A High-Performance Solution for Data Security and Traceability in Civil Production and Value Networks through Blockchain

Erik Neumann
Faculty Applied Computer Sciences and Biosciences
University of Applied Sciences Mittweida
Mittweida, Germany
e-mail: neumann3@hs-mittweida.de

Kilian Armin Nölscher
Department Digitalization in Production
Fraunhofer IWU
Chemnitz, Germany
e-mail: kilian.noelscher@iwu.fraunhofer.de

Gordon Lemme
Department Digital Production Twin
Fraunhofer IWU
Dresden, Germany
e-mail: gordon.lemme@iwu.fraunhofer.de

Adrian Singer
Department Digitalization in Production
Fraunhofer IWU
Chemnitz, Germany
e-mail: adrian.singer@iwu.fraunhofer.de

Abstract—**This paper presents a blockchain-based solution for secure distribution of product, process and machine data across value networks. The data is stored in a high-performance private blockchain, which is a self-development as part of the federal funded project "safe-UR-chain". The infrastructure is secured by design through distributed ledger with a selectable consensus mechanism. In addition to the architectural overview of the concept, a system evaluation follows based on machine tool data.**

Keywords—Private Blockchain; Data Security; Traceability; Value Chain.

I. Introduction

Both, vertical and horizontal value chains have been increasingly threatened by cybercrime, sabotage and industrial espionage in recent years. The German Federal Criminal Police Office identified a total of 82,649 cases of cybercrime in the narrower sense (+80.5% compared to the previous year) in Germany. Studies by the digital association Bitkom and the Federal Office for the Protection of the Constitution (BfV) estimate an annual damage of 55 billion Euros for the German economy due to cybercrime, its consequences and defense measures. Of around 1,000 companies surveyed in Germany, 53% said they had been affected by cybercrime in the last two years, with the proportion of affected companies increasing steadily with company size (60% for 500+ employees) [1]. The origin of these crimes ranges from own or former employees, competitors to organized crime. Due to the general drive towards digitalization, this trend will continue in the future, posing an enormous threat to the civil infrastructure. As a countermeasure to this development, simply improving IT security step by step, e.g., by "hardening" software, is not enough. The project "safe-UR-chain" [2] researches new solutions for the described challenges.

A. Motivation

The basic protection objectives for digital communication include confidentiality, integrity, and availability [3]. There are numerous approaches to guaranteeing these, but in the past it has not been possible to implement these objectives with appropriate solutions in such a holistic way that they are applied across the board in operational practice. Communication between networked systems can be protected, e.g., by means of "end-to-end encryption". Production-specific data can also be encrypted to ensure confidentiality. This already poses an increasing dilemma when it comes to designing data-transparent value creation networks across company boundaries. Companies, especially Small and Medium-sized Enterprises (SMEs), are also increasingly having to make use of cloud solutions to maintain availability, as these guarantee high availability, which would only be possible as an inhouse solution with cost-intensive effort. This service provided by third-party providers is in competition with real-time requirements and confidentiality aspects. What is not taken into account here is data integrity, i.e., ensuring that stored data is correct. This goes hand in hand with a less strong protection goal of so-called non-repudiation (bindingness). Here, it is important to design communications in such a way that they are indisputable to a third party in retrospect.

This means that value networks consisting of production and logistics lack a practical, encrypted, traceable and tamper-proof solution for storing production-related data. Currently, production-relevant data is stored by various network nodes in a central database. Due to the already high and increasingly growing data density in the manufacturing industry, with simultaneous necessary data distribution and conditional data disclosure in distributed value chains, this approach appears to be increasingly disproportionate and impractical, especially for SMEs and with a lack of trust among the companies. Furthermore, future systems (machines, plants) will consist of a number of individual systems (control, measuring system, etc.), which is why new approaches are needed to save and synchronize the data recorded by the subsystems and, if necessary, to make it available to other applications within or outside the company.

In the field of transparent and tamper-proof data exchange and data storage, blockchain technology, as a representative of distributed ledger technologies, has become an increasingly relevant tool. By its very nature, a blockchain is a distributed stored linked list with the unique property that the addition of new data packets (blocks) is decided by a pseudo-democratic consensus process. The current main applications of blockchain technology are digital payment systems (e.g., Bitcoin [4]) and project financing. Due to their decentralized architecture and the consensus mechanisms used, these so-called public blockchains fulfill the requirements for availability and bindingness of the stored data. However, due to their lack of bandwidth, high costs and the fact that they are public, these public blockchains are unsuitable for practical use as storage locations for large and sensitive data volumes. Therefore, the use of so-called private blockchains, such as Hyperledger [5], is emerging in the enterprise environment. These differ in that access to them can be restricted. Furthermore, provided that the participating entities trust each other, a costly consensus algorithm for transaction verification can be avoided, thus significantly increasing bandwidth. The security properties of such a private blockchain (depending on the number of network nodes involved) is significantly lower compared to public blockchains [6].

B. Objectives

This motivation gave rise to the mentioned project “safe-UR-chain”, whose backbone is a private blockchain with inherent protection mechanisms. This paper provides the description, design and testing of the same. The primary goal was to increase IT security beyond the current state of the art, while consuming few resources and providing a transferable concept for a wide range of applications. In the subsequent evaluation, the deployment in a value network will be considered. The result is thus the provision of a blockchain-based architecture for the traceable and tamper-proof storage of selected data in the private blockchain, without being bound to data models. In particular, the following data is relevant:

- relevant master data of both companies,
- process and sensor data of the plant,
- movement and quality data of the products along the production and
- product-related data for end customers.

After the presentation of motivation and objectives in the Section I, the further structure of the work is as follows: After Section II “Architectural overview” presents the blockchain system and explains how it is implemented, the Section III “Setup of the example scenario” follows, which provides a testbed for the overall system in an industrial environment that is as close to reality as possible. The insights gained from this are presented and evaluated in the Section IV “Evaluation”, after which the Section V “Conclusion and Future Work” completes the paper.

II. Architectural overview

The primary purpose of the system is to store data in such a way that the integrity of individual records can be verified at a later date. To achieve this, the participating companies each use private blockchain networks that store both local records and block hashes from the blockchains of the other networks.

Each record goes through the same process until its existence at a certain point in time can be verified by all participating companies:

- intake of the data set
- distribution over the network
- inclusion into the blockchain
- “countersigning” by the other parties

This process is carried out on different layers of the system, these layers are the focus of this section.

A. Nodes

Nodes form the backbone of each local blockchain network. All of them perform basic tasks such as verifying cryptographic signatures and forwarding network messages - these essential tasks do not place high demands on the hardware. However, other tasks require either computational power or mass storage and are therefore implemented in a way that allows their use to be configurable. The node software is divided into several modules, as shown in Figure 1.

The **Ingest** module provides multiple interfaces for feeding data into the system. The simplest of which is a file ingest that watches a particular directory and reads the contents of all files that match the intake criteria (such as file name or type). This interface can be easily included into most existing systems since it only involves writing data to files. Other ingest interfaces can be added to this module, e.g., proprietary network based protocols that may already be used in some companies (see Section II-E). Within the ingest module, records are also extended with metadata and signed with the node’s private key. All nodes are assigned a public/private key pair, which they use to sign data within the system. By using private/public key cryptography, each node gets a unique, verifiable identity. This signature can later be used to trace records back to their origin. This signed bundle of data is generally referred to as a transaction. It is passed on to the **Processing** module where a configurable amount of worker threads perform a multitude of parallelizable tasks that are relevant for the node’s operation. These tasks include the creation and queuing of network messages, as well as the processing of incoming messages. The messages are sent and received by the **Networking** module. This module maintains a list of nodes in the network and establishes keep-alive connections to some of them over which data is sent to the network through the use of a flooding protocol [7]. Received and local transactions are bundled up

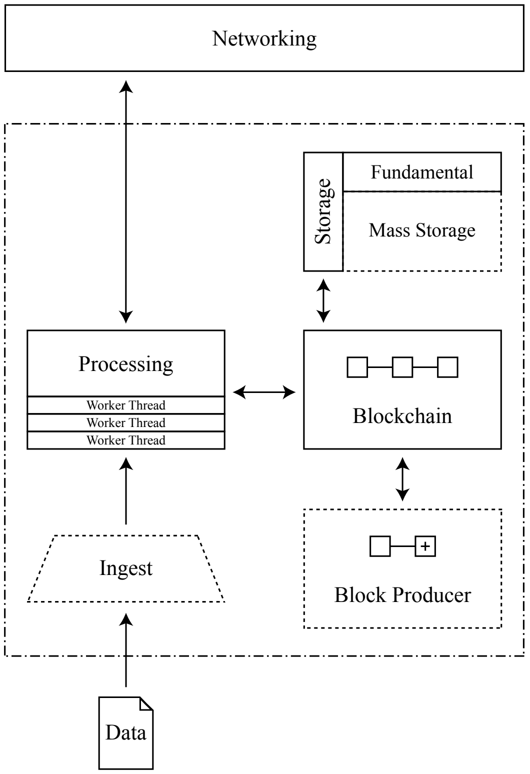

Figure 1. Software modules of a node; dashed lines indicate optional modules; the dash-dotted border signifies the system boundaries to other nodes and external data sources.

into blocks by nodes that have the **Block Producer** module enabled. The production of new blocks and their inclusion into the blockchain is governed by a generic interface that defines block validity and block work, which is used to decide upon the canonical (i.e., the "correct"/"longest" chain). Newly created blocks are then broadcast to the network and included in all nodes' blockchains. The transactions within these new blocks, are not necessarily stored on all nodes since this could use up the available storage on some of them. Instead, each node stores only the data that is absolutely necessary to verify the integrity of received data (i.e., block headers) and discards all other data based on a configurable filter. This way, the nodes' mass storage is only used for data that is relevant to their operation. Nodes that store *all* transaction data, can be used as archives within the network and can make this data quickly accessible to any program that consumes the node's API, e.g., GraphQL. By enabling only certain modules, four main node types can be created (see Table I). Using these types, the system can be integrated into existing infrastructures. E.g., in factories thin nodes can be used to ingest data from machine tools, Block Producers to add data into the blockchain and archive nodes for long term storage.

B. Blockchain

Within the system, each company uses a separate blockchain to store their own records, as well as data, which can later be used to verify the existence of remote records. Data is stored in a block by grouping transactions together into a merkle tree, by using this data structure the inclusion of single records in the blockchain can be proven by providing the block header, the merkle path, and the record itself [8]. This means that any future proof will only reveal the data in question, also proofs of this nature are efficient size wise, even if many records are stored in a particular block.

TABLE I
NODE TYPES

	Networking & Blockchain	Block Creation	Mass Storage
Thin Node	yes	no	no
Block Producer	yes	yes	no
Archive	yes	no	yes
Full Node	yes	yes	yes

The `Block` structure itself only contains the fields `header` and `data` (see Figure 2), its hash is not included and will be calculated on each node individually. The hash is calculated by serializing and then hashing the block header, which includes the the root of the merkle tree.

```
struct Block {
header: BlockHeader,
data: MerkleTree,
}
```

Figure 2. Block data structure.

The block header (see Figure 3) contains all fields that are necessary to verify a block and place it in the blockchain. Additionally, it contains fields that can be used by the generic interface that governs block validity and block work (therefore the consensus mechanism), e.g., the `nonce` field can be used to manipulate what hash the block has in proof of work and derivative mechanisms. And the `signatures` field can be used for protocols in which blocks become valid only when a certain amount of validators sign them. The fields in this data structure were chosen to facilitate many different consensus algorithms, so the system could potentially even be used in a non-private blockchain network.

```
struct BlockHeader {
timestamp: u128,
previous_digest: Vec<u8>,
difficulty: Difficulty,
nonce: Vec<u128>,
height: usize,
merkle_root: Vec<u8>,
signatures: Vec<SignedData>,
}
```

Figure 3. Block header data.

Further, transactions can be *stripped*, such transactions loose their payload and only retain a signed hash, as well as some metadata. These transactions allow for the construction of selectively stripped blocks (see Figure 4), which are used on most non-archive nodes to save space while keeping enough data to know what transaction to ask the network for, if additional information is ever needed.

Blocks themselves are stored in a tree like data structure (see Figure 5), which uses whatever consensus protocol was defined to create the canonical chain. It also keeps track of orphaned blocks and resolves them whenever possible. This `Block Tree` also contains a generic interface for storing block headers, merkle trees and transactions, each company can either use the supplied file system database or integrate their own storage solution into the system. The current implementation allows lookups in near constant time.

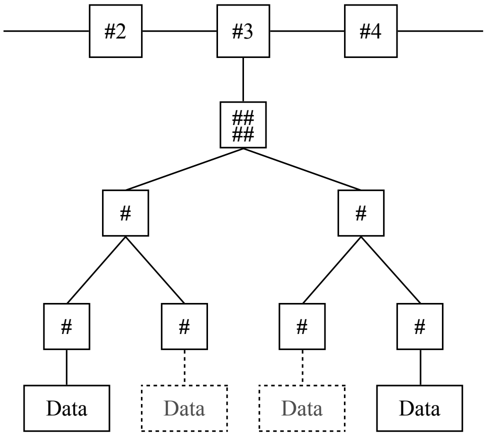

Figure 4. Blockchain (top) with the merkle tree shown for block No. 3, transactions (bottom) with a dotted outline are *stripped*.

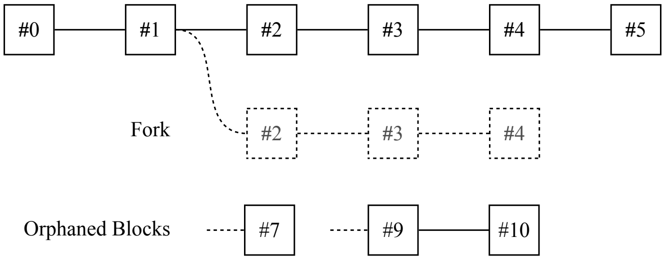

Figure 5. Block tree structure with the canonical chain (top), a fork (center) and orphaned blocks (bottom).

C. Local Network

A peer-to-peer network protocol is used to facilitate communications between nodes. It is constructed in a way that reduces manual maintenance by implementing automated bootstrapping and self repairing capabilities. The bootstrapping process uses so called "seed" nodes, which are nodes that have a high availability within the network (i.e., archive nodes). If at least one seed node is online, new nodes within the network will obtain information about the other peers and in turn request even more information from them. This method of bootstrapping as chosen to allow for automatic bootstrapping in networks which do not allow broadcast messages to be sent.

All nodes will try to maintain a complete list of all nodes, which are currently online in the network but only communicate to some of them. If any node goes offline, the list can be used to immediately increase the number of active connections to the desired amount. This behavior in addition to the use of a flooding protocol ensure the delivery of messages to wide parts of the network.

D. Global Network

Companies regularly share block hashes from their respective blockchains, these hashes are included into the other companies' blockchains, which removes the possibility for one company to retroactively change any data and recompute their local blockchain (some consensus protocols would allow this). This has the effect, that companies essentially "entangle" their local blockchains and in effect, provide an acknowledgment that they now possess the means to verify any proof of data up to this point. Example: company A creates a new block #1A and sends the block's hash to company B. Company B then includes a transaction with the remote block hash in block #2B. When the hash of #2B (or of any successor block) is sent to company A and included in their blockchain, all data from both blockchains is linked up to the shared block hashes. Since all companies on the global network do this, no peer will be able to change their blockchain and therefore be fully accountable for any records included in it. Figure 6 visualizes this concept.

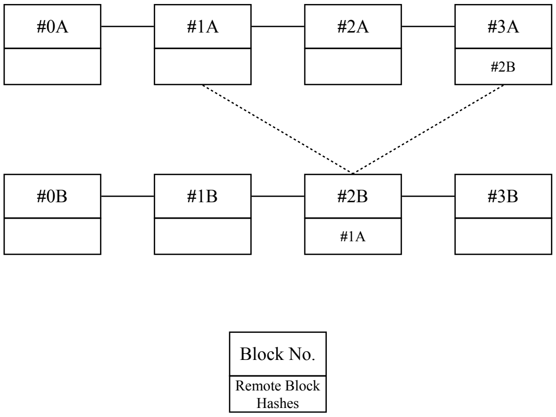

Figure 6. Two blockchains, both include block hashes from the other.

The exchange of these hashes is made possible by the use of an HTTPs message broker. This broker runs on a server that is accessible by certain nodes on all of the participants' networks. HTTPs was chosen as connections via this protocol are usually allowed by the firewalls used within an industry setting. All data sent to this broker is network-to-network encrypted, making it impossible to read messages even if an attacker were to gain access to the broker.

E. Extensibility

The nodes also provide an Application Programming Interface (API) to ingest any kind of data as new record into the blockchain. This API implements a transparent protocol called “Profichain” (Production and Factory Information over Blockchain). The Profichain protocol may be implemented in any kind of programming language, in order to ensure the highest compatibility to factory specific environments. The data is transmitted over the Transmission Control Protocol (TCP). The reference implementation also takes place within the evaluation and demonstrates that any process data of machinery or files are ingested safely. The protocol implements a 2-tier encryption with the Advanced Encryption Standard (AES). All tiers are optional and can be configured on clientside. The first tier represents an end-to-end encryption between the clientside and the node. The second tier provides a private encryption of the data that none of the network participants is able to decrypt except the original sender. The 2-tier encryption enables participants to work with strictly confidential data within the overall blockchain networks.

III. Setup of the Example Scenario

For the testbed, the complex construct of modern value chains is reduced to a minimal example and the delivery to a customer is simulated. This results in four stations, along which critical data is generated, see Figure 7. The raw material is turned into a semi-finished product (1), which is then further processed (2). This is followed by the assembly of the semifinished product with supplier components (3) and a final quality control (4). Transport takes place between each of the stations. During all steps, the product is clearly identifiable by an applied code. All data is assigned to this code, which enables the purchaser of the product in the event of an audit to seamlessly track product manufacture in retrospect.

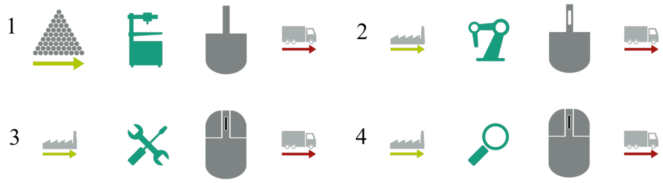

Figure 7. A simple value chain as a test scenario.

Physically, the value chain is represented for this research as follows: The origin of semifinished parts production lies in a 3D printer. By means of additive manufacturing, this generates a structure that is roughly similar to the final product. The background to this is increased productivity of the overall process, since a more finely finished structure must also be reworked to achieve high-precision requirements, but this entails a significantly longer printing time. Thus, the first data sets with relevance for storage in the blockchain result in information about the filament from source material, the 3D CAD model, or the converted machine code, as well as production data dropped during the process. The code for identifying the component is generated during printing and applied to a surface of the component that does not require any processing. This code is also stored in the blockchain. For the transport between the stations, an Automated Guided Vehicle (AGV) is used, which, equipped with a scanning unit and mobile blockchain nodes, can acknowledge the transport. Subsequently, iterative machining and comparison of actual to target geometry is used to evaluate the part accordingly. Depending on the number of iterations, a lot of data is generated here, which is stored in the blockchain. Subsequently, the assembly to the finished product takes place. Supplier parts, which consequently cannot show any data history in the blockchain, are identified via batch or part numbers. Finally, the quality inspection follows - its result and an inspection report are the last data records for the blockchain.

In order to be independent of different manufacturers of machine and automation controllers, data sources are transferred to OPC UA (Open Platform Communications Unified Architecture) servers: OPC UA is a standardized data exchange protocol for machine-to-machine communication [9]. Here, the data can be stored using suitable logger applications before being made available to the ingester.

As listed in Table I, the blockchain is based on different types of nodes. To create an executable instance of the blockchain, a Full Node is built into the network and thin nodes are built on each of the machines in the value chain. The physical component for the full node is a server with 8 cores and 32 GB RAM, as well as SSD mass storage, and for each of the thin nodes a single-board computer with 4 cores (x86), 8 GB RAM and SSD mass storage. Ubuntu 20.4 LTS is used as the operating system on all IT devices.

IV. Evaluation

A. Proceeding

The following system evaluation is to be seen as a first test of the fusion of blockchain system and testbed, while further and more extended investigations are ongoing. Therefore, the following evaluation was primarily limited to the basic questions regarding the performance of the blockchain system in conjunction with OPC UA data sources. The following questions had to be answered:

- Do packets get lost, especially during high transaction loads?
- What is the effect of varying the payload of a transaction?
- How big is the latency between transaction and block creation?

In advance, the blockchain system was tested in an isolated manner. For this, random transactions with a payload of 1kByte were generated and passed to the Ingester. The block time here, as in the following runs, was 15 seconds, and the experimental duration was 660 seconds, or 44 blocks.

For the test with real-world components, a data handler was written in Python3, which, as an OPC UA client, retrieves data from the OPC UA servers assigned to it, transfers it to a file and passes it on to the ingester. On each of the thin nodes such a data handler is running.

For this purpose, several test runs with different configurations were performed in a semi-automated way. The difference in the configuration refers to the size of the payload: 100Byte, 10x 100Byte, 100x 100Byte and 1000x 100Byte. This means that either a record of the machine with 100Byte was passed to the ingester immediately, or multiples were collected first and then passed as one transaction. In addition, each transaction that was passed to the ingester was also stored locally. This makes it possible to find possible packet losses. Before each launch, all existing data regarding the blockchain was deleted, so that a new blockchain was used each time.

B. Results

During the simulated tests of the blockchain system, up to 100 transactions per second could be processed. This is thus considered by us to be the limit of what is possible, determined by the load test.

The tests under the machine shop conditions delivered an average time between two data packets of $22.1\,\mathrm{ms} \pm 0.4\,\mathrm{ms}$ based on the thin node, with hardly any deviations occurring in the different configurations and no correlation between low and high payload could be found. At this point, the authors refer to the higher overhead for data retrieval between OPC UA server and client than for data storage. The delay between the times of transaction and block creation, on the other hand, is at least one block, i.e., 15 seconds, and depends on the number of transactions in the transaction pool and thus on the size of the payload. When few transactions with large payloads were created, they could usually be found within the next block. Many smaller transactions however were included within two blocks.

Finally, it should be noted that no packet was lost during the entire evaluation, which means that all data was transferred to the blockchain without errors.

V. Conclusion and Future Work

The connectivity of blockchain technology, has significant potential across all major value-added industries. These include, but are not limited to:

- automotive industry,
- machinery and plant engineering,
- aerospace industry,
- medical technology and the medical sector.

All of the industries mentioned are already characterized by a value chain in which upstream and downstream processes are linked via sensitive data processing. Due to the practicable and highly flexible implementation, the developed overall system is suitable for future integrations into existing production facilities, as it was developed independently of specifications regarding the data structure of the payload. Thus, the blockchain network is estimated to be easily transferable.

During operation, a stable sampling rate could be proven within the scope of the naturally occurring deviations due to the network communication of the OPC UA protocol. For high-frequency data acquisition, the Profichain API mentioned under Section II-E must be used. An evaluation of this is pending.

The described realization of the target system makes an important contribution to securing civil production and value creation networks, since faulty or manipulated product data are detected before products can cause damage in further processing or pose a threat to civil security at the end consumer in the public. Particularly noteworthy compared to other solutions is the combination of slimness, flexibility and high performance.

However, the mere safeguarding of data alone does not yet qualify it for use as a functional tool in the manufacturing industry. As global value networks grow ever closer together, the companies involved need tamper-proof and transparent production data with changing contractual partners. To increase trust in the authenticity of the data stored in the blockchain, hash values of blocks from the private blockchain are to be stored cyclically in a public blockchain. In this way, the advantages of both solutions (high performance for the private, high trustworthiness for the public) can be combined in a target-oriented manner.

As further work, on the one hand, a procedure is to be described to authenticate domain-specific data across locations and to distribute it in a tamper-proof manner. Furthermore, the exchange of relevant data between two sites or companies in a horizontal value chain is necessary. On the other hand, a detailed investigation must be carried out to gain knowledge regarding the possible attack vectors on the estimated system.

Acknowledgment

The project is funded by the German Federal Ministry of Education and Research within the framework of the announcement “Civil Security - Critical Structures and Processes in Production and Logistics” under the funding codes 13N15150 to 13N15153.

References

[1] R. Klatt, “Danger from cyber attacks has increased sharply in Germany.” [Online]. Available from: https://www.forschung-undwissen.de/nachrichten/oekonomie/gefahr-durch-cyberangriffe-hat-indeutschland-stark-zugenommen-13375090, June 2021.

[2] Fraunhofer IWU, “safe-UR-chain Webpage.” [Online]. Available from: https://safe-ur-chain.de, August 2021.

[3] G. Lemme, D. Lemme, K. A. Nölscher, and S. Ihlenfeldt, “Towards safe service ecosystems for production for value networks and manufacturing monitoring,” in Journal of Machine Engineering, Vol. 20 No. 1, March 2020, pp. 4–5.

[4] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System.” [Online]. Available from: https://bitcoin.org/bitcoin.pdf August 2021.

[5] “An Introduction to Hyperledger.” [Online]. Available from: https://www.hyperledger.org/wp-content/uploads/2018/07/HL_Whitepaper_IntroductiontoHyperledger.pdf August 2021.

[6] G. Lemme, K. A. Nölscher, E. Bei, C. Hermeling, and S. Ihlenfeldt, “Secure data storage and service automation for cyber physical production systems through distributed ledger technologies,” in Journal of Machine Engineering, Vol. 21 No. 1, March 2021, p. 4.

[7] A. S. Tanenbaum and D. J. Wetherall, “Computer Networks (5th ed.),” Pearson Education, 2010, p. 368.

[8] R. Merkle, “Protocols for Public Key Cryptosystems,” IEEE Symposium on Security and Privacy, 1980, pp. 125–127.

[9] OPC Foundation, “OPC 10000-1: OPC Unified Architecture - Part 1: Overview and Concepts.” [Online]. Available from: https://opcfoundation.org/about/opc-technologies/opc-ua/, July 2021.