solving similar problems exist, none of them are suitable for the comparison of mHAZOPs. Therefore, we propose a new secure-computation protocol that compares *HAZOPElements* based on their guidewords, parameters, and SIL classifications. It employs an additively homomorphic encryption scheme. We suggest using Paillier's cryptosystem. To the best of our knowledge, this protocol is the first of its kind.

4.1. Adversary model

Our secure-computation protocol enables the privacy-preserving computation of the intersection of safety information held by Module Vendor and by Owner/Operator. We assume both parties, Module Vendor and Owner/Operator, to behave semi-honestly (Lindell, 2017). That is, both parties follow the protocol but try to infer non-trivial knowledge about the other party's confidential inputs.

An assumption of a semi-honest adversary is reasonable in the described interaction between Module Vendors and a plant Owners/Operators. Both aim for a legal agreement, like a sales contract, and therefore have an intrinsic motivation to correctly execute the protocol. Moreover, protocols that are secure in the semi-honest model can be extended to be secure also under stronger notions of security through standard transformations, for example, as those described in Goldreich et al. (1987).

4.2. Prerequisites

We refer to the Module Vendor as P_1 and to the Owner/ Operator as P_2. We further encode both guideword and parameter as single digits in the interval $[0,9]$ and the SIL classification in the interval $[0,4]$. P_1 holds a list H_1 of M equipment *HAZOPElements*, each consisting of guideword, parameter, and SIL classification. Similarly, P_2 holds a list H_2 of N process *HAZOPElements*, each consisting of guideword, parameter, and SIL classification. P_1 holds a Paillier public–private key pair (pk, sk) and provides the public key to P_2 at set-up time prior to protocol execution.

In the protocol, we denote concatenation by "||". For the sake of readability, we use simplified notations of encryption and decryption by discarding the keys, i.e., $E(m)$ and $D(c)$ rather than $E_{pk}(m)$ and $D_{sk}(c)$. We enable the representation of negative plaintexts by allocating the upper half of the plaintext space $\mathcal{M} = \mathbb{Z}_n$ for negative values. In a two's-complement fashion, the plaintext space then covers the range from $-(\frac{n-1}{2})$ to $(\frac{n-1}{2})$.

The security of our protocol partially relies on a technique called blinding, where confidential data is obfuscated through addition or multiplication with random values. The values used for multiplicative blinding are sampled uniformly at random from $\mathbb{Z}_{2^{l_1}}$ whereas values used for additive blinding are sampled uniformly at random from $\mathbb{Z}_{2^{l_2}}$. To prevent overflows of plaintexts, we require $2^{l_2} \ll 2^{l_1} \ll \frac{n}{2}$, e.g., $l_1 = 256$ and $l_2 = 64$ for 2048-bits long n.

4.3. Secure-computation protocol

Our protocol is formally provided in Table 2. It works as follows. The protocol starts in Step 0 with both the Module Vendor (P_1) and the Owner/Operator (P_2) preparing their confidential inputs as follows. For all equipment *HAZOPElements* $hE_{1_i} \in H_1, 1 \le i \le M$, P_1 combines the encoded guideword g_{1_i} and parameter p_{1_i} into a single two-digit number through concatenation and encrypts the resulting two-digit number with the key pk. P_1 further encrypts the encoded SIL classification for each equipment *HAZOPElement* with pk. P_1 obtains two lists of ciphertexts, one containing the encrypted guideword-parameter encodings and one containing the encoded SIL classifications.

Similarly, for all process *HAZOPElements* $hE_{2_j} \in H_2, 1 \le j \le N$, P_2 combines the encoded guideword and parameter into a single two-digit number. P_2 performs all subsequent steps as ciphertext–plaintext operations (see Section 3.3) and, therefore, does not need to encrypt the encoded guideword-parameter values and SIL classifications. P_2 obtains two lists of plaintexts, one containing the guideword-parameter encodings and one containing the encoded SIL classifications.

In Step 1, P_1 sends the two resulting lists of ciphertexts computed in Step 0 to P_2. For all possible combinations of P_1's and P_2's guideword-parameter encodings, P_2 samples two numbers $0 < r_{2_{i,j}} \ll r_{1_{i,j}}$ uniformly at random and homomorphically computes the blinded difference of the guideword-parameter combination as follows. Due to the fact that the $x_{2_j}, r_{1_{i,j}}, r_{2_{i,j}}$ are known to P_2, this computation can be performed as ciphertext–plaintext operations.

$$E(y_{i,j}) = (E(x_{1_i}) \oplus (-x_{2_j})) \odot r_{1_{i,j}} \oplus r_{2_{i,j}} \tag{9}$$

Similarly, for all possible combinations of P_1's and P_2's SIL classifications, P_2 samples two numbers $0 < r_{4_{i,j}} \ll r_{3_{i,j}}$ uniformly at random and homomorphically computes the blinded difference of SIL classifications by subtracting the SIL value of the *HAZOPElements* from process and equipment as follows.

$$E(t_{i,j}) = (E(s_{1_i}) \oplus (-s_{2_j})) \odot r_{3_{i,j}} \oplus r_{4_{i,j}} \tag{10}$$

Blinding ensures that P_1 later does not learn the exact differences but only whether it is positive or negative. To do so, the difference is homomorphically multiplied by a large, positive random number. To prevent factorization, a second smaller positive random number is added to the product. This function maps the difference to a random number while preserving the algebraic sign.

P_2 then shuffles the two resulting lists of encrypted, blinded differences with a random permutation π_1 and sends the shuffled lists to P_1. Shuffling prevents P_1 from learning which elements match and which do not.

In step 2, P_1 decrypts all blinded differences of the guideword-parameter combinations and SIL combinations with the decryption key sk by computing

$$y_{\pi_1(i,j)} = D(E(y_{\pi_1(i,j)})) \tag{11}$$

and

$$t_{\pi_1(i,j)} = D(E(t_{\pi_1(i,j)})). \tag{12}$$

Since P_2 is supposed to learn the same results but does not know the secret key sk, P_1 provides the plaintext differences to P_2. However, to prevent P_2 from learning which elements matched and which did not, P_1 again applies blinding and shuffling to the plaintext differences. That is, for each combination of guideword-parameter encodings, P_1 samples $0 < r_{6_{i,j}} \ll r_{5_{i,j}}$ uniformly at random and computes

$$y'_{i,j} = y_{\pi_1(i,j)} \cdot r_{5_{i,j}} + r_{6_{i,j}}. \tag{13}$$

Similarly, for each SIL combination, P_1 samples two values $0 < r_{8_{i,j}} \ll r_{7_{i,j}}$ uniformly at random and computes

$$t'_{i,j} = t_{\pi_1(i,j)} \cdot r_{7_{i,j}} + r_{8_{i,j}}. \tag{14}$$

P_1 then shuffles the resulting lists of guideword-parameter combinations and SIL combinations with a random permutation π_2 and sends them to P_2.

In Step 3, both participants individually compute the protocol output as follows. They create a set O_1 containing all pairs of guideword-parameter differences and SIL differences that indicate matches, i.e., all pairs with a small random guideword-parameter difference and a positive SIL difference. This can be formalized as follows.

$$O_1 = \{(y''_{i,j}, t''_{i,j}) | 0 < y''_{i,j} < 2^{l_1} \wedge 0 < t''_{i,j}\} \tag{15}$$

Similarly, they create a second set O_2 containing all guideword-parameter differences that indicate matches regardless of the SIL difference, i.e.,

$$O_2 = \{y''_{i,j} | 0 < y''_{i,j} < 2^{l_1}\}. \tag{16}$$

The participants then count the elements in O_1. If the cardinality $|O_1|$ equals the number of process *HAZOPElements*, the protocol returns $\top$, indicating suitability. Otherwise, if $|O_1| < N$ but the overall number of guideword-parameter matches is correct, i.e., $|O_2| = N$, the protocol returns $\perp^*$, indicating that the equipment and the process are not suitable but further adjustment of the equipment could make them suitable. In all other cases, the protocol ultimately returns $\perp$, indicating that equipment and process are not suitable. This concludes our protocol.

4.4. Protocol outcome

Both parties learn the number of matches and the number of elements of the other party, which is trivial knowledge. Furthermore, both learn the number of elements for which the SIL comparison indicates non-suitability, which is the desired output of the protocol. Neither party learns the guideword, parameter, or SIL values.

5. Illustrative example

To show the individual steps of the protocol, an illustrative example of the application is shown in Table 3.

In this example, MV has three *HAZOPElements* and O/O has two *HAZOPElements*, which are to be compared. In the preparation, the encoding and combination of guideword and parameter is done and

Table 2
Algorithm of the protocol for checking the suitability of process and equipment mHAZOPs.

Step	Module Vendor		Owner/Operator
0.	$\forall hE_{1_i} \in H_1, 1 \le i \le M$:		$\forall hE_{2_j} \in H_2,\ 1 \le j \le N$:
	$x_{1_i} = g_{1_i} \| p_{1_i}$		$x_{2_j} = g_{2_j} \| p_{2_j}$
	$E(x_{1_i})$		
	$E(s_{1_i})$		
1.	$X = (E(x_{1_1}), \dots, E(x_{1_i}), \dots, E(x_{1_M}))$	$\xrightarrow{X}$	
	$S = (E(s_{1_1}), \dots, E(s_{1_i}), \dots, E(s_{1_M}))$	$\xrightarrow{S}$	
			$\forall i, j$:
			$E(y_{i,j}) = (E(x_{1_i}) \oplus (-x_{2_j})) \odot r_{1_{i,j}} \oplus r_{2_{i,j}}$
			$E(t_{i,j}) = (E(s_{1_i}) \oplus (-s_{2_j})) \odot r_{3_{i,j}} \oplus r_{4_{i,j}}$
		$\xleftarrow{Y}$	$Y = \pi_1(E(y_{1,1}), \dots, E(y_{i,j}), \dots, E(y_{M,N}))$
		$\xleftarrow{T}$	$T = \pi_1(E(t_{1,1}), \dots, E(t_{i,j}), \dots, E(t_{M,N}))$
2.	$(\dots, y_{\pi_1(i,j)}, \dots) = (\dots, D(E(y_{\pi_1(i,j)})), \dots)$		
	$(\dots, t_{\pi_1(i,j)}, \dots) = (\dots, D(E(t_{\pi_1(i,j)})), \dots)$		
	$\forall i, j$:		
	$y'_{i,j} = y_{\pi_1(i,j)} \cdot r_{5_{i,j}} + r_{6_{i,j}}$		
	$t'_{i,j} = t_{\pi_1(i,j)} \cdot r_{7_{i,j}} + r_{8_{i,j}}$		
	$Y'' = (\dots, y''_{i,j} = y'_{\pi_2(i,j)}, \dots) = \pi_2(y'_{1,1}, \dots, y'_{i,j}, \dots, y'_{M,N})$	$\xrightarrow{Y''}$	
	$T'' = (\dots, t''_{i,j} = t'_{\pi_2(i,j)}, \dots) = \pi_2(t'_{1,1}, \dots, t'_{i,j}, \dots, t'_{M,N})$	$\xrightarrow{T''}$	
3.	$O_1 = \{(y''_{i,j}, t''_{i,j}) \| 0 \le y''_{i,j} < \wedge 0 < t''_{i,j}\}$		$O_1 = \{(y''_{i,j}, t''_{i,j}) \| 0 \le y''_{i,j} < 2^{l_1} \wedge 0 < t''_{i,j}\}$
	$O_2 = \{y''_{i,j} \| 0 < y''_{i,j} < 2^{l_1}\}$		$O_2 = \{y''_{i,j} \| 0 < y''_{i,j} < 2^{l_1}\}$
	$output = \begin{cases} \|O_1\| = N & \top \\ \|O_1\| < N \wedge \|O_2\| = N & \perp^* \\ otherwise & \perp \end{cases}$		$output = \begin{cases} \|O_1\| = N & \top \\ \|O_1\| < N \wedge \|O_2\| = N & \perp^* \\ otherwise & \perp \end{cases}$

the values are encrypted. The encrypted lists X and S are sent to the O/O. The O/O combines each element of the lists X and S with its own encoded elements by performing Eq. (9) and (10) for the respective lists. The combinations result in six comparisons for the guideword-parameter combination and the SIL values stored in lists Y and T, which are shuffled with the same random permutation π_1 and sent to the MV. As the guideword-parameter combination and SIL lists are shuffled with the same permutation, for every element of the guideword-parameter combination set there is a corresponding value from the masked SIL set. That allows for a preservation of correspondence of compared elements. In the second step, the MV decrypts the lists. The number of potential matches can already be seen. Suitable matches for the guideword-parameter combination must be small random numbers and positive numbers for the SIL values. Large or negative numbers for the decrypted Y indicate no match. Negative results in the decrypted T indicate that the process SIL is larger than the equipment SIL.

The decrypted results are masked again by the MV and sent back to the O/O shuffled with the same random permutation π_2.

As a result, there are two potential suitable matches ($R_2 = 2$), but the SIL values only match for one. In this example, the final result is "possibility to adjust".

A simulation of an illustrative example confirms that the protocol for comparison of modular *HAZOPElements* can compute the compatibility between process and equipment in a confidential manner.

5.1. Input data

The example in Fig. 4 shows a possible collection of input data from the Module Vendor and Owner/Operator. The combinations of guidewords and parameters correspond to certain values of SIL from the classification provided on Fig. 3. Here, the transformation from *HAZOPCases* to a collection of *HAZOPElements* is already done to solely focus on the comparison covered in the proposed protocol. The task of the developed confidential protocol is to compare the input data without allowing any of the parties to discover the raw inputs. The total amount of equipment *HAZOPElements* (Fig. 4a) and process *HAZOPElements* (Fig. 4b) is 24 and 18, respectively. One element of the process *HAZOPElements* collection has a higher SIL class than the respective element in the modular equipment and is highlighted in the figure. The rest of the elements comply with the safety requirements.

Fig. 4 highlights the comparison of the "less temperature" guideword and parameter combination that has an SIL class higher in the *Process HAZOP* compared to the *Equipment HAZOP*. During homomorphic computation, blinding, and permutation procedures, the underlying plaintexts are very large (e.g., 1024 bits) or large (e.g., 64 bits) values. The output of the comparison for the considered element is a recommendation for adjustment. If the Module Vendor is willing to increase the SIL value for this element, the integration between modular equipment and process becomes viable.

5.2. Result

Since guideword and parameter combinations are unique for each mHAZOP collection of elements, the comparison of all equipment and process *HAZOPElements* with each other can be conducted without fear of spotting bogus matches. Overall, 432 comparisons were computed for the current example and the results are shown in Fig. 5. The elements were considered as a match when the SIL value for the *Process HAZOP* element was less or equal than the respective value in the *Equipment HAZOP* element. These matches are marked with a tick in Fig. 5. In cases where the total amount of matches and possible adjustments is less than the total amount of elements in *Process HAZOP*, the equipment is not suitable for the process and integration of modular HAZOPs is not possible.

5.3. Discussion

The proposed protocol showed that homomorphic encryption could be used to compare the mHAZOP models of MV and O/O. However, for the results to be suitable for application, there are some prerequisites to be fulfilled. At first, the structure of the mHAZOP and the choice of guidewords and parameters must match for the different parties. In the provided illustrative example, the compatibility of the *HAZOPElements* from MV and O/O is given. This might not always be the case since

Table 3
Illustrative application of the protocol for the comparison of *HAZOPElements*.

Step	Module Vendor		Owner/Operator
0.	$H_1 = \{$[More Pressure, SIL 3], [No Flow, SIL 2], [Less Temperature, SIL 2]$\}$ $H_1 = \{x_{1_1}, x_{1_2}, x_{1_2}\}$ $x_{1_1} = [11],\ s_{1_1} = [3]$ $x_{1_2} = [03],\ s_{1_2} = [2]$ $x_{1_3} = [22],\ s_{1_3} = [2]$ $E(x_{1_1}) = 42... \ldots 8756$ $E(s_{1_1}) = 10... \ldots 1292$		$H_2 = \{$[More Pressure, SIL 1], [No Flow, SIL 3]$\}$ $H_2 = \{x_{2_1}, x_{2_2}\}$ $x_{2_1} = [11],\ s_{2_1} = [1]$ $x_{2_1} = [03],\ s_{2_2} = [3]$
1.	$X = (42... \ldots 8756, 52... \ldots 8472, 73... \ldots 8254)$ $S = (10... \ldots 1292, 82... \ldots 9573, 83... \ldots 7254)$	$\xrightarrow{X}$ $\xrightarrow{S}$	
			$\forall i, j$: $E(y_{1,1}) = (E(x_{1_1}) \oplus (-x_{2_1})) \odot r_{1_{1,1}} \oplus r_{2_{1,1}}$ $= (42... \ldots 8756 \oplus 63... \ldots 8762) \odot 87... \ldots 124 \oplus 5...3$ $= 37... \ldots 8354$ $E(t_{1,1}) = (E(s_{1_1}) \oplus (-s_{2_1})) \odot r_{3_{1,1}} \oplus r_{4_{1,1}}$ $= (10... \ldots 1292 \oplus 41... \ldots 9046) \odot 53... \ldots 423 \oplus 3...2$ $= 54... \ldots 1269$
		$\xleftarrow{Y}$	$Y = \pi_1(37... \ldots 8354, E(y_{1,2}), E(y_{2,1}), E(y_{2,2}), E(y_{3,1}), E(y_{3,2}))$
		$\xleftarrow{T}$	$T = \pi_1(54... \ldots 1269, E(t_{1,2}), E(t_{2,1}), E(t_{2,2}), E(t_{3,1}), E(t_{3,2}))$
2.	$(..., y_{\pi_1(1,1)}, \ldots) = (..., D(E(y_{\pi_1(1,1)})), \ldots)$ $= (23... \ldots 124, 62... \ldots 518, 5...3, 94... \ldots 614, 7...5, 18... \ldots 873)$ $(..., t_{\pi_1(1,1)}, \ldots) = (..., D(E(t_{\pi_1(1,1)})), \ldots)$ $= (64... \ldots 295, 41... \ldots 874, 6...5, 82... \ldots 734, 53... \ldots 256, 32... \ldots 861)$ $\forall i, j$: $y'_{i,j} = y_{\pi_1(i,j)} \cdot r_{5_{i,j}} + r_{6_{i,j}}$ $y'_{1,1} = 23... \ldots 124 \cdot 12... \ldots 531 + 7...8$ $t'_{i,j} = t_{\pi_1(i,j)} \cdot r_{7_{i,j}} + r_{8_{i,j}}$ $t'_{1,1} = 64... \ldots 295 \cdot 76... \ldots 324 + 5...1$		
	$Y'' = (..., y''_{i,j} = y'_{\pi_2(i,j)}, \ldots) = \pi_2(10... \ldots 415, \ldots, y'_{i,j}, \ldots, y'_{M,N})$	$\xrightarrow{Y''}$	
	$T'' = (..., t''_{i,j} = t'_{\pi_2(i,j)}, \ldots) = \pi_2(41... \ldots 027, \ldots, t'_{i,j}, \ldots, t'_{M,N})$	$\xrightarrow{T''}$	
3.	$O_1 = \{(y''_{i,j}, t''_{i,j}) \mid 0 \le y''_{i,j} < \wedge\, 0 < t''_{i,j}\}$ $\lvert O_1 \rvert = 1$ $O_2 = \{y''_{i,j} \mid 0 < y''_{i,j} < 2^{I_1}\}$ $\lvert O_2 \rvert = 2$ $output = \perp^*$		$O_1 = \{(y''_{i,j}, t''_{i,j}) \mid 0 \le y''_{i,j} < \wedge\, 0 < t''_{i,j}\}$ $\lvert O_1 \rvert = 1$ $O_2 = \{y''_{i,j} \mid 0 < y''_{i,j} < 2^{I_1}\}$ $\lvert O_2 \rvert = 2$ $output = \perp^*$

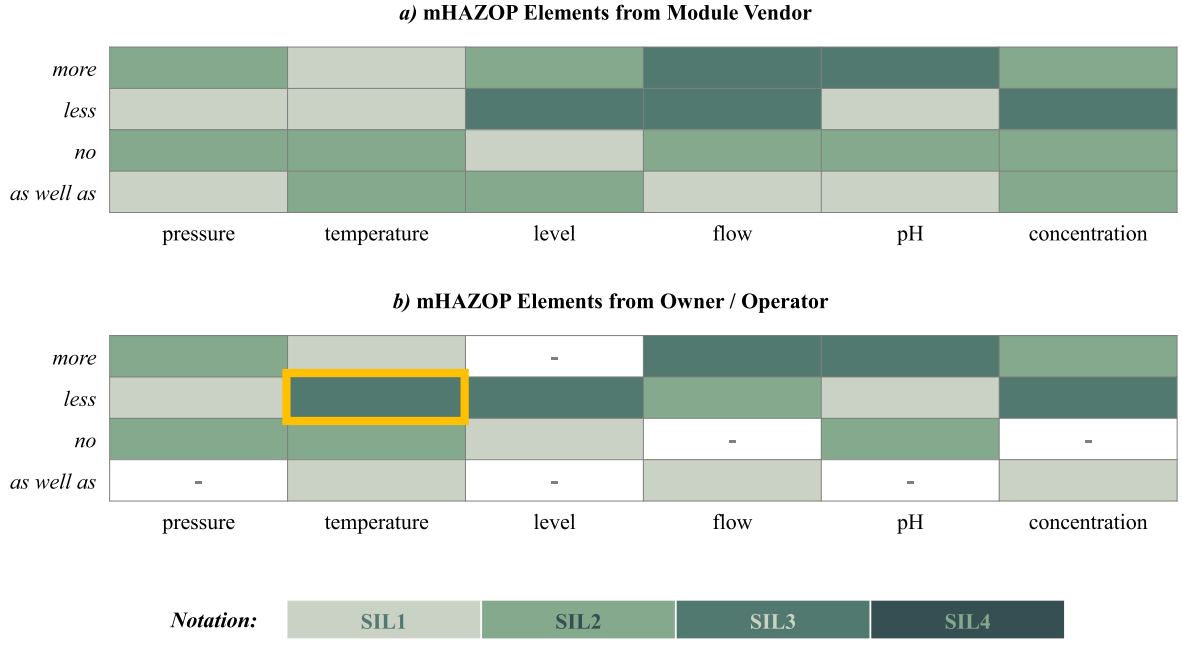


Fig. 4. Correspondence of risk parameters to the SIL classes.

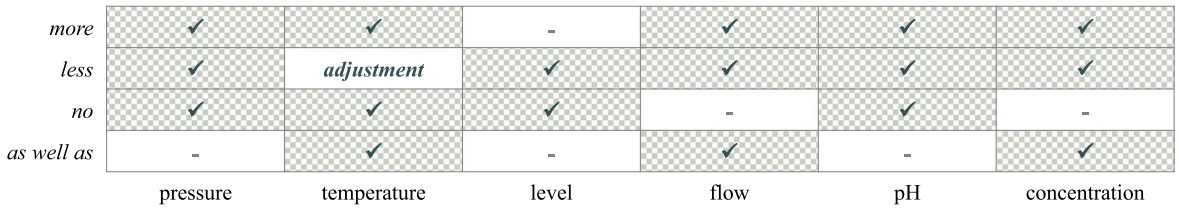


Fig. 5. Result of the simulated case.

in modular plants it is likely that they develop the mHAZOPs independently. A mismatch of the characterization of the *HAZOPElements* for the same scenario would affect the compatibility check. This issue can be addressed through standardization of the mHAZOP content and description. Furthermore, there is a high chance that the comparison results in no match, but the ability to adapt the equipment to fit the process can remain. At this point both parties need to decide if they want to disclose more information, since the protocol itself will not provide information on how a mismatch could be improved.

5.4. Security guarantees

For our protocol to be secure in the semi-honest model, it needs to ensure both correctness and privacy. We demonstrate correctness and privacy of our mHAZOP-comparison protocol through the following correctness and security arguments.

5.4.1. Correctness

Correctness is guaranteed if, for all valid inputs, the protocol returns the same output as a plaintext comparison of *HAZOPElements*. That is, it needs to return "suitable" if all process *HAZOPElements* have a corresponding equipment *HAZOPElement* with equal or greater SIL value, "not suitable with potential for adjustment" if all process *HAZOPElements* have a corresponding equipment *HAZOPElement* with SIL value that fits or can be adjusted to fit, and "not suitable" in all remaining cases. Therefore, it suffices to show that the protocol correctly recognizes guideword-parameter matches and correctly compares their SIL classifications. This is achieved by computing all possible *HAZOPElement* combinations, homomorphically comparing their guideword and parameter values, and homomorphically comparing their SIL classifications.

Homomorphic comparison Homomorphic comparison of two integers x, y is performed by subtraction, i.e., $\delta = x - y$, in combination with multiplicative blinding of δ with a large random number $r_1 \in \mathbb{Z}_{2^{l_1}}$, and additive blinding of δ with a small random value $r_2 \in \mathbb{Z}_{2^{l_2}}$ such that $0 < r_2$, i.e., $\delta' = \delta \cdot r_1 + r_2$. Given $2^{l_2} \ll 2^{l_1}$ (see Section 4.2), r_1 is exponentially larger than r_2, except with negligible probability. To demonstrate correctness, we need to show that $\delta' < 0$ indicates $\delta < 0$, $0 < \delta' < 2^{l_2}$ indicates $\delta = 0$, and $\delta > 2^{l_1}$ indicates $\delta > 0$, except with negligible probability.

In the first case, where $\delta < 0$, the maximal value is $\delta = -1$ for $\delta < 0$, since $\delta \in \mathbb{Z}$. Hence, multiplying the negative δ by the positive r_1 yields a negative multiple of r_1, i.e., $|\delta \cdot r_1| \geq r_1$. From $r_1 > r_2$, it follows that $|\delta \cdot r_1| > r_2$. Therefore, adding the positive r_2 to the negative $\delta \cdot r_1$ cannot result in a positive sum. Hence, if δ is negative, so is δ'.

In the second case, where $\delta = 0$, δ' is r_2, which is in $\{1, \ldots, 2^{l_2} - 1\}$. Therefore, if $\delta = 0$, $0 < \delta' < 2^{l_2}$.

The third case, where $\delta > 0$, is trivial. Both r_1 and r_2 are greater than zero. Hence, the product of the two positive values δ and r_1 is a random number larger than 2^{l_2}. Adding the small positive r_2 to $\delta \cdot r_1$ cannot change that. Consequently, if δ is positive, $\delta' > 2^{l_2}$, except with negligible probability.

Further considerations The permutations π_1, π_2 have no effect on the values' integrity as they only rearrange the *HAZOPElement* combinations.

Homomorphic computations with encrypted negative values work correctly since plaintexts are encoded in a two's-complement manner where overflows are prevented since all random values are exponentially smaller than $\mathbb{Z}_{n/2}$.

Decryption is correct if all underlying intermediate results are in $\mathbb{Z}_n$, which is guaranteed since the guideword-parameter encoding is a two-digit number, the SIL classification is a one-digit number, and all random values are exponentially smaller than $\mathbb{Z}_{n/2}$.

5.4.2. Privacy

Prerequisites We first define a participant's *view* $\mathcal{V}$ as their inputs, internal random tapes, and all messages they receive during protocol execution.

For a secure computation protocol to be secure in the semi-honest model 4.1, it is sufficient to demonstrate that anything an adversary $\mathcal{A}$ can learn during protocol execution can as well be learned given only the inputs and outputs of the protocol (Lindell, 2017). That is, it is sufficient to show that the view $\mathcal{V}_\mathcal{A}$ of $\mathcal{A}$ can be generated by a polynomial-time algorithm $\mathcal{S}$, referred to as *simulator*. The simulator performs this computation based solely on the inputs and outputs of the participant that is corrupted by $\mathcal{A}$.

For the sake of readability, the notation of random blinding values used in this Section differs from the notation of random blinding values used in the protocol description.

The simulator creates the protocol input by taking the original input. Furthermore, it simulates the coin tosses by employing the same pseudorandom generator that is used for sampling random numbers in the protocol. Hence, only the messages that the participants receive during protocol execution are relevant. For these, the simulator has to generate a simulated message for each message of the view such that both are computationally indistinguishable. We now briefly argue on the simulatability of all messages.

Simulation of X, S The lists of cipher texts transferred to O/O in step 1 can be simulated by sampling random numbers from $\mathbb{Z}_{n^2}$, which are computationally indistinguishable from Paillier ciphertexts due to the fact that Paillier's cryptosystem is semantically secure.

Simulation of Y, T MV can decrypt the list of guideword-parameter-difference ciphertexts received in step 1 since MV has access to the decryption key. Due to the fact that the underlying plaintexts are plaintext values that are multiplicatively and additively blinded, the simulator simulates the underlying plaintexts as follows. As it knows the presumed number N of guideword-parameter matches, it samples N values uniformly at random from $\mathbb{Z}_{2^{l_1}}$ and simulates the other plaintexts by generating valid guideword-parameter differences such that $\delta \neq 0$, sampling r_1 and r_2 from $\mathbb{Z}_{2^{l_1}}$ and $\mathbb{Z}_{2^{l_2}}$, respectively, and computing $m_{i,j} = \delta \cdot r_1 + r_2$ as the simulated plaintext. It proceeds similarly for the list of SIL-difference ciphertexts.

Simulation of Y'', T'' The messages received by O/O in step 2 can be simulated as before but with blinding the simulated δ's twice, i.e., $m_{i,j} = (\delta \cdot r_1 + r_2) \cdot r_3 + r_4$, where r_1 and r_3 are sampled uniformly at random from $\mathbb{Z}_{2^{l_1}}$ and r_2, r_4 are sampled uniformly at random from $\mathbb{Z}_{2^{l_2}}$, respectively.

Summary Given the above computations, the described simulators for MV and O/O generate outputs that are computationally indistinguishable from real views, which implies privacy of our protocol.

6. Summary and conclusion

In this paper, a privacy-preserving protocol for mHAZOP-based integration between process and equipment in modular plants was developed. Two parties, namely the Module Vendor and Owner/Operator, were involved in the privacy-preserving computation. The Module Vendor was involved in the design, manufacturing and, eventually, selling or outsourcing the modular equipment for various processes, such as chemical reaction, separation, or mixing. The Owner/Operator employed modular equipment to operate processes in their value chain. Both parties possessed sensitive information in their mHAZOP models.

Since risk parameters of mHAZOP could be confidential, this model considered what risk parameters are necessary for the computation of compatibility between EquipmentHAZOP and ProcessHAZOP and described a procedure for the integration within mHAZOPs. In order to prevent parties from exposing their sensitive data to one another, suitable cryptographic techniques, including partially homomorphic encryption, blinding, and permutations, were integrated into the privacy-preserving protocol. Additive homomorphic properties of Paillier's encryption scheme allow for comparison of the encoded mHAZOP values and, together with blinding and permutations, provide privacy of inputs and intermediate results.

The illustrative example based on the simulation of the protocol application in a modular process and equipment integration scenario was provided. Based on the amount of matches in comparison of *HAZOPElements*, the protocol provides the decision on compatibility between modular process and equipment. In addition, the security guaranties, including correctness and privacy aspects of the protocol, have been demonstrated.

To the best of our knowledge, this protocol is the first privacy-preserving approach for mHAZOP combination based on comparison of risk parameters and subsequent procedure of counting the matches. Therefore, this provides further opportunities for research in this area and raises more privacy awareness in modular process industries. Industrial partners collaborating through modular production concepts need reliable privacy protection tools for important information that comprise their competitive advantage. As a recommendation, various cryptographic protocols can be developed and optimized to tackle this issue in case-specific applications. In future work, the proposed protocol and its application need to be evaluated for industrial applications with larger HAZOPs from different MVs and O/Os and more restrictions on compatibility to explore further limitations. Additionally this concept can be integrated further into existing methodologies, such as the comparison of HAZOPCases using case-based reasoning Zhao et al. (2009).

CRediT authorship contribution statement

Zarina Chokparova: Conceptualization, Methodology, Software, Visualization, Validation, Writing – Original Draft, Review, Editing. **Kilian Becher:** Conceptualization, Methodology, Software, Visualization, Validation, Writing – Original Draft, Review, Editing. **Anselm Klose:** Conceptualization, Methodology, Validation, Writing – Original Draft, Review, Editing. **Thorsten Strufe:** Supervision, Review, Editing. **Leon Urbas:** Supervision, Review, Editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

Acar, A., Aksu, H., Uluagac, A.S., Conti, M., 2018. A survey on homomorphic encryption schemes: Theory and implementation. ACM Comput. Surv. 51 (4), http://dx.doi.org/10.1145/3214303.

Ahmad, A., Bosua, R., Scheepers, R., 2014. Protecting organizational competitive advantage: A knowledge leakage perspective. Comput. Secur. 42, 27–39. http://dx.doi.org/10.1016/j.cose.2014.01.001.

Alloghani, M., Alani, M.M., Al-Jumeily, D., Baker, T., Mustafina, J., Hussain, A., Aljaaf, A.J., 2019. A systematic review on the status and progress of homomorphic encryption technologies. J. Inf. Secur. Appl. 48, 102362. http://dx.doi.org/10.1016/j.jisa.2019.102362.

Atallah, M.J., Li, J., 2005. Secure outsourcing of sequence comparisons. Int. J. Inf. Secur. 4 (4), 277–287. http://dx.doi.org/10.1007/s10207-005-0070-3.

Baezner, M., Robin, P., 2017. Stuxnet. Technical Report, ETH Zurich, http://dx.doi.org/10.3929/ETHZ-B-000200661.

Baldea, M., Edgar, T.F., Stanley, B.L., Kiss, A.A., 2017. Modular manufacturing processes: Status, challenges, and opportunities. AIChE J. 63 (10), 4262–4272. http://dx.doi.org/10.1002/aic.15872.

Baldimtsi, F., Ohrimenko, O., 2015. Sorting and searching behind the curtain. In: Financial Cryptography and Data Security. Springer Berlin Heidelberg, pp. 127–146. http://dx.doi.org/10.1007/978-3-662-47854-7_8.

Bardhan, I., Whitaker, J., Mithas, S., 2006. Information technology, production process outsourcing, and manufacturing plant performance. J. Manage. Inf. Syst. 23 (2), 13–40. http://dx.doi.org/10.2753/mis0742-1222230202.

Beaver, D., Micali, S., Rogaway, P., 1990. The round complexity of secure protocols. In: Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing - STOC'90. ACM Press, http://dx.doi.org/10.1145/100216.100287.

Becher, K., Beck, M., Strufe, T., 2019. An enhanced approach to cloud-based privacy-preserving benchmarking. In: 2019 International Conference on Networked Systems (NetSys). IEEE, pp. 1–8.

Becher, K., Lagodzinski, J.A.G., Strufe, T., 2020. Privacy-preserving public verification of ethical cobalt sourcing. In: 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). IEEE Computer Society, pp. 998–1005.

Blake, I.F., Kolesnikov, V., 2004. Strong conditional oblivious transfer and computing on intervals. In: Advances in Cryptology - ASIACRYPT 2004. Springer Berlin Heidelberg, pp. 515–529. http://dx.doi.org/10.1007/978-3-540-30539-2_36.

Bogdanov, D., Kamm, L., Laur, S., Sokk, V., 2018. Rmind: A tool for cryptographically secure statistical analysis. IEEE Trans. Dependable Secure Comput. 15 (3), 481–495. http://dx.doi.org/10.1109/tdsc.2016.2587623.

Chokparova, Z., Urbas, L., 2021-09. Utilization of homomorphic cryptosystems for information exchange in value chains. In: 2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). pp. 01–07. http://dx.doi.org/10.1109/ETFA45728.2021.9613439.

Corallo, A., Lazoi, M., Lezzi, M., 2020. Cybersecurity in the context of industry 4.0: A structured classification of critical assets and business impacts. Comput. Ind. 114, 1–15. http://dx.doi.org/10.1016/j.compind.2019.103165, URL https://linkinghub.elsevier.com/retrieve/pii/S0166361519304427.

Cramer, R., 1999. Introduction to secure computation. In: Lectures on Data Security. Springer Berlin Heidelberg, pp. 16–62. http://dx.doi.org/10.1007/3-540-48969-x_2.

Cui, L., Zhao, J., Zhang, R., 2010. The integration of HAZOP expert system and piping and instrumentation diagrams. Process Saf. Environ. Protect. 88 (5), 327–334. http://dx.doi.org/10.1016/j.psep.2010.04.002.

Damgård, I., Geisler, M., Krøigaard, M., 2007. Efficient and secure comparison for on-line auctions. In: Pieprzyk, J., Ghodosi, H., Dawson, E. (Eds.), Information Security and Privacy. ACISP 2007. vol. 4586, Springer, Berlin, Heidelberg, http://dx.doi.org/10.1007/978-3-540-73458-1_30.

D'Amours, S., Montreuil, B., Lefranc, P., 1999. Networked manufacturing: The impact of information sharing. p. 17.

Dunjó, J., Fthenakis, V., Vílchez, J.A., Arnaldos, J., 2010. Hazard and operability (HAZOP) analysis. a literature review. J. Hard Mater. 173 (1–3), 19–32. http://dx.doi.org/10.1016/j.jhazmat.2009.08.076.

Fleischer, C., Wittmann, J., Kockmann, N., Bieringer, T., Bramsiepe, C., 2015. Sicherheitstechnische aspekte bei planung und bau modularer produktionsanlagen. Chem. Ing. Tech. 9 (87), 1258–1269. http://dx.doi.org/10.1002/cite.201400188.

Gaonkar, R., Viswanadham, N., 2001. Collaboration and information sharing in global contract manufacturing networks. IEEE/ASME Trans. Mechatronics 6 (4), 366–376. http://dx.doi.org/10.1109/3516.974850, URL http://ieeexplore.ieee.org/document/974850/.

Gentry, C., 2009. A Fully Homomorphic Encryption Scheme (Ph.D. thesis). Stanford University.

Goldreich, O., Micali, S., Wigderson, A., 1987. How to play any mental game or a completeness theorem for protocols with honest majority. In: Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing. STOC '87, Association for Computing Machinery, pp. 218–229.

Ioannidis, I., Grama, A., 2003. An efficient protocol for Yao's millionaires' problem. In: 36th Annual Hawaii International Conference on System Sciences, 2003. Proceedings of the. IEEE, http://dx.doi.org/10.1109/hicss.2003.1174464.

Kantarcioglu, M., Jiang, W., Liu, Y., Malin, B., 2008. A cryptographic approach to securely share and query genomic sequences. IEEE Trans. Inf. Technol. Biomed. 12 (5), 606–617. http://dx.doi.org/10.1109/TITB.2007.908465.

Kenyon, G., Meixell, M., Westfall, P., 2016. Production outsourcing and operational performance: An empirical study using secondary data. Int. J. Prod. Econ. 171, 336–349. http://dx.doi.org/10.1016/j.ijpe.2015.09.017.

Kletz, T.A., Institution of Chemical Engineers (Great Britain), 2001. Hazop and Hazan: Identifying and Assessing Process Industry Hazards. Institution of Chemical Engineers.

Kletz, T., 2018. Hazop and Hazan: Identifying and Assessing Process Industry Hazards, fourth ed. CRC Press, http://dx.doi.org/10.1201/9780203752227.

Klose, A., Bramsiepe, C., Szmais, S., Schafer, C., Krink, N., Welscher, W., Urbas, L., 2019. Safety-lifecycle of modular process plants - information model and workflow. In: 2019 4th International Conference on System Reliability and Safety (ICSRS). IEEE, pp. 509–517. http://dx.doi.org/10.1109/icsrs48664.2019.8987685.

Klose, A., Kessler, F., Pelzer, F., Rothhaupt, M., Kostiuk, D., Kabashi, A., Forkel, V., Urbas, L., 2021. Representing causal structures in HAZOP studies. In: 2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). IEEE, pp. 1–8. http://dx.doi.org/10.1109/ETFA45728.2021.9613357.

Kockmann, N., Thenée, P., Fleischer-Trebes, C., Laudadio, G., Noël, T., 2017. Safety assessment in development and operation of modular continuous-flow processes. Reaction Chem. Eng. 2 (3), 258–280. http://dx.doi.org/10.1039/c7re00021a.

Lin, H.-Y., Tzeng, W.-G., 2005. An efficient solution to the millionaires' problem based on homomorphic encryption. In: Applied Cryptography and Network Security. Springer Berlin Heidelberg, pp. 456–466. http://dx.doi.org/10.1007/11496137_31.

Lindell, Y., 2017. Tutorials on the Foundations of Cryptography: Dedicated to Oded Goldreich, first ed. Springer Publishing Company, Incorporated.

Micali, S., Rogaway, P., 1992. Secure computation. In: Advances in Cryptology — CRYPTO '91. Springer Berlin Heidelberg, pp. 392–404. http://dx.doi.org/10.1007/3-540-46766-1_32.

Paillier, P., 1999. Public-key cryptosystems based on composite degree residuosity classes. In: Advances in Cryptology – EUROCRYPT'99.

Pelzer, F., Klose, A., León, S.V., Horch, A., Knab, J., Langehegermann, M., Menck, U., Oehlert, R., Hauff, T., Kotsch, C., Knödler, M., Hensel, S., Menschner, A., Urbas, L., 2021a. Sicherheitslebenszyklen für die modulare automation in der prozessindustrie. Atp Magazin.

Pelzer, F., Klose, A., Miesner, J., Schmauder, M., Urbas, L., 2021b. Safety in modular process plants: demonstration of safety concepts. E & I Elektrotechnik und Informationstechnik 138 (7), 462–468. http://dx.doi.org/10.1007/s00502-021-00928-8.

ProcessNet, 2016. Modular plants: Flexible chemical production by modularization and standardization – status quo and future trends. DECHEMA, Gesellschaft für Chemische Technik und Biotechnologie.

Rabin, M.O., 1981. How to exchange secrets with oblivious transfer. Tech. Memo TR-81, Aiken Computation Laboratory, Harvard University, (See Cryptology ePrint Archive: Report 2005/187).

Rane, S., Boufounos, P.T., 2013. Privacy-preserving nearest neighbor methods: comparing signals without revealing them. IEEE Signal Process. Mag. 30 (2), 18–28. http://dx.doi.org/10.1109/msp.2012.2230221.

Rivest, R.L., Adleman, L., Dertouzos, M.L., 1978a. On data banks and privacy homomorphisms. Found. Secure Comput. 4 (11), 169–180.

Rivest, R.L., Shamir, A., Adleman, L., 1978b. A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM 21 (2), 120–126. http://dx.doi.org/10.1145/359340.359342.

Roy, S., 2017. Consider modular plant design. In: Process Development. American Institute of Chemical Engineers (AIChE).

Schindel, Polyakova, Harding, Weinhold, Stenger, Grünewald, Bramsiepe, 2021. General approach for technology and process equipment assembly (PEA) selection in process design. Chem. Eng. Process. - Process Intensif. 159, 108223. http://dx.doi.org/10.1016/j.cep.2020.108223, URL http://www.sciencedirect.com/science/article/pii/S0255270120306851.

Vaidhyanathan, R., Venkatasubramanian, V., 1996. Experience with an expert system for automated HAZOP analysis. Comput. Chem. Eng. 20, S1589–S1594. http://dx.doi.org/10.1016/0098-1354(96)00270-0.

VDI 2776-3, 2022. Verfahrenstechnische Anlagen - Modulare Anlagen - Sicherheit modularer Anlagen.

VDI/VDE 2180, 2019. Funktionale Sicherheit in Der Prozessindustrie - Einführung, Begriffe, Konzeption. Functional Safety in the Process Industry - Introduction, Terms, Conception. VDI Verein Deutscher Ingenieure e.V., VDI - The Association of German Engineers.

Yao, A.C., 1982. Protocols for secure computations. In: 23rd Annual Symposium on Foundations of Computer Science (Sfcs 1982). IEEE, http://dx.doi.org/10.1109/sfcs.1982.38.

Zhao, J., Cui, L., Zhao, L., Qiu, T., Chen, B., 2009. Learning hazop expert system by case-based reasoning and ontology. Comput. Chem. Eng. 33 (1), 371–378. http://dx.doi.org/10.1016/j.compchemeng.2008.10.006.